

## Section 8

---

# HP PCL Soft Font Design

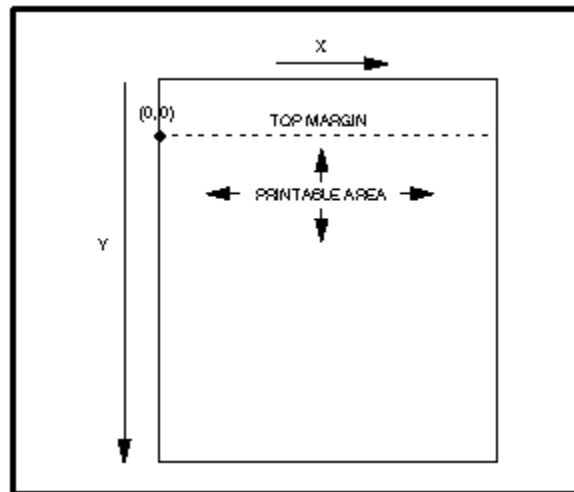
A font descriptor and one or more character descriptors define a soft font. A font descriptor is a block of data used to communicate font characteristics to the printer.\*

A character descriptor specifies the position and shape of an individual character and the implied movement of the cursor after printing the character. Following each character descriptor is a block of raster data containing the dot-per-inch image of the character.

### COORDINATE SYSTEM

Characters of a font are designed within a rectangular area called a cell. The dimensions of the cell are in PCL Coordinate system dots and are therefore page orientation-independent (refer to “Page Characteristics” in Section 2 for a complete description of the PCL coordinate system).

Character design dimensions within the cell are in physical coordinate system dots (rather than PCL Coordinate System dots). The physical coordinate system is defined in terms of the directions of raster scan (X) and paper motion (Y) as illustrated in Figure 8-1.

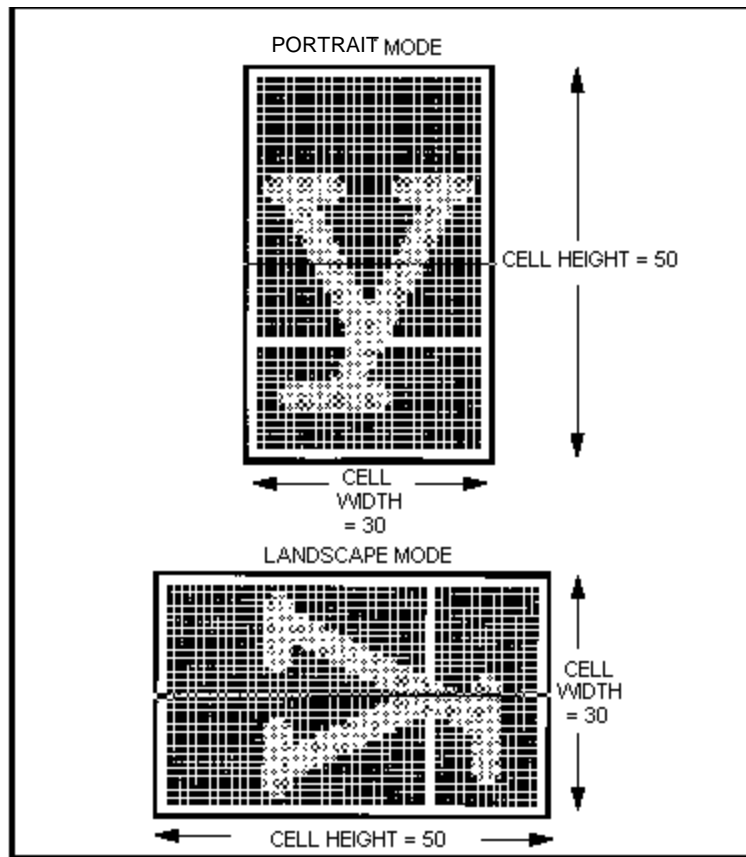


**Figure 8-1 Physical Coordinate System**

Since the raster scan and paper motion directions of a device are fixed, the physical coordinate system is page-orientation dependent. Character design dimensions within the cell are in physical coordinate system dots and therefore depend on the print orientation.

---

Figure 8-2 shows a character cell.



