

Chapter 12: Unbound Fonts and Downloaded Symbol Sets

Contents of this Chapter

- Introduction 12-1
- Unbound Font Operations 12-2
- Downloading Symbol Sets 12-5
- Managing User-Defined Symbol Sets..... 12-11

This chapter describes the following PCL commands:

Symbol Set Code.....	<i>Esc*c#R</i>	12-5
Download Symbol Set.....	<i>Esc(f#W[symbol set definition]</i>	12-6
Symbol Set Control	<i>Esc*c#S</i>	12-11

12.1 Introduction

Bound and Unbound Fonts

A **bound** font is restricted to a single symbol set. An **unbound** font contains the union of multiple symbol sets described by a Symbol Index such as Hewlett-Packard's Master Symbol List (MSL) or Unicode (TrueType). Unbound fonts typically contain 300, 400, or even more characters.

Downloading Symbol Sets

Symbol sets may be downloaded like fonts. Both customized and standard HP-supported symbol sets may be downloaded.

12.2 Unbound Font Operations

Font Selection and Unbound Fonts

When a font is requested, the printer selects an available font that most closely matches the current font selection characteristics — symbol set, spacing, pitch, height, style, stroke weight, and typeface. (See Chapter 9 for font selection.)

Since the symbol set attribute has the highest priority, the printer first makes a list of all the available fonts — bitmap, bound, and unbound — that contain the requested symbol set. Bitmap and bound fonts are easily checked because they contain only one symbol set. However, to determine which unbound fonts match a symbol set, the printer must identify the appropriate *symbol collections*.

Symbol Collections

The symbols in an unbound font can be divided into symbol collections that classify symbols according to language or application. The symbols in each collection do not change from one unbound font to another — that is, the Basic Latin collection always contains the same symbols. However, different fonts may contain different collections. For example, the internal Univers font in LaserJet IIIIP contains the Latin, Math, and Semi-graphic collections, which together contain all the symbols required for the 35 symbol sets that Univers provides. On the other hand, the Dingbats font contains only the Dingbats collection, which contains all the symbols required for the five supported Dingbat symbol sets.

Matching Unbound Fonts to Requested Symbol Sets

The *Character Complement* number, which is a 64-bit field in an unbound font definition (see Chapter 11), identifies the symbol collections contained in the font. Each bit in this field corresponds to a symbol collection. For example, if bit 63 is cleared, an unbound Intellifont contains the Basic Latin collection. If bit 31 is cleared, an unbound TrueType font contains ASCII.

The *Character Requirements* number, which is a 64-bit number provided to the printer for each symbol set, identifies the character collections needed by the symbol set. The printer matches bit-by-bit the Character Requirements number for the requested symbol set with the Character Complement number of every unbound font in the printer.

After matching the Character Complement number with the required symbol collections, the printer will contain a list of all fonts (bitmap, bound, and unbound) that support the requested symbol set. If no fonts are found for the requested symbol set, the printer continues with the font selection algorithm described in Chapter 9.

Summary of Symbol Set Matching

Symbol set matching for unbound fonts uses the following general procedure:

1. The host requests a font through font attribute selection by updating the font attribute table.
2. Since the symbol set attribute has the highest priority, the printer looks for the requested symbol set.
3. The printer uses the *Character Requirements* number of the requested symbol set to identify the symbol collections needed by the symbol set.
4. The printer checks the *Character Complement* field of available fonts for symbol collections that contain symbols in the requested symbol set.
5. Fonts whose symbol collections together contain the necessary symbols are included in the next round of attribute matching.

Symbol set or point size are not specified when an unbound scalable font is downloaded. Designation by ID is equivalent to a PCL font select string containing parameters with fixed values (spacing, stroke weight, style, and typeface), but not point size and symbol set. To designate an unbound scalable font by ID, first select the desired symbol set and point size, and then select the font by ID.

Symbol Indexes (MSL and Unicode)

Symbol Indexes identify HP printer symbols by a unique number. Two symbol indexes are used for unbound fonts. Unbound Intellifonts use MSL (Master Symbol List) numbers; unbound TrueType fonts use Unicode numbers. Character collections differ between Intellifont and TrueType fonts.

