

**Microsoft<sub>R</sub> Windows**

**Version 2.1**

---

**Technical Notes on the PCL Driver**

Microsoft Corporation

## **CONTENTS**

---

|          |  |            |
|----------|--|------------|
| <b>1</b> | <b>INTRODUCTION</b>                          | <b>1-3</b> |
| 1.1      | Version 3.0 Enhancements                     | 1-3        |
| 1.2      | Version 3.1 Enhancements                     | 1-3        |
| 1.3      | Version 3.2 Enhancements                     | 1-4        |
| <b>2</b> | <b>WIN.INI FLAGS</b>                         | <b>2-3</b> |
| 2.1      | Summary of Flags                             | 2-3        |
| 2.2      | Details on Flags                             | 2-4        |
| <b>3</b> | <b>PERMANENT SOFT FONTS</b>                  | <b>3-3</b> |
| 3.1      | Setting Up Fonts for Download                | 3-3        |
| 3.2      | Tracking Permanent Fonts in the WIN.INI File | 3-7        |
| <b>4</b> | <b>SFINSTAL.DIR</b>                          | <b>4-3</b> |
| 4.1      | Logical Drive Definition                     | 4-3        |
| 4.2      | Font Family Definition                       | 4-4        |
| 4.3      | Installing Without the SFINSTAL.DIR File     | 4-7        |
| 4.4      | Sample SFINSTAL.DIR Directory File           | 4-8        |
| <b>5</b> | <b>DEVELOPERS' TOOLS</b>                     | <b>5-3</b> |
| 5.1      | Build SFINSTAL.DIR File                      | 5-3        |
| 5.2      | Add Fonts                                    | 5-4        |
| 5.3      | Enable Edit Button                           | 5-6        |
| <b>6</b> | <b>.PFM GENERATOR</b>                        | <b>6-3</b> |
| 6.1      | Installer PFM Versus Vendor-Supplied PFM     | 6-3        |
| 6.2      | File Naming Scheme                           | 6-3        |
| 6.3      | Re-generating .PFM Files                     | 6-3        |
| 6.4      | PFM Data From Font Data                      | 6-4        |

## **CONTENTS**

---

|          |  |            |
|----------|--|------------|
| <b>7</b> | <b>INSTALLER SCENARIOS</b>                             | <b>7-3</b> |
| 7.1      | Selecting Printer Fonts                                | 7-3        |
| 7.2      | Selecting Screen Fonts                                 | 7-5        |
| 7.3      | Recovering Soft Fonts From a WIN.INI File              | 7-6        |
| 7.4      | A Quick Method for Moving Fonts                        | 7-7        |
| 7.5      | How to Build a Floppy Disk Set of Fonts                | 7-9        |
| 7.6      | Setting Up Fonts on a Network                          | 7-10       |
| 7.7      | Setting Up .PFM Files for Resident and Cartridge Fonts | 7-12       |
| 7.8      | Sample SFINSTAL.DIR File                               | 7-15       |

# **CHAPTER 1**

## **INTRODUCTION**

### **CONTENTS**

---

|     |                          |     |
|-----|--------------------------|-----|
| 1.1 | Version 3.0 Enhancements | 1-3 |
| 1.2 | Version 3.1 Enhancements | 1-3 |
| 1.3 | Version 3.2 Enhancements | 1-4 |



# 1 INTRODUCTION

This document presents technical aspects of the version 3.2 Windows PCL / HP LaserJet printer driver. It should be read by technical support specialists, font vendors, Windows application developers, and advanced users. Because most of the issues with PCL printing under Windows involve fonts, most of this document is dedicated to a discussion of fonts and font management.

We assume the audience of this document has knowledge of the following:

- How to use the PCL driver's Soft Font Installer, documented in the *Microsoft Windows User's Guide*.
- How to build a Windows .PFM (Printer Font Metrics) file, documented in *Microsoft Windows Fonts Guide*.
- How to use DOS and a text editor.

Knowledge of the Soft Font Installer is the basic requirement for understanding Chapters 3 through 7. Knowledge of .PFM files is important for developers and font vendors reading the sections involving PFM file generation. Knowledge of DOS is important for executing the recommendations outlined in Chapter 7, "Installer Scenarios."

## 1.1 Version 3.0 Enhancements

Version 3.0 of the driver contained several enhancements over previous versions. Most important, it contained the Soft Font Installer, a utility that enables users to install and manage their PCL soft fonts in Windows without having to understand DOS or WIN.INI files.

Other enhancements to the driver were:

1. Additional printers: HP LaserJet 2000, NEC SilentWriter LC860, Tandy LP-1000, and Epson GQ-3500
2. Additional HP cartridges: S1, S2, and Z1a
3. Improved bitmap condensing code (replaces large blocks of white space with cursor moves)

## 1.2 Version 3.1 Enhancements

In addition to some minor bug fixes, the version 3.1 PCL driver included three primary enhancements:

1. Modification of the image area sizes to use the maximum possible printable image region on the page. This meant:
  - Overhauling the driver's banding mechanism to correct for software deficiencies that prevented previous versions from accessing the maximum possible printable region.
  - Removing the paper capabilities from the printer capabilities flag, *prtcaps*, which the driver writes to the WIN.INI file. The driver no longer uses the printer capabilities flag to determine available paper sizes; it uses a different, internal mechanism.
  - Taking advantage of the maximum possible print region for each printer. The image areas differ among the LaserJet Plus, LaserJet Series II, and the LaserJet 2000. Previous versions of the driver enforced the same image area for all three printer types. Now the user must select the correct printer, or else the driver will not correctly place the image area on the target printer.

2. Support of envelope feed. Because the image area change freed up several bits in the *prtcaps* flag in the WIN.INI file, two bits have been re-used for handling the envelope feed. This change is described in Section 2.2.12, "Prtcaps."
3. Support for three new printers: HP LaserJet IID, Olivetti LP 5000, and Toshiba PageLaser12. It also supports a new add-on card for the LaserJet, the Intel Visual Edge.

### **1.3 Version 3.2 Enhancements**

Here again, a few bugs were found and fixed after releasing the version 3.1 PCL driver. The major enhancement to this release has two aspects:

1. Giving the user the ability to add additional font support for cartridges by using the Soft Font Installer.
2. Providing an easy mechanism, with our new PFM Editor, for technicians with some font experience to create the information file needed by the driver to support the cartridge.

With the PFMEDIT utility, printer and font vendors will find it easy to create .PFM files by simply plugging in the relevant font data. They can then define all the fonts in a cartridge in multiple .PFM files and consolidate this information into one .PCM file.

The .PCM file represents one font cartridge and is recognized as such in the Soft Font Installer. When you choose to "Add" a cartridge, the Add listbox will indicate .PCM files with a "(c)" next to the font cartridge title.

Once the .PCM file is installed, you can "Exit" to the main device mode dialog box and see the cartridge listed at the end of the "Cartridges" listbox. The cartridge can now be selected in the same manner as the other cartridges listed in the box.

## **CHAPTER 2**

### **WIN.INI FLAGS**

#### **CONTENTS**

---

|        |                  |      |
|--------|------------------|------|
| 2.1    | Summary of Flags | 2-3  |
| 2.2    | Details on Flags | 2-4  |
| 2.2.1  | Cartindex $n$    | 2-4  |
| 2.2.2  | Cartridgen $n$   | 2-5  |
| 2.2.3  | Copies           | 2-5  |
| 2.2.4  | Duplex           | 2-6  |
| 2.2.5  | <Filename>       | 2-7  |
| 2.2.6  | FontSummary      | 2-7  |
| 2.2.7  | Fsvers           | 2-8  |
| 2.2.8  | MaxFontSummary   | 2-8  |
| 2.2.9  | Numcart          | 2-9  |
| 2.2.10 | Options          | 2-9  |
| 2.2.11 | Orient           | 2-11 |
| 2.2.12 | Paper            | 2-12 |
| 2.2.13 | Prtcaps          | 2-12 |
| 2.2.14 | Prtindex         | 2-15 |
| 2.2.15 | Prtresfac        | 2-16 |
| 2.2.16 | Smdir            | 2-16 |
| 2.2.17 | Sfdlbat          | 2-16 |
| 2.2.18 | Sfdlstyle        | 2-16 |
| 2.2.19 | SoftFont $n$     | 2-17 |
| 2.2.20 | SoftFonts        | 2-18 |
| 2.2.21 | Tray             | 2-19 |
| 2.2.22 | White_text       | 2-19 |



## 2 WIN.INI FLAGS

This section describes the flags that the PCL / HP LaserJet printer driver writes to and reads from the WIN.INI file. Most of these flags are controlled either directly or indirectly by the user via the driver-specific dialog for the Soft Font Installer. The user should have no need to modify these flags manually.

The *options* and *white\_text* flags are the only flags to which the user does not have access via a dialog. The user would have to edit the WIN.INI file to modify these flags.

The driver places and updates flags in the WIN.INI file via the routines provided by Windows. This means that:

1. The driver has no control over the order of entries in the WIN.INI file.
2. The driver cannot "delete" entries from the WIN.INI file.

The order of entries in the WIN.INI file is insignificant and has no effect on the operation of the driver. The driver cannot delete an entire entry line from WIN.INI; it can only remove everything to the right of the equal sign (=). The title text to the left of the equal sign remains there forever, or until the user manually removes it.

### 2.1 Summary of Flags

The WIN.INI flags are summarized below. A detailed description of each flag and its use is provided in the next section.

| Flag                    | Description   |
|-------------------------|---|
| <i>cartindexn</i>       | The cartridge(s) selected in the cartridge listbox  |
| <i>Cartridgen</i>       | <i>The name of a .PCM file.</i>   |
| <i>copies</i>           | The number of uncollated copies of each page  |
| <i>duplex</i>           | The duplex printing option (0=simplex)  |
| <i>&lt;filename&gt;</i> | Used by Soft Font Installer to track permanently downloaded font file names   |
| <i>FontSummary</i>      | The name of the font information file built by the driver   |
| <i>fsvers</i>           | The version of the FontSummary file   |
| <i>MaxFontSummary</i>   | The maximum allowable size (in Mb) of the FontSummary file  |
| <i>numcart</i>          | The number of cartridges the user has selected  |
| <i>options</i>          | On/off settings for printer rest, enable DP-TEK LaserPort, force load of soft font information, and allow vertical clipping |
| <i>orient</i>           | Printer orientation   |
| <i>paper</i>            | Paper size selection  |

| Flag              | Description  |
|-------------------|--|
| <i>prtcaps</i>    | A bit field representing the printer capabilities                                      |
| <i>prtindex</i>   | The index to the currently selected printer (includes memory option)                   |
| <i>prtresfac</i>  | The printer resolution factor (determines resolution in dpi)                           |
| <i>sfdir</i>      | The path to the directory containing soft fonts  |
| <i>sfdlbat</i>    | The path and file name of the batch file that downloads permanent fonts to the printer |
| <i>sfdlstyle</i>  | The manner in which permanent soft fonts should be downloaded                          |
| <i>SoftFontn</i>  | The soft font entry or entries in the WIN.INI file                                     |
| <i>SoftFonts</i>  | The number of soft fonts and cartridges listed in the WIN.INI file                     |
| <i>tray</i>       | The input paper tray   |
| <i>white_text</i> | Used for setting the driver's sensitivity to detecting white text                      |

## 2.2 Details on Flags

This section contains a detailed description of each flag and how to use it.

### 2.2.1 Cartindex*n*

This flag provides the cartridge selected in the cartridge listbox. The number of selectable cartridges depends on the printer. The driver is capable, though, of selecting up to eight cartridges at one time.

The flag for the first cartridge the user selects is:

`cartindex=n`

The flag for the second cartridge the user selects is:

`cartindex1=n`

Subsequent flags are *cartindex2*, *cartindex3*, *cartindex4*, *cartindex5*, *cartindex6*, and *cartindex7*. The number of flags in use (i.e., the number of cartridges the user has selected) is contained in the *numcarts* flag.

The cartridge index ("n" in the above examples) is a number representing the selected font cartridge. The values are as follows:

| WIN.INI Value | Cartridge Letter | Cartridge Description        |
|---------------|------------------|------------------------------|
| 0             |                  | None (no cartridge selected) |
| 1             | A                | Courier 1                    |
| 2             | B                | Tms Proportional 1           |
| 3             | C                | International 1              |
| 4             | D                | Prestige Elite               |
| 5             | E                | Letter Gothic                |
| 6             | F                | Tms Proportional 2           |
| 7             | G                | Legal Elite                  |
| 8             | H                | Legal Courier                |
| 9             | J                | Math Elite                   |
| 10            | K                | Math Tms                     |
| 11            | L                | Courier P&L                  |
| 12            | M                | Prestige Elite P&L           |
| 13            | N                | Letter Gothic P&L            |
| 14            | P                | Tms Rmn P&L                  |
| 15            | Q                | Memo 1                       |
| 16            | R                | Presentations 1              |
| 17            | S1               | Courier Document 1           |
| 18            | S2               | Tms/Helv Report 1            |
| 19            | T                | Tax 1                        |
| 20            | U                | Forms Portrait               |
| 21            | V                | Forms Landscape              |
| 22            | Y                | PC Courier 1                 |
| 23            | Z                | Microsoft 1                  |
| 24            | Z                | Microsoft 1A                 |

### 2.2.2 Cartridges

This flag is used in conjunction with the *cartindexn* flag. If another value is used in the *cartindexn* flag, the PCL driver looks for a cartridge<number> = PCMfile entry, which defines an external cartridge selected by the driver.

For example, cartindex=100 causes the driver to select the cartridge defined by the Cartridge100= line.

### 2.2.3 Copies

This flag provides the number of uncollated copies of each page that the driver should request from the printer. It may be a number between 1 and 99. "Uncollated copies" is a printer feature; the driver sends down one image of the page and requests the printer to repeat it for the requested number of copies. For example, if the user specifies two copies of a three page document, the driver will print two copies of page 1, two copies of page 2, and two copies of page 3 (i.e., the printing order would be pages 1, 1, 2, 2, 3, 3).

Collated copies (printing order 1, 2, 3, 1, 2, 3) is not a function of the driver. It is a function of the application. The application sends an image of the page to the printer for each copy requested, which is considerably slower than printing uncollated copies.

The driver always writes "*copies=1*" to the WIN.INI file, even if the user requested more than one copy. The driver remembers that the user requested more than one copy (via the driver's memory-resident device mode structure), but it does not write the actual number of copies requested to the WIN.INI file. The result is the driver will print the desired number of copies as long as the user stays in Windows. If the user exits Windows and re-enters, the driver reverts back to 1 copy.

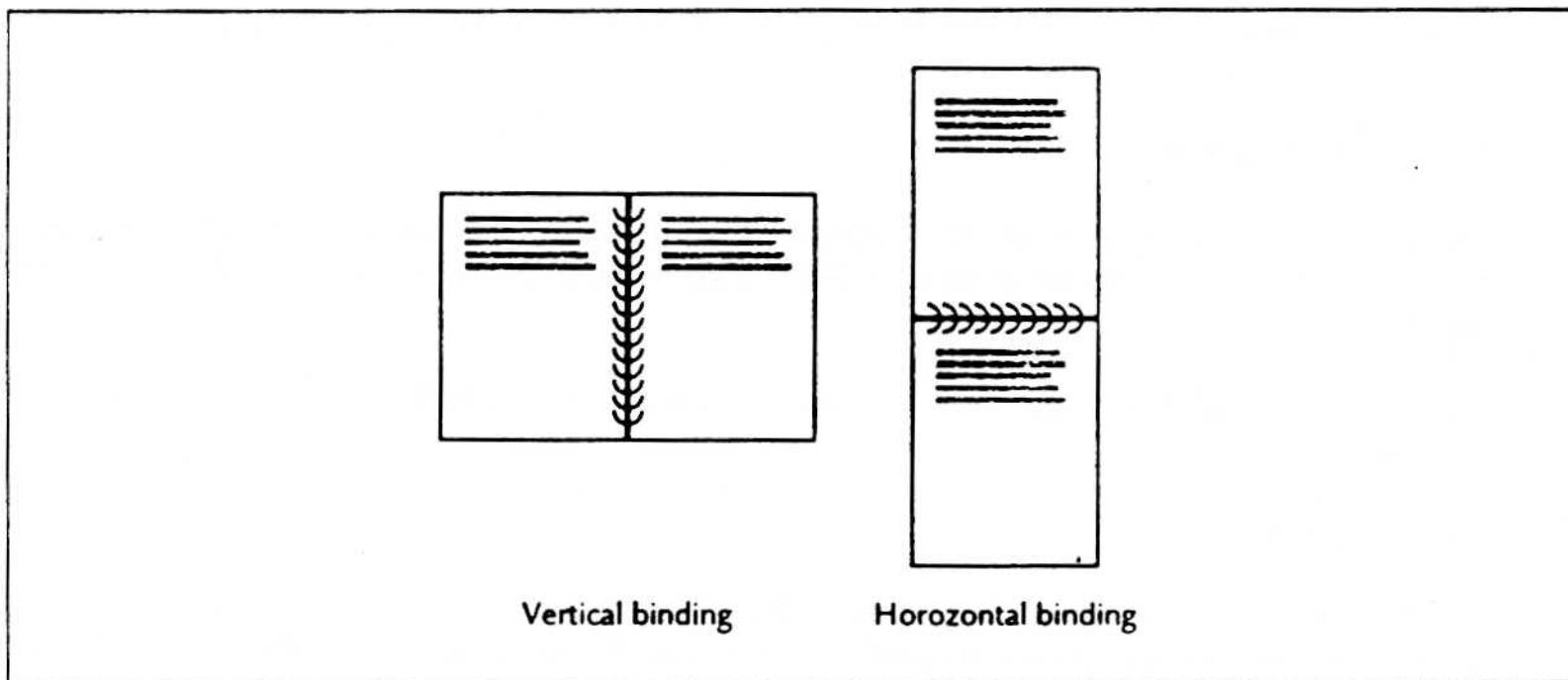
If the user manually changes the *copies=* line in the WIN.INI file before entering Windows, the driver will use that as the default number of copies. However, the next time the user pulls up the driver-specific dialog, the driver will overwrite the user's manually entered value with a 1. The next time the user exits and re-enters Windows, the default number of copies will revert to 1.

#### **2.2.4 Duplex**

This flag contains a value between 0 and 3 that indicates the type of duplex printing:

| Value | Printing Option           |
|-------|---------------------------|
| 0     | Simplex                   |
| 1     | Duplex/Vertical binding   |
| 2     | Duplex/Horizontal binding |

*Duplexing* is the ability of the printer to print on both sides of the page. Vertical (or long-edge) binding means the printer turns the sheet along the long edge, horizontal (or short-edge) binding means the printer turns the sheet along the short edge. See Figure 2.1 for examples of vertical and horizontal binding.



