0013

COLLABORATORS			
	TITLE : 0013		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		August 8, 2022	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

0013

1.1 Personal Fonts Maker - 13. Tutorial

```
                13.        Tutorial

   This chapter contains step-by-step examples of "real world"
applications of the Personal Fonts Maker package.


13.1       Starting with an Amiga Font

   Before the Personal Fonts Maker was developed, Amiga fonts could be
printed by text processing software only in graphics mode. This is
generally of poorer quality and reduced print speed, compared with the
letter quality text mode standards of the printer.

   This example shows how an Amiga font can be loaded with the Personal
Fonts Maker and converted into a Personal Fonts Maker font ready to be
downloaded to a printer. Section
                13.4
                 ("Downloading a Font to the
Printer") continues explaining how to adapt the font to the format of the
printer being used. Also in this example, the conversion of a font (in
this case the Amiga "Topaz 8" font) into a different format.

   Before the Amiga font file is loaded, a font format must be specified.
When the program starts, default values are assigned to the font format
(sections 12.6.1, 1.11 and 2.6). Sections 7.1 ("Load Preferences") and
7.3 ("Font Description") explain how to modify the font format. The font
format is used by the Personal Fonts Maker to determine, for example, the
width to height ratio. This affects the way a font is displayed. If the "Y
Max" parameter is set to a different value from the height of the font
being loaded, for example, the Personal Fonts Maker allows the user to
stretch the character images to the new size.

   In this example, the font format parameters should be set as follows:
"X Max" to 36, "Y Max" to 24, "X Dpi" to 360 and "Y Dpi" to 180. This
format works very well if the font is to be downloaded as a letter quality
font to a 24-pin printer. "Y Max" could be set to 16 if the font was to be
downloaded to a printer using 8 or 9 pins, which normally prints letter
quality text in two passes, and can therefore use fonts whose height is at
```

least 16. In general, multiples of 8 should be assigned to "Y Max" to define a font format used for printer download.

Once the desired font format has been determined, an Amiga font can be selected. The "Import Amiga Font" (section 4.5) displays a font requester listing the names of the different Amiga fonts which are available. A double-click on the "topaz 8 of" gadget will cause the program to load the Amiga "Topaz" font whose height is 8. Since the current font format specifies a font height of 24 (or 16, depending on the printer being used in this example), a requester will appear asking the user how the difference between the font's format and the program's format should be handled. If the "Stretch" option is selected, the program will stretch the image of each character in the "Topaz" font to the new format. Section 7.10 ("Stretch") explains how different stretch options can be selected.

As described in section 2.8 ("Character Sets"), the Personal Fonts Maker can work with character sets other than the Amiga set. This makes it possible to design fonts which can be interpreted by programs and printers using different character sets. During loading, the characters of the Amiga font are reordered so that they are placed in the correct positions according to the character set which is being used. Sections 4.8 ("Load Character Set"), 4.10 ("Define Character Set") and 4.11 ("Edit Character Set") have more on this subject. This example assumes that the "PC_Usa2" set (the default set) is used.

After the load operation is completed, the character editing box will contain an "enlarged" version of one of the characters in the "Topaz" font. Most characters need to be retouched manually by the user, as new details can be added to the characters thanks to the higher resolution. These changes can easily be made by using the mouse to edit the characters directly, as described in chapter 3.

A macro like "SetSpcKrn" (section 12.5.5) can be executed on all characters in the font with the "Controlled Execution" function (section 6.8). This will quickly set the "Space" and "Kerning" parameters of all the characters to a standard value.

Finally, the font can be saved. The "Save PFM Font" (section 4.4) function allows the user to specify the device, drawer and name of the file to be saved, as explained in section 3.23 ("The File Requester"). A suffix can be appended immediately after the font's name, as described in section 1.4 ("File Names and Titles"). A correct file name for the font of this example could be "Topaz_24.fnt".

13.2 A Simple Macro

This example shows how easy it is to record a macro which places a shadow under a character image. As described in section 12.5.1 ("The 'Shadow' Macro"), the same macro is already included with the Personal Fonts Maker package.

As explained in chapter 6, to record a macro it is sufficient to execute the operations manually while the record mode is active. A character should be displayed in the editing box, so that the result of each change can be verified immediately.