**Figure 8-2 Character Cell**

## FONT DESCRIPTOR FORMAT

A unique ID number should be designated prior to the download of a font descriptor. If an existing font is already associated with the ID, the existing font is deleted during the download of the font descriptor. Unless otherwise specified, inappropriate values in a font descriptor field invalidate the font download process; a font is not created and the associated font data is discarded.

Those font descriptor fields identified as "reserved" should be set to 0.

Table 8-1 describes the format of the font descriptor.

**Table 8-1 Font Descriptor Format**

Byte	15 (MSB)	8	7	(LSB) 0
0	Font Descriptor Size (64)			
2	Header format (0)		Font type	
4	Style MSB		Reserved	
6	Baseline position			
8	Cell width			
10	Cell height			
12	Orientation		Spacing	
14	Symbol set			
16	Pitch (default HMI)			
18	Height			
20	x - Height			
22	Width type		Style LSB	
24	Stroke weight		Typeface LSB	
26	Typeface MSB		Serif style	
28	Quality		Placement	
30	Underline position (Distance)		Underline thickness (height)	
32	Text height			
34	Text width			
36	First code			
38	Last code			
40	Pitch extended		Height extended	
42	Cap height			
44 - 47	Font number . . .			
48 - 63	Font name . . .			
64	Copyright (optional)			

---

**NOTE:** The following notation is used to define the data type of each field in the font descriptor described in the following subsections.

---

- (B) :Boolean (0,1)  
(UB) :Unsigned Byte (0 # 225)  
(SB) :Signed Byte (-128 # 127)  
(UI) :Unsigned Integer (0 # 65535)  
(SI) :Signed Integer (-32768 # 32767)  
(ASCxx) :ASCII string array (0 # xx-1) of characters

### Font Descriptor Size (UI)

Specifies the number of bytes in the font descriptor. The standard font descriptor size is 64 bytes.

### Font Type (UB)

Specifies one of the font types defined in Table 8-2. The font type should be set appropriately for the font's symbol set.

**Table 8-2 Font Type Values**

Value	Font Type
0	7-bit (96 characters) – character codes 32 to 127 [decimal] are printable.
1	8-bit (192 characters) – character codes 32 to 127 [decimal] and 160 to 255 [decimal] are printable.
2	8-bit (256 characters) – All characters are printable, however 0, 7 to 15, 27, and 28 [decimal] must be in transparency mode to print.

Use the transparent print data command to access those codes that are unprintable, yet have a character defined.

### Descriptor Format

The descriptor format byte should be set to zero.

---

**Baseline Distance (UI)**

Specifies the distance from the top of the cell to the baseline. The baseline is an imaginary dot row on which all of the characters in a line stand. The measurement of this distance is in PCL coordinate system dots. The valid range for the baseline distance is 0 to cell height minus 1.

**Cell Width (UI)**

Specifies the width of the cell in PCL coordinate system dots.

**Cell Height (UI)**

Specifies the height of the cell in PCL coordinate system dots.

**Orientation (UB)**

Specifies the orientation of the font. All characters within the font must have the same orientation; otherwise they are discarded as they are downloaded. Zero specifies portrait and one specifies landscape.

**Spacing (B)**

Specifies the spacing of the font. Zero specifies fixed spacing and one specifies proportional spacing.

**Symbol Set (UI)**

Specifies the symbol set for the font. This value (shown in Table 8-3) is computed by taking the "value field" value for the symbol set, multiplying it by 32, adding the ASCII decimal value of the escape sequence termination character, and subtracting 64.

For example, the ISO Symbol Set 61 has a value field of 0 and a termination character of D (ASCII decimal 68). Therefore the symbol set value for ISO 61 is

$$(0 \times 32) + 68 - 64 = 4$$

The valid range of symbol set values is 0 to 2047. Refer to Table 8-3 for the font descriptor symbol set values. The Hewlett-Packard emulation uses the font descriptor symbol set values from 0 to 1023. Symbol set escape sequence value field values 1024 to 2047 are available for use by independent font vendors. Symbol set escape sequence termination characters can be any uppercase ASCII character "A" through "V".

