Chapter 11: Downloading Characters

Contents of this Chapter

This chapter describes the following PCL commands:

11.1 Introduction

The Second Part of Font Creation

As described in Chapter 10, font downloading consists of (1) downloading the font definition and (2) downloading the individual characters in the font. Chapter 10 described font definition; this chapter describes character definition.

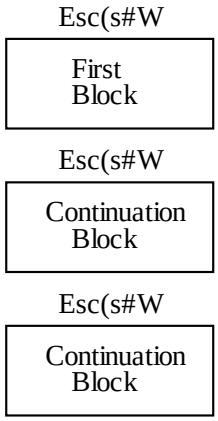
Each character is treated as a separate entity that can be downloaded and deleted. Individual characters are identified by a **character code** and defined by a **character definition**.

Character Code

The character code is an identification number — analogous to Font ID — attached to each character by *Esc*c#E*. A unique decimal character code (e.g., ASCII 33) must be designated prior to downloading a character's definition. If the font being downloaded already contains a character with this code, the existing character is deleted when the character is downloaded.

Character Definition

A character is defined by *Esc(s#W[character definition]*. This command can send a maximum of 32767 bytes. If more bytes are needed, the command is used again as many times as necessary to send the additional data. The data sent by a single command is called a **block**. Additional data for a given character is sent as ***continuation blocks***.



Character Definition

A ***character definition*** is composed of the following:

- Header (one per block)
- Descriptor
- Data

The character ***header*** consists of the first two byte fields (Format and Continuation) in every character definition. These two bytes are repeated in the continuation blocks, if any. The Format field describes the type of character to follow (e.g., bitmap, Intellifont, TrueType, etc). The Continuation field indicates previous blocks in the character definition: zero means none.

The character ***descriptor*** contains information such as character width and height, left and top offsets, etc.

The binary ***data*** following the descriptor specifies the character shape — raster data for bitmap fonts or outline data for scalable fonts.

11.2 The Character Code

Before each individual character is downloaded, a unique identification code must be assigned so the character can be referenced.

Character Code *Esc * c # e/E*

Establishes a decimal ASCII code for the next character downloaded.

Value(#) = Character code (decimal)
Default = 0
Range = 0 to 2^{32}-1

The character code is a state variable that must be designated prior to the download of a character descriptor. Any existing characters with the same code are deleted.

Unbound Intellifont Fonts - For unbound Intellifont fonts, the character code for a given character will equal the MSL (HP Master Symbol List) number for that character (see the *Book of Characters* for character code MSL assignments).

Unbound TrueType Fonts - For unbound TrueType fonts, the character code for a given character will equal its Unicode index as provided in the cmap table of the TrueType font file. A special code (0xffff) must be used to download vertical substitution characters, missing character glyphs, and glyph pieces that never stand alone as characters.

EXAMPLE: *Esc*c103E* sets the character code to 103. If followed by the Character Descriptor command (*Esc(s#W*) with a valid character descriptor and data, a character is defined in the code location corresponding to the ASCII lower case "g".

DEVICE NOTE: DJs below 1000 and pre-LJIIs have a range of 255. LJIII has a range of 32767.

11.3 The Character Definition

After downloading the font definition, each character in the font must be defined.

Download Character *Esc (s # W [Character Definition]*

Downloads a character definition with the character code assigned by *Esc*c#E*.

Value(#) = Number of bytes following the terminating character
Default = NA
Range = 0 to 2^{32} -1

The value field (#) contains the number of bytes to be downloaded. If more bytes are needed, this command is used again as many times as necessary. The group of bytes sent by one command is called a ***block***. A character definition consists of a first block and zero or more ***continuation blocks***.

An unsupported or invalid character definition is ignored and discarded. An out-of-memory condition during character download deletes the entire font. Reserved fields should be set to 0.

LaserJet Bitmap Character Definition

The format for the LaserJet bitmap character definition and continuation block is shown below. Format is set to 4, and Descriptor Size is set to 14. The character descriptor is grayed.

Byte	15 (MSB) 8	7 0 (LSB)
0	Format (4)	Continuation (0)
2	Descriptor Size (14)	Class (1)
4	Orientation	Reserved (0)
6	Left Offset	
8	Top Offset	
10	Character Width	
12	Character Height	
14	Delta X	
16	Bitmap Character Data: (in bytes) ...	

Byte	15 (MSB) 8	7 0 (LSB)
0	Format (4)	Continuation (non-zero)
2	Bitmap Character Data: (in bytes) ...	

LaserJet Bitmap Character Definition and Continuation Block Format

DeskJet Bitmap Character Definition

The format for the DeskJet5xx series (except 540) character definition is shown below. These DeskJets do not allow continuation blocks.

