PFM Files & Device Font Formats

Author:		Format:	**MS Word 97**
File:	**PCL 6 PFM Format.doc**	Template:	**hp_doc.dot**
Directory:		Creation Date:	**Feb 4, 1999**
		Last Saved:	**Apr 9, 1999 -**

Latest Revision: 0.03

Rev	Revisions	Date	Modified By
0.01	Initial Draft	02/04/99	
0.02	Updated ROM XL selection string requirements	02/17/99	
0.03	Added the requirement of the win3.1 PCM/PFM Sepcification	04/09/99	

This Page Intentionally Left Blank.

Table of Contents

1. Introduction

The purpose of this document is to describe the format of the enhanced/multi-pdl/PCL 6 PFM file format, and the ROM font formats that are supported by both the PCL 5 and PCL 6 (XL) languages. Given the direction of our language strategy (PCL 6), we want to make sure that device font vendors are enabling this language, while maintaining support for PCL 5.

1.1 PFM Format

PFM stands for Printer Font Metrics, and a PFM describes the characteristics of a device font, typically on DIMM, SIMM, or cartridge. It is not the intent of this document to describe each element of the PFM file, rather to briefly describe the differences of a PFM file that supports the standard PCL 5 printer definition language (PDL), and one that supports both PCL 5 and PCL6 (XL). The only difference in the two file formats is the information contained in the PDL specific selection string.

1.2 ROM Entity Formats

The second purpose of this document is to point out the ROM entity formats that are supported by both the PCL 5 and XL languages. XL supports a subset of the ROM entity formats that PCL 5 supports, and it is beneficial that font DIMM vendors understand this so that their DIMM fonts can be selected from either a PCL 5 driver or a PCL 6 driver.

2. Standard PFM file format

The standard PFM file (as listed in the Windows 3.1 DDK PFM Specification) has the following generic form:

PFMHEADER	Header;	*// Font Header*
WORD	WidthTable[];	*// Character Widths*
PFMEXTENSION	Extensions;	*// Extensions*
Char	DeviceName[];	*// Printer-Device Na*me
Char	FaceName;	*// Font Name*
EXTTEXTMETRIC	ExtTextMetrics;	*//Extended-Text Metrics*
WORD	ExtentTable[];	*// Unscaled-Character Widths*
DRIVERINFO	DriverInfo;	*// Driver-Specific Information*
PAIRKERN	KerningPairs[];	*// Pair-Kerning Ta*ble
KERNTRACK	KerningTracks[];	*// Track-Kerning Table*

Every PFM file that represents a device font in a printer must at least have the **PFMHEADER** structure, the **PFMEXTENSION** structure, **DRIVERINFO** structure, and the font name. This document is most conerned with the **DRIVERINFO** structure, more specifically the **epEscape** member of this structure.

The **DRIVERINFO.epEscape** member is a **DWORD** value that is offset from the top of the PFM file to an escape string that is used to select the font from the DIMM,SIMM, or cartridge.

All PFM files created on or before this document was written, were created to support a PCL 5 driver. That is, the selection string contains an escape sequence unique to PCL 5 that is used to select the font from the device.

2.1 PCM and PFM File Format Requirements

The new HP drivers will **only** support PCM and PFM files that adhere to the specifications listed in the Windows 3.1 DDK. The new Windows 95 DDK PFM/PCM specifications have not yet been adopted by the new HP drivers.

Your PCM and PFM files must adhere to the Windows 3.1 DDK specifications in order to be read in and usable by the new HP drivers.

3. Enhanced /PCL6 PFM file format

The standard PFM file format remains unchanged as described in the previous section, with one minor addition/modification: the string that the **DRIVERINFO.epEscape** points to could be PCL 6 selection string, or a selection string that supports both PCL 5 and PCL 6. This slight modification is described below.

