9 Font Management

Introduction

Font management provides mechanisms for manipulating soft fonts. It provides the means for controlling which soft fonts are saved in user memory (RAM) or deleted. This is accomplished by assigning a font as either temporary or permanent, or deleting a soft font. In addition, font management includes the command for assigning ID numbers to RAM fonts. It also provides a mechanism for copying ROM fonts (internal, cartridge, or SIMM) to RAM for the purpose of assigning ID numbers.

Downloading Soft Fonts

The process of transferring soft fonts from a host computer to the printer's user memory (RAM) is called downloading. Designate a unique identification (ID) number prior to the download of a font. This number is then associated with the soft font. This number is assigned using the Font ID command, described later in this chapter. Subsequent manipulation of the soft font is accomplished using the font's ID number. If a font is already associated with this ID number in the printer, the existing font is deleted during the download.

Several commands are required to define a font before downloading it to the printer. These commands are described in detail in Chapter 11. Hewlett-Packard font files include the necessary commands that define the symbols of a font. Assigning a font ID number and then copying the font file to the printer downloads the font. Scalable fonts and typefaces may be prepared by font management software such as HP's Type Director. Once prepared, scalable fonts are downloaded in much the same manner as bitmap fonts.

Once downloaded, a soft font occupies a portion of user memory (RAM). The number of soft fonts that can be stored in user memory is limited only by the amount of available user memory.

Temporary vs. Permanent Fonts

Once downloaded, a font is automatically designated as temporary. A temporary soft font is deleted from user memory during a printer reset or when a Typeface List, a Font Printout or a self-test is performed from the printer's control panel. A soft font can be designated as permanent to prevent the printer from deleting it during a printer reset. A soft font is designated as temporary or permanent by referencing its ID number and using the Font Control command (refer to "Font Control Command" later in this chapter).

Note

Both temporary and permanent fonts are deleted from user memory whenever the printer's power is turned off.

Switching printer languages ("personalities"), changing resolution, or changing the Page Protection setting also deletes temporary and permanent fonts from user memory in some printers (refer to the appropriate printer *User's Manual* for specifics).

An existing font is deleted when a new font with the same ID number is downloaded. The new font replaces the existing font (whether temporary or permanent).

Deleting Fonts

There are several mechanisms provided by PCL font management that delete soft fonts from user memory. These include commands to delete all soft fonts, all temporary soft fonts, or an individual soft font by reference to its font ID number (refer to the Font ID and the Font Control commands described on the following pages).

Font ID Command

The Font ID command is used to specify an ID number for use in subsequent font management commands. The ID number of a font can be used to select the font for printing (refer to “Font Selection by ID” in Chapter 8).

$^{E}_{C}$ ***c#D**

\# = ID number

Default = 0

Range = 0 - 32767

The font ID number is used during subsequent soft font downloads, selections or deletions.

The factory default font ID is 0 (if no Font ID command is sent, an ID of 0 is assigned).

Note The font number assigned by the printer and used from the printer’s control panel is not the same as the ID number assigned using the Font ID Command.

Example

To specify a font ID number of 1, send:

$^{E}_{C}$***c1D**

Font Control Command

The Font Control command provides mechanisms for manipulating soft fonts.

$^{E}_{C}$ *** c # F**

\# = 0 - Delete all soft fonts
1 - Delete all temporary soft fonts
2 - Delete soft font (last ID specified)
3 - Delete Character Code (last ID and Character Code specified)
4 - Make soft font temporary (last ID specified)
5 - Make soft font permanent (last ID specified)
6 - Copy/Assign current invoked font as temporary (last ID specified)

Default = N/A

Range = 0 - 6 (values outside range are ignored)

Note If the primary or secondary font is deleted, a new primary or secondary font is selected automatically from the remaining fonts.

Examples

To remove all soft fonts from user memory, send:

$^{E}_{C}$ ***c0F**

To remove only those soft fonts that are temporary, send:

$^{E}_{C}$ ***c1F**

To delete the soft font with an ID of 1, send:

$^{E}_{C}$ ***c1d2F**

To delete the character “p” (112 decimal) in a bitmap or bound scalable font with an ID of 1, send:

$^{E}_{C}$ ***c1d112e3F**

(A space is printed in place of the deleted character. Also, the $^{E}_{C}$***c#E** Character Code command used in the above sequence “...112e...,” is described in Chapter 11.)