Byte	15 (MSB) 8	7 0 (LSB)
0	Format (5, 9, 12)	Continuation (0)
2	Descriptor Size	Character Type
4	Width (normal)	Compressed Width
6	Left PS Pad	Right PS Pad

DeskJet Character Definition

Intellifont Character Definition

The format for the Intellifont character definition and continuation block is shown below. Format is set to 10, and Descriptor Size is set to 2. The character descriptor is grayed.

Byte	15 (MSB) 8	7 0 (LSB)
0	Format (10)	Continuation (0)[1]
2	Descriptor Size (2)	Class (3 or 4)
4	Contour Character Data: (in bytes) (see the "Class" description) …	
	Reserved (0) [2]	Checksum [2]

Byte	15 (MSB) 8	7 0 (LSB)
0	Format (10)	Continuation (non-zero)[1]
2	Contour Character Data — resumed: (in bytes) (see the "Class" description) …	
	Reserved (0) [2]	Checksum [2]

[1] Continuation is supported for classes 1, 2, 3, and 15 only (compound characters cannot be continued.

[2] These bytes appear only on the last (continuation) block of data.

Intellifont Character Definition and Continuation Block Format

TrueType Character Definition

The format for the TrueType character definition and continuation block format is shown below. The Format field is set to 15. Descriptor Size, which includes everything after the continuation byte but prior to the Character Data Size field, may vary; the minimum is 2. The character descriptor is grayed.

Byte	15 (MSB) 8	7 (LSB) 0	Byte
0	Format (15)	Continuation (0)	1
2	Descriptor Size	Class (15)	3
4	{Additional descriptor data may be inserted here}		
...	...		1+Desc Size
2+Desc Size	Character Data Size		3+Desc Size
4+Desc Size	Glyph ID		5+Desc Size
6+Desc Size	TrueType Glyph Data		
	...		
			# - 3
# - 2	Reserved (0)	Checksum	# - 1

TrueType Character Descriptor (no continuation block required)

Byte	15 (MSB) 8	7 (LSB) 0	Byte
0	Format (15)	Continuation (0)	1
2	Descriptor Size	Class (15)	3
4	{Insert additional descriptor data here}		
…	…		1+Desc Size
2+Desc Size	Character Data Size		3+Desc Size
4+Desc Size	Glyph ID		5+Desc Size
6+Desc Size	Beginning of TrueType Glyph Data		
	…		# - 1

Byte	15 (MSB) 8	7 (LSB) 0	Byte
0	Format (15)	Continuation (1)	1
2	Conclusion of TrueType Glyph Data		
	…		
			# - 3
# - 2	Reserved (0)	Checksum	# - 1

TrueType Character Descriptor (multiple character blocks)

Format (UBYTE)

Specifies the character descriptor format.

Value	Format
0	82906A
1	82450A
3	QuietJet
4	LaserJet bitmap
5	DeskJet
6	PaintJet
7	PaintJet XL
8	RuggedWriter
9	DeskJet Plus
10	Intellifont
12	DeskJet 500
15	TrueType

The character is discarded if the format is different from that expected by the device.

Continuation (BOOL)

Specifies whether the subsequent data is a character descriptor block (0) or a continuation (non-zero) of the data associated with the previous character descriptor.

If the byte count in the value field of the Download Character command exceeds 32767, the character must be sent in two or more blocks. The additional bytes are sent in as many *continuation blocks* as needed (except compound characters).

NOTE: Compound characters (e.g. accented characters) cannot be continued.

A character that has not received all the character data is an "incomplete" character. There is at most one incomplete character at a time. If an incomplete character is deleted, any subsequent continuation downloads are ignored.

A continuation block that is downloaded before the first block was received is ignored.

DEVICE NOTE: Set to 0 for DJs below 1200.

Descriptor Size (UBYTE)

Specifies character descriptor size in bytes. The descriptor follows the character header, which consists of the first two bytes of the character definition (the Format and Continuation fields). For bitmap characters, the descriptor size includes Descriptor Size through Delta X. For Intellifont characters, the descriptor size includes only Descriptor Size and Class. For TrueType characters, the descriptor size includes Descriptor Size and Class, but additional descriptor information can follow; therefore, the minimum TrueType descriptor size is 2.

Value	Device
8	DeskJet 5xx except 540 (Format 5 or 9 character descriptor)
8	DeskJet 5xx except 540 (Format 12 character descriptor)
2	Intellifont
2+	TrueType (additional descriptor information can be added)
14	LaserJet bitmap

Class (UBYTE)

Specifies the format of the character data.