*Figure 2.1 Vertical and Horizontal Binding*

### 2.2.5 <Filename>

The <filename> flag is used by the PCL driver's Soft Font Installer to keep track of the names of permanently downloaded font files. In the Window's convention for listing soft fonts (described in Section 2.2.19, "SoftFont\n"), a permanently downloaded font file is specified by a SoftFont entry without a download file. The installer stores the name under the <filename> flag. For example, SoftFont\n = PFM File and, then, PFM File = Download File.

### 2.2.6 FontSummary

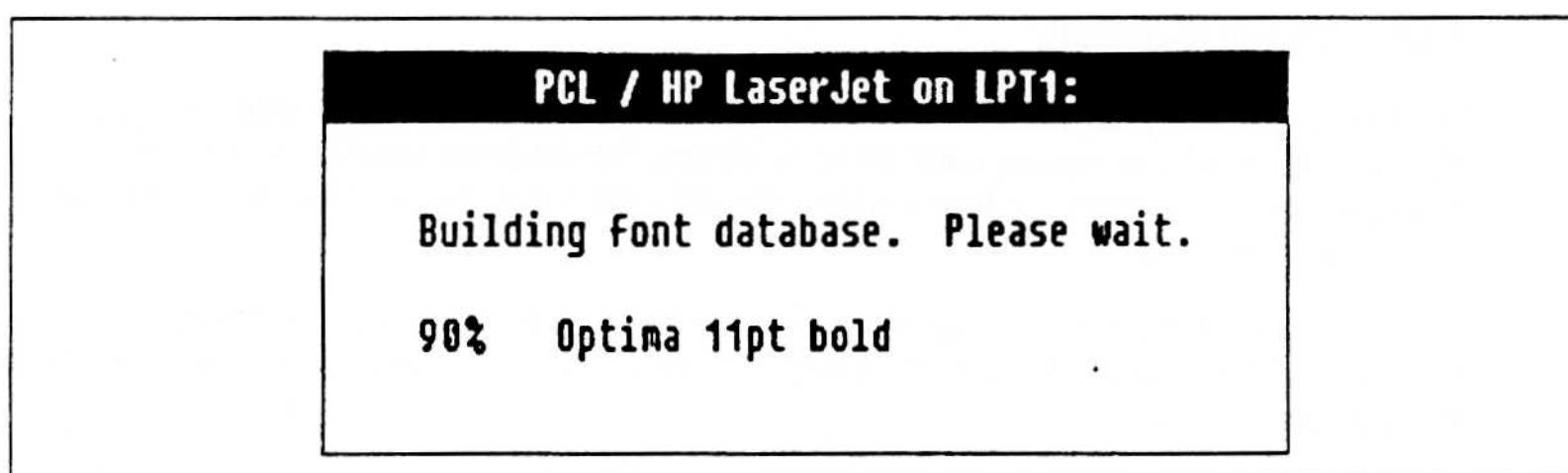
This flag is the name of the font information file that the driver builds. The file contains the internal font data structure used by the driver. Because it may take a long time to build this data structure, the driver saves it to file whenever it builds it. The driver constructs the name of the file from the prefix "FS" combined with the port name. It then truncates the resultant name to eight characters (maximum DOS file name length) and concatenates the file extension ".PCL" to the name. For example, the FontSummary file for LPT1 is called FSLPT1.PCL.

The driver maintains a different copy of the data structure in the FontSummary file for every configuration the user has selected from the driver-specific dialog. For example, if the user changed the printer in the printer listbox, the driver will build a new font data structure and add it to the FontSummary file. If the user reverts to the originally selected printer, the driver will revert to the original font data structure (still in the FontSummary file).

The driver maintains a "most recently used" list of the data structures in the FontSummary file. It will delete least recently used structures to shorten the size of the FontSummary file if necessary. By default, the driver limits the file to 100 kilobytes, but the user may change it with the *MaxFontSummary* flag.

The driver also maintains a different FontSummary file for every port. Because the user may have loaded soft fonts or cartridges under certain ports, the font information may change from port to port.

If the user has loaded soft fonts, it may take the driver a long time to build its internal font data structure, which is why the structure is saved to file once it is built. If the user has more than 15 soft fonts loaded under a given port, the driver will show a "Building font database" message whenever it has to build its font data structure (see illustration). When the driver reads the font data structure from the FontSummary file, it will not show the "Building font database" message.



The driver places the FontSummary file in the directory that contains the driver executable file *HPPCLDRV*. For example, a typical entry for the PCL / HP LaserJet on LPT1: would be:

FontSummary=C:\WINDOWS\FSLPT1.PCL

If the user has several ports with similar names, the driver will assign the same name to different FontSummary files. Suppose the user has added a list of output files to the [ports] section of the WIN.INI file:

```
[ports]
LPT1:=
LPT2:=
LPT3:=
COM1:=9600,n,8,1
COM2:=9600,n,8,1
EPT:=
AppleTalk=
output1.prn=
output2.prn=
output3.prn=
```

In this example, the driver would assign the name "FSOUTPUT.PCL" for the PCL / HP LaserJet on output1.prn, output2.prn, and output3.prn. The driver writes the full string for the port into the FontSummary file header. If the port named in the FontSummary file does not match the port to which the driver is connected, the driver will discard that FontSummary file and rebuild it. The result of having multiple ports using the same FontSummary file is that the driver will delete the FontSummary file every time the user switches ports.

#### **2.2.7 Fsvers**

The FontSummary file version number is a value that starts at 1 and is incremented every time the driver modifies the FontSummary file. The value of *fsvers* resides in both the WIN.INI file and the header of the FontSummary file. The driver verifies that the two version numbers are the same every time it attempts to read a font data structure from the FontSummary file. If the version numbers are different, then the driver discards the FontSummary file and builds a new one.

This flag exists primarily for the Soft Font Installer. Whenever the user modifies soft font information via the installer, it increments the value in *fsvers*. This causes the driver to rebuild its internal font database instead of reading an invalid one from the FontSummary file.

#### **2.2.8 MaxFontSummary**

The *MaxFontSummary* flag is the maximum allowable size (in kilobytes) of the FontSummary file. The driver keeps a "most recently used" list of the internal font databases it builds and saves to the FontSummary file. It deletes the least recently used structures to ensure that the file size is less than *MaxFontSummary*.

The size of the FontSummary file depends upon the number of soft fonts the user has loaded. On the average, the file will range from 3 to 10 kilobytes. By default, the maximum allowable size of the file is 100 kilobytes.

If *MaxFontSummary* is 1 or greater, the minimum FontSummary file size is equal to the size of the most recently used font data structure. In other words, if there is only one data structure in the file, the driver will not trim the FontSummary file even if its size exceeds *MaxFontSummary*.

To force the driver never to build a FontSummary file, the user should add "*MaxFontSummary=0*" to the driver-specific section of the WIN.INI file.

### 2.2.9 Numcart

The *numcart* flag contains the number of cartridges the user has selected. If it is zero (0), no cartridges have been selected (i.e., "none" in the cartridge listbox). If it is 1, then the flag *cartindex* contains the index to the selected cartridge. If it is 2, then *cartindex* and *cartindex1* contain the cartridge indexes. This pattern repeats up to *cartindex7* for the 8th cartridge selected.

The driver *can* allow the user to select up to eight cartridges at one time. However, the number of cartridges the driver *does* allow the user to select depends upon the printer selected in the printer listbox.

The printers and their corresponding number of selectable cartridges are as follows:

| Printer                | # Cartridges |
|------------------------|--------------|
| HP LaserJet            | 1            |
| HP LaserJet Plus       | 1            |
| HP LaserJet 500+       | 1            |
| HP LaserJet Series II  | 2            |
| HP LaserJet IID        | 2            |
| HP LaserJet 2000       | 3            |
| Apricot Laser          | 0            |
| Epson GQ-3500          | 1            |
| Kyocera F-Series       | 0            |
| NEC SilentWriter LC890 | 1            |
| Okidata LaserLine 6    | 0            |
| Olivetti ETV 5000      | 0            |
| Olivetti LP 5000       | 0            |
| QuadLaser I            | 0            |
| Tandy LP-1000          | 0            |
| Tegra Genisis          | 3            |
| Toshiba PageLaser12    | 3            |
| Wang LDP8              | 2            |

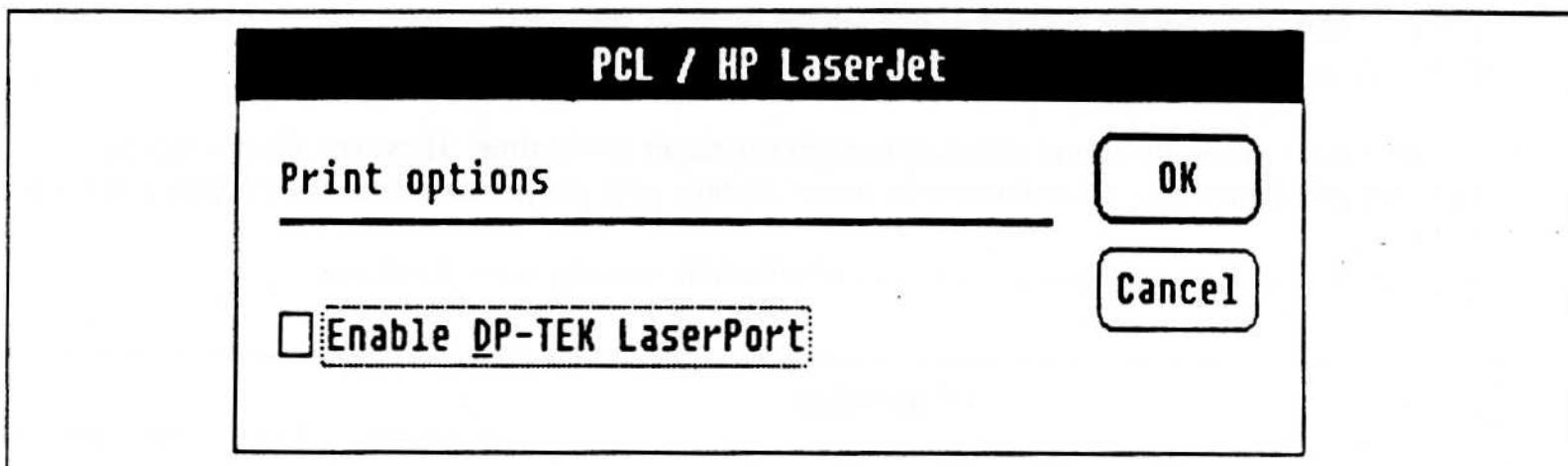
### 2.2.10 Options

The *options* flag is a bit-field of ON/OFF settings. The bits (in hexadecimal) and their meaning are as follows:

| Value  | Meaning  | Default Setting |
|--------|--|-----------------|
| 0x0001 | Enable use of the DP-TEK LaserPort (if present)    | ON              |
| 0x0002 | Reset the printer (escape+ "E") between print jobs | ON              |
| 0x0004 | Force the driver to load soft fonts                | OFF             |
| 0x0008 | Enable vertical text clipping                      | OFF             |

The default *options* value is 0x0003. The version 3.0 PCL driver gave the user the ability to modify the first bit and enable DP-TEK LaserPort, if the driver detected the cards presence. To modify the other bits, the user must manually edit the WIN.INI file.

When the driver detects the presence of the DP-TEK LaserPort, it makes an *options* button visible on the driver-specific dialog. When the user clicks on this button, an additional *options* dialog pops up (see illustration). This dialog contains a check-box for "Enable DP-TEK LaserPort." This dialog is only visible when the LaserPort is installed.



The DP-TEK LaserPort is a card the user can purchase that improves the quality of bitmap images sent to the LaserJet. If the driver detects the presence of the card, then it will use it.

By *not* selecting "Enable DP-TEK LaserPort," you change the bit to zero which makes the driver ignore the LaserPort. This feature is intended to be used by those who own the LaserPort but would like to print to file (users cannot print to file with the LaserPort enabled because all image data would automatically be sent to the printer regardless of the output port).

To modify the other bits, the user must edit the WIN.INI file. Here are some examples:

To disable printer reset between jobs:

```
options=2
```

To force the driver to load soft fonts on a standard LaserJet, as well as options 1 and 2:

```
options=7 (4+2+1)
```

To enable vertical text clipping, as well as options 1 and 2:

```
options=11 (8+2+1)
```

To force the loading of soft fonts and to enable vertical text clipping, as well as everything else:

```
options=15 (8+4+2+1)
```

By default, the driver always resets the printer between print jobs if bit 1 is set. This clears any macros and temporarily downloaded fonts that resided in the printer's memory before the job was sent. If the user does not want to have the printer reset between print jobs, this bit may be set to zero.

**Note:** The driver *always* assumes it has *all* the printer's memory available to it to print the job. It also always assumes it must download any temporary soft fonts in the job.

The "force the driver to load soft fonts" bit causes the driver to load soft font information even for printers that cannot handle soft fonts. Normally, the driver loads soft font information based upon the printer's ability to handle them. But sometimes users list .PFM files for *cartridge* fonts or *printer-resident* fonts that the driver does not know about (for more information, see Section 7.7, "Setting Up .PFM Files for Resident and Cartridge Fonts"). In this situation, the user would want the driver to load soft font information even on a printer that does not normally handle soft fonts.

Enabling this bit will cause the driver to assume that the printer can handle all the soft font information listed in the WIN.INI file. In other words, if there are normal soft fonts listed in the WIN.INI along with the special .PFM files for *cartridge* or *printer-resident* fonts, the driver will load *all* the fonts. If the user selects one of the real soft fonts, the driver will attempt to download it to the printer.

Because PCL does not allow text clipping, the driver must simulate text clipping for the application. Text clipping is the ability of the driver to *not print* text that lies outside of a clip region defined by the application. The driver always clips text horizontally, but it only clips text vertically if the "enable vertical text clipping" bit is set.

The driver clips text in an all-or-none fashion. If any part of the letter is outside of the clip rectangle, the whole letter is not printed. If any part of the line is on the top or bottom edge of the clip rectangle, the whole line is not printed (if vertical text clipping is enabled).

In most cases, vertical text clipping is unnecessary. It is a nuisance for Microsoft Excel users. If they modify the height of a row in Excel by a small amount, the driver will (vertically) clip all the text in that row. It is not much of an issue for Aldus PageMaker. PageMaker contains its own logic to avoid sending lines to the driver that it thinks will end up being clipped.

However, PageMaker's clipping logic assumes the driver can clip partial lines. If any part of the line is within the clip rectangle, PageMaker will send it to the driver. There may be situations (most likely to occur if the user is printing a tiled publication) where PageMaker will send a line to the driver that straddles the top or bottom of the page image area. If vertical text clipping is disabled, the driver will send the line to the printer. Since it is beyond the printer's image area, the printer will randomly print the line somewhere on the page. This problem may be corrected by enabling vertical text clipping which will simply clip the line and not print it.

### 2.2.11 Orient

The *orient* flag is the orientation in which the document should be printed. The values are as follows:

| Value | Orientation |
|-------|-------------|
| 15    | Portrait    |
| 16    | Landscape   |

### 2.2.12 Paper

The *paper* variable selects the paper size on which to print. The values are as follows:

| Value | Paper     |
|-------|-----------|
| 20    | Letter    |
| 21    | A4        |
| 22    | Legal     |
| 23    | B5        |
| 24    | Executive |
| 25    | A3        |
| 26    | Ledger    |

### 2.2.13 Prtcaps

This flag is a bit-field of the capabilities of the currently selected printer. The driver reads its value from an internal data structure; it *never* reads this flag from the WIN.INI file. It then writes it to the WIN.INI file so other applications may read it.

This flag exists for applications which need to know the capabilities of the printer the user has selected. For example, font generation utilities may need to know if the printer can handle soft fonts, or if the printer can handle soft fonts above 30 points.

The values of this field changed between the version 3.0 and 3.1 driver. The fields that represented paper information in the 3.0 driver were removed. Two new fields (for handling envelope feed) were added where paper information fields used to exist. The definition of some fields was slightly modified.

The *old* list of capabilities bits (in hexadecimal) was as follows:

|     |        |   |
|-----|--------|---|
|     | 0x0001 | printer has capabilities of a standard LaserJet                   |
|     | 0x0002 | printer has capabilities of a LaserJet Plus                       |
|     | 0x0004 | printer has capabilities of a Laserjet 500                        |
|     | 0x0008 | lower tray is handled   |
|     | 0x0010 | printer does <i>not</i> support downloadable fonts                |
|     | 0x0020 | manual feed is <i>not</i> supported                               |
|     | 0x0040 | printer cannot support internal bit stripping                     |
| old | 0x0080 | printer can handle B5 paper                                       |
| old | 0x0100 | printer emulates an HP LaserJet                                   |
| old | 0x0200 | printer can handle Exec paper                                     |
| old | 0x0400 | printer can handle A3 paper                                       |
| old | 0x0800 | printer can handle ledger paper                                   |
|     | 0x1000 | printer can print duplex  |
|     | 0x2000 | printer selects paper bin based on paper size (auto paper select) |
|     | 0x4000 | printer can print fonts in any orientation (auto font rotation)   |
| old | 0x8000 | printer uses new paper select strings                             |

The *new* list of capabilities bits (in hexadecimal) is as follows:

|     |        |   |
|-----|--------|---|
|     | 0x0001 | printer has capabilities of a standard LaserJet                   |
|     | 0x0002 | printer has capabilities of a LaserJet Plus                       |
|     | 0x0004 | printer has capabilities of a Laserjet 500                        |
|     | 0x0008 | lower tray is handled   |
|     | 0x0010 | printer does <i>not</i> support downloadable fonts                |
|     | 0x0020 | manual feed is <i>not</i> supported                               |
|     | 0x0040 | printer cannot support internal bit stripping                     |
| new | 0x0080 | printer supports manual/envelope feed                             |
| new | 0x0100 | printer is an HP PCL emulation printer                            |
| new | 0x0200 | printer supports new (LaserJet IID) envelope feed                 |
| new | 0x0400 | printer can print duplex like the LaserJet IID                    |
| new | 0x0800 | <i>undefined</i>  |
|     | 0x1000 | printer can print duplex  |
|     | 0x2000 | printer selects paper bin based on paper size (auto paper select) |
|     | 0x4000 | printer can print fonts in any orientation (auto font rotation)   |
| new | 0x8000 | printer has the capabilities of a LaserJet Series II              |

The bits for envelope feed are new to the version 3.1 driver. Bit 0x0080 is set if the printer selects envelopes using the manual/envelope feed escape defined in the *HP LaserJet Technical Reference Manual*. Even though most HP printers and clones support this escape, the bit is set only for printers that actually have an envelope feed attachment for the printer.