Since the printer identifies symbols by their Symbol Index Number (0-65535), but receives character codes (0-255), the printer has a symbol set mapping table for each available symbol set. Using this mapping, the printer identifies which indexed character will be printed for the requested character code. A partial mapping for Roman-8 set is shown below.

Character Code	MSL Number	Unicode Number (hex)
32	0000	(no HMI variable space)
33	0001	0x0021
34	0002	0x0022
35	0003	0x0023
36	0004	0x0024
37	0005	0x0025
38	0006	0x0026
39	0008	0x2019
40	0009	0x0028
41	0010	0x0029
...	...	
252	0189	0x25a0
253	0190	0x00bb
254	0191	0x00b1

Printing a Character

After an unbound font has been selected and the printer receives a character code for printing, the printer must access the mapping table to get the Symbol Index Number for that character. The printer then searches the selected unbound font for the correct MSL or Unicode number and prints that character.

For example, if Roman-8 is requested and the printer receives character code 254, the printer accesses the Roman-8 mapping table. In the Roman-8 mapping table, character code 254 is mapped to MSL number 191 or Unicode number 0x00b1, corresponding to the plus-over-minus symbol, which is then printed.

12.3 Downloading Symbol Sets

Symbol sets may be downloaded and used to bind an unbound font. The downloaded symbol set may be one of the standard HP-supported symbol sets listed in Chapter 9 that is not internal to the printer, or it may be a user-defined or modified symbol set. The following process is used to download a symbol set:

1. Specify an identification number for the symbol set — *Esc*c#R*.
2. Download the symbol set definition — *Esc(f#W [data]*.
3. Select the symbol set for printing — *Esc(ID*.
4. Delete the symbol set or designate it as permanent — *Esc*c#S*.

Symbol Set Identification: Before a symbol set can be downloaded or manipulated as an individual entity, it must be assigned a unique identification number. *Esc*c#R* designates an identification number prior to downloading the symbol set. An existing symbol set with this code is deleted at download.

Symbol Set Definition: *Esc(f#W* downloads a group of symbol set attributes as well as a list that maps each symbol set character code to a character ID number by which the given character may be specified in unbound scalable fonts.

Symbol Set Selection: After downloading the symbol set definition, the user may select the symbol set by *Esc(ID*.

Symbol Set Management: Once a user-defined symbol set is downloaded, the Symbol Set Control command (*Esc*c#S*) can assign symbol sets as temporary or permanent, or delete them.

Symbol Set Code *Esc * c # r/R*

Assigns an identification code to a downloadable symbol set.

Value(#) = Identification code
 Default = 0
 Range = 0 to 65535

DEVICE NOTE: LJIIIP and 4 have an upper range of 32767, corresponding to 1023Z. They do not allow values above 1023Z for user-defined symbol sets.

The Symbol Set Code is analogous to Font ID and Character Code. It is used to download and manage symbol sets.

Any downloaded symbol set already associated with this code is deleted when the symbol set definition is received.

The value (#) used for this command must match the Encoded Symbol Set Designator field in the downloaded symbol set definition.

The Symbol Set Code is derived from the identification number (ID) used by *Esc(ID)* in font selection:

$$\text{Symbol Set Code} = (\# * 32) + (L - 64)$$

where # is the number portion of the ID, and L is the ASCII value of the letter portion. (See Chapter 9 for a list of symbol set IDs.)

EXAMPLE

Assume that a symbol set has an ID of 1Q, which will be used as a font select parameter by *Esc(ID)* — that is, *Esc(1Q)*. The Symbol Set Code command for this symbol set would be *Esc*c49R* — that is, $(1 * 32) + (81 - 64) = 49$.

Then the Download Symbol Set command, *Esc(#W)*, will create a symbol set with a symbol set code of 49. And the Symbol Set Control command (*Esc*c#S*) will execute the specified action for any symbol set with a symbol set code of 49.

Download Symbol Set *Esc (f # W [symbol set definition]*

Defines the characters and character mapping for a downloaded symbol set.

Value(#) = Number of bytes in symbol set definition
 Default = NA
 Range = 0 to 32767 (command ignored if invalid definition or out-of-memory)

This command must be sent subsequent to Symbol Set Code (*Esc*c#R*). The last symbol set code sent is used; if no code has been sent, the default (0) is assigned.