---

**Table 8-3 Symbol Set Values**

<b>Symbol Set Name</b>	<b>Value Field</b>	<b>Termination Character</b>	<b>Symbol Set Value</b>
HP Math-7	0	A	1
HP Line Draw	0	B	2
ISO 60: Norwegian version 1	0	D	4
*ISO 61: Norwegian version 2	0	D	36
HP Roman Extensions	0	E	5
ISO 4: United Kingdom	1	E	37
*ISO 25: French	0	F	6
ISO 69: French	1	F	38
*HP German	0	G	7
ISO 21: German	1	G	39
HP Greek-8	8	G	263
ISO 15: Italian	0	I	9

\*Not recommended for future use.

---

**Table 8-3 Symbol Set Values (cont'd)**

<b>Symbol Set Name</b>	<b>Value Field</b>	<b>Termination Character</b>	<b>Symbol Set Value</b>
*ISO 14: JIS ASCII	0	K	11
*ISO 57: Chinese	2	K	75
Technical-7	1	M	45
HP Math-8	8	M	269
ECMA-94 Latin 1	0	N	14
OCR A	0	O	15
OCR B	1	O	47
ISO 11: Swedish Names	0	S	19
*HP Spanish	1	S	51
ISO 17: Spanish	2	S	83
*ISO 10: Swedish	3	S	115
*ISO 16: Portuguese	4	S	147
*ISO 84: Portuguese	5	S	179
*ISO 85: Spanish	6	S	211
ISO 6: ASCII	0	U	21
HP Legal	1	U	53
*ISO 2: Intl Reference Version	2	U	85
OEM-1	7	U	245
HP Roman-8	8	U	277
PC-8	10	U	341
PC-8 (Danish/Norwegian)	11	U	373
HP PI Font	15	U	501

\* Not recommended for future use.

## **Pitch (UI)**

Specifies the pitch of the font in quarter-dot units (four quarter-dot units equal one dot). This defines the default HMI for the font. The IntelliBar supports a pitch range of 0 to 16,800 quarter-dots. Any values greater than 16,800 are set to 16,800.

**Height (UI)**

Specifies the design height of the font in quarter-dot units. This value, converted to points, is used as the height characteristics value of the font. A PCL point is 1/72 (0.01389) inch. The IntelliBar supports a height range of 0 to 10,922 quarter-dots. Any values greater than 10,922 are set to 10,922.

**x#Height (UI)**

Specifies the height of the lowercase "x" in quarter-dot units. The IntelliBar does not use this field.

**Width Type (SB)**

Specifies the proportionate width of characters in the font. The IntelliBar does not use this field.

**Style (UB)**

Specifies the style of the font. Upright is specified by 0 and italics by 1.

**Stroke Weight (SB)**

Specifies the thickness of the strokes used in designing the font. The supported stroke weight values are -7 through 7 (see Table 8-4). The thinnest stroke available is -7; and 7 is the thickest stroke weight available. The standard stroke weight for a medium font is 0; the standard stroke weight for a bold font is 3; and the standard stroke weight for a light font is -3.

**Table 8-4 Stroke Weights**

<b>Value(#)</b>	<b>Typeface</b>
-7	Ultra Thin
-5	Thin
-3	Light
0	Medium, or Book
+3	Bold
+5	Black
+7	Ultra Black

---

## Typeface

Specifies the least significant byte of the typeface of the font (see Table 8-5).

**Table 8-5 Typeface Values**

Value	Typeface
0	Line Printer
3	Courier
4	Helv
5	TmsRmn
6	Letter Gothic
8	Prestige
11	Presentations

## Serif Style (UB)

Specifies one of the serif styles defined in Table 8-6.

**Table 8-6 Serif Style Values**

Value	Serif Style
0	Sans Serif Square
1	Sans Serif Round
2	Serif Line
3	Serif Triangle
4	Serif Swath
5	Serif Block
6	Serif Bracket
7	Rounded Bracket
8	Flair Stroke

## Underline Distance (SB)

Specifies the distance from the baseline to the top dot row of the underline in dots. Zero specifies an underline position at the baseline. A positive value specifies an underline position above the baseline. A negative value specifies an underline position below the baseline.

**Underline Height (UB)**

Specifies the thickness of the underline in dots. The IntelliBar always uses a three-dot thickness for an underline.

**Text Height (UI)**

Specifies the font's optimum inter-line spacing in quarter-dot units. The IntelliBar does not use this field.

**Text Width (UI)**

Specifies the font's average lowercase character width in quarter-dot units. The IntelliBar does not use this field.