The second envelope feed escape, bit 0x0200, is set for the HP LaserJet IID. The driver uses the escape defined by HP which is special for this printer's envelope feeder.

The second duplex bit is also new to the version 3.1 driver. The first duplex bit, 0x8000, supports duplex for the LaserJet 2000. The second duplex bit, 0x0400, supports duplex in the same manner that the LaserJet IID prints two-sided pages.

As stated earlier, the driver only writes the *prtcaps* flag to the WIN.INI file. It reads it from its internal data structure, based upon the currently selected printer. The capabilities of each of the printers are as follows:

| Value  | Printer Capabilities   |
|--------|--|
| 0x0011 | HP LaserJet (does not support soft fonts, standard LaserJet)   |
| 0x0002 | HP LaserJet Plus (LaserJet Plus)   |
| 0x000C | HP LaserJet 500+ (lower tray, LaserJet 500+)   |
| 0x8080 | HP LaserJet Series II (manual/envelope feed, LaserJet Series II)   |
| 0xC608 | HP LaserJet IID (lower tray, new envelope feed, auto font rotation, LaserJet IID duplex, LaserJet Series II)                 |
| 0xF028 | HP LaserJet 2000 (auto font rotation, auto paper select, duplex, no manual feed, lower tray, LaserJet Series II)             |
| 0x010A | Apricot Laser (emulation, lower tray, Laserjet plus)   |
| 0x010C | Epson GQ-3500 (emulation, lower tray, LaserJet 500+)   |
| 0x010A | Kyocera F-Series (emulation, lower tray, LaserJet Plus)  |
| 0x014C | NEC SilentWriter LC890 (emulation, cannot do internal bitstripping, lower tray, LaserJet 500+)                               |
| 0x0102 | Okidata LaserLine 6 (emulation, LaserJet Plus)   |
| 0xC100 | Olivetti ETV 5000 (emulation, auto font rotation, LaserJet Series II)  |
| 0xC100 | Olivetti LP 5000 (emulation, auto font rotation, LaserJet Series II)   |
| 0x0102 | QuadLaser I (emulation, LaserJet Plus)   |
| 0x0102 | Tandy LP-1000 (emulation, LaserJet Plus)   |
| 0xE188 | Toshiba PageLaser12 (emulation, auto font rotation, auto paper select, lower tray, manual envelope feed, LaserJet Series II) |
| 0x8100 | Tegra Genesis (emulation, LaserJet Series II)  |
| 0x8100 | Wang LDP8 (emulation, LaserJet Series II)  |

If the *prtcaps* flag is not present, outside applications should assume that the capabilities of the standard LaserJet are in use.

### 2.2.14 PrtIndex

The *prtindex* flag is the index to the currently selected printer. Current values are as follows:

| <b>Index</b> | <b>Printer</b>         | <b>Memory</b> |
|--------------|------------------------|---------------|
| 0            | HP LaserJet            | 128 Kb        |
| 1            | HP LaserJet Plus       | 512 Kb        |
| 2            | HP LaserJet Plus       | 2 Mb          |
| 3            | HP LaserJet 500+       | 512 Kb        |
| 4            | HP LaserJet 500+       | 2 Mb          |
| 5            | HP LaserJet Series II  | 512 Kb        |
| 6            | HP LaserJet Series II  | 1.5 Mb        |
| 7            | HP LaserJet Series II  | 2.5 Mb        |
| 8            | HP LaserJet Series II  | 4.5 Mb        |
| 9            | HP LaserJet IID        | 512 Kb        |
| 10           | HP LaserJet IID        | 1.5 Mb        |
| 11           | HP LaserJet IID        | 2.5 Mb        |
| 12           | HP LaserJet IID        | 4.5 Mb        |
| 13           | HP LaserJet 2000       | 1.5 Mb        |
| 14           | HP LaserJet 2000       | 2.5 Mb        |
| 15           | HP LaserJet 2000       | 3.5 Mb        |
| 16           | HP LaserJet 2000       | 4.5 Mb        |
| 17           | HP LaserJet 2000       | 5.5 Mb        |
| 18           | Apricot Laser          | 512 Kb        |
| 19           | Apricot Laser          | 1 Mb          |
| 20           | Apricot Laser          | 1.5 Mb        |
| 21           | Apricot Laser          | 3.5 Mb        |
| 22           | Epson GQ-3500          | 640 Kb        |
| 23           | Epson GQ-3500          | 1.5 Mb        |
| 24           | Epson GQ-3500          | 2 Mb          |
| 25           | Kyocera F-Series       | 512 Kb        |
| 26           | Kyocera F-Series       | 1 Mb          |
| 27           | Kyocera F-Series       | 1.5 Mb        |
| 28           | Kyocera F-Series       | 3.5 Mb        |
| 29           | NEC SilentWriter LC890 | 512 Kb        |
| 30           | NEC SilentWriter LC890 | 1.3 Mb        |
| 31           | Okidata LaserLine 6    | 512 Kb        |
| 32           | Olivetti ETC 5000      | 2 Mb          |
| 33           | Olivetti LP 5000       | 2 Mb          |
| 34           | QuadLaser I            | 2 Mb          |
| 35           | Tandy LP-1000          | 1.5 Mb        |
| 36           | Tegra Genisis          | 5.5 Mb        |
| 37           | Toshiba PageLaser12    | 512 Kb        |
| 38           | Toshiba PageLaser12    | 2 Mb          |
| 39           | Wang LDP8              | 512 Kb        |
| 40           | Wang LDP8              | 1.5 Mb        |
| 41           | Wang LDP8              | 2.5 Mb        |
| 42           | Wang LDP8              | 4.5 Mb        |

These values changed between the version 3.0 and 3.1 printer driver, and will most likely change again in the future. As printers are added, the index values will shuffle. No application outside of the PCL driver should access these numbers. Applications that want to determine the capabilities of the selected printer should look at the *prtcaps* flag.

#### 2.2.15 Prtresfac

The *prtresfac* flag is the printer resolution factor. 300 dpi shifted right by this number yields the printer resolution. Possible values are as follows:

| Value | Resolution |
|-------|------------|
| 0     | 300 dpi    |
| 1     | 150 dpi    |
| 2     | 75 dpi     |

#### 2.2.16 Sfdir

The *sfdir* flag is the path to the directory containing the soft fonts. This is typically C:\PCLFONTS, C:\PCLPFM, or C:\FONTS.

#### 2.2.17 Sfdlbat

The *sfdlbat* flag contains the path and name of the batch file that downloads permanent soft fonts to the printer. The Soft Font Installer generates this file whenever the user sets up permanent soft fonts.

A typical entry would look like the following:

*sfdlbat=C:\PCLPFM\SF\PT1.BAT*

The name of the file is constructed from the prefix "SF" combined with the port name. The resultant name is truncated to eight characters (maximum DOS filename length) and the file extension ".BAT" is concatenated to the name.

#### 2.2.18 Sfdlstyle

The *sfdlstyle* is the manner in which soft fonts should be downloaded. This flag is a bit-field, the values (in hexadecimal) are as follows:

| Value | Meaning  |
|-------|--|
| 0x10  | Download fonts "now" (when the user exits the installer) |
| 0x20  | Download fonts when the user turns on the computer       |

Potential *sfdlstyle* values would be as follows:

Do not download fonts:

**sfdlstyle=0**

Download fonts at startup only:

**sfdlstyle=32**

Download fonts when the user exits the installer and at startup:

**sfdlstyle=48**

### 2.2.19 SoftFont*n*

This is the soft font entry in the WIN.INI file. The format for listing soft fonts in the WIN.INI file is a Microsoft Windows standard (i.e., all printer drivers should list soft fonts in this manner). This format is described in the *Microsoft Windows Fonts Guide*. The mechanism is described briefly here. A typical soft font entry would look like this:

```
SoftFonts=9
SoftFont1=C:\PCLFONTS\OPPR0090.PFM
SoftFont2=C:\PCLFONTS\OPPB0090.PFM
SoftFont3=C:\PCLFONTS\OPPI0090.PFM
SoftFont4=C:\PCLFONTS\OPPR0110.PFM
SoftFont5=C:\PCLFONTS\OPPB0110.PFM,C:\PCLFONTS\OP110BPN.R8P
SoftFont6=C:\PCLFONTS\OPPI0110.PFM,C:\PCLFONTS\OP110IPN.R8P
SoftFont7=C:\PCLFONTS\OPPR0240.PFM,C:\PCLFONTS\OP240RPN.R8P
SoftFont8=C:\PCLFONTS\OPPB0240.PFM,C:\PCLFONTS\OP240BPN.R8P
SoftFont9=C:\PCLFONTS\OPPI0240.PFM,C:\PCLFONTS\OP240IPN.R8P
```

In addition to these entries, which are defined as the "standard" format for listing soft fonts, the PCL driver's Soft Font Installer would add the following entries:

```
C:\PCLFONTS\OPPR0090.PFM=C:\PCLFONTS\OP090RPN.R8P
C:\PCLFONTS\OPPB0090.PFM=C:\PCLFONTS\OP090BPN.R8P
C:\PCLFONTS\OPPI0090.PFM=C:\PCLFONTS\OP090IPN.R8P
C:\PCLFONTS\OPPR0110.PFM=C:\PCLFONTS\OP110RPN.R8P
```

The *SoftFont*n** entry lists the two files necessary for a soft font to be used in Windows. The first file is the Windows Printer Font Metric (PFM) file. The second file is the downloadable font file. The .PFM file contains the metrics used by the driver to provide Windows applications with information about the fonts. Information such as font height, character widths, and pair kern tables are contained in the .PFM file. The *Microsoft Windows Fonts Guide* describes the contents of .PFM files.

The .PFM portion of the *SoftFont*n** entry must always be listed in the WIN.INI file. The downloadable font file, if listed, follows the name of the .PFM file. The two names are separated by a comma. The presence of the downloadable font file name indicates that the font is set up for *temporary* download. This means that the driver should download the font to the printer the first time it encounters text in the font during a print job. The font will be deleted at the end of the print job.

The absence of the downloadable font file name indicates to the driver that the font has been permanently downloaded to the printer (it is also used to indicate that the font is a printer-resident or cartridge font, see Section 7.7, "Setting Up .PFM Files for Resident and Cartridge Fonts"). Permanently downloaded fonts are sent to the printer when it is turned on and remain in the printer's memory until it is turned off. The driver uses the font by sending the Font ID string to the printer. The Font ID must equal the value of "n" in the *SoftFontn* flag.

The PCL driver's Soft Font Installer adds one more convention: for every font set up for permanent download, it adds an entry to the WIN.INI file in the form:

<PFM filename>=<downloadable font filename>

The installer does this because it needs to keep track of the downloadable font file name. If a font is set up for permanent, that means that the downloadable font file name is not present. To keep track of the download file name, the installer adds the "<PFM filename>=<downloadable font filename>" entry.

The PCL driver also supports two other aspects of the soft font entries. These are not considered to be part of the standard for setting up soft fonts in Windows:

1. The driver allows the user to abbreviate the *SoftFontn* flag.
2. The driver does not require the *SoftFontn* entries to be in order.

The PCL driver looks for a match in the *SoftFontn* flag up to the length of the flag in the WIN.INI file. If, for example, the flag in the WIN.INI were "soft1=<etc>," or even "so1=<etc>," the driver would recognize the flag as a *SoftFontn* entry. However, this capability exists in the driver only for historical reasons; it is not recommended that users abbreviate the *SoftFontn* flag.

The PCL driver does not require that soft fonts be listed in one contiguous range. Some drivers, on the other hand, require contiguous ranges. For example, if "SoftFonts=3," the driver would expect the three soft font entries to be:

SoftFont1=<etc>  
SoftFont2=<etc>  
SoftFont3=<etc>

For the PCL driver, though, the entries could just as easily be *SoftFont2*, *SoftFont8*, and *SoftFont10*, or any other non-contiguous set of flags. It scans the WIN.INI file for all soft font entries. It will also load *all* the soft fonts regardless of how many soft fonts the *SoftFonts* flag indicates are available.

## **2.2.20 SoftFonts**

This gives the number of soft fonts listed in the WIN.INI file.

### 2.2.21 Tray

The *tray* flag contains the input paper tray. Possible values are as follows:

| Value | Paper Tray                      |
|-------|---------------------------------|
| 30    | Default paper tray ("top tray") |
| 31    | Lower paper tray                |
| 32    | Manual feed                     |
| 33    | Auto paper feed                 |

The "Lower" paper tray on the LaserJet 2000 is the middle tray. The user gains access to the paper deck by selecting "Auto paper feed."

### 2.2.22 White\_text

The *white\_text* flag controls the value of "white" text. Because PCL does not have white text (e.g., for reverse text), the driver attempts to "synthesize" white text by not sending it to the printer. The *white\_text* flag gives the user the ability to adjust the driver's sensitivity to white text.

Please notice that PCL printers are not capable of actually printing white text. This section describes how the driver behaves when the application assigns the color "white" to text. The *white\_text* flag is used by the driver to determine how to detect white text. When the driver detects the color of the text is white, it does not send the text to the printer.

The driver detects white text by looking at the RGB value passed to it by the application. An RGB value consists of three numbers indicating the red, green, and blue components of the color. Possible values range from 0 to 255 for each color. 0 means full saturation (or black) of the color, 255 means no color (or white). A value between 0 and 255 means partial saturation of the color. If the R, G, and B components are 0, the color is black. If all three numbers are 255, the color is white.

The driver applies the following logic to detect white text:

If all three values of red, green, and blue are greater than or equal to the value of *white\_text*, then the color is white and the text should *not* be printed. If at least one of the R, G, or B components is less than *white\_text*, then the color is printed.

By default, the driver uses a *white\_text* value of 255. This means that the text must be truly white for the driver not to print it. The driver regards any text that has a value close to white, but not exactly white, as black and prints it.

The user may make the driver regard all text as white text with:

`white_text=0`

And always print all text with:

`white_text=256`

This flag exists in the event an application wants a finer control over white text printing than the driver's default approach to determining what constitutes white text.



# **CHAPTER 3**

## **PERMANENT SOFT FONTS**

### **CONTENTS**

---

|     |  |     |
|-----|--|-----|
| 3.1 | Setting Up Fonts for Download                | 3-3 |
| 3.2 | Tracking Permanent Fonts in the WIN.INI File | 3-7 |



### 3 PERMANENT SOFT FONTS

Permanent soft fonts are downloaded to the printer when the printer is turned on and remain there until the printer is turned off. Because permanent fonts are not sent to the printer during the print job, they pose some interesting problems. This chapter explores some of these problems and how the Soft Font Installer solves them.

Permanent fonts are downloaded to the printer *sometime* after power up. The driver does not have to be active, nor does Windows need to be running for permanent fonts to be downloaded to the printer. In fact, permanent fonts are typically downloaded to the printer when the user first turns on his or her computer.

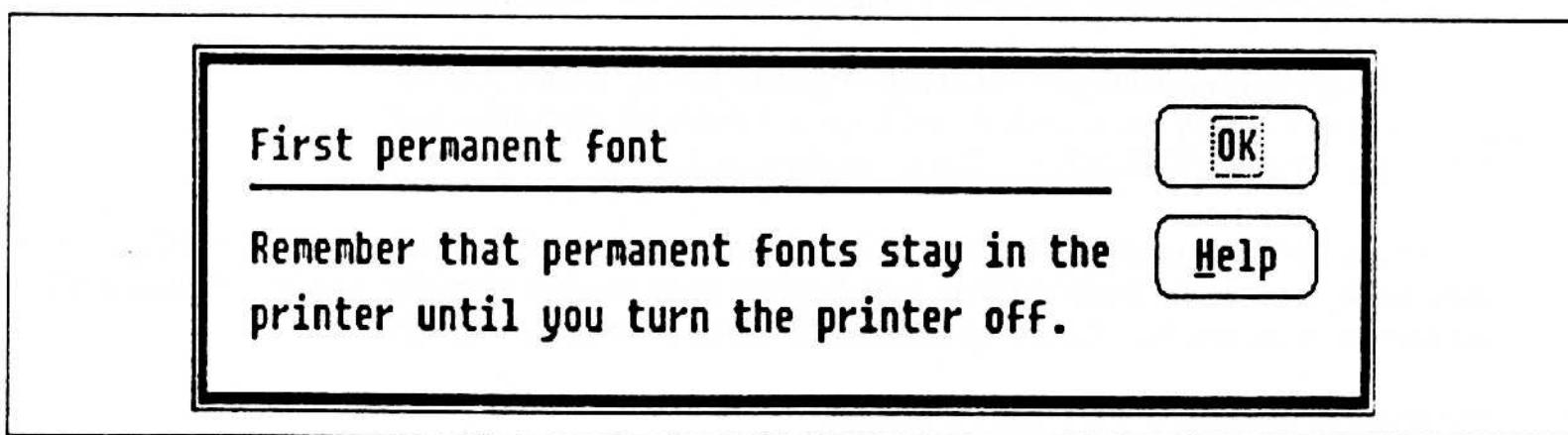
Previous versions of the driver assumed that the user took responsibility for setting up fonts for permanent download. In the version 3.0 PCL driver, the Soft Font Installer manages the downloading of permanent soft fonts.

