FONT FORMAT VERSIONS 1.0, 1.3 and 2.0

November 19, 2009

The printer will accept fonts via download. Each V 1.0 font consists of a 54 byte header followed by the graphic bit map of each character, in ASCII order (e.g. the first character might be a space, followed by an exclamation point, followed by double quotes etc). The file must be in binary.

Each row of the character represented contains an integer number of bytes even if all bits within that byte are not used. A "one" will turn the paper dark; a "zero" will leave the paper white. There must be a bit mapped entry for each character within the sequence to be represented.

Each font has a unique one character and five character name (e.g. PT10B, 10CPI, FONT1). The one character name is used in the line printer mode to select the font via the ESC w command, and as an abbreviated method to refer to the font in Easy Print protocol. The five character name is used in Easy Print protocol only. You may select any one and five character names, but they MUST BE UNIQUE. If they are NOT unique, then the first font downloaded with that name will be selectable and the rest will be unusable.

To speed finding fonts, each five character name has a MOD 256 summation of that name in the header. To calculate this value, ADD the ASCII value of each character (in HEX) and use the lower order byte. For example, the characters in the name PT10B have ASCII values 50H, 54H, 31H, 30H, and 42H which add to 147H. Using just the lower order byte, we would enter 47H in the table where it says "MOD 256 summation..."

THE V1.0 HEADER:

The 54 bytes (ALL bytes must be present) within the header are as follows:

4 BYTES	May be anything, rewritten internally
3 BYTES	Font version number (must be "1.0")
1 BYTE	Mod 256 summation of the five character name
5 BYTES	Five character name for this font
1 BYTE	One character name for this font
1 BYTE	Must be 00 (says this table is a FONT)
2 BYTES, LSB 1st	Number of dots wide
2 BYTES, LSB 1st	Number of dots high
1 BYTE	Number of bytes in each row
2 BYTES	Number of bytes in each character
1 BYTE	First ASCII character represented in this font
1 BYTE	Last ASCII character represented in this font
1 BYTE	Reserved
1 BYTE	USER version number
8 BYTES	USER creation date
20 BYTES	USER description

THE V1.3 HEADER:

The 71 bytes (ALL bytes must be present) within the header are as follows:

4 BYTES	May be anything, rewritten internally
4 BYTES	Font version number (must be "1.3") plus NUL terminator
1 BYTE	Mod 256 summation of the five character name
6 BYTES	Five character name for this font plus NUL terminator
1 BYTE	One character name for this font
1 BYTE	One character name for this font (impact only - PICA)
1 BYTE	One character name for this font (impact only - ELITE)
1 BYTE	One character name for this font (impact only – italic PICA)
1 BYTE	One character name for this font (impact only – italic ELITE)
1 BYTE	Must be 00 (says this table is a FONT)
1 BYTE	Display Code (0=do not display on self test; 1=display)
2 BYTES, LSB 1st	Number of dots wide
2 BYTES, LSB 1st	Number of dots wide (impact only – PICA)
2 BYTES, LSB 1st	Number of dots wide (impact only – ELITE)
2 BYTES, LSB 1st	Number of dots wide (impact only – italic PICA)
2 BYTES, LSB 1st	Number of dots wide (impact only – italic ELITE)
2 BYTES, LSB 1st	Number of dots high
1 BYTE	Number of bytes in each row
2 BYTES	Number of bytes in each character
1 BYTE	First ASCII character represented in this font
1 BYTE	Last ASCII character represented in this font
1 BYTE	Dot row to place underline
1 BYTE	USER version number
9 BYTES	USER creation date plus NUL terminator
21 BYTES	USER description plus NUL terminator

THE V2.0 HEADER:

The 96 bytes (ALL bytes must be present) within the header are as follows:

4 BYTES	May be anything, rewritten internally
4 BYTES	Font version number (must be "2.0") plus NUL terminator
4 BYTES	Must be 96 (size of the header)
1 BYTE	Mod 256 summation of the five character name
6 BYTES	Five character name for this font plus NUL terminator
1 BYTE	One character name for this font
1 BYTE	One character name for this font (impact only - PICA)
1 BYTE	One character name for this font (impact only – PICA cond)
1 BYTE	One character name for this font (impact only – ELITE)
1 BYTE	One character name for this font (impact only – ELITE cond)
1 BYTE	Must be 00 (says this table is a FONT)
1 BYTE	Display Code (0=do not display on self test; 1=display)

2 BYTES, LSB 1st	Number of dots wide
2 BYTES, LSB 1st	Number of dots wide (impact only – PICA)
2 BYTES, LSB 1st	Number of dots wide (impact only – PICA cond)
2 BYTES, LSB 1st	Number of dots wide (impact only – ELITE)
2 BYTES, LSB 1st	Number of dots wide (impact only – ELITE cond)
2 BYTES, LSB 1st	Number of dots high
2 BYTES	Number of bytes in each row
2 BYTES	Number of bytes in each character
1 BYTE	First ASCII character represented in this font
1 BYTE	Last ASCII character represented in this font
2 BYTES	Dot row to place underline
2 BYTES	Baseline Position
1 BYTE	USER version number
11 BYTES	USER creation date plus NUL terminator
21 BYTES	USER description plus NUL terminator
15 BYTES	Must be 0xFF – reserved bytes

EACH CHARACTER:

Characters are represented within a "cell". The cell represents a character position on the printed line. The cell must extend upwards to the top of the highest character *and* down to the bottom of the lowest character. Characters are left justified within this cell; white space between characters can be adjusted but exist on the right side within this cell.

The "cell" for each font may be any width, but the downloaded image width must be a multiple of 8 wide (since 8 is the number of bits within a byte), and may be any height. Each character then contains a series of bytes; one or more bytes constitute each dot line row of the character; a sequence of these byte(s), then will build the entire character one dot line row at a time.

Each dot within the cell is 1/200 inch tall and 1/200 inch wide. To build a character approximately 0.07 inches tall and .06 inches wide, we would use 14 dotlines high and 12 dots wide. Since the 12 dot width of the character would need to be represented by two bytes, we could specify a character width of 12 dots for a zero dot spacing between characters, 13 dots for a one dot spacing, on up to 16 dots for a 4 dot spacing. This size character would probably look best with a 2 dot spacing, so we would specify a width of 14 dots total in the font header.

The letter "A" then might be built as follows (note that even though the character is only 12 dots wide, we must represent it with two full bytes, or 16 dots):

CELL EXAMPLE #1:

0000011000000000	->	006h & 000h
0000011000000000	->	006h & 000h
0000111100000000	->	00fh & 000h
0000111100000000	->	00fh & 000h
0001111110000000	->	01fh & 080h

0001100110000000 -> 019h & 080h
0011100111000000 -> 039h & 0c0h
0011111111000000 -> 03fh & 0c0h
0111111111100000 -> 07fh & 0e0h
0110000001100000 -> 060h & 060h
1110000001110000 -> 0e0h & 070h
1100000000110000 -> 0c0h & 030h
1100000000110000 -> 0c0h & 030h
1100000000110000 -> 0c0h & 030h

The "cell" for this character is 14 dots high and 12 dots wide. If, in the same font, we wanted to also represent the lower case "j", with descenders (that part of the character below an imaginary line along the bottom of all upper case letters) we would need to increase the cell size to 20 high from 14 high:

CELL EXAMPLE #2:

00000001 10000000 -> 001h & 080h
00000001 10000000 -> 001h & 080h
00000000 00000000 -> 000h & 000h
00000000 00000000 -> 000h & 000h
00000001 10000000 -> 001h & 080h
00000001 10000000 -> 001h & 080h
00000001 10000000 -> 001h & 080h
00000001 10000000 -> 001h & 080h
00000001 10000000 -> 001h & 080h
00000001 10000000 -> 001h & 080h
00000001 10000000 -> 001h & 080h
00000001 10000000 -> 001h & 080h
00000001 10000000 -> 001h & 080h
00000001 10000000 -> 001h & 080h
00000001 10000000 -> 001h & 080h
00000001 10000000 -> 001h & 080h
00000001 10000000 -> 001h & 080h
00000001 10000000 -> 001h & 080h
01100001 10000000 -> 061h & 080h
01110011 10000000 -> 073h & 080h
00111111 00000000 -> 03fh & 000h
00011110 00000000 -> 01eh & 000h

The letter "A" would need to allow for the increased cell size, and would become:

0000011000000000 -> 006h & 000h
0000011000000000 -> 006h & 000h
0000111100000000 -> 00fh & 000h
0000111100000000 -> 00fh & 000h

```
0001111110000000   ->   01fh & 080h
0001100110000000   ->   019h & 080h
0011100111000000   ->   039h & 0c0h
0011111111000000   ->   03fh & 0c0h
0111111111100000   ->   07fh & 0e0h
0110000001100000   ->   060h & 060h
1110000001110000   ->   0e0h & 070h
1100000000110000   ->   0c0h & 030h
1100000000110000   ->   0c0h & 030h
1100000000110000   ->   0c0h & 030h
0000000000000000   ->   000h & 000h
0000000000000000   ->   000h & 000h
0000000000000000   ->   000h & 000h
0000000000000000   ->   000h & 000h
0000000000000000   ->   000h & 000h
0000000000000000   ->   000h & 000h
```

FONT FILE EXAMPLE (V1.0):

Since the file must be in binary, it is often easiest to create using an assembler. The following example shows the header, and two of the characters in a form ready to assemble. The object code output, then, can be sent to the printer using the Oneil WINDOWS configuration program, or a special command.

```
        dl      PT10B_LINK  ; Value is reset internally and can be any 4 byte number
        db      "1.0"       ; font version number (1.0 flags this header and respresentation)
        db      047H        ; code resulting from mod 256 summation of name of font below
        db      "PT10B"     ; name of this font
        db      'E'         ; single letter used as an alternate to refer to this font
        db      00d         ; use 00 for straight text
        dw      14d         ; (ABSOLUTE = 12) number of dots wide for this font
        dw      20d         ; number of dots high for this font
        db      02d         ; number of bytes in each dot row of each character
        dw      40d         ; number of bytes in each character represented in this font
        db      'A'         ; The letter "A" is first ASCII character represented
        db      'B'         ; The letter "B" is last ASCII character represented
        db      00d         ; reserved
        db      '1'         ; USER font version number
        db      "04/30/96"  ; USER font creation date
        db      "2 CHARS EXAMPLE FONT"      ; USER 20 character description
; BIT MAPPED PATTERN OF LETTER "A":
        db      006h,000h
        db      006h,000h
        db      00fh,000h
```

```
        db      00fh,000h
        db      01fh,080h
        db      019h,080h
        db      039h,0c0h
        db      03fh,0c0h
        db      07fh,0e0h
        db      060h,060h
        db      0e0h,070h
        db      0c0h,030h
        db      0c0h,030h
        db      0c0h,030h
        db      000h,000h
        db      000h,000h
        db      000h,000h
        db      000h,000h
        db      000h,000h
        db      000h,000h
; BIT MAPPED PATTERN OF LETTER "B":
        db      0ffh,0e0h
        db      0ffh,0f0h
        db      0c0h,070h
        db      0c0h,030h
        db      0c0h,030h
        db      0c0h,070h
        db      0ffh,0e0h
        db      0ffh,0e0h
        db      0c0h,070h
        db      0c0h,030h
        db      0c0h,030h
        db      0c0h,070h
        db      0ffh,0f0h
        db      0ffh,0e0h
        db      000h,000h
        db      000h,000h
        db      000h,000h
        db      000h,000h
        db      000h,000h
        db      000h,000h
PT10B_LINK:
        END
```