**Pitch Extended (UB)**

This is an addition to the pitch field that extends the pitch an extra eight bits. The value of this field is 1/1024 of one dot. For example, a 17 pitch font would have a pitch field of 70 (17.5 dots, or 17.1429 cpi) and a pitch extended field of 150 (0.1465 dots additional, which adds to 17.6465 dots, or 17.0005 pitch).

**Height Extended (UB)**

This is an addition to the height field that extends the height an extra eight bits. The value of this field is 1/1024 of one dot. For example, a 10 point font would have a height field of 166 (41.5 dots, or 9.96 points) and a height extended field of 170 (0.1660 dots additional, which adds to 9.9998 points).

**Font Name (ASC16)**

This is a 16-character ASCII field that you use to assign a font name. The IntelliBar prints this font name on the test print when you select font sample (see the *Intellibar User's Guide*).

**FONT DESCRIPTOR (FONT HEADER) COMMAND**

The font descriptor (font header) command is used to download font header data to the printer.

```
ESC)s#W [font header data]
```

The value field (#) identifies the number of bytes in the font header (see Table 8-7).

---

To download a font header for a portrait Roman-8, 10 pitch, 12 point, upright, medium, Courier font, with an ID number of one, send:

```
ESC*c1D (set Font ID to 1)
ESC)s64W [64 bytes of font header data]
```

Table 8-7 lists the font header data.

**Table 8-7 Font Header Data**

Field Name	Value	Description
Font Descriptor Size	64	Bytes
Reserved	0	
Font Type	1	Eight bit
Reserved	0	
Baseline Distance	35	
Cell Width	30	
Cell Height	50	
Orientation	0	Portrait
Spacing	0	Fixed Pitch
Symbol Set (8U)	277	(8x32 + (85-64))
Pitch	120	30 dots
Height	200	50 dots
xHeight	9	223 dots
Width Type	0	Medium
Style	0	Upright
Stroke Weight	0	Normal
Typeface (LSB)	3	Courier
Reserved	0	
Serif Style	2	Serif Line
Reserved	0	
Underline Distance	-5	
Underline Height	3	
Text Height	200	(50 dots)
Text Width	120	(30 dots)
Reserved		0
Reserved		0

**Table 8-7 Font Header Data (cont'd)**

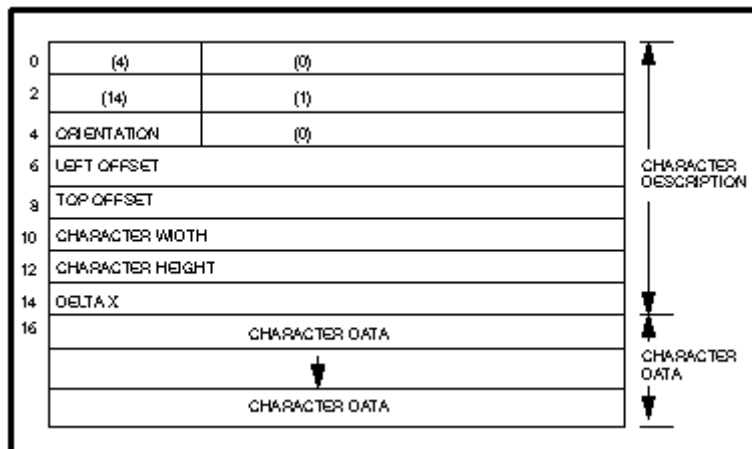
Field Name	Value	Description
Pitch Extended		0
Height Extended		0
Reserved		0
Reserved		0
Reserved		0
Font Name	Courier 10	0

## CHARACTER DESCRIPTOR AND DATA FORMAT

A unique character code must be designated prior to the download of a character descriptor and data. If the font being downloaded already contains a character with this code, the existing character is deleted during the download of the character descriptor and data. Unless otherwise specified, inappropriate values in a character descriptor field invalidate the character download process; a character is not created, and the associated descriptor and data is discarded.

Those character descriptor fields identified as “reserved” should be set to 0.

Figure 8-3 illustrates the format of the character descriptor and data.