#### 3.1 Setting Up Fonts for Download

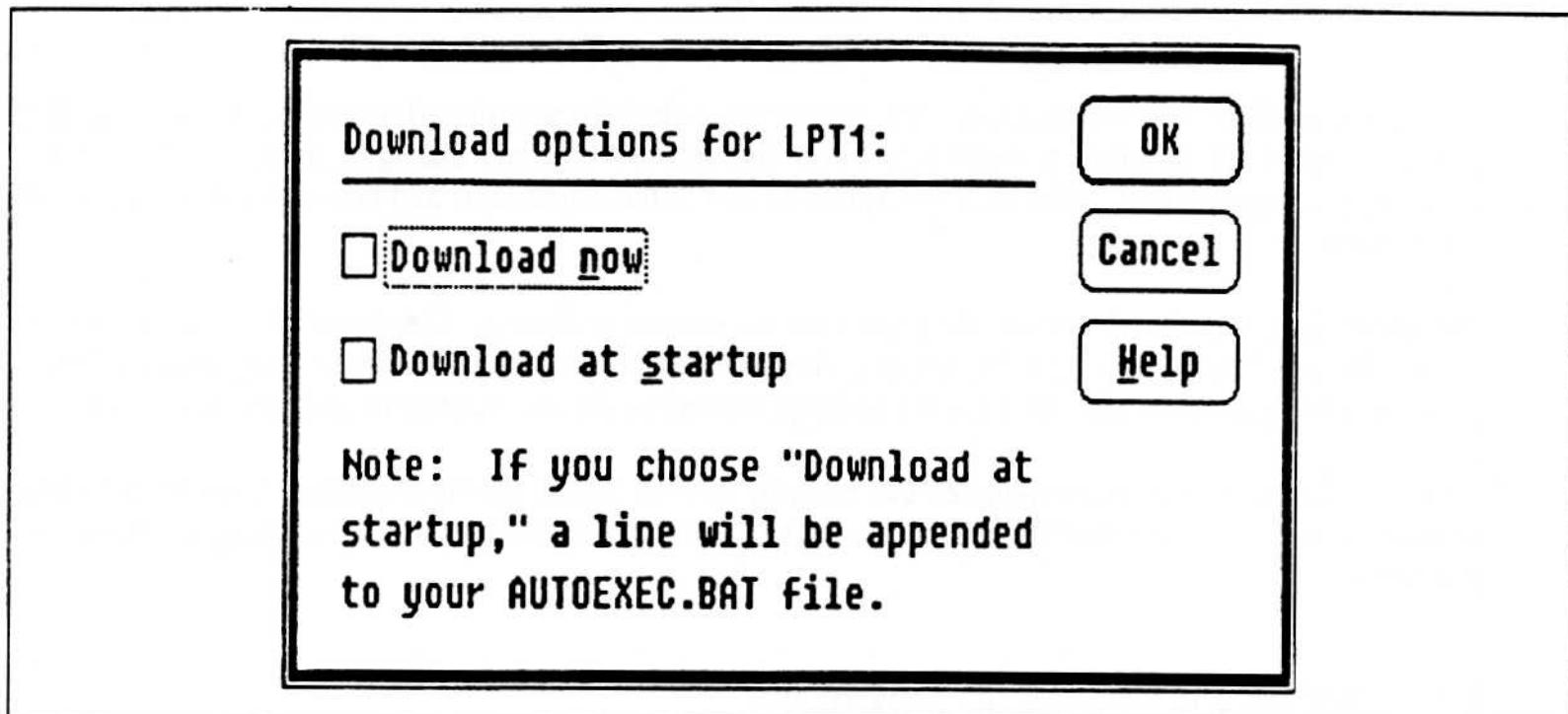
The user tells the Soft Font Installer that a font should be set up for permanent download by selecting the font in the installer's listbox and clicking on the "Permanent" button at the bottom of the listbox. See the *Microsoft Windows User's Guide* for a description of the user interface to the installer.

The installer allows the user to change the status (i.e., from temporary to permanent) of one font at a time. The font receiving the status change is the one whose name appears on the status line at the bottom of the installer dialog. The installer only allows one font at a time to be modified to prevent the user from converting a large number of fonts to permanent download.

The first time a user changes a font from temporary to permanent status, the installer pops up a warning dialog (see illustration). This dialog reminds the user that permanent fonts take up printer memory and, therefore, only the most frequently used fonts should be permanently downloaded to the printer.



When the user exits the installer, it prompts the user for the download style (see illustration on next page). If the user selects "Download now," the installer will send a print job consisting of the permanent fonts to the printer. If the user selects "Download at startup," the installer will build a batch file that downloads the fonts to the printer and edits the AUTOEXEC.BAT file to call the batch file.



The user may select both, one, or none of these options. As long as the user has permanent soft fonts, the installer pops up this prompt every time the user exits the installer.

As part of downloading permanent fonts, the installer sends a "delete all fonts" escape to the printer. This forces the printer to delete all permanently downloaded fonts. After downloading the new fonts, a banner page (in portrait format) is printed that shows the fonts that have been downloaded. Portrait fonts are shown in the typeface that was downloaded.

Downloading permanent fonts "now" (i.e., upon exiting the installer) is not very difficult for the driver. The installer simply opens a Windows print job and sends down the fonts. Downloading fonts "at startup" (i.e., when the user turns on his or her machine) is more complex.

To set up the downloading of permanent fonts at startup, the installer performs the following tasks:

1. Writes an executable program that prompts the user to download fonts.
2. Creates a batch that downloads the fonts and writes out the banner page.
3. Edits the AUTOEXEC.BAT file to call the download batch file.

The executable program presents the user with a yes/no prompt for downloading permanent soft fonts. The program is stored in the driver's resources and written to its own file when the installer is setting up fonts for permanent download. The download batch file calls this executable program.

The program presents the user with the prompt:

Download PCL fonts to <port name>? [y/n]

If the user responds "no," the program returns 1, and the batch file tests the DOS ERRORLEVEL and exits without downloading fonts.

If the user responds "yes," the program returns a DOS ERRORLEVEL of 0, and the batch file proceeds to download fonts. For example, this sample download batch file downloads one permanent font, "Tms Rmn 12pt," to the printer:

```

rem HPPCL -- Downloading fonts
echo off
C:\PCLFONTS\PCLSF0YN.EXE LPT1:
if ERRORLEVEL 1 goto nodownload
echo {ESC}E{ESC}*c0F > LPT1:
echo {ESC}(s3t0b0s12v10h0P{ESC}&a0c0RPermanently downloaded
    font(s):{ESC}&12D > %tmp%\pcl3.tmp
echo {ESC}*c1D > %tmp%\pcl11.tmp
echo {ESC}*c1d5F > %tmp%\pcl12.tmp
echo {ESC}(1X*Tms Rmn 12pt >> %tmp%\pcl3.tmp
copy %tmp%\pcl11.tmp+C:\PCLFONTS\TR120RPN.USP/b+%tmp%\pcl12.tmp/a
LPT1:/b
echo {FF} >> %tmp%\pcl3.tmp
copy %tmp%\pcl3.tmp LPT1:
erase %tmp%\pcl11.tmp
erase %tmp%\pcl12.tmp
erase %tmp%\pcl3.tmp
:nodownload

```

The download batch file creates three temporary files:

| File | Function                                    |
|------|---|
| tmp1 | Sets up permanent Font ID.                  |
| tmp2 | Assigns the Font ID to the downloaded font. |
| tmp3 | Downloads the banner.                       |

The sample download batch file shows how these temporary files are used. Notice that the batch file takes advantage of the DOS batch file string substitution feature. That is, every reference to a temporary file is preceded with %tmp%. If the user has an environment variable "tmp," then DOS will replace the string with the temporary directory path. If the environment variable does not exist, DOS will replace the string with a null string. The temporary files will be written to the user's root directory.

The batch file is invoked from the user's AUTOEXEC.BAT file. The installer edits the AUTOEXEC.BAT file and appends the following strings to the file:

```

rem The Windows PCL / HP LaserJet driver added the next line
command /c C:\PCLFONTS\SFLPT1.BAT

```

The string that invokes the download batch file uses the DOS "command /c" option. This allows DOS to suspend execution of the AUTOEXEC.BAT file to execute the download file, then return to the execution of the AUTOEXEC.BAT file. Without "command /c," DOS would not return to executing the AUTOEXEC.BAT file (any commands after the line in the AUTOEXEC.BAT file would not be executed).

The installer appends this string to the AUTOEXEC.BAT file by stepping to the end of the file and then backing up until it hits a character greater than or equal to a space. This circumvents a problem some users may have if they use Windows Notepad to edit their AUTOEXEC.BAT file. Windows Notepad sometimes writes an end-of-file character as the last valid character in the file. An application that blindly appends text to the AUTOEXEC.BAT will append it *after* the end-of-file character. The added text would be ignored when DOS executed the AUTOEXEC.BAT file.

There are always potential problems when an application "automatically" edits the user's AUTOEXEC.BAT file. These programs typically will either append text to the file (like the installer) or place text at the beginning of the file. Both approaches present some problems:

1. Some users have special "menu templates" that are executed from the AUTOEXEC.BAT file. These templates are typically set up by dealers for the users. From the template, the user executes the applications he or she wishes to use. The user never exits this template utility to return to the execution of the AUTOEXEC.BAT file.
2. Other users set up soft fonts to be permanently downloaded to a port which is a remote printer on a network. Typically, the network software is executed somewhere at the beginning of the AUTOEXEC.BAT file. The download commands must appear *after* the network software runs.

If an application appends text to the AUTOEXEC.BAT file, it runs the risk of the commands never being executed. If the application inserts the commands at the beginning of the file, it runs the risk of executing the commands before certain other vital commands are executed.

If a user sets up fonts for permanent download, but they are not being downloaded to the printer, the commands in the AUTOEXEC.BAT file are probably not being executed. The problem may be solved by moving the commands in the AUTOEXEC.BAT file.

It is important that the command line the installer writes to the AUTOEXEC.BAT file never be modified. The line may be moved in the file, but it should not be modified. This is because the installer always looks for the line and appropriately modifies it. The installer searches the AUTOEXEC.BAT file for the line and changes it without changing the file size. It does not assume it is at the bottom of the file.

For example, if the user sets up some soft fonts for permanent download at startup, the installer will build the download batch file and add this line to the AUTOEXEC.BAT file:

```
rem The Windows PCL / HP LaserJet driver added the next line
command /c C:\PCLFONTS\SFLPT1.BAT
```

If the user then re-enters the installer and changes all the permanent fonts to temporary, the installer (upon exit) parses the AUTOEXEC.BAT file and modifies the command line to:

```
rem The Windows PCL / HP LaserJet driver added the next line
rem      C:\PCLFONTS\SFLPT1.BAT
```

This has the effect of disabling the download command without deleting it. If the user once again enters the installer and makes some fonts permanent, the installer (upon exit) will locate the commented line in the AUTOEXEC.BAT file and uncomment it:

```
rem The Windows PCL / HP LaserJet driver added the next line
command /c C:\PCLFONTS\SFLPT1.BAT
```

### 3.2 Tracking Permanent Fonts in the WIN.INI File

The format for listing soft fonts in the WIN.INI file is a Microsoft Windows standard (i.e., all printer drivers should list soft fonts in this manner). This format is described in the *Microsoft Windows Fonts Guide*. The mechanism is briefly described in this document's Section 2.2.19, "SoftFontn."

Users who added soft fonts using PCLPFM must deal with this problem . PCLPFM is the utility that used to be shipped with the PCL driver (version 1.05a and earlier). This utility generated PFM files from downloadable font files and created an APPNDWIN.INI file. After running PCLPFM, the user would manually edit the WIN.INI file to add the *SoftFontn* entries.

By default, PCLPFM set up all fonts for temporary download. If the user wanted to set up fonts for permanent download, then he or she was instructed to *remove* the name of the downloadable font file from the *SoftFontn* line in the WIN.INI file. The user was responsible for setting up the mechanism for downloading the permanent fonts to the printer.

For example, suppose the following entries were added to the WIN.INI by the Soft Font Installer:

```
[HPPCL, LPT1]
SoftFonts=2
SoftFont1=C:\PCLFONTS\OPPR0110.PFM
SoftFont2=C:\PCLFONTS\OPPB0110.PFM, C:\PCLFONTS\OP110BPN.R8P
C:\PCLFONTS\OPPR0110.PFM=C:\PCLFONTS\OP110RPN.R8P
```

The same fonts added from PCLPFM (with help from the user) would look like this:

```
[HPPCL, LPT1]
SoftFont1=C:\PCLPFM\OP110RPP.PFM
SoftFont2=C:\PCLPFM\OP110BPP.PFM, C:\PCLPFM\OP110BPN.R8P
```

This mechanism posed a problem for the Soft Font Installer. With the name of the permanently downloaded font missing, how could the installer locate the font? Also, since the user may have already set up a mechanism for downloading those fonts to the printer, how could the installer work around it?

The best approach to resolving the problem is to inform users that they should go back into their WIN.INI file and restore the fonts to temporary status (add *back* the name of the permanent font file). The users should also remove any utilities they had that downloaded the fonts.

To handle fonts set up for permanent download by PCLPFM, the installer applies the following logic:

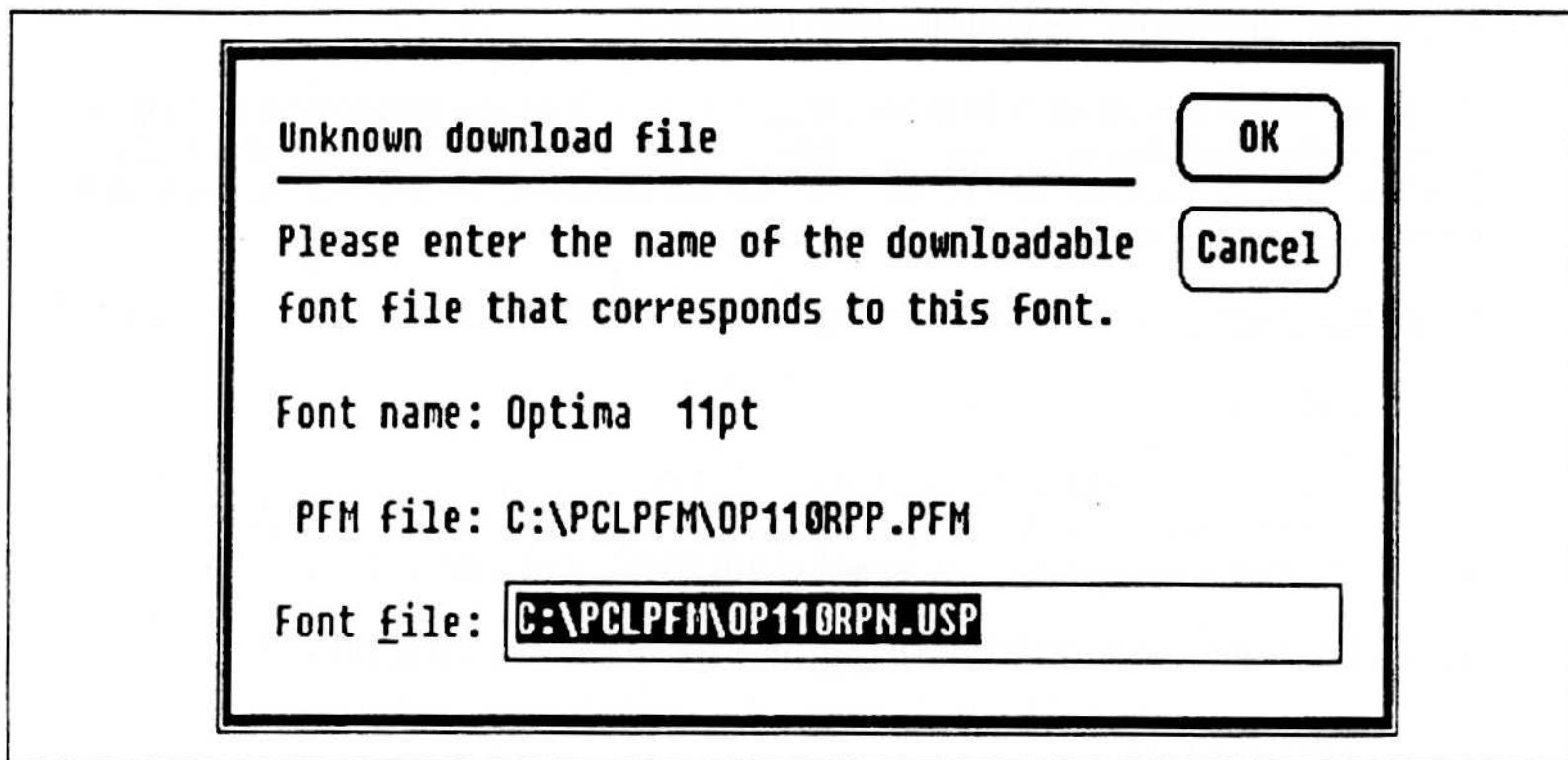
1. If the installer has no record of the downloadable font file name, then it simply skips it when it sets up the permanent download batch file.
2. If the user attempts to change the font from permanent to temporary status, the installer requires the user to provide a name for the downloadable font file.

When the installer creates its download batch file, it will not report an error if it cannot find the name of the downloadable font file. It does this for two reasons:

1. The installer assumes the user removed the file name. The user also probably has some utility that automatically downloaded those fonts. Those fonts will continue to be downloaded via the old mechanism, while any new fonts changed to permanent by the installer will be downloaded by the installer.

2. There is another mechanism by which the user can provide the driver with .PFM files for *cartridge* or *printer-resident* fonts unknown to the driver. In this mechanism, only the .PFM file is listed. The downloadable font file does not exist. The installer will not report that it cannot download fonts, which prevents confusion for the user.

When the user attempts to change a font from permanent to temporary status, the installer will prompt the user to provide the name of the downloadable font file (see illustration). If the user cannot provide the name, then the user cannot change the font's status.



The installer attempts to manufacture the name of the downloadable font file by assuming it resides in the same directory as the .PFM file and the name was automatically assigned by PCLPFM. Since PCLPFM derived the .PFM file name from the downloadable font file name, the installer simply reverses the logic to get the downloadable font file name from the .PFM file name. This manufactured name is presented to the user as part of the prompt for the file name.

Once the user has provided the correct downloadable font file name (the installer does verify that the file actually exists before it accepts the file name), the installer can set up the font for temporary download. If the user ever reverts the font's status back to permanent, the installer preserves the name using the <PFM filename>=<downloadable font filename> convention. The installer also assumes that, since the user knowingly reverted the font back to temporary status, the user removed whatever utilities existed on the machine to download the permanent fonts.

## **CHAPTER 4**

### **SFINSTAL.DIR**

#### **CONTENTS**

---

|     |  |     |
|-----|--|-----|
| 4.1 | Logical Drive Definition                 | 4-3 |
| 4.2 | Font Family Definition                   | 4-4 |
| 4.3 | Installing Without the SFINSTAL.DIR File | 4-7 |
| 4.4 | Sample SFINSTAL.DIR Directory File       | 4-8 |



## 4 SFINSTAL.DIR

The soft font install directory file, SFINSTAL.DIR, provides a standard way for font manufacturers to describe the contents of a font distribution package to the Soft Font Installer. The basic purpose of this file is to link each downloadable font file to its corresponding .PFM and screen font file.

The soft font install directory file should be placed on the first disk in a package of soft fonts, or its location should be clearly marked on the disk that contains it.

This chapter describes the format of the file. In this explanation, the following syntax conventions are used:

CAPS indicates a reserved keyword required by the SFINSTAL.DIR file  
[ ] (square brackets) indicate the enclosed item is optional  
,... means the description may be repeated

Comments, if included in SFINSTAL.DIR, have the format:

```
/* comment */
```

The comment must begin with /\* and end with \*/. Comments may not be nested (i.e., /\* within a comment will be ignored, and the first occurrence of \*/ will end the comment).