Value	Class
1	Bitmap
2	Compressed bitmap
3	Intellifont
4	Compound Intellifont
15	TrueType

DEVICE NOTE: LJ+, LJII, LJ2000 use a value of 1. DJs below 1200 do not use this field.

Class 1: Bitmap Data — Class 1 character data is a string of bytes containing the dot-per-bit image of the character, with no compression. A "1" bit causes the dot to be printed.

The data is grouped in dot rows describing a one-dot high strip of the character from left to right in the direction of the printer's raster scan. The dot rows are organized from top to bottom of the character (in portrait orientation): the first dot row of data corresponds to the first dot row of the character. The end of each row is padded with zero bits so it contains an integral number of bytes.

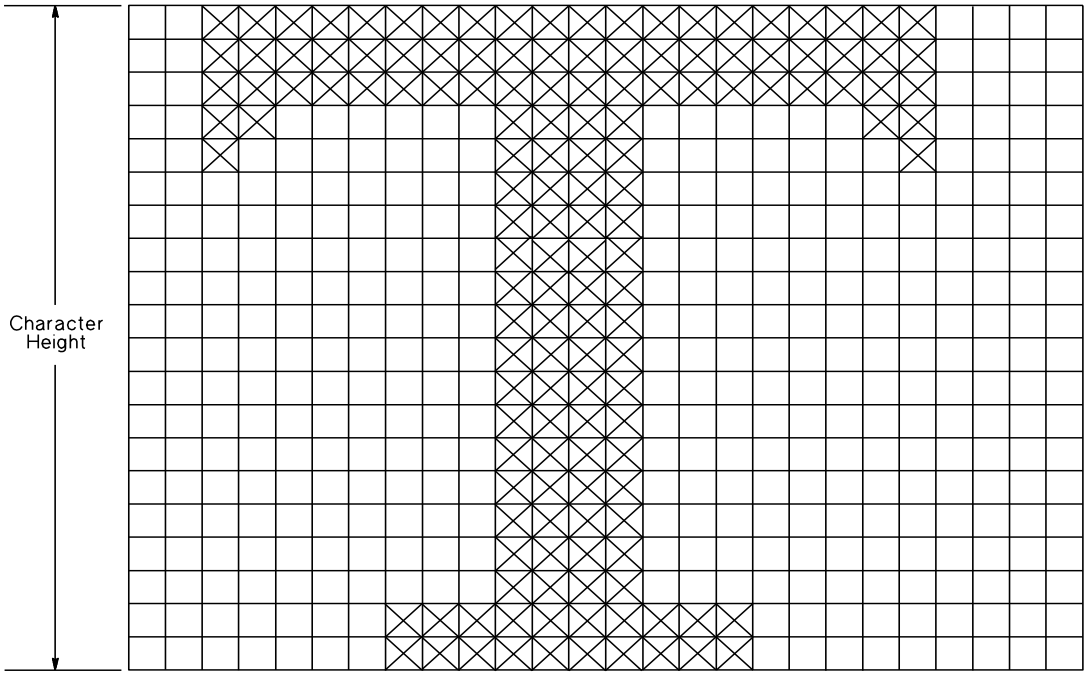
The number of bytes of character data should be: $\mathrm{TRUNC}\left(\frac{\mathrm{CharWidth}+7}{8}\right)\times\mathrm{CharLength}$

Additional data is discarded. The character will consist only of the downloaded character data, even if this is insufficient; the missing part is undefined.

Class 2: Compressed Bitmap Character Data — The string of bytes composing Class 2 character data is in compressed run-length-with-line-repetition format. A row's first byte tells how many times the row is repeated. The second byte tells how many white dots start the row (it is 0 if the first dot is black). The third byte tells how many black dots follow, the fourth byte tells how many white dots follow that, etc.

The *character width* (dots) field in the character descriptor determines the row width. The dot count for each row in the character cell must equal the character width. In the following figure the cell width is 20, so each row (excluding the repetition count byte) adds up to 20.

More than 255 dots of the same type is indicated by a byte containing 255 followed by a byte containing 0 (meaning there are none of the opposite type of dots), followed by a byte containing the count of the remaining dots of the current type.



Line Repetition	Number White Pixels	Number Black Pixels	Number White Pixels	Number Black Pixels	Number White Pixels	Number Black Pixels
2	0	20	–	–	–	–
0	0	2	6	4	6	2
0	0	1	7	4	7	1
12	8	4	8	–	–	–
1	5	10	5	–	–	–

* Byte alignment is necessa
only for raster data (ie, not
for compressed raster dat

Uncompressed – 60 Bytes
Compressed – 25 Bytes

Class 2: Compressed Bitmap Character Data

Class 3: Intellifont Scalable Character Contour Data — The Intellifont character data is organized as shown below. The data area is grayed.