Here are the following possible formats of the **DRIVERINFO.epEscape** selection strings:

1. If the PFM is PCL5 specific and only supports PCL5, then the **DRIVERINFO.epEscape** remains unchanged, and this member points to the PCL5 escape (←) string. (e.g. "←(9Y←(s1#blahblah77T")

2. If the PFM is XL specific and only supports XL, then the **DRIVERINFO.epEscape** remains unchanged, and this member points to the XL escape string that is **!** (bang) separated. (e.g. "**!**19U**!**MT SomeFont **!**Monotype Some Font")

   -Where the format of the bang separated string is: "**!**symbol set id**!**xl font selection string**!**system font name"

   Symbol set id - This is the alpha-numeric symbol set id (e.g. 19U, 581L)

   XL font selection string - This is typically a 16byte (space padded) string. HP internal printer fonts are setup to be selected by a 16byte string, but other DIMM vendors do not have this requirement. The selection string can be any length.

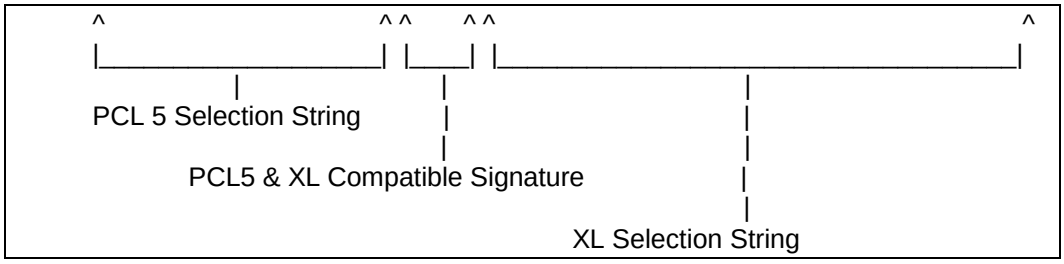
   System font name - This is the same value as the system font name that is specified in the **PFMHEADER.dfFace** structure member

3. If the PFM supports both languages, then the **DRIVERINFO.epEscape** will have an additional 4 byte binary signature to follow the string NULL terminator, followed by the other PDL escape string. (The quotes signify a NULL terminated string, and are not part of the selection string) For example:

   DRIVERINFO.epEscape = "1st SelectionString"**~XL~**"2nd PDL Selection String"

   DRIVERINFO.epEscape =

   "←(9Y←(s1#blahblah77T"**~XL~**"**!**19U**!**MT SomeFont **!**Monotype Some Font"



The example above could be flipflopped so that the default PDL selection string is XL, and it would also support PCL5 provided the **~XL~** signature is present, followed by the PCL5 selection string.

DRIVERINFO.epEscape =

"**!**19U**!**MT Sorts **!**Monotype Sorts"**~XL~**"←(9Y←(s1#8skdfjs77T"

This new format enables the PFM file to support both PCL5 and XL, while maintaining the old PCL5 PFM format, so that PFM file utilities (PFM Editor and legacy drivers) would be able to read in the newly formatted PFM files. It is up to the font DIMM vendors to determine which PDL will be the default, by specifying which selection string comes first. For those vendors that have been supporting PCL5, it is in their best interest to keep the PCL5 string the first selection string in the new format.

The steps required to add PCL6 driver support to their existing PCL5 PFM files are:

1. Locate the end of the selection string (NULL terminator) in the existing PFM file
2. Append the **~XL~** (4 byte binary) signature
3. Append the PCL6 NULL terminated selection string
4. Update the **PFMHEADER.dfSize** member so that it accurately represents the new file size

The next major step in providing PCL6 support to a device font, is to make sure the ROM Entity format is compatible with PCL5 and XL.

4. ROM Entity Formats

A ROM entity font that supports PCL5 doesn't necessarily support XL. XL supports a subset of the ROM entity formats that are supported by PCL5. For a device font to be supported by both PCL5 and XL, it must be one of the following formats:

1. One of the TrueType 300 or 305 Entity Formats (16 byte space padded XL selection string)
2. Infinifont 400 Level Entity Format (16 byte space padded XL selection string)
3. HP SX Bitmap ROM format (No limit on XL selection string length)

These 3 formats are the only formats that are supported by both XL and PCL5. The specifications for these formats are available from the firmware teams.