**Figure 8-3 Character Descriptor and Data Format**

---

**NOTE:** The following notation is used to define the data type of each field in the character descriptor.

---

(B)	:Boolean	(0,1)
(UB)	: Unsigned Byte	(0 — 255)
(SB)	:Signed Byte	(-128 — 127)
(UI)	:Unsigned Integer	(0 — 65535)
(SI)	:Signed Integer	(-32768 — 32767)

### Format (UB)

Specifies the format of the character descriptor and data. The format number used by the printer is 4.

### Continuation (B)

Specifies whether the following data is a character descriptor block (0) or a continuation of the data (1) associated with the previous character descriptor.

Because the escape sequence value field is limited to 32767 bytes, characters whose number of descriptor and data block bytes exceed this limit must be downloaded in two or more blocks. Figure 8-4 illustrates the format of a character data continuation block.

Byte	15 – MSB	8	7	LSB – 0
0	Format (4)		Continuation (1)	
2	Character Data			
4	(in bytes)			
	–			
	–			
	–			

**Figure 8-4 Character Descriptors/Data Continuation Block**

---

### **Descriptor Size (UB)**

Specifies the size of the character descriptor in bytes. The IntelliBar uses a descriptor size of 14.

### **Class (UB)**

Specifies the format of the character data. The IntelliBar uses 1 as the character data format number.

### **Orientation (UB)**

Specifies the orientation of the character. Zero specifies portrait and one specifies landscape. The orientation of the character must match the orientation of the font.

### **Left Offset (SI)**

Specifies the distance in dots from the reference point to the left side of the character pattern on the physical page coordinate system (i.e., this value is orientation dependent). The left and top offsets locate the character reference point about the current active position (see ).

The valid range for the left offset is -4200 to 4200.

### **Top Offset (SI)**

Specifies the distance in dots from the reference point to the top of the character pattern on the physical coordinate system (i.e., this value is orientation dependent). The left and top offsets locate the character reference point about the current active position (see Figure I-5 and Figure I-6).

The valid range for the top offset is -4200 to 4200.

### **Character Width (UI)**

Specifies the width of the character in dots on the physical coordinate system.

The valid range for the character width is 1 to 4200. If you exceed the cell boundary defined in the font descriptor, the character will not download.

### **Character Height (UI)**

Specifies the height of the character in dots on the physical coordinate system.

The valid range for character height is 1 to 4200. If you exceed the cell boundary defined in the font descriptor, the character will not download.

---

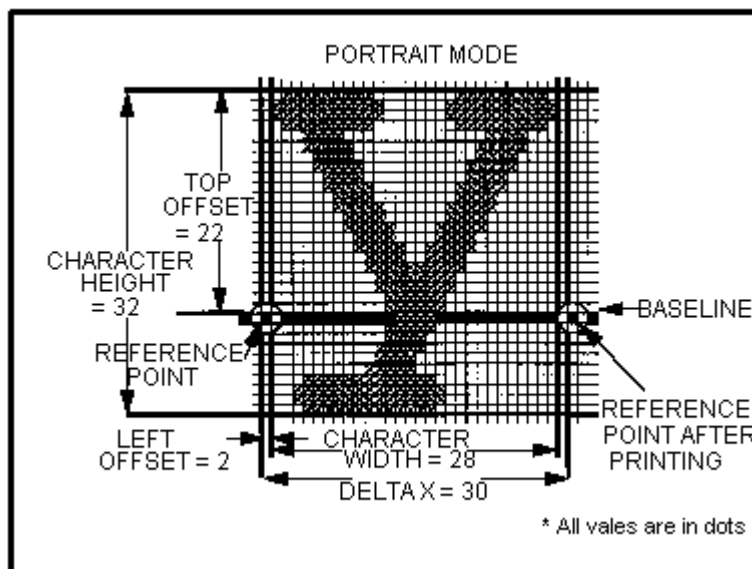
## Delta X (SI)

Specifies the number of quarter-dot units by which the horizontal position within the logical page coordinate system will be incremented after printing the character. This value is only used by the printer when the font is proportionally spaced.