The SFINSTAL.DIR file consists of two primary items:

1. The logical drive definition
2. The font family definition

A description of each of these items follows.

### 4.1 Logical Drive Definition

The purpose of this command is to allow the font vendor to describe the location of files that are not on the current disk. The DRIVE definition must appear before it is used anywhere in the file. It is highly recommended that a logical drive be defined for each floppy disk in the distribution package, and that all files referenced be preceded by a logical drive ID.

The format of the logical drive definition is:

```
DRIVE id[:] = label-file [, "label-descriptor" ]
```

A description of the fields follows:

| Field                   | Description  |
|-------------------------|--|
| <b>id</b>               | The logical drive's identifier. The <i>id</i> is used as a drive specifier to the file name. For example:<br><br><b>DRIVE cen1:=CEN1.LBL, "Century Schoolbook disk #1"</b><br><br>sets up a logical drive, cen1.   |
| <b>label-file</b>       | The file for which the SoftFont Installer will look. The label file must use the .LBL file extension (for an explanation, see Section 4.3, "No SFINSTAL.DIR file").<br><br>When a filename is referenced as CEN1:CN010RPN.USP, the Soft Font Installer will look for the file CEN1.LBL. If present, the installer will load CN010RPN.USP; if not present, it will prompt the user, for example:<br><br>Please place Century Schoolbook disk #1 into drive A:<br><br>After the user clicks OK on the prompt (assuming the disks have been switched), the installer will again look for CEN1.LBL and, if present, will load CN010RPN.USP. If not present, the installer will repeat the prompt until the user has inserted the correct disk. |
| <b>label-descriptor</b> | A general description of the floppy disk. This argument is optional. If not present, the Soft Font Installer will prompt:<br><br>Please place cen1 into drive A:<br><br>The installer uses the name of the label file as the descriptor string. Quotes around the description are required.  |

## 4.2 Font Family Definition

The format of the font family definition is:

```
FAMILY [ "family-name" ] {  
    aspect-ratio = "description", screen-font-file  
    "font-description" = orient, [ download-file ] [, PFM-file ]  
}
```

Where:

The first line is the format for specifying a screen font file.

The second line is the format for specifying a downloadable font file.

Both of these lines can be repeated or eliminated.

A description of the fields follows:

| Field        | Description  |
|--------------|--|
| FAMILY       | The FAMILY keyword groups together a list of download files and their .PFM files to their corresponding screen font files. Each sub-entry, <i>aspect-ratio=</i> and <i>font-description=</i> , may appear many times and in any order within the braces { }.   |
| family-name  | A general description of the font family.  |
| aspect-ratio | The beginning of a screen font definition. Its form is as follows:<br><br>screen-width-in-lines/inch : screen-depth-in-lines/inch,...<br><br>Notice that both the aspect ratio and the screen resolution are built into the aspect-ratio field. Examples of valid fields are as follows: <ul style="list-style-type: none"> <li>• 72:72, 96:96, 108:108, and 144:144 for 1:1 screens at different resolutions (IBM Personal System/2 is 72:72)</li> <li>• 96:72 for the IBM Enhanced Graphics Adapter</li> <li>• 96:48 for the IBM Color Graphics Adapter</li> </ul> Multiple aspect ratios may be listed in one aspect-ratio field, separated by commas. For example, 72:72, 96:96 = would be a valid aspect-ratio specification.<br><br>If the installer installs fonts on a display device with no matching resolution, but with matching aspect ratios, then the installer will choose the aspect-ratio field whose resolution is less than or equal to the desired resolution. (If there is no resolution less than or equal to the desired resolution, then it will choose the lowest resolution.) If there is no screen font with a matching aspect ratio, then no screen fonts will be loaded. |
| description  | The "descriptive-text" string used to describe the screen font when it was compiled as a resource. That is, when the screen font was made, a special entry in a Windows .DEF file was made for the font in the form:<br><br><b>DESCRIPTION FONTRES FontTypeList : "descriptive-text"</b><br><br>The descriptive-text string from the .DEF file should be repeated for the description field in the SFINSTAL.DIR file. The Windows 1.04 Programming Guide, Appendix B, describes screen font resources in detail. If you are unsure of the string, then use the Windows Control Panel to load the screen font. In the WIN.INI file, examine the section labeled [FONTS]. Whatever you see to the left of the equal sign (=) for your screen font is the string that should appear in the description field.   |

| Field              | Description  |
|--------------------|--|
|                    | Examples of description fields are as follows:   |
|                    | <ul style="list-style-type: none"><li>• "Courier 8,10,12 (Set #3)"</li><li>• "Roman (Set #1)"</li><li>• "PageMaker Fonts (1:1)"</li></ul>  |
|                    | Quotes around the description field are required.  |
| screen-font-file   | The name and location of the screen font file. If you use a logical drive as part of the name, then you guarantee that the installer will be able to find the file (i.e., it will prompt the user to change disks if the incorrect disk is in drive A). Also, the path should be included in the filename if it is not in the root directory on the disk.  |
| "font-description" | The name that is listed in the installer's listbox. It should include the following:   |
|                    | <ul style="list-style-type: none"><li>• Exact face name</li><li>• Point size (abbreviating "point" to "pt" is recommended)</li><li>• "bold" if the font is a bold face</li><li>• "italic" if the font is italic face</li></ul>   |
|                    | Examples of font-description strings are as follows:   |
|                    | <ul style="list-style-type: none"><li>• "Tms Rmn 10pt"</li><li>• "Helv 14pt bold"</li><li>• "Century Schoolbook 8pt bold italic"</li></ul>   |
|                    | Quotes around the font-description string are required.  |
| orient             | The font's orientation (portrait or landscape). <i>Orient</i> is either P (or p) for portrait orientation or L (or l) for landscape orientation. PL or LP may be used to indicate either orientation.  |
| download-file      | The name and location of the downloadable font file. If you use a logical drive as part of the name, then you guarantee that the installer will be able to find the file (i.e., it will prompt the user to change disks if the incorrect disk is in drive A). Also, the path should be included in the filename if it is not in the root directory on the disk. To omit the download file, use two commas before the name of the PFM-file: |
|                    | <p>"font-description" = orient,,PFM-file</p>   |
|                    | If the download file name is omitted, the PFM-file name must be present.   |

| Field     | Description  |
|-----------|--|
| PFM-file  | <p>The name and location of the printer font metrics file. If you use a logical drive as part of the name, then you guarantee that the installer will be able to find the file (i.e., it will prompt the user to change disks if the incorrect disk is in drive A). Also, the path should be included in the filename if it is not in the root directory on the disk.</p>  |
|           | <p>If the PFM-file is omitted, then the Soft Font Installer will attempt to generate a .PFM file by reading metrics from the downloadable font file. In general, this approach is discouraged as the font manufacturer then relies on the driver's PFM Generator to correctly build .PFM files. This is also an inconvenience to the user as he or she has to sit through the process of watching the Soft Font Installer build the PFM files. Because we have a PFM Editor, you should be able to create a .PFM file very easily.</p> |
|           | <p>The font manufacturer can demonstrate more control over the .PFM files and provide a convenience to the user by supplying the .PFM files on the distribution disks.</p>   |
| CARTRIDGE | <pre>CARTRIDGE {     aspect-ratio = "description", screen font filename     "cartridge title" = PCM-file }</pre>   |
|           | <p>This defines an external cartridge for installation.<br/> Where: The aspect-ratio, "description", and font filename are the same as for soft fonts. The cartridge title is the title of a cartridge that can be installed. This title will appear in the installer's listbox. Both of these lines can be repeated or eliminated.<br/> The PCM-file is the file containing the cartridge information (the collection of PFMs). This file can also include a label.</p>   |

### 4.3 Installing Without the SFINSTAL.DIR File

If there is no soft font directory file, the installer will perform the following actions:

1. If a file on the disk exists with the .LBL extension, the installer assumes the SFINSTAL.DIR file is on another disk. It prompts the user to switch disks, while allowing the user the option to read the disk without the directory file.
2. If there is no .LBL file, or the user chooses the option in #1, the installer assumes the directory file does not exist. It then scans the disk looking for downloadable font files.

When the installer has to scan the disk for soft fonts, it opens each file on the disk and looks for the header of a downloadable font file. If it finds the header, it derives the font name and point size, and displays this to the user in the listbox of soft fonts available for installation. When the font is installed, the installer will generate a .PFM file from the downloadable font file. Without the SFINSTAL.DIR file, it will ignore any .PFM files already on the disk.

#### 4.4 Sample SFINSTAL.DIR Directory File

```
/* Acme Corporation's font package.  
 *  
 * This package consists of two floppy disks, one  
 * containing the Tms Rmn soft font set, the other  
 * containing the Helvetica soft font set.  
 */  
  
/* Logical drives  
 */  
DRIVE TR1: = TRSET1.LBL, "Tms Rmn set (disk 1 of 2)"  
DRIVE HV1: = HVSET1.LBL, "Helv set (disk 2 of 2)"  
  
/* Tms Rmn set  
 */  
FAMILY "Tms Rmn" {  
    /* screen fonts */  
    1:1 = "Acme Tms 1:1", TR1:TR11.FON  
    4:3 = "Acme Tms 4:3", TR1:TR43.FON  
    2:1 = "Acme Tms 2:1", TR1:TR21.FON  
  
    /* printer fonts */  
    "Tms Rmn 6pt"      = P,TR1:TR060RPN.USP,TR1:TR060RPP.PFM  
    "Tms Rmn 6pt"      = L,TR1:TR060RPN.USL,TR1:TR060RPL.PFM  
    "Tms Rmn 6pt bold" = P,TR1:TR060BPN.USP,TR1:TR060BPP.PFM  
    "Tms Rmn 6pt bold" = L,TR1:TR060BPN.USL,TR1:TR060BPL.PFM  
    .  
    .  
    .  
    "Tms Rmn 30pt bold" = L,TR1:TR300BPN.USL,TR1:TR300BPL.PFM  
    "Tms Rmn 30pt italic" = P,TR1:TR300IPN.USP,TR1:TR300IPP.PFM  
    "Tms Rmn 30pt italic" = L,TR1:TR300IPN.USL,TR1:TR300IPL.PFM  
}  
  
/* Helv set  
 */  
FAMILY "Helv" {  
    /* screen fonts */  
    1:1 = "Acme Helv 1:1", HV1:HV11.FON  
    4:3 = "Acme Helv 4:3", HV1:HV43.FON  
    2:1 = "Acme Helv 2:1", HV1:HV21.FON  
  
    /* printer fonts */  
    "Helv 6pt"          = P,HV1:HV060RPN.USP,HV1:HV060RPP.PFM  
    "Helv 6pt"          = L,HV1:HV060RPN.USL,HV1:HV060RPL.PFM  
    "Helv 6pt bold"     = P,HV1:HV060BPN.USP,HV1:HV060BPP.PFM  
    "Helv 6pt bold"     = L,HV1:HV060BPN.USL,HV1:HV060BPL.PFM  
    .  
    .  
    .  
    "Helv 30pt bold"    = L,HV1:HV300BPN.USL,HV1:HV300BPL.PFM  
    "Helv 30pt italic"   = P,HV1:HV300IPN.USP,HV1:HV300IPP.PFM  
    "Helv 30pt italic"   = L,HV1:HV300IPN.USL,HV1:HV300IPL.PFM
```

```
)  
  
/* Cartridge version  
 */  
CARTRIDGE {  
    /* screen fonts */  
    1:1 = "Acme Helv 1:1", HV1:HV11.FON  
    4:3 = "Acme Helv 4:3", HV1:HV43.FON  
    2:1 = "Acme Helv 2:1", HV1:HV21.FON  
    1:1 = "Acme Tms 1:1", TR1:TR11.FON  
    4:3 = "Acme Tms 4:3", TR1:TR43.FON  
    2:1 = "Acme Tms 2:1", TR1:TR21.FON  
  
    /* cartridge containing the fonts */  
    "Acme Font Ser" = TR1:ACMECART.PCM  
}
```



## **CHAPTER 5**

### **DEVELOPERS' TOOLS**

#### **CONTENTS**

---

- |     |                         |     |
|-----|-------------------------|-----|
| 5.1 | Build SFINSTAL.DIR File | 5-3 |
| 5.2 | Add Fonts               | 5-4 |
| 5.3 | Enable Edit Button      | 5-6 |

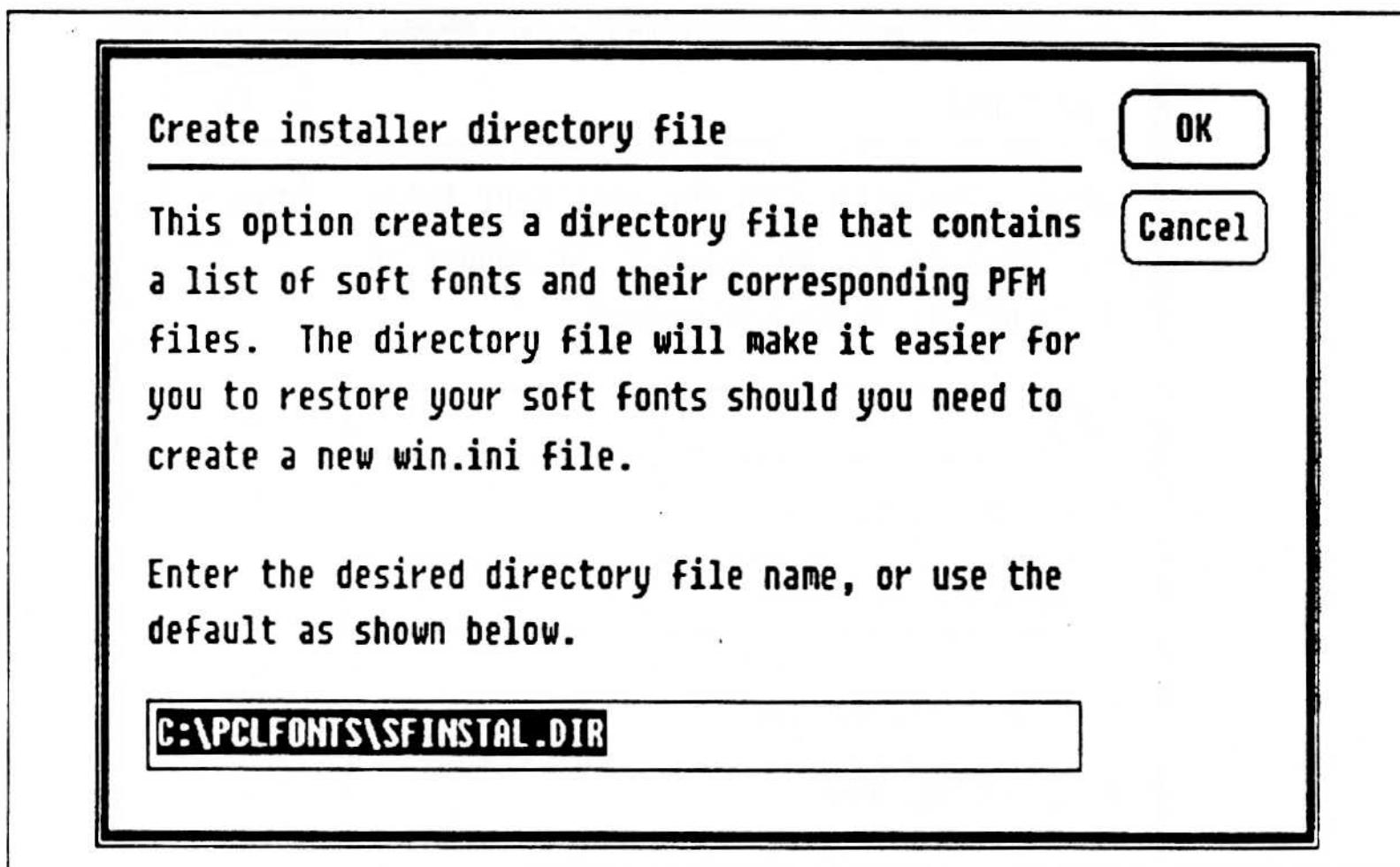


## 5 DEVELOPERS' TOOLS

This chapter presents the features of the Soft Font Installer that are not in the user documentation. These features were specifically put in the installer to help font vendors and technical support specialists.

### 5.1 Build SFINSTAL.DIR File

The installer will build a SFINSTAL.DIR file for the fonts listed in the *left* listbox of the installer dialog. The developer should hold down the CTRL and SHIFT keys while clicking the mouse on the "Exit" button. The installer will prompt the user for the name and path of the SFINSTAL.DIR file (see illustration). If a file by the same name already exists, the installer will prompt to replace the file.



This feature has several uses:

1. To preserve the soft font and cartridge entries when the user wishes to re-install Windows.
2. To expedite moving fonts to another machine.
3. As a starter file for developers or power users who wish to set up a distribution disk of fonts.

This feature is primarily intended to be used by technical support specialists who are directing users to perform certain operations. A user may be advised to erase part or all of the contents of his or her machine and start over. This feature is very useful in preventing the user from losing soft font information.