Byte	15 - MSB 8	7 LSB - 0	Byte
0	Format (10)	Continuation[1]	1
2	Descriptor Size (2)	Class (3)	3
4	Contour Data Size		5
6	Metric Data Offset		7
8	Character Intellifont Data Offset		9
10	Contour Tree Offset		11
12	XY Data Offset		13
14	Tangent Data Offset ...		15
	Metric Data ...		
	Character Intellifont Data ...		
	Contour Tree Data ...		
	XY Coordinate Data ...		
	Reserved (0)	Checksum	

[1] Continuation is supported for classes 1, 2, 3, and 15 only.

Class 3: Intellifont Contour Character Data Format

Class 4: Intellifont Scalable Compound Character Data — A class 4 compound character is a combination or two or more different characters (e.g., accented characters). Continuation should be set to 0, since compound characters cannot be continued. The character data is grayed.

Byte	15 - MSB 8	7 LSB - 0
0	Format (10)	Continuation (0)[1]
2	Descriptor Size (2)	Class (4)
4	Compound Character Escapement	
6	Number of Components	
8	Component List (See the "Component List" description for the format) ...	
	Reserved (0)	Checksum

[1] Continuation is not supported for class 4.

Class 4: Intellifont Compound Character Descriptor and Data Format

Class 15: TrueType - Since all TrueType scalable characters are handed to the TrueType font scaler in the same format, this field does not provide vital information. For TrueType characters, set this field to 15.

Character Type (UBYTE)

DeskJets below 1200 only. Specifies the type of character.

Value	Class
0	Normal (single pass)
2	Pass 1 of multi-pass
3	Pass 2 of multi-pass
4	Pass 3 of multi-pass
5	Pass 4 of multi-pass

Orientation (UBYTE)

Bitmap fonts only. Specifies the orientation of the character. Character orientation must match the orientation in the font descriptor, as follows:

Value	Orientation
0	Portrait
1	Landscape
2	Reverse-portrait
3	Reverse-landscape

The character is discarded if the orientation is unsupported or different than font orientation.

DEVICE NOTE: LJ, LJ+, LJII support only 0,1.

Left Offset (SINT16)

Bitmap fonts only. 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 CAP.

DEVICE NOTE: The left-offset range is -16384 to 16384 on LJIIIs later, -4200 to 4200 on LJII, and -128 to 127 for LJ+. Values outside these ranges cause the character to be discarded.

Top Offset (SINT16)

Bitmap fonts only. Specifies the distance in dots from the reference point to the top 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 CAP.

DEVICE NOTE: The legal range is -16384 to 16384 on LJIIIs and later, -4200 to 4200 on LJII, and -128 to 127 for LJ+. Values outside these ranges cause the character to be discarded.

Character Width (UINT16)

Bitmap fonts only. Specifies the width of the character in dots on the physical coordinate system (i.e., this value is orientation dependent). Generally, this width is from the farthest left black dot to the farthest right black dot.

DEVICE NOTE: The legal range for character width is 1 to 16384 on LJIII and LJ2000, 1 to 4200 on LJII, and 1 to 128 for LJ+. Values outside these ranges cause the character to be discarded.

Character Height (UINT16)

Bitmap fonts only. Specifies the height of the character in dots on the physical coordinate system (i.e., this value is orientation dependent).

DEVICE NOTE: The legal range is 1 to 16384 on LJIIIs and later, 1 to 4200 on LJII, and 1 to 128 for LJ+. Values outside these ranges cause the character to be discarded.

Delta X (SINT16)

Bitmap fonts only. Specifies the number of quarter-dot units (radix dots) by which the horizontal position within the PCL logical page coordinate system is incremented after printing the character. If the value field is negative, the value is set to 0. This value is used by the printer only when the font is proportionally spaced.

DEVICE NOTE: The legal range is -32768 to 32767 on LJIIIs and later, and 0 to 16800 on LJII and LJ2000. Values outside these ranges cause the character to be discarded.

Width (UBYTE)

DeskJet 5xx (except 540) only. This field holds the normal width of the character (in 600ths of an inch). The normal width of a character can take on values from 0 to 254. A value of 255 specifies that the character is a non-printing, non-spacing character. Non-printing, non-spacing characters print a blank space in transparent print mode and display functions mode.