A previously downloaded symbol set with the same symbol set code is deleted. An internal symbol set with the same code is overridden by the new symbol set.

If the symbol set definition is invalid, if the Encoded Symbol Set Designator field of the definition does not match the symbol set code, or if there is insufficient memory to create the symbol set, the command is ignored and the symbol set discarded.

The format for a downloaded symbol set definition is shown below:

Byte	15 (MSB)	8	7	0 (LSB)	Byte
0	Header Size				1
2	Encoded Symbol Set Designator				3
4	Format		Symbol Set Type		5
6	First Code				7
8	Last Code				9
10 to 16	Character Requirements				11 to 17
#-2	Symbol Map [Last Code - First Code +1]				#-1
	...				

User-defined Symbol Set Definition Format

Header Size (UINT16)

Specifies the number of bytes in the header of the symbol set definition. This is the number of bytes preceding the Symbol Map.

For a format 1 (MSL) symbol set definition, the header size must be 18 or greater; otherwise, the symbol set is invalidated.

Encoded Symbol Set Designator (UINT16)

This field must match the value designated by Symbol Set Code (*Esc*c#R*).

Format (UBYTE)

This field specifies the symbol index and format of the symbol set definition:

Value	Format
1	MSL (Intellifont)
3	Unicode (TrueType)

Unrecognized values invalidate symbol set creation.

DEVICE NOTE: LJIIIIP recognizes only a format of 1.

Symbol set Type (UBYTE)

Defines printable and unprintable codes for the symbol set:

Value	Symbol Set Organization
0	7-bit (32-127 are printable) *
1	8-bit (32-127 and 160-255 are printable) *
2	PC-8 (All codes are printable except 0, 7 - 15, and 27) *

* All character code positions print in transparency mode.

First Code (UINT16)

Designates the first character code in the set. In a Format 1 or 3 symbol set, this value can be 0 to 255. The symbol set is invalid unless the First Code is less than or equal to the Last Code.

Last Code (UINT16)

Designates the last character code in the set. The value must be between 0 and 255 inclusive, and must not be smaller than the First Code.

Together, the First Code through the Last Code identify the range of character codes that map to the Symbol Index numbers (characters) in the Symbol Map field.

Character Requirements (Array of UBYTE)

This 8-byte field, in conjunction with the Character Complement field in the unbound font definition, determines the compatibility of the unbound font to a symbol set. Each bit in the field represents a specific collection of symbols. Setting a bit to 1 indicates that collection is required; setting the bit to 0 indicates that collection is not required. (Bit 63 is the most significant bit of the first byte; bit 0 is the least significant bit of the 8-byte field.) A font and a symbol set are compatible only if the result of AND'ing the Character Complement field of the font definition with the Character Requirements field of the symbol set definition is 8 bytes of zero.

As described below, Character Requirements differ between MSL-based symbol sets and Unicode-based symbol sets. Unbound Intellifonts are ordered in MSL numbers; unbound TrueType fonts are ordered in Unicode numbers.

MSL Symbol Index Character Requirements

Bit Field	Designated Use
58-63	Reserved for Latin fonts
55-57	Reserved for Cyrillic fonts
52-54	Reserved for Arabic fonts
50-51	Reserved for Greek fonts
48-49	Reserved for Hebrew fonts
3-47	Miscellaneous Uses (South Asian fonts, East Asian fonts, Armenian, Georgian, Amharic, other alphabets, bar codes, OCR, Math, PC Semi-graphics, Dingbats, etc.)
0-2	Symbol Index Indicator

Individually defined bits include:

Bit	Value	Meaning
63	1	Basic Latin required (such as Roman-8 and ISO8859/1 Latin 1)
	0	Basic Latin not required
62	1	East European Latin required (such as ISO8859/2 Latin 2);
	0	East European Latin not required
61	1	Turkish required (such as ISO 8859/9 Latin 5)
	0	Turkish not required
57	1	Cyrillic required (such as ISO 8859 Latin/Cyrillic)
	0	Cyrillic not required
34	1	Math required (such as Math-8)
	0	Math not required
33	1	Semi-graphic required (such as PC-8 D/N)
	0	Semi-graphic not required
32	1	Dingbats required (such as ITC Zapf Dingbats series 100)
	0	Dingbats not required
2,1,0	000	MSL Symbol Index

Unicode Symbol Index Character Complements

Bit Field	Designated Use
28-31	Reserved for symbols in Latin fonts.
26-27	Reserved for accents and publishing symbols.
22-25	Reserved for application/environment specific symbols.
3-21	Reserved for miscellaneous uses.
0-2	Reserved for Symbol Index indicator

Individually defined bits include:

Bit	Value	Meaning
31	1	ASCII symbols required (various definitions).
	0	No basic Latin alphabet required.
30	1	Latin 1 extensions required.
	0	No West European added symbols required.
29	1	Latin 2 extensions required.
	0	No East European added symbols required.
28	1	Latin 5 extensions required.
	0	No Turkish and West European added symbols required.
27	1	Publishing symbols required.
	0	No added publishing symbols required.
26	1	Accents (often missing from some sets) required.
	0	No added accents required.
25	1	PCL (Roman-8, Legal, etc.) required.
	0	No added symbols for the above required.
24	1	Macintosh symbols required.
	0	No added symbols for MacIntosh text required.
23	1	PostScript required
	0	No added symbols for PostScript text required.
22	1	Code page required.
	0	No added symbols for PC-8, PC-8 D/N required.
2,1,0	001	Unicode symbol index indicator.

There are no invalid Character Requirements field values. Examples of values are:

Value (Hex)	Meaning
0x0000000000000000	Default requirement (MSL); symbol set can be used with any typeface indexed by MSL.
0x8000000000000000	Symbol set (MSL) requires only the Basic Latin Symbol Collection. (e.g., Roman-8)
0x8000000400000000	Symbol set (MSL) requires both Latin and math characters. (e.g., Math-8)
0x8000000200000000	Symbol set (MSL) requires both Latin and PC semi-graphic characters. (e.g., PC-8)
0x0000000100000000	Symbol set (MSL) requires only the Dingbat Collection.
0x0000000000000001	Default requirement (Unicode); symbol set can be used with typeface indexed by Unicode.
0x00000000a0000001	Symbol set (Unicode) requires ASCII and East European collections (ISO 8859/2 Latin 2)
0x0000000088000001	Symbol set (Unicode) requires ASCII and publishing collections (e.g., Ventura US)

Symbol Map (Array of UINT16)

Maps each character code to a symbol index number. The number of symbol index characters in the array must match the number of character codes in the range, First Code through Last Code. If no printable symbol is associated with a given character code (e.g., codes 128 through 160 of Roman-8), the corresponding entry in the Symbol Map is 65535 (0xffff).

12.4 Managing User-Defined Symbol Sets

Symbol set management involves the same types of operations as font management:

- Designating a symbol set temporary or permanent
- Deleting a symbol set

Symbol set management operations are performed on the symbol set designated by the current symbol set code (*Esc*c#R*).

Symbol Set Control *Esc * c # s/S*

Designates user-defined symbol sets as permanent or temporary, or deletes them.

Value(#)	=	0	Delete all temporary and permanent user-defined symbol sets
	=	1	Delete all temporary user-defined symbol sets
	=	2	Delete current user-defined symbol set (last Symbol Set Code specified)
	=	4	Make current user-defined symbol set temporary
	=	5	Make current user-defined symbol set permanent
Default	=	NA	
Range	=	0 to 2, 4, 5	

A downloaded symbol set is temporary by default. Internal symbol sets cannot be deleted or made temporary; however, a downloaded symbol set can *overlay* (redefine) an internal symbol set, but this is not recommended. An overlaying symbol set must be deleted to access an overlaid ROM-based symbol set. The priority scheme is (highest priority to lowest):

1. Downloaded symbols set (lowest ID)
2. Read/Write removeable disk (lowest ID)
3. Read/Write removeable flash (lowest ID)
4. Read/Write permanent disk (lowest ID)
5. Read/Write permanent flash (lowest ID)
6. Cartridge (lowest unit to highest unit)¹
7. SIMM (lowest ID)
8. Internal

¹ DEVICE NOTE: On LJIII the left cartridge has priority over the right cartridge. On DeskJets below 1200, the back cartridge has priority over the front.