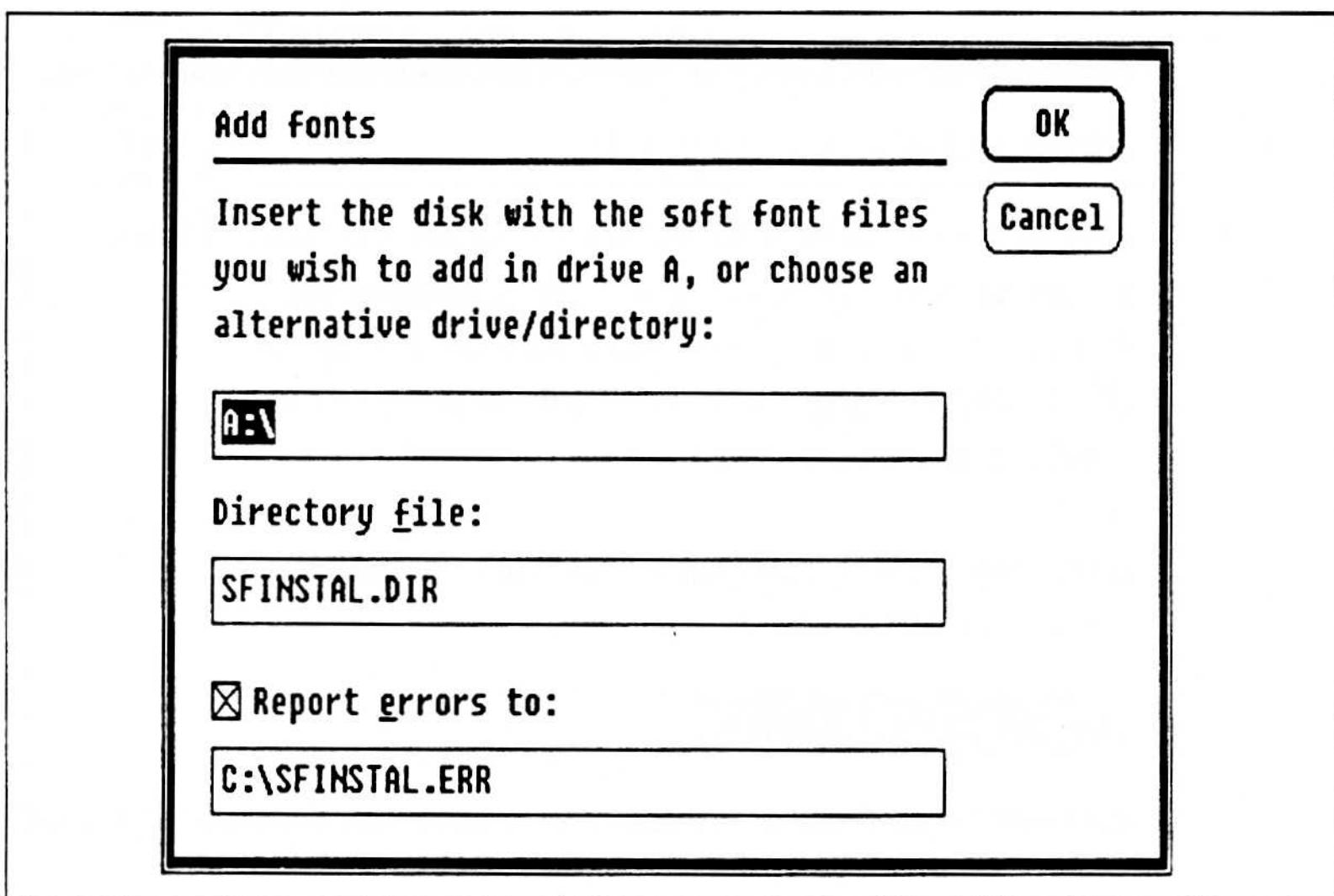
Chapter 7, "Installer Scenarios," provides sample uses of this feature.

## 5.2 Add Fonts

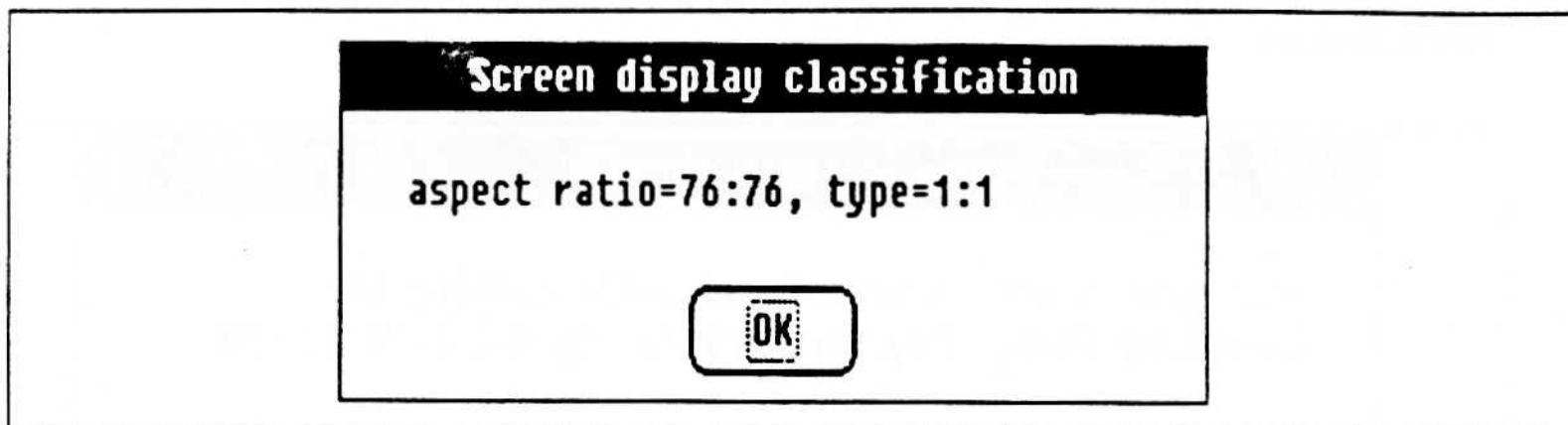
Developers can gain access to an advanced "Add fonts..." dialog (see illustration) by holding down the CTRL and SHIFT keys when clicking the mouse on the "Add fonts..." button. This dialog gives developers the option to:

1. Specify the path from which the installer should read fonts.
2. Rename the SFINSTAL.DIR file to something else.
3. Indicate if an error file should be written and specify its name.

The first item is available to users via the normal "Add fonts..." dialog. The other two items are intended to be used by font vendors or power users setting up SFINSTAL.DIR files for distribution.



Before parsing the SFINSTAL.DIR file, the installer will report the screen aspect ratio (see illustration on next page). The installer will classify the screen type as either CGA (2:1), EGA(4:3), or VGA(1:1). It uses the same algorithm to classify the screen fonts listed in the SFINSTAL.DIR file. The fonts that match the category in which the screen was classified would be loaded by the installer.



The formula the installer uses to classify screens is as follows:

1. Compute the screen width in lines per inch (horizontal resolution divided by horizontal size converted from millimeters to inches):

$$\text{width} = (\text{GetDeviceCaps(HORZ_RES)} * 25) / \text{GetDeviceCaps(HORZSIZE)}$$

2. Compute the screen height in lines per inch (vertical resolution divided by vertical size converted from millimeters to inches):

$$\text{height} = (\text{GetDeviceCaps(VERT_RES)} * 25) / \text{GetDeviceCaps(VERTSIZE)}$$

3. Take the inverse ratio of width to height:

$$\text{ratio} = (\text{height} * 10000) / \text{width}$$

4. Categorize the screen type based upon the ratio:

|                |           |
|----------------|-----------|
| 0 to 6249      | CGA (2:1) |
| 6250 to 9374   | EGA (4:3) |
| 9375 and above | VGA (1:1) |

It is highly recommended that font vendors use the advanced "Add fonts..." dialog to verify that there are no syntax errors in the SFINSTAL.DIR file. If the installer finds errors in the SFINSTAL.DIR file, it will write error messages in the SFINSTAL.ERR file. The messages will be of the form:

line <line-number>, near character <character position>: <error message>

The installer also displays a dialog indicating that errors were found and written to the file (see illustration).



### **5.3 Enable Edit Button**

By default, the installer dialog is pulled up with the "Edit" button invisible. The user has to pull down the system menu and select "Enable edit button." However, the installer dialog may be pulled up with the "Edit" button enabled by holding down the **CTRL** and **SHIFT** keys while clicking the mouse on the "Fonts" button from the printer-specific dialog.

The purpose of this feature is to expedite the use of the installer for developers who often exit and re-enter the installer and are editing fonts in the process.

# **CHAPTER 6**

## **.PFM GENERATOR**

### **CONTENTS**

---

|       |  |     |
|-------|--|-----|
| 6.1   | Installer PFM Versus Vendor-Supplied PFM | 6-3 |
| 6.2   | File Naming Scheme                       | 6-3 |
| 6.3   | Re-generating .PFM Files                 | 6-3 |
| 6.4   | PFM Data From Font Data                  | 6-4 |
| 6.4.1 | PFM Header                               | 6-5 |
| 6.4.2 | Extended Text Metric Structure           | 6-8 |
| 6.4.3 | Device-Specific Structure                | 6-9 |



## 6 .PFM GENERATOR

This chapter describes the Soft Font Installer's .PFM file generator. When the installer scans a disk for soft fonts and there is no SFINSTAL.DIR file, it assumes it will have to build .PFM files from the downloadable font files. It reads the header from the font file plus the widths of each individual character and uses that information to build the PFM file.

Please notice that when the installer scans a disk for fonts and there is no SFINSTAL.DIR file, it will ignore any .PFM files already on the disk and will attempt to generate .PFM files from the font files. It will, however, recognize cartridge .PCM files.

### 6.1 Installer PFM Versus Vendor-Supplied PFM

It is highly recommended that font vendors provide the .PFM files. The installer will supply a generic .PFM file that contains width table information and a guess at the font name. .PFM files provided by the vendor are guaranteed to contain the correct name and may include pair kern tables. .PFM files from the vendor will more accurately represent the font and give the vendor a competitive advantage over fonts from other vendors. We have included the PFM Editor as a tool now for generating or editing .PFM files. See the *Microsoft Windows Fonts Guide*, Chapter 4, "The PFM Editor," for a complete description of this new tool.

### 6.2 File Naming Scheme

The installer builds the .PFM file name as follows:

| Characters | Definition   |
|------------|--|
| 1, 2       | First two characters from the font file name                     |
| 3          | P, L, or X for portrait, landscape, or both orientations         |
| 4          | B, R, I, or E for bold, roman, italic, or bold/italic (enhanced) |
| 5, 6, 7    | Point size in points   |
| 8          | Magic number   |

The "magic number" is used by the installer to ensure that the file name is unique. It starts out at 0. If a file already exists by that name, it is incremented to 1. The installer continues to increment the number to 9, then A through Z. If it fails to find a unique file name out of the 36 possibilities, then it fails to build the .PFM file (it reports a "Cannot build PFM file" error message).

### 6.3 Re-generating .PFM Files

This section deals with situations in which the user is re-installing existing fonts onto themselves without the aid of the SFINSTAL.DIR file. The installer contains a protection mechanism that prevents it from generating duplicate .PFM files. There are two potential situations:

1. The user re-installs fonts with .PFM files that were initially generated by the installer.
2. The user re-installs fonts that were initially provided by the font vendor (i.e., either on the distribution disks or built by a utility provided by the font vendor).

When the user "re-installs" fonts, he or she pulls up the Soft Font Installer, selects "Add fonts..." and directs the installer to read font information from the directory containing already installed fonts. Then the user installs fonts into the same directory. The installer will prompt the user if fonts should be replaced (see illustration). If the user responds "yes," the installer proceeds to replace every font with itself.



As described in the previous section, the installer contains a mechanism that guarantees it will find a unique name for each .PFM file. In this situation, the installer would uniquely name a .PFM file that was a duplicate of an existing .PFM file. For example, suppose the first time a user installed a font, the installer generated a .PFM file name XYZ0.PFM. When the user re-installed the font, the installer would generate a second identical .PFM file named XYZ1.PFM.

To prevent duplicate .PFM files on the disk, the installer contains a protection mechanism:

If the "magic number" in the .PFM file name is not 0, then the installer compares the contents of the .PFM file to the other .PFM files sharing the same name. If it finds a duplicate, it erases the .PFM file it just built and uses the already existing .PFM file.

This mechanism prevents the installer from placing duplicate .PFM files on the disk. However, it does not take care of the problem when .PFM files were originally supplied by the font vendor.

If the user originally installs fonts provided by a font vendor and then re-installs without use of a SFINSTAL.DIR file, the installer will *replace* the vendor-supplied .PFM files with its own generated .PFM files. It does not remove the .PFM files from disk; rather it adds its own .PFM files and uses them. Because the installer's PFM generator is not as accurate as the font vendor's .PFM files, the installer may actually end up *renaming* the fonts, which causes confusion for the user.

The only solution to this problem is "don't do that!" The user should never re-install soft fonts from and to the same directory without the aid of a SFINSTAL.DIR file. Because the installer contains the ability to automatically generate a SFINSTAL.DIR file of the already loaded fonts, there really is no reason for re-installing fonts without the use of a SFINSTAL.DIR file.

Section 7.3, "Recovering Soft Fonts From a WIN.INI Change," presents how the user would preserve the installed fonts using a SFINSTAL.DIR file.

#### **6.4 PFM Data From Font Data**

The installer's PFM generator derives the .PFM file from the data in the downloadable font file. The structure of the .PFM file is documented in the *Microsoft Windows Fonts Guide*. The following sections list where the data comes from for each field in the .PFM file structures.

#### 6.4.1 PFM Header

The installer derives the PFM header from the download file as follows:

| Field             | Computation   |
|-------------------|---|
| dfVersion         | 256   |
| dfSize            | Size of file  |
| dfCopyright       | Copyright string from font file header                                      |
| dfType            | 128   |
| dfPoints          | Height of the font in points (height from header * 72 / 1200)               |
| dfVertRes         | 300   |
| dfHorizRes        | 300   |
| dfAscent          | Baseline position from font header  |
| dfInternalLeading | (Cell height - height + 2) / 4 from font header                             |
| dfExternalLeading | (height / 4) - baseline from font header                                    |
| dfItalic          | Set if the style byte in the header = 1                                     |
| dfUnderline       | 0   |
| dfStrikeOut       | 0   |
| dfWeight          | Derived from stroke weight (see table on following page)                    |
| dfCharSet         | Based upon HP symbol set (see table on following page)                      |
| dfPixWidth        | If variable width, set to zero; if fixed pitch, set to width                |
| dfPixHeight       | Cell height from the font header  |
| dfPitchAndFamily  | Pitch bit is set if fixed pitch, family is derived from the typeface number |
| dfAvgWidth        | Average of all the character widths (cursor move)                           |
| dfMaxWidth        | Maximum of all the character widths (cursor move)                           |
| dfFirstChar       | First character in the font file  |
| dfLastChar        | Last character in the font file   |

| Field              | Computation   |
|--------------------|---|
| dfDefaultChar      | 127 - first character   |
| dfBreakChar        | 32 - first character  |
| dfWidthBytes       | 0   |
| dfDevice           | Offset from beginning of file to device string "PCL / HP LaserJet"                                  |
| dfFace             | Offset from beginning of file to face name, derived from the typeface number in the font header     |
| dfBitsPointer      | 0   |
| dfBitsOffset       | 0   |
| dfCharOffset       | If variable pitch, the width table goes here; if fixed pitch, this field is not written to the file |
| dfSizeFields       | Size of this part of the PFM structure  |
| dfExtMetricsOffset | Offset from beginning of file to EXTTEXTMETRIC (Extended Text Metrics) structure                    |
| dfExtentTable      | 0   |
| dfOriginTable      | 0   |
| dfPairKernTable    | 0   |
| dfTrackKernTable   | 0   |
| dfDriverInfo       | Offset from beginning of file to DRIVERINFO (driver-specific) structure                             |
| dfReserved         | 0   |

The installer derives the Windows weight value from the weight value in the font file header as follows:

| HP Weight | Windows Weight    |
|-----------|-------------------|
| -7        | 100 (Thin)        |
| -6        | 100 (Thin)        |
| -5        | 200 (Extra light) |
| -4        | 200 (Extra light) |
| -3        | 300 (Light)       |
| -2        | 300 (Light)       |
| -1        | 400 (Normal)      |
| 0         | 400 (Normal)      |
| 1         | 500 (Medium)      |
| 2         | 600 (Semi-bold)   |
| 3         | 700 (Bold)        |
| 4         | 700 (Bold)        |
| 5         | 800 (Extra bold)  |
| 6         | 800 (Extra bold)  |
| 7         | 900 (Heavy)       |

The installer determines the Windows character set (dfCharSet) and the type of translation (in the driver-specific structure, xtbl.xtSymbolSet) from the symbol set field in the font file:

| HP Symbol Set     | Windows Character Set | Translation              |
|-------------------|-----------------------|--------------------------|
| 8U (Roman-8)      | 0 (ANSI)              | Roman-8                  |
| 0U (USASCII)      | 0 (ANSI)              | USASCII                  |
| 11Q, 0N (ECMA-94) | 0 (ANSI)              | 8-bit pass through       |
| 8M (Math-8)       | 180 (Math-8)          | 8-bit pass through       |
| 15U (PI Font)     | 181 (PI Font)         | 8-bit pass through       |
| 0B (LineDraw)     | 182 (LineDraw)        | 7-bit pass through       |
| 4Q (PC Line)      | 183 (PC Line)         | 7-bit pass through       |
| 0B (TaxLnDrw)     | 184 (Tax Line)        | 7-bit pass through       |
| 1U (US Legal)     | 185 (US Legal)        | 7-bit pass through       |
| <i>all others</i> | 0 (ANSI)              | 7- or 8-bit pass through |

For the generic case, the installer checks the font type byte in the font header. If it is non-zero, the font uses an 8-bit pass through; if zero, the font uses a 7-bit pass through.

#### 6.4.2 Extended Text Metric Structure

The installer derives extended text metrics from the download header file as follows:

| Field                         | Computation   |
|-------------------------------|---|
| etmSize                       | Size of EXTTEXTMETRIC structure   |
| etmPointSize                  | Cell Height *1440 / 300 from font file header   |
| etmOrientation                | 2 if orientation byte set in header, 1 if not   |
| etmMasterHeight               | Cell height from font file header   |
| etmMinScale                   | etmMasterHeight   |
| etmMaxScale                   | etmMasterHeight   |
| etmMasterUnits                | etmMasterHeight   |
| etmCapHeight                  | Baseline from header - dfInternalLeading  |
| etmXHeight                    | Top offset from character record for 'x', left offset if landscape font                     |
| etmLowerCaseAscent            | Top offset from character record for 'd', left offset if landscape font                     |
| etmLowerCaseDescent           | Character width - left offset from character record for 'p', height - top if landscape font |
| etmSlant                      | 0   |
| etmSuperScript                | etmCapHeight - etmXHeight   |
| etmSubScript                  | Cell height from header   |
| etmSuperScriptSize            | Cell height from header   |
| etmSubScriptSize              | Cell height from header   |
| etmUnderlineOffset            | Top offset from character record for '_', left offset if landscape font                     |
| etmUnderlineWidth             | Character width from character record for '_', character height if landscape font           |
| etmDoubleUpperUnderlineOffset | etmUnderlineOffset + etmUnderlineWidth * 2  |
| etmDoubleLowerUnderlineOffset | etmUnderlineOffset  |
| etmDoubleUpperUnderlineWidth  | etmUnderlineWidth   |
| etmDoubleLowerUnderlineWidth  | etmDoubleUpperUnderlineWidth  |

| Field              | Computation   |
|--------------------|---|
| etmStrikeOutWidth  | Character width from record for '-', character height if landscape font   |
| etmStrikeOutOffset | Left offset - etmStrikeOutWidth from character record for '-', top offset - etmStrikeOutWidth if landscape font |
| etmKernPairs       | 0   |
| etmKernTracks      | 0   |

#### 6.4.3 Device-Specific Structure

The format of this structure is presented in Section 7.7, "Setting Up .PFM Files for Resident and Cartridge Fonts."

| Field            | Computation  |
|------------------|--|
| epSize           | Size of DRIVERINFO structure   |
| epVersion        | 1  |
| epMemUsage       | ((sum of all character widths + 7) >> 3) * height + 63, function from the <i>HP Technical Reference Manual</i> |
| epEscape         | 0  |
| xtbl.xtSymbolSet | Based upon font symbol set (see HP symbol set table on previous page)  |
| xtbl.xtOffset    | 0  |
| xtbl.xtLen       | 0  |
| xtbl.xtFirstChar | 0  |
| xtbl.xtLastChar  | 0  |



## **CHAPTER 7**

### **INSTALLER SCENARIOS**

#### **CONTENTS**

---

|     |  |      |
|-----|--|------|
| 7.1 | Selecting Printer Fonts                                | 7-3  |
| 7.2 | Selecting Screen Fonts                                 | 7-5  |
| 7.3 | Recovering Soft Fonts From a WIN.INI File              | 7-6  |
| 7.4 | A Quick Method for Moving Fonts                        | 7-7  |
| 7.5 | How to Build a Floppy Disk Set of Fonts                | 7-9  |
| 7.6 | Setting Up Fonts on a Network                          | 7-10 |
| 7.7 | Setting Up .PFM Files for Resident and Cartridge Fonts | 7-12 |
| 7.8 | Sample SFINSTAL.DIR File                               | 7-15 |



## 7 INSTALLER SCENARIOS

This chapter presents some sample uses of the Soft Font Installer. It starts with recommendations for selecting printer and screen fonts, then presents some "cookbook" approaches to performing certain tasks with the Soft Font Installer.