In this example, an 'A' character of the "Gemini_24" font is used to record the macro. In practice, it does not matter which character is used, as the macro is independent from the character and the font on which it was recorded.

Before the recording of the macro is started, all special modes like the "OR" mode, the "Brush" mode or the "Define Brush" mode (sections 3.16, 3.17 and 3.18) should be disabled. The position of the brush handle (section 5.9) should be set to "Upper Left" (section 5.9.1). The current character must be "On" (section 3.10). The coordinates should be displayed on the title bar in the "Start 0:0" mode (section 7.5.1). The "Copy to Buffer" function (section 3.11) should be selected, so that a copy of the original character image is always available. This can be useful if the image is to be restored after a mistake is made, or before the macro is re-executed on the entire font (including the current character, which would be "shadowed" a second time).

To start the macro record mode, the "Start Record" command must be selected (section 6.3). In the macro requester, which appears automatically, the 'A' element in the list box (or any other one) should be selected with the mouse, to indicate that the macro which will be recorded should be assigned to the <A> key of the keyboard. The name of the macro, "Shadow" in this example, must be written in the string gadget at the bottom of the requester. After the "Proceed" gadget is selected, the macro record mode begins.

The first instruction to be recorded is the selection of the entire character as a brush. To do this, the "brush definition" mode must be activated, by selecting the "Define Brush" gadget, as described in section 3.18. To mark the brush rectangle, the left mouse button must be pressed when the mouse pointer is over the top left dot in the character editing box. The mouse must then be moved (with the left button still pressed) until the pointer is both to the right of the character editing box and below it. The mouse button must not be released when the mouse pointer is over the bottom right dot of the box. The mouse button should be released only when the highlighted region includes the entire character image and the mouse pointer is beyond the bottom right corner of the character editing box. This ensures that the macro records a command like "The entire character should be defined as a brush", rather than "Mark brush from 0:0 to 26:23". This latter command would leave out parts of characters larger than 27 by 24.

The definition of the brush automatically activates the "Draw with Brush" mode. The "OR" mode must also be activated, by selecting the "OR" gadget (section 3.16).

A shadow will make the character wider. To make room for the shadow of this example the right arrow gadget associated with "X Size" must be pressed twice. This will add two columns to the character width.

At this point, the brush can be used to draw a shadow to the right of the character. To do this, the mouse pointer must be moved until the coordinate box on the title bar displays "2:2". In this position, the left mouse button should be pressed and released. The shadow has been drawn. Now the dots between the shadow and the character need to be cleared, otherwise the shadow cannot be distinguished from the character. To do this, the brush outline must be moved by one dot up and to the left,

so that the coordinates on the title bar are "1:1". The right mouse button must be pressed and released in this position to clear the dots. Finally, the original character must be restored, as the previous operation has cleared most of the character image. This can be done by pressing and releasing the left mouse button when the brush coordinates are "0:0".

To terminate the macro gracefully, the "OR" mode and the "Draw with Brush" mode should be terminated (sections 3.16 and 3.17). This ensures that after the macro is executed the normal environment is restored. The "End Record" function (section 6.4) must be selected to end the recording of the macro.

The macro can now be tested. To do this on the same character, the "Paste from Buffer" option (section 3.12) can be used to restore the original character image, if it was copied to the buffer before the recording of the macro began. The "Execute Macro" function (section 6.7) can be used to execute the macro on the current character, while "Controlled Execution" (section 6.8) will repeat the macro on all "On" characters in the font, or on another user-defined range. The "Examine Macro" (section 6.6) and "Step Execution" (section 6.10) options can be selected for a better understanding of the macro.

The "Save Macro" function (section 6.2) will save the macro to a file, ready to be loaded whenever necessary.

In this example, the "OR" mode and the "Brush Definition" mode are disabled before the start of the macro recording and activated during the recording. This is absolutely correct and makes the macro more readable and elegant, since all the commands necessary for the macro to work properly are explicitly executed during the recording of the macro. It is, however, possible to create a functionally equivalent macro, with two commands less. This can be done by activating the "OR" mode and the "Brush Definition" mode only before the start of the recording. This can be done because the Personal Fonts Maker always places at the beginning of each macro a set of commands which describe the initial editing environment.