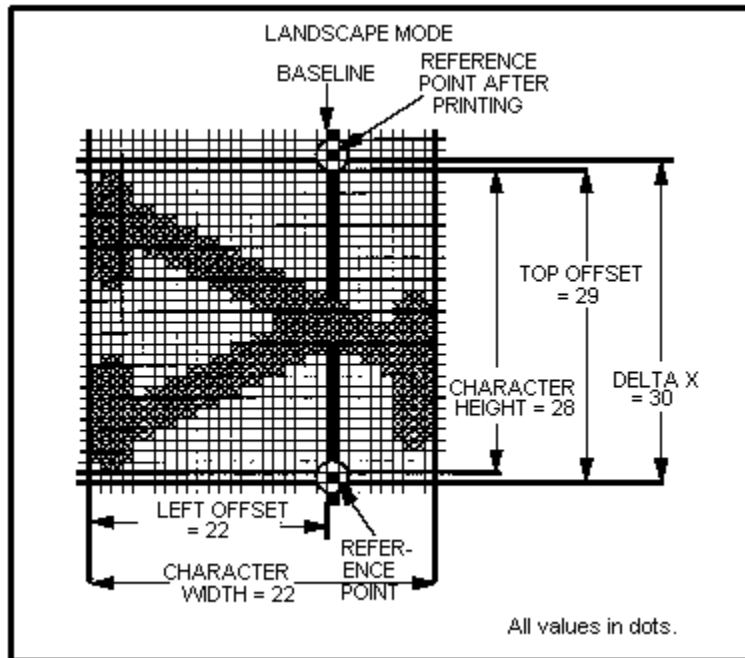
The valid range for delta X is 0 to 16800.

## Character Data

Character data is a string of bytes containing the dot-per-bit image of the character. If a bit is set to one, the corresponding dot will be printed. The data is grouped in dot rows; a row describes a one-dot high strip of the character from left to right, parallel to the printer's raster scan direction (see Figure 8-5 and Figure 8-6). Zeroed bits must be added to the end of each row to make it contain an integral number of bytes. The dot rows are organized from top to bottom of the character, i.e., the first dot row of data corresponds to the top dot row of the character.



**Figure 8-5 Portrait Character Example**



**Figure 8-6 Landscape Character Example**

## CHARACTER CODE

The character code sequence establishes the decimal code that will be associated with the next character downloaded. This single byte value is used to reference the character for printing.

ESC\*c#E

Decimal: 027 042 099 # 069

Hex: 1B 2A 63 # 45

#: character code.

For example, to designate the character code for an ASCII lowercase "y", send:

ESC\*c103E

## DOWNLOAD CHARACTER

The download character sequence downloads a character descriptor and associated character data to the printer.

ESC(s#W [character descriptor and data]

Decimal: 027 040 115 # 087

Hex: 1B 28 73 # 57

The value field (#) identifies the number of bytes in the character descriptor and data. The maximum number is 32767.

For example, to download the character descriptor and data for a portrait, 10 Pitch, 12 point, upright medium, Courier lowercase “y”, send:

ESC\*c121E (121 is the decimal character code for an ASCII lowercase “y”)

ESC(s144W [character descriptor and data]

Table 8-8 lists the character format, continuation, and descriptor data.

**Table 8-8 Portrait Character Descriptor Data**

Field Name	Value	Description
Format	4	
Continuation	0	
Descriptor Size	14	
Class	1	
Orientation	0	Portrait
Reserved	0	
Left offset	2	
Top Offset	22	
Character Width	27	
Character Height	32	
Delta X	120	30 dots

Figure 8-7 shows a portrait character data example.