We assume that the reader understands how to use the Soft Font Installer and has a basic understanding of DOS. The *Microsoft Windows User's Guide* describes the user interface to the installer.

We recommend that all technical support specialists also read this chapter.

### 7.1 Selecting Printer Fonts

Fonts take up disk space and printer memory. Planning which fonts are needed and installing only those fonts can save disk space. Selecting fonts is a highly subjective process. It requires making trade offs among document design and computer memory, disk space, printer memory, and printing speed.

The recommendations presented in this section and Section 7.2, "Selecting Screen Fonts," are based upon experience gained from working with Windows and soft fonts. The intent of these recommendations is to help you optimize the amount of disk space and printer memory used by soft fonts. Some of the recommendations indicate you should not load certain variations of fonts; you may want to experiment to determine if the additional expense in print time and disk space is worth including these fonts.

We recommend loading the typographic range of point sizes and limiting documents to using these sizes. The typographic range is:

6, 7, 8, 9, 10, 11, 12, 14, 18, 24, 30, 48, 60, 72

Better yet, if you know exactly which sizes you normally use in a document, load only those sizes. For example, a document may use only the following point sizes of Bitstream Optima (Zapf Humanist):

11 point for body text  
9 point for running headers, running footers, and footnotes  
14, 24, and 30 point bold for subheads and headlines.

By loading only these sizes, you would reduce the number of fonts taking up disk space.

We recommend generating normal and italic for all point sizes, and bold for point sizes equal to and above 14 points. The PCL driver has the capability to simulate bold text. It will not look the same as the true bold face, so you may want to experiment first (e.g., print a document with bold type but with no bold face loaded and, then, load the bold face and reprint the document). Normally, however, the driver-simulated bold face is adequate for the small point sizes.

To save on even more space, we recommend you load bold *only* for the larger sizes (i.e., 24 points and above). The assumption is you will only use bold sizes for headlines.

Avoid loading bold italic for a font. Bold italic consumes disk space and printer memory and, because bold italic is rarely used, it does not warrant the space it requires. As an alternative, the driver will simulate bold italic by synthetically bolding the italic face.

Character sets can make a big difference in the amount of disk space and printer memory used by a font. If the document never uses accented characters or special symbols like the bullet (centered period) and copyright, we recommend using the USASCII character set. If the document requires these characters, use the Windows ANSI, ECMA-94, or Roman-8 character sets (listed in order of preference, depending upon which sets you have access to).

Fonts in the USASCII character set contain a little more than half the number of characters contained in the Windows ANSI, ECMA-94, and Roman-8 character sets. By selecting USASCII, you effectively halve the amount of disk space and printer memory used by the font. We recommend using, in order of preference, Windows ANSI, ECMA-94, or Roman-8 for point sizes below 14 points and USASCII for point sizes 14 points and above. Figure 7.1 summarizes these recommendations.

|           |    | <b>Normal</b> | <b>Bold</b> | <b>Italic</b> | <b>ANSI</b> | <b>USASCII</b> |
|-----------|----|---------------|-------------|---------------|-------------|----------------|
| Footnotes | 6  | x             |             | x             | x           |                |
|           | 7  | x             |             | x             | x           |                |
| Body Text | 8  | x             |             | x             | x           |                |
|           | 9  | x             |             | x             | x           |                |
|           | 10 | x             |             | x             | x           |                |
|           | 11 | x             |             | x             | x           |                |
|           | 12 | x             |             | x             | x           |                |
| Headers   | 14 | x             | x           | x             |             | x              |
|           | 18 | x             | x           | x             |             | x              |
|           | 24 | x             | x           | x             |             | x              |
|           | 30 | x             | x           | x             |             | x              |
|           | 48 | x             | x           | x             |             | x              |
|           | 60 | x             | x           | x             |             | x              |
|           | 72 | x             | x           | x             |             | x              |

Further optimizations:

- Choose 1 footnote size (recommended = 6 point)
- Choose 2 or 3 body text sizes (recommended = 8, 10, and 12 point)
- Choose 4 headline sizes (recommended = 14, 24, 30, and 72 point)
- Generate only bold for the headline sizes

*Figure 7.1 Recommended Printer Font Sizes*

## 7.2 Selecting Screen Fonts

Selecting screen fonts is considerably different from selecting printer fonts. The selection of screen font sizes should not be based upon the selected printer font sizes! It does not make sense to build one corresponding screen font for every printer font for the following reasons:

1. Screen fonts in Windows are sized and modified by the screen driver. The screen driver can derive normal, bold, italic, and bold italic variations from one screen font.
2. Screen fonts are selected by way of a generic selection process controlled by the application and the screen driver. One cannot assume that a screen font specifically targeted for a printer font will be used to display that font.
3. Some applications allow the user to see the font at different views (e.g., "fit in window," 100%, and 200%). A screen font should be provided for each view.
4. Most important, screen fonts take dynamic memory away from applications, thus slowing the overall performance of Windows.

Poorly selected screen fonts can both degrade the performance of the system and produce highly confusing and ugly screen display.

Regardless of the printer font sizes and variations selected, we recommend loading a shortened typographic range of screen fonts:

8, 9, 10, 12, 14, 18, 24

Load these fonts in normal face only; the screen driver can simulate bold, italic, and bold italic.

Do not load screen fonts for printer fonts that are like *Tms Rmn*, *Helv*, or *Courier*. Windows already provides those screen fonts. Additional screen fonts would use up memory without additional visual benefits.

If you generate one printer font, you should generate the whole recommended range. Applications typically offer more than one view of the page, so the screen driver will need the different sizes to display the font in each view.

The screen driver does an excellent job of synthesizing bold, italic, and bold italic from the normal face font. There is no need to double, triple, or quadruple the memory used by screen fonts to get these special fonts.

If you feel you want the typographic quality of pre-built bold and italic screen fonts, then you should experiment to see if these fonts look better than fonts synthesized by the screen driver. Display a page of text with only the normal face loaded, then display it with the normal, bold, and italic faces loaded. Never load a bold or italic screen font without a corresponding normal face font. Without the normal face, the screen driver will always use the bold or italic face, producing an extremely confusing screen display.

For example, suppose you decide to create a document that uses 24-point italic Century Schoolbook. You load these fonts:

one printer font, Century Schoolbook 24-point italic  
one screen font, Century Schoolbook 24-point italic

Every time the screen driver determines it needs a 24-point screen font that is close to Century Schoolbook, it uses the 24-point italic face. Regardless of the variation (normal, bold, italic), the font will display in italic.

As you add more fonts to your computer, the situation will become more confusing. To continue the example, suppose you decide to add 48-point normal. You load these fonts:

one printer font, Century Schoolbook 48-point normal  
one screen font, Century Schoolbook 48-point normal

When the application displays the 48-point font at 100% view on the screen, the font will display as 48-point normal (correct behavior). At 50% view, the font will display as 24-point italic (because that is the only font available). When switching views, the font changes from normal to italic.

You can avoid this problem by simply *not* loading anything other than the normal face of the screen font. Normally, the difference between the synthesized font and the "true" font is barely noticeable, if noticeable at all. If you must load the italic or bold variations, then you must also load the normal variation in the same size.

### **7.3 Recovering Soft Fonts From a WIN.INI Change**

Sometimes you need to remove the contents of your WINDOWS directory and reinstall Windows. In the process of doing this, you could lose all the soft font information. To prevent losing data:

1. Build a SFINSTAL.DIR file.

From inside Windows, start up the Control Panel, select Setup, Printer, and the PCL / HP LaserJet printer driver. If you have an application, such as PageMaker, and runtime Windows, run the application and select "Change printer" or "Printer setup" from the file menu or start up the Control Panel from the PageMaker system menu. Select "PCL / HP LaserJet" and click on "Setup."

Once inside the driver-specific dialog, click on the "Fonts" button to invoke the Soft Font Installer. The fonts you plan to save should be listed in the left listbox. Hold down the CTRL and SHIFT keys while clicking the mouse on the "Exit" button. The installer prompts for a file name and path. Click "OK" to accept the name of the SFINSTAL.DIR file. Exit back to DOS by clicking "OK" on the driver-specific dialog and selecting "Close" from the MS-DOS Executive system menu (or exit out of the application if you are running one).

2. Reinstall Windows.

Perform whatever steps are necessary to reinstall Windows (i.e., remove everything in the WINDOWS directory and run Windows install).

3. Install from the directory to itself to get the soft fonts listed.

Run the Soft Font Installer on the new machine. Run Windows Control Panel and select Setup, Printer, and PCL / HP LaserJet to get to the driver-specific dialog. Or, run your application and select File and Printer Setup.

From the driver-specific dialog, click on the "Fonts" button to invoke the installer. If the "Fonts" button is not highlighted, then you need to select a printer that supports soft fonts (lower left listbox in the driver-specific dialog). Select "Add fonts..." and specify \PCLFONTS as the source for copying fonts. If your fonts are stored in another directory (such as FONTS or PCLPFM), then specify that directory.

Hold down the CTRL and SHIFT keys while selecting all the fonts in the right listbox. Select the "Add..." button between the two listboxes. The installer prompts for a target directory. *This must be equivalent to the directory from which you are installing the fonts.* If you are installing from \PCLFONTS, then click "OK" to accept \PCLFONTS as the target directory. Otherwise, change the name to that of the directory that contains the soft fonts (such as FONTS or PCLPFM).

The installer then *very quickly* shuffles the fonts from the right listbox to the left listbox. It does not copy any fonts; it simply updates the WIN.INI file. If this does not happen quickly, then you probably did not build a SFINSTAL.DIR file as described in Step 1, or did not install fonts from and to the same directory.

#### 7.4 A Quick Method for Moving Fonts

The purpose of this section is to describe how to move fonts from one machine to another. It is not intended for distributing fonts to several users. That procedure is described in Section 7.5, "How to Build a Floppy Disk Set of Fonts". This description is not intended to encourage illegal copying of fonts. We assume you are *removing* fonts from one machine and adding them to another.

The steps are as follows:

1. Build a SFINSTAL.DIR file.

From inside Windows, start up the Control Panel, select Setup, Printer, and the PCL / HP LaserJet printer driver. If you have an application and runtime Windows, run the application and select "Change printer" or "Printer setup" from the file menu or start up Control Panel from the application's system menu. Select "PCL / HP LaserJet" and click on "Setup."

Once inside the driver-specific dialog, click on the "Fonts" button to invoke the Soft Font Installer. The fonts you plan to save should be listed in the left listbox. Hold down the CTRL and SHIFT keys while clicking the mouse on the "Exit" button. The installer prompts for a file name and path. Click "OK" to accept the name of the SFINSTAL.DIR file. Exit back to DOS by clicking "OK" on the driver-specific dialog and selecting "Close" from the MS-DOS Executive system menu (or exit out of the application if you are running one).

2. Use the DOS xcopy command to copy files to a floppy disk.

Change to the directory containing the soft fonts. Type:

```
cd \PCLFONTS
```

Format as many floppy disks as you think are necessary to hold the soft fonts. Type:

```
format A:
```

Make sure that all the files in the directory are *not* marked for archive. Type:

```
attrib -a *.*
```

Copy files to the first disk. Type:

```
xcopy *.* A:/m
```

If DOS reports an "insufficient disk space" error, format another floppy disk (if necessary) and repeat the xcopy command. The /m option on the xcopy command instructs DOS to mark the files it has copied to disk, so when you repeat the command it does not recopy the files that are already on disk.

Repeat the xcopy command until all the files are on the disk(s).

**3. Copy the files from the floppy disk(s) to the new machine.**

On the new machine, make a directory for the fonts. Type:

```
cd \
mkdir \PCLFONTS
cd \PCLFONTS
```

Copy the fonts from the floppy disk into the new directory. Type:

```
copy A:/*.*
```

Repeat this step for every floppy disk.

**4. Install from the directory to itself to get the soft fonts listed.**

Run the Soft Font Installer on the new machine. Run Windows Control Panel and select Setup, Printer, and PCL / HP LaserJet to get to the driver-specific dialog. Or, run your application and select File and Printer Setup.

From the driver-specific dialog, click on the "Fonts" button to invoke the installer. Select "Add fonts..." and specify \PCLFONTS as the source for copying fonts. The installer should list all the fonts you just copied into the \PCLFONTS directory.

Hold down the CTRL and SHIFT keys while selecting all the fonts in the right listbox. Select the "Add..." button between the two listboxes. The installer will prompt for a target directory. *This must be the same directory from which you are installing the fonts.* If you are installing from \PCLFONTS, as these instructions indicate, then click "OK" to accept \PCLFONTS as the target directory.

The installer then *very quickly* shuffles the fonts from the right listbox to the left listbox. It does not copy any fonts; it simply updates the WIN.INI file. If this does not happen quickly, then you probably did not build a SFINSTAL.DIR file as described in Step 1, or did not install fonts from and to the same directory.

The fonts will be moved to the new machine. Notice that this procedure is probably the fastest and simplest way to move fonts from one machine to the next. However, we do not recommend it as a general method of distributing fonts. The installer has a more elegant and less error-prone way of loading fonts from floppy disks as described in the next section.

## 7.5 How to Build a Floppy Disk Set of Fonts

This section describes how to set up a floppy disk set of fonts for distribution. This procedure is useful, for example, if a company chooses to use a range of fonts available from a font generation utility. One person could generate the fonts and create a disk set of fonts for internal distribution. Then, all the other users could quickly and easily install the pre-generated fonts without having to learn how to use the font generation utility.

By providing this procedure, we do not encourage illegal copying of fonts. We assume that everyone receiving the fonts has either paid some kind of a licensing fee or has his or her own copy of the font generation utility used to build the fonts.

To use the following procedure, you must know how to use DOS and a text editor. In addition, you should understand the advanced installer features described in Chapter 4, "SFINSTAL.DIR," and Chapter 5, "Developer's Tools."

The steps are as follows:

1. Install and/or generate the soft fonts.

Load the fonts onto your machine. If you are generating soft fonts, follow the recommendations outlined in Section 7.1, "Selecting Printer Fonts."

If you have control over the creation of screen fonts, follow the recommendations outlined in Section 7.2, "Selecting Screen Fonts." You should build at least two sets of screen fonts, one each for:

- EGA with enhanced color and 64k+
- VGA

If you have users with CGA displays, you may want to add a set of screen fonts for CGA displays.

When you make changes to the SFINSTAL.DIR file (see Step 4), you can instruct the installer to load different screen fonts depending upon the aspect ratio of the machine receiving the fonts.

2. Build a SFINSTAL.DIR file.

From inside Windows, start up the Control Panel, select Setup, Printer, and the PCL / HP LaserJet printer driver. If you have an application and runtime Windows, run the application and select "Change printer" or "Printer setup" from the file menu or start up Control Panel from the application's system menu. Select "PCL / HP LaserJet" and click on "Setup".

Once inside the driver-specific dialog, click on the "Fonts" button to invoke the Soft Font Installer. The fonts you plan to save should be listed in the left listbox. Hold down the CTRL and SHIFT keys while clicking the mouse on the "Exit" button. The installer prompts for a file name and path. Click "OK" to accept the name of the SFINSTAL.DIR file: Exit back to DOS by clicking "OK" on the driver-specific dialog and selecting "Close" from the MS-DOS Executive system menu (or exit out of the application if you are running one).

3. Arrange the font files, .PFM files, .PCM files, screen font files, and SFINSTAL.DIR file onto disks.

If all the files fit on one disk, this is an easy task. If they do not and the number of fonts is great, this may be a major task. You will find recommendations for arranging them after this list.

4. Edit the SFINSTAL.DIR file to add logical drives.

If the files require more than one disk, edit the SFINSTAL.DIR file and add logical drives so the installer will know where all the files are. With logical drives properly set up, the installer will prompt the user to switch disks when necessary.

If you loaded screen fonts, add entries in the SFINSTAL.DIR file for screen fonts as well. Place the screen fonts in the appropriate FAMILY block for which the screen fonts were made. The format of the SFINSTAL.DIR file is described in Chapter 4, "SFINSTAL.DIR."

5. Test the font installation process thoroughly.

Make sure that all the fonts are correctly loaded. If you have screen fonts for different aspect ratios, test loading fonts on machines with different screen displays.

Use the advanced "Add fonts..." button to check the syntax of your SFINSTAL.DIR file and verify your screen display type as described in Chapter 5, "Developers' Tools."

Arranging font files onto disks can be an extremely tedious task, especially if you have to set up disks for 360 Kb, 720 Kb (3.5 inch), and 1.2 Mb builds and if you need to conserve disk space.

Here are some recommendations:

- Keep the screen font files for one FAMILY together on the same disk and make sure the lowest point size font is also on that disk.
- Keep each .PFM file on the same disk as its corresponding printer font file.
- Conserve disk space by putting the largest fonts on the disk with the smallest fonts.
- If you are building the set for different disk densities, build the 360 Kb set first, then put the files from two disks onto each 720 Kb disk and the files from three disks onto each 1.2 Mb disk.