13.3 Modifying a PFM Font

This example shows how to create a full 8-bit font starting from a Personal Fonts Maker font containing only characters defined by the 7-bit US ASCII set, whose code is smaller than 128. In general, it is more convenient to first design the basic letters of the US ASCII set, and then copy these letters to the positions where national variants of these characters should appear. It is easier to modify an existing character rather than designing a new character, especially if it is important to maintain a style which is coherent with the other characters in the font. All the fonts which come with the Personal Fonts Maker package have the national characters and other signs which are not part of the US ASCII character set already defined. Other fonts, originally designed to be used in a less international environment, can be extended as explained here.

Before any font is loaded, it is important that the correct font format is selected. This can be done with the requester described in section 7.3 ("Font Description"). Normally, the format does not need to be changed manually, as the most used formats can be saved to and recalled from a file (sections 7.1 and 7.2). As explained in sections 1.11, 2.6 and

12.6 it is even possible to define two different font formats to become the default formats when the program is loaded.

If only the characters defined by the US ASCII set are used, it does not matter very much whether the current character set is the Amiga set, the IBM PC set or the Macintosh set, as most 8-bit sets share the subset of codes defined by the 7-bit US ASCII set. When, however, a 7-bit font has to be extended to an 8-bit character encoding format, it is important to decide which set to choose. Section 2.8 ("Character Sets") has more on character sets. It is possible to convert a font from one character set to a different set using the Amiga font format as an intermediate step in the conversion, as described in sections 4.5 ("Import Amiga Font") and 4.6 ("Export Amiga Font"). Characters not shared by the source set, the target set and the Amiga set are not converted.

Once the font with the 7-bit encoded characters is loaded, the longest and most annoying work is the creation of all variations of the existing letters, i.e. all the characters (capitals and lower case letters) with accents and other diacritical signs. Copying the existing letters to all the positions where these characters only need to be modified greatly simplifies this job. If this operation has to be done only once, as described in the introduction to chapter 6, the repetitive work is reduced to a minimum.

As described in section 12.5.4, the set of "CopyOver127" macros was developed to ease the creation of fonts which are a superset of existing fonts. Similar macros can be designed by the user to work with different character sets.

The macro "CopyOver127" does exactly what the user would do. It copies some of the existing letters to the positions where similar letters are to be designed. For example, the ’u’ letter is copied to those positions in the font where the letters ’ü’, ’ù’, ’ú’ and ’û’ will have to appear. Once the basic ’u’ image is in these positions, it is extremely easy to add an umlaut or an accent. It is even possible to design a set of diacritical signs which can be pasted over the "normal" characters whenever necessary. These diacritical signs could be temporarily stored in the other font environment, and automaticaly pasted over the images written by the "CopyOver127" macro with a second macro. Of course, some manual retouching will always be necessary, as different letters may require the signs to be placed at different heights.

To execute the "CopyOver127" macro, it is sufficient to select the "Execute Macro" function (section 6.7). There is no need to use the "Controlled Execution" function (section 6.8), as the macro already contains the instructions to select the different characters.

It is important that the selected "CopyOver127" macro works on the character set being used, as different character sets have the letters with diacritical marks at different positions. The sets which are most used with the Personal Fonts Maker are the printer’s character set and the set of the host computer.

Once the font has been completed, it can be saved as described in sections 4.4 ("Save PFM Font") and 4.6 ("Export Amiga Font").

13.4 Downloading a Font to the Printer

Once a font has been loaded, created or modified, as shown in the examples of sections 13.1 to 13.3, it can be downloaded to the printer. Its ability to output the data from any font in the format required by different printers is one of the most interesting features of the Personal Fonts Maker. Section 2.5 ("Downloaded Printer Fonts") contains a general introduction to downloaded fonts.

For this example, the "Andromeda" font will be used. The font will be downloaded as a letter quality, proportionally spaced font. Depending on the printer being used, either the "Andromeda_24" or the "Andromeda_16" font can be used. The first is ideal for 24-pin printers, while the second is more suited for 8/9-pin and 18-pin printers.

Before the font is loaded, the correct font format must be specified. This can be done through the "Font Description" requester (section 7.3) or by loading an existing parameter file (section 7.1, "Load Preferences" and 12.6.4, "Parameter Files