**Character data:**

Dot Row	Bit Map				Decimal Equivalent			
01	01111111	11000000	00111111	11100000	127	192	63	224
02	11111111	11100000	01111111	11110000	255	224	127	240
03	11111111	11100000	01111111	11110000	255	224	127	240
04	01111111	11000000	00111111	11100000	127	192	63	224
05	00011110	00000000	00001111	10000000	30	0	7	128
06	00011111	00000000	00001111	10000000	31	0	15	128
07	00001111	00000000	00001111	00000000	15	0	15	0
08	00001111	10000000	00011111	00000000	15	128	31	0
09	00000111	10000000	00011110	00000000	7	128	30	0
10	00000111	11000000	00111110	00000000	7	192	62	0
11	00000011	11000000	00111100	00000000	3	192	60	0
12	00000011	11100000	01111100	00000000	3	224	124	0
13	00000001	11100000	01111000	00000000	1	224	120	0
14	00000001	11110000	11111000	00000000	1	240	248	0
15	00000000	11110000	11110000	00000000	0	240	240	0
16	00000000	11111001	11110000	00000000	0	249	240	0
17	00000000	01111001	11100000	00000000	0	121	224	0
18	00000000	01111111	11100000	00000000	0	127	224	0
19	00000000	00111111	11000000	00000000	0	63	192	0
20	00000000	00111111	11000000	00000000	0	63	192	0
21	00000000	00011111	10000000	00000000	0	31	128	0
22	00000000	00011111	10000000	00000000	0	31	128	0
23	00000000	00001111	00000000	00000000	0	15	0	0
24	00000000	00011111	00000000	00000000	0	31	0	0
25	00000000	00011110	00000000	00000000	0	30	0	0
26	00000000	00111110	00000000	00000000	0	62	0	0
27	00000000	00111100	00000000	00000000	0	60	0	0
28	00000000	01111100	00000000	00000000	0	124	0	0
29	00011111	11111111	00000000	00000000	31	255	0	0
30	00111111	11111111	10000000	00000000	63	255	128	0
31	00111111	11111111	10000000	00000000	63	255	128	0
32	00011111	11111111	00000000	00000000	31	255	0	0

**Figure 8-7 Portrait Character Data Example**

To download the character descriptor and data for a landscape, 10 pitch, 12 point, upright, medium, Courier lowercase “y”, send:

ESC\*c121E (121 is the decimal character code for an ASCII lowercase “y”)

ESC(s124W [character descriptor and data]

Character format, continuation, and descriptor data are listed in Table 8-9.

**Table 8-9 Landscape Character Descriptor Data**

<b>Field Name</b>	<b>Value</b>	<b>Description</b>
Format	4	
Continuation	0	
Descriptor Size	14	
Class	1	
Orientation	1	
Reserved	0	
Left Offset	-22	
Top Offset	28	
Character Width	32	
Character Height	27	
Delta X	120	30 dots

Figure 8-8 shows a landscape character data example.

---

**Character data:**

Dot Row	Bit Map				Decimal Equivalent			
01	01100000	00000000	00000000	00000000	96	0	0	0
02	11110000	00000000	00000000	00000000	240	0	0	0
03	11110000	00000000	00000000	00000000	240	0	0	0
04	11111100	00000000	00000000	00000000	252	0	0	0
05	11111111	00000000	00000000	00000000	255	0	0	0
06	11111111	11000000	00000000	00000000	255	192	0	0
07	11111111	11110000	00000000	00000000	255	240	0	0
08	11110111	11111100	00000000	00000000	247	252	0	0
09	11110001	11111111	00000000	00000000	241	255	0	0
10	11110000	01111111	11000000	00000000	240	127	192	0
11	01100000	00011111	11110000	00000000	96	31	240	0
12	00000000	00000111	11111100	00000110	0	7	252	6
13	00000000	00000001	11111111	00001111	0	1	255	15
14	00000000	00000000	01111111	11001111	0	0	127	207
15	00000000	00000000	01111111	11111111	0	0	127	255
16	00000000	00000001	11111111	11111111	0	1	255	255
17	00000000	00000111	11111101	11111111	0	7	253	255
18	01100000	00011111	11110000	01111111	96	31	240	127
19	11110000	01111111	11000000	00011111	240	127	192	31
20	11110001	11111111	00000000	00001111	241	255	0	15
21	11110111	11111100	00000000	00001111	247	252	0	15
22	11111111	11110000	00000000	00001111	255	240	0	15
23	11111111	11000000	00000000	00001111	255	192	0	15
24	11111111	00000000	00000000	00001111	255	0	0	15
25	11111100	00000000	00000000	00001111	252	0	0	15
26	11110000	00000000	00000000	00000110	240	0	0	6
27	11110000	00000000	00000000	00000000	240	0	0	0
28	01100000	00000000	00000000	00000000	96	0	0	0

**Figure 8-8 Landscape Character Data Example**

\* "This information is subject to change without notice. This information is provided "as is" without either express or implied warranty. IntelliTech International, Inc. disclaims any and all warranties with regard to this information. IntelliTech shall not be liable in any event for any special, indirect or consequential damages or any damages whatsoever resulting from loss of data, profits or use, for any reason or in any action, arising out of or in connection with the use or performance of this information. "