Compressed Width (UBYTE)

DeskJet 5xx (except 540) only. If the font supports algorithmic compressed printing, this field holds the compressed width of the character (in 1/600ths of an inch). If the width of the character is equal to 255 (designating a non-spacing, non-printing character), this value is also 255. In landscape fonts, this field is the left pad.

Left PS PAD (UBYTE)

DeskJet 5xx (except 540) only. If the font is proportionally spaced, this field holds the left pad value of the character (in 600ths of an inch). The value can range from 0 to 255 or -127 to +127.

DEVICE NOTE: This field is a UBYTE for DeskJet Classic and Plus, but an SBYTE for all other DeskJets in the 5xx series, except the 540, which does not use it.

Right PS PAD (UBYTE)

DeskJet 5xx (except 540) only. If the font is proportionally spaced, this field holds the right pad value of the character (in 600ths of an inch). The value ranges from 0 to 255 or -127 to +127.

DEVICE NOTE: This field is a UBYTE for DeskJet Classic and Plus, but an SBYTE for all other DeskJets in the 5xx series, except the 540, which does not use it.

Character Data

The character data is in the format specified by the class field.

Contour Data Size (UINT16)

Intellifont only. The size of the contour data including this size field.

Metric Data Offset (SINT16)

Intellifont only. The offset to the Metric Data relative to the Contour Data Size field address.

Character Intellifont Data Offset (SINT16)

Intellifont only. The offset to the Character Intellifont Data relative to the address of the Contour Data Size field.

Contour Tree Offset (SINT16)

Intellifont only. The offset to the contour Tree Data relative to the address of the Contour Data Size field.

XY Data Offset (SINT16)

Intellifont only. The offset to the XY data relative to the address of the Contour Data Size field.

Tangent Data Offset (SINT16)

Intellifont only. The offset to the Tangent Data relative to the address of the Contour Data Size field. This field should be set to -1. This field appears in HQ2 characters but not in HQ3 characters. HQ2 and HQ3 formats are described in *Intellifont SubSystem Specification 2.0 version 6*.

Metric Data

Intellifont only. See *Intellifont SubSystem Specification 2.0 version 6*.

Character Intellifont Data

Intellifont only. See *Intellifont SubSystem Specification 2.0 version 6*.

Contour Tree Data

Intellifont only. See *Intellifont SubSystem Specification 2.0 version 6*.

XY Coordinate Data

Intellifont only. See *Intellifont SubSystem Specification 2.0 version 6*.

Compound Character Escapement (SINT16)

Intellifont only. The escapement in dimensional units of a resulting compound character.

Number of Components (UBYTE)

Intellifont only. The number of components of the compound character.

Component List

Intellifont only. The component list contains a 6-byte record (shown below) for each compound character component. The number of 6-byte records is determined by the Number of Components field described above.

Byte	15 (MSB) 8	7 (LSB) 0	Byte
0	Character Code		1
2	X Offset		3
4	Y Offset		5

The *Character Code* is the character code number of a component of a compound character. *X Offset* is the offset of a component from the origin in the x direction in design window units (as defined in the Scale Factor field of the font descriptor).
Y Offset is the offset of a component in the y direction from the origin in design window units.

NOTE: The character code may be greater than the last code of the symbol set that is implied by the font type, since a compound character can include components that are not part of the symbol set.

Character Data Size (UINT16)

TrueType only. This value should equal the sum of the sizes of the Character Data Size, Glyph ID, and TrueType Glyph Data fields. It allows the PCL interpreter to know when a continuation block is needed. The minimum value is 4.

The value of Character Data Size plus Descriptor Size plus 4 (for the Format, Continuation, Reserved and Checksum bytes) is never less than the value in the Define Character command. If the sum is equal to the value in the command, no continuation block is expected for the given character; if the sum is greater, a continuation block is needed.

The validity of a downloaded TrueType character requires that the sum of the Define Character command values for all of that character's blocks equals the sum of Descriptor Size and Character Data Size plus 2 (for Reserved and Checksum), plus 2 times the number of character blocks (for Format and Continuation bytes).

Glyph ID (UINT16)

TrueType only. This field is used by the TrueType font scaler as an ID number for the glyph data associated with the given character.

TrueType Glyph Data

TrueType only. This field contains the data segment associated with the given character as found in the glyph table of the original TrueType font file. (See the *TrueType Font File Specification*).

Checksum (UBYTE)

Intellifont. The checksum of all the contour character data. The checksum is for all the blocks added together.

TrueType. The value of this byte, when added to the sum of all of the bytes in the Character Data Size, Glyph ID, and TrueType Glyph Data fields, should equal 0 (modulo 256). The Checksum is found in the last block for a given character.