To make the soft font with an ID of 2 temporary, send:

$^{E}_{C}$ ***c2d4F**

To make the soft font with an ID of 2 permanent, send:

$^{E}_{C}$ ***c2d5F**

To make a copy of the currently invoked (selected) font, with an ID of 9, send:

$^{E}_{C}$ ***c9d6F**

The Copy/Assign font control feature can be used to copy either ROM or RAM fonts into RAM assigning them ID numbers.

Note

When the currently selected font is a scalable TrueType ROM font, $^{E}_{C}$*c#d6F **assigns** a font ID number, but makes **no copy** of the font in RAM. Any attempts to download or delete characters within the font are ignored. An attempt to delete the font merely results in the loss of the ID number.

Font Management Example

This example illustrates several typical font management operations. It assumes a bitmap soft font is stored and available on an MS-DOS based hard disk.

1 Set the font ID number to 2:

$^{E}_{C}$***c2D**

2 Download a soft font file using the MS-DOS COPY command with the /B option:

COPY /B *filename* PRN

Note that the soft font is associated with font ID 2.

3 Make the soft font permanent to prevent its deletion during a printer reset:

$^{E}_{C}$***c5F**

4 Designate the permanent soft font as primary:

$^{E}_{C}$**(2X**

Unbound Scalable Fonts

Prior to introduction of the HP LaserJet IIIP printer, a downloaded scalable font was restricted to a single symbol set. Now scalable fonts with no symbol set affiliation can be downloaded. These new fonts are called unbound fonts.

To download unbound fonts, the “PCL Font Header for Intellifont Unbound Scalable Fonts” or the “Format 15 Font Header for Scalable Fonts” (TrueType) must be used (see Chapter 11).

Bound and Unbound Fonts

The terms “bound” and “unbound” refer to the symbol set capacity of a font. A bound font identifies a font which is restricted (bound) to a single symbol set. An unbound font (or unbound *typeface*) indicates the capacity to be bound to a set of symbols selected from a complementary **symbol index** (such as the Master Symbol List (**MSL**), or the **Unicode** symbol index).

Font Selection and Unbound Fonts

When a font is requested for printing, the printer selects a font which most closely matches the current font selection characteristics (symbol set, spacing, pitch, height, style, stroke weight, and typeface). Refer to “Summary of Font Selection by Characteristics” in Chapter 8 for detailed font characteristic selection information.

Since symbol set is the highest font selection priority and typeface is the lowest, the printer searches for the symbol set first. A list of all fonts that match the requested symbol set is made. This includes bitmap, bound, and unbound fonts. Since bitmap and bound scalable fonts contain only one symbol set, they can be easily identified. However, determining which unbound fonts match a symbol set is more complex. Symbol set compatibility for unbound fonts is determined by identification of groups of symbols referred to as symbol collections.

Symbol Collections

The symbols in an unbound font (typeface) can be divided into symbol collections. These symbol collections identify the symbols according to some language basis or special application usage. Some symbol collections include: Basic Latin, East European, Turkish, Math, Semi-Graphic, and Dingbats. If a symbol collection is included in an unbound font, all of the symbols of that collection are included.

Note See Appendix D in the *PCL 5 Comparison Guide* to identify symbols in the various collections.

The symbols within a symbol collection do not change from one unbound font to the next. For example, the Basic Latin collection always contains the same symbols. Different fonts may contain different symbol collections. For example, the internal Univers typeface contains the Latin, Math, and Semi-Graphic collections (these collections contain all the symbols required for the 35 symbol sets that Univers supports). The ITC Zapf Dingbats typeface, on the other hand, contains only the Dingbats collection, which includes all the symbols required for the five supported symbol sets.

When searching unbound scalable fonts (during font selection) for those that match the requested symbol set, the printer actually searches for symbol collections. To identify symbol collections which meet the needs of the requested symbol set, the printer uses two numbers: the **Character Requirements** number and the **Character Complement** number.

Character Complement Numbers

The “Intellifont Unbound Scalable Font Header” (header) includes a 64 bit field (bytes 78-85) which contains the Character Complement number. For TrueType fonts, in the “Format 15 Font Header for Scalable Fonts” (unbound), the Character Complement number is included in the accompanying “Segmented Font Data” section of the header.