The easiest way to build the disk set is to place files on the disks in the exact order they appear in the SFINSTAL.DIR file. That is, the SFINSTAL.DIR file goes on first, followed by the screen fonts for the first family listed, followed by the .PFM and font files for that family. Make sure that all the screen font files for one family go on the same disk, and that at least one printer font (the first one listed in the FAMILY block) with its PFM also goes on that disk.

Once the disk set is built and working, reorganize the files to optimize for disk space usage. For a sample SFINSTAL.DIR file that shows logical drives and screen fonts, see Section 7.8, "Sample SFINSTAL.DIR File."

## **7.6 Setting Up Fonts on a Network**

This section describes how a network administrator can set up fonts for use on a network. To use this procedure, you must know how to use DOS and a text editor. In addition, you must understand the advanced installer features described in Chapter 3, "Permanent Soft Fonts," Chapter 4, "SFINSTAL.DIR," and Chapter 5, "Developers' Tools."

To set up fonts for use on a network, follow these steps:

1. Install and/or generate the soft fonts.

Load the fonts onto your machine. If you are generating soft fonts, follow the recommendations outlined in Section 7.1, "Selecting Printer Fonts."

If you have control over the creation of screen fonts, follow the recommendations outlined in Section 7.2, "Selecting Screen Fonts." You should build at least two sets of screen fonts, one each for:

- EGA with enhanced color and 64k+
- VGA

If you have users with CGA displays, you may want to add a set of screen fonts for CGA displays.

When you make changes to the SFINSTAL.DIR file (see Step 5), you can instruct the installer to load different screen fonts depending upon the aspect ratio of the machine receiving the fonts.

2. Select permanent fonts.

To use permanently downloaded soft fonts, enter the Soft Font Installer (described in the next step) and select the fonts you want permanently downloaded to the printer. When you exit the installer, it will prompt you for download options. Make sure download "at startup" (i.e., when your machine is turned on) is checked.

3. Build a SFINSTAL.DIR file.

From inside Windows, start up the Control Panel, select Setup, Printer, and the PCL / HP LaserJet printer driver. If you have an application and runtime Windows, run the application and select "Change printer" or "Printer setup" from the file menu or start up Control Panel from the application's system menu. Select "PCL / HP LaserJet" and click on "Setup."

Once inside the driver-specific dialog, click on the "Fonts" button to invoke the Soft Font Installer. The fonts you plan to save should be listed in the left listbox. Hold down the CTRL and SHIFT keys while clicking the mouse on the "Exit" button. The installer prompts for a file name and path. Click "OK" to accept the name of the SFINSTAL.DIR file. Exit back to DOS by clicking "OK" on the driver-specific dialog and selecting "Close" from the MS-DOS Executive system menu (or exit out of the application if you are running one).

4. If you are not working from the file server, copy the fonts to the file server.

If you created and installed fonts from the network server, you can skip this step.

Set up a directory on the file server for the fonts and copy all the files to it. It is assumed you know the DOS commands to do this. If you set up permanent fonts, move the files created by the installer from your machine to the network machine and edit the AUTOEXEC.BAT file on the network to transfer the command line invoking the download batch file. Chapter 3, "Permanent Soft Fonts," describes how permanent soft fonts are set up.

5. Edit SFINSTAL.DIR to add screen fonts.

If you generated screen fonts, edit the SFINSTAL.DIR file to add references to the fonts. Chapter 4, "SFINSTAL.DIR," describes the format of the SFINSTAL.DIR file. There is also an example in Section 7.8, "Sample SFINSTAL.DIR File."

You do not need to add logical drives to the SFINSTAL.DIR file if all the font files and .PFM files reside in one directory.

**6. Test the font installation process thoroughly.**

Make sure that all the fonts are correctly loaded. If you have screen fonts for different aspect ratios, test loading fonts on machines with different screen displays.

Use the advanced "Add fonts..." button to check the syntax of your SFINSTAL.DIR file and verify your screen display type as described in Chapter 5, "Developers' Tools."

**7. Announce the availability of soft fonts to network users.**

Instruct your users that the fonts are available. They can load the fonts by running the Soft Font Installer and selecting "Add fonts..." They should indicate the network drive and path to the directory containing the soft fonts (in response to the "Add fonts..." request for a directory). Installation of the fonts this way will be much faster than loading from floppy disks.

If you set up permanent soft fonts, then your instructions must list those fonts and their ID numbers for the users. Instruct the users to enable the "Edit" button and edit each of the permanent fonts, verifying the ID numbers. They must change the ID numbers on any fonts that do not match your list. See the *Microsoft Windows User's Guide* for a description of the user interface to the installer.

The only "tricky" part to setting up fonts on a network is making sure that all the users have assigned the correct IDs to their permanent fonts. This is not as difficult as it seems. There usually are only a few permanent fonts, and the installer resolves ID conflicts. If the user attempts to assign an ID to a font that is used by another font, the installer will ask the user if the other font should be assigned a new ID. The installer will select the first available ID number when assigning a new ID.

## **7.7 Setting Up .PFM Files for Resident and Cartridge Fonts**

It is possible to build special .PFM files for printer-resident and cartridge fonts. These fonts should be set up as permanently downloaded fonts in the WIN.INI file or collected into a cartridge .PCM file. The special .PFM files contain the escape sequences that the driver will send to the printer to use the fonts.

Font or printer vendors will most likely build the .PFM files. A knowledge of .PFM files is necessary. It is assumed the developer will write a utility that *builds* .PFM files. The *Microsoft Windows Fonts Guide* describes the format of .PFM files.

In short, to set up cartridge and printer-resident fonts, the developer should:

1. Build special .PFM files.
2. If appropriate, combine PFM's into a PCM.
2. Make a special SFINSTAL.DIR file for loading PFM-only fonts or PCM files.

In addition to the structures described for the .PFM files, the driver has a driver-specific data structure which must be added to the file. The PFMEVENTION structure contains a *dfDriverInfo* variable. This

variable is a file offset to the driver-specific structure. (See the *Microsoft Windows Fonts Guide* for more information on PFM and PCM file formats and the new PFM Editor.) For the PCL driver, the driver-specific structure looks like this:

```
typedef enum
{
    epsymUserDefined,
    epsymRoman8,
    epsymKana8,
    epsymMath8,
    epsymUSASCII,
    epsymLineDraw,
    epsymMathSymbols,
    epsymUSLegal,
    epsymRomanExt,
    epsymISO_DenNor,
    epsymISO_UK,
    epsymISO_France,
    epsymISO_German,
    epsymISO_Italy,
    epsymISO_SwedFin,
    epsymISO_Spain,
    epsymGENERIC7,
    epsymGENERIC8,
    epsymECMA94
} SYMBOLSET;

typedef struct
{
    SYMBOLSET symbolSet;           /* kind of translation table */
    DWORD offset;                 /* location of user-defined table */
    WORD len;                     /* length (in bytes) of table */
    BYTE firstchar, lastchar;     /* table ranges from firstchar to lastchar */
} TRANSTABLE;

typedef struct
{
    WORD epSize;                  /* size of this data structure */
    WORD epVersion;               /* number indicating version of struct */
    DWORD epMemUsage;             /* amt of memory font takes up in printer */
    DWORD epEscape;                /* pointer to escape that selects the font */
    TRANSTABLE xtbl;              /* character set translation info */
} DRIVERINFO;
```

The values of the fields in the DRIVERINFO structure should be filled in as follows:

| Field            | Value   |
|------------------|---|
| epSize           | size of(DRIVERINFO)   |
| epVersion        | 1   |
| epMemUsage       | An approximation of the printer memory used by the font   |
| epEscape         | An offset from the top of the file to the escape string that the driver should send to the printer to invoke the font |
| xtbl.xtSymbolSet | <i>epsymGENERIC7</i> for 7-bit fonts and <i>epsymGENERIC8</i> for 8-bit fonts   |
| xtbl.xtOffset    | (long)0   |
| xtbl.xtLen       | 0   |
| xtbl.xtFirstChar | (BYTE)0   |
| xtbl.xtLastChar  | (BYTE)0   |

In addition, the escape string for invoking the fonts must be written to the .PFM file. The driver will use the escape string pointed to by *epEscape* to select the font. Once the .PFM files are built, you must make a .PFM-only SFINSTAL.DIR file. Here is the SFINSTAL.DIR file that was made to support a special cartridge:

```
/* FORD FONT CARTRIDGE
*
* Installer directory file for the Ford font cartridge.
*/
FAMILY "Ford font cartridge"
{
    "Ford Math/Greek 10pt" = PL, , FORD_MTH.PFM
    "Ford Logo 36pt"      = PL, , FORD_LGO.PFM
    "LineDraw 10pt"       = PL, , FORD_LD.PFM
    "LinePrinter 8pt"     = PL, , FORD_LP.PFM
    "Prestige Elite 10pt" = PL, , FORD_PE.PFM
    "Tms Rmn 12pt"        = PL, , FORD_TMS.PFM
}
```

For driver versions 3.2 and above, it is preferable to combine the .PFM files into a single .PCM file and use the following syntax:

```
CARTRIDGE
{
    "Ford font cartridge" = FORD.PCM
}
```

The absence of the downloadable font file is indicated by the two commas in the font strings. The installer loads these fonts as permanently downloadable soft fonts. This has one side effect. When the user exits the installer, it will prompt for download options. Because the installer thinks the fonts are permanent soft fonts, it will try to download them to the printer. However, it will not find the permanent font files because they do not exist. The installer will simply ignore the files, it will not make a download batch file or open a print job. If these fonts are mixed in with other "real" permanent soft fonts, then the installer will correctly download those fonts.

If the target printer is a standard LaserJet, the driver will not load the .PFM files. The driver will not load the .PFM files when the target printer cannot handle soft fonts. However, the user may instruct the driver to load soft font information to gain access to the cartridge font information. The user would place the following flag in the driver-specific section of the WIN.INI file:

options=7

This flag, described in Chapter 2, "WIN.INI Flags," instructs the driver to load soft font information regardless of the printer's abilities.

## 7.8 Sample SFINSTAL.DIR File

This file was built for the Aldus *PageMaker Classroom*. The fonts were generated with Bitstream Fontware (and are distributed with permission from Bitstream). The file can be used on 360 Kb, 720Kb (3.5 inch), and 1.2 Mb disks. To do this, three sets of logical drives were created. Depending on the disk density, one of the logical drive sets was uncommented. In the example, the logical drives for the 1.2 Mb disk set are uncommented:

```
*****
**
**          PAGEMAKER CLASSROOM
**
**          Soft Font Directory File
**
**          Copyright 1988, Aldus Corporation
**
*****
```

\* This file is read by the Windows' PCL / HP LaserJet printer driver.  
\* It lists all the fonts available in PageMaker Classroom's soft font  
\* package.  
\*  
\* The fonts in this package are used under special permission by  
\* Bitstream Incorporated. They were generated using Fontware. These  
\* fonts may not be used for any purpose other than training with  
\* PageMaker Classroom.  
\*/

```
/* High Density (1.2 Mb) Disk set
*/
DRIVE P1: = PMDSK1.LBL, "PageMaker Classroom Soft Font Disk 1 of 3"
DRIVE P2: = PMDSK2.LBL, "PageMaker Classroom Soft Font Disk 1 of 3"
DRIVE P3: = PMDSK3.LBL, "PageMaker Classroom Soft Font Disk 1 of 3"
DRIVE P4: = PMDSK4.LBL, "PageMaker Classroom Soft Font Disk 2 of 3"
DRIVE P5: = PMDSK5.LBL, "PageMaker Classroom Soft Font Disk 2 of 3"
DRIVE P6: = PMDSK6.LBL, "PageMaker Classroom Soft Font Disk 2 of 3"
DRIVE P7: = PMDSK7.LBL, "PageMaker Classroom Soft Font Disk 3 of 3"
DRIVE P8: = PMDSK8.LBL, "PageMaker Classroom Soft Font Disk 3 of 3"
****

/* Low Density (360 Kb) Disk set
*
DRIVE P1: = PMDSK1.LBL, "PageMaker Classroom Soft Font Disk 1 of 8"
DRIVE P2: = PMDSK2.LBL, "PageMaker Classroom Soft Font Disk 2 of 8"
DRIVE P3: = PMDSK3.LBL, "PageMaker Classroom Soft Font Disk 3 of 8"
DRIVE P4: = PMDSK4.LBL, "PageMaker Classroom Soft Font Disk 4 of 8"
DRIVE P5: = PMDSK5.LBL, "PageMaker Classroom Soft Font Disk 5 of 8"
DRIVE P6: = PMDSK6.LBL, "PageMaker Classroom Soft Font Disk 6 of 8"
DRIVE P7: = PMDSK7.LBL, "PageMaker Classroom Soft Font Disk 7 of 8"
DRIVE P8: = PMDSK8.LBL, "PageMaker Classroom Soft Font Disk 8 of 8"
****

/* PS/2 (720 Kb) Disk set
*
DRIVE P1: = PMDSK1.LBL, "PageMaker Classroom Soft Font Disk 1 of 4"
DRIVE P2: = PMDSK2.LBL, "PageMaker Classroom Soft Font Disk 1 of 4"
DRIVE P3: = PMDSK3.LBL, "PageMaker Classroom Soft Font Disk 2 of 4"
DRIVE P4: = PMDSK4.LBL, "PageMaker Classroom Soft Font Disk 2 of 4"
DRIVE P5: = PMDSK5.LBL, "PageMaker Classroom Soft Font Disk 3 of 4"
DRIVE P6: = PMDSK6.LBL, "PageMaker Classroom Soft Font Disk 3 of 4"
DRIVE P7: = PMDSK7.LBL, "PageMaker Classroom Soft Font Disk 4 of 4"
DRIVE P8: = PMDSK8.LBL, "PageMaker Classroom Soft Font Disk 4 of 4"
****

/* The font package contains an assortment of downloadable printer fonts
 * in Dutch, Swiss, and Swiss Condensed. It also contains corresponding
 * screen fonts for EGA (4:3 aspect ratio) and VGA (1:1 aspect ratio)
 * displays.
 */
FAMILY "Dutch"
{
/* screen fonts: roman 8, 9, 10, 12, 14, 18, 24 */
4:3 = "Bitstream Dutch Roman ai000wbp", P2:AI000WBP.FON
1:1 = "Bitstream Dutch Roman ai000wlp", P2:AI000WLP.FON
```

```

    "Dutch  8pt"          = P, P2:AI0080WH.HPF, P2:AI0080WH.PFM
    "Dutch  9pt"          = P, P2:AI0090WH.HPF, P2:AI0090WH.PFM
    "Dutch  10pt"         = P, P2:AI0100WH.HPF, P2:AI0100WH.PFM
    "Dutch  10pt italic"  = P, P3:AJ0100WH.HPF, P3:AJ0100WH.PFM
    "Dutch  10pt"         = L, P6:AI0100W@.HPF, P6:AI0100W@.PFM
    "Dutch  11pt"         = P, P6:AI0110WH.HPF, P6:AI0110WH.PFM
    "Dutch  11pt"         = L, P6:AI0110W@.HPF, P6:AI0110W@.PFM
    "Dutch  12pt"         = P, P4:AI0120WH.HPF, P4:AI0120WH.PFM
    "Dutch  14pt"         = P, P6:AI0140WH.HPF, P6:AI0140WH.PFM
    "Dutch  24pt"         = P, P3:AI0240WH.HPF, P3:AI0240WH.PFM
    "Dutch  24pt"         = L, P3:AI0240W@.HPF, P3:AI0240W@.PFM
    "Dutch  30pt bold"    = P, P6:AK0300WH.HPF, P6:AK0300WH.PFM
    "Dutch  30pt bold"    = L, P7:AK0300W@.HPF, P7:AK0300W@.PFM
    "Dutch  30pt italic"  = P, P3:AJ0300WH.HPF, P3:AJ0300WH.PFM
}

FAMILY "Swiss"
{
    FAMILY "Swiss 721"
    {
        /* screen fonts: roman 8, 9, 10, 12, 14, 18, 24 */
        4:3 = "Bitstream Swiss Roman aa000wbp", P4:AA000WBP.FON
        1:1 = "Bitstream Swiss Roman aa000wlp", P4:AA000WLP.FON

        "Swiss 11pt bold italic" = P, P4:AD0110WH.HPF, P4:AD0110WH.PFM
        "Swiss 14pt bold italic" = P, P4:AD0140WH.HPF, P4:AD0140WH.PFM
        "Swiss 30pt"            = P, P7:AA0300WH.HPF, P7:AA0300WH.PFM
        "Swiss 30pt bold italic" = P, P4:AD0300WH.HPF, P4:AD0300WH.PFM
    }

    FAMILY "Swiss Condensed"
    {
        /* screen fonts: roman 8, 9, 10, 12, 14, 18, 24 */
        4:3 = "Bitstream Swiss Condensed Roman ay000wbp", P5:AY000WBP.FON
        1:1 = "Bitstream Swiss Condensed Roman ay000wlp", P5:AY000WLP.FON

        "Swiss Condensed 8pt"      = P, P5:AY0080WH.HPF, P5:AY0080WH.PFM
        "Swiss Condensed 8pt italic" = P, P7:AZ0080WH.HPF, P7:AZ0080WH.PFM
        "Swiss Condensed 10pt"     = P, P7:AY0100WH.HPF, P7:AY0100WH.PFM
        "Swiss Condensed 10pt bold" = P, P6:BA0100WH.HPF, P6:BA0100WH.PFM
        "Swiss Condensed 10pt italic" = P, P6:AZ0100WH.HPF, P6:AZ0100WH.PFM
        "Swiss Condensed 11pt"     = P, P6:AY0110WH.HPF, P6:AY0110WH.PFM
        "Swiss Condensed 11pt bold" = P, P7:BA0110WH.HPF, P7:BA0110WH.PFM
        "Swiss Condensed 11pt italic" = P, P8:AZ0110WH.HPF, P8:AZ0110WH.PFM
        "Swiss Condensed 12pt"     = P, P6:AY0120WH.HPF, P6:AY0120WH.PFM
        "Swiss Condensed 12pt italic" = P, P5:AZ0120WH.HPF, P5:AZ0120WH.PFM
        "Swiss Condensed 12pt"     = L, P5:AY0120W@.HPF, P5:AY0120W@.PFM
        "Swiss Condensed 22pt bold" = P, P5:BA0220WH.HPF, P5:BA0220WH.PFM
        "Swiss Condensed 30pt"     = P, P5:AY0300WH.HPF, P5:AY0300WH.PFM
        "Swiss Condensed 30pt bold" = P, P8:BA0300WH.HPF, P8:BA0300WH.PFM
        "Swiss Condensed 30pt italic" = P, P8:AZ0300WH.HPF, P8:AZ0300WH.PFM
    }
}

```