The Character Complement number identifies the symbol collections in the font. Each bit in this field corresponds to a symbol collection (not all bits are currently defined; refer to Appendix D in the *PCL 5 Comparison Guide*).

Intellifont example: If bits 63 and 34 are cleared (set to zero) it indicates that the unbound font contains the Basic Latin (bit 63) and Math (bit 34) symbol collections and that the character index is in HP’s MSL numbers (bit 0).

TrueType example: If bits 31, 30, and 0 are cleared (set to zero), it indicates that the unbound font contains ASCII, Latin 1 extensions and is based on Unicode numbers.

Character Requirements Number

The other number the printer uses to determine symbol set compatibility, the Character Requirements number, is provided as part of the information contained in the symbol set. The Character Requirements number is a 64-bit number analogous to the Character Complement number; however, it identifies the symbol collections needed by the symbol set.

Intellifont example: If a symbol set based on HP’s MSL numbers requires one or more characters from the standard Latin collection and some of the characters from the Math collection then bits 63 and 34 are set to one (refer to Appendix D in the *PCL 5 Comparison Guide* for information regarding the various symbol collections).

TrueType example: If a symbol set based on Unicode numbers requires one or more characters from the standard ASCII collection and some of the characters from the Latin 5 collection, then bits 31, 28 and 0 are set to one (refer to Appendix D in the *PCL 5 Comparison Guide* for information regarding the various symbol collections).

As stated above, to determine which unbound scalable fonts contain the symbols for the specific symbol set, the printer must identify those unbound fonts that contain the symbol collections of the requested symbol set. To do this, the printer accesses the Character Requirements number for the requested symbol set. If, for example, the Roman-8 symbol set was requested, the printer would access the Character Requirements number from the Roman-8 symbol set information in the printer. This number is then compared with the Character Complement number of each unbound font in the printer. If any matches are found, those unbound fonts are included in the list of potential fonts for selection.

Final Font Selection

After the process above is complete, the printer contains a list of all fonts (bitmap, bound, and unbound) which support the requested symbol set. (If no fonts are found for the specified symbol set, Roman-8 is used. If more than one font remains, the printer continues comparing font selection characteristics, eliminating fonts, until only one remains. Whenever only one font remains, it is selected for printing.

Symbol Set Mapping Table

The printer receives character codes in the range 0-255 which, depending on the selected symbol set, identify the symbols to print or control codes to execute.

There are hundreds of symbols available in unbound fonts in HP LaserJet printers, more than can be identified by the character code range (0 - 255). A list of these symbols is provided in the **symbol indexes**, such as the Master Symbol List (MSL) and the Unicode list In Appendix D of the *PCL 5 Comparison Guide*). Each symbol in the list is identified by a unique MSL or Unicode number. Symbols in unbound fonts are identified by this number.

Since the printer identifies symbols by their symbol index number (range from 0 to 65535), but receives character codes (range 0-255), a relation must be made between the character codes and the larger range of symbol index numbers. This relation is defined by the symbol set mapping table.

The printer contains a symbol set mapping table for each available symbol set. These tables list the character code range and corresponding list of symbol index numbers. Using this mapping the printer identifies which indexed character is printed for the character code in the current symbol set.

A partial symbol set mapping table is shown in Table 9-1 for the Roman-8 symbol set.

Table 9-1 Roman-8 Symbol Index Mapping

Character Code	MSL Index (decimal)	Unicode Index (hexadecimal)
32	0	
33	1	0021
34	2	0022
35	3	0023
36	4	0024
37	5	0025
38	6	0026
39	8	2019
40	9	0028
41	10	0029
⋮	⋮	⋮
252	189	25a0
253	190	00bb
254	191	00b1

Printing a Character

When an unbound font is selected for printing and a character code is received, the printer accesses the requested symbol set mapping table to identify the MSL or Unicode number. For example, if the Roman-8 symbol set is selected and the printer receives character code 254, the printer accesses the Roman-8 symbol set mapping table (Table 9-1). In the Roman-8 mapping table, character code 254 is mapped to MSL number 191 or Unicode number 00b1 (plus-over-minus symbol). Thus, to print character code 254, the printer searches the selected unbound font for the correct MSL or Unicode number and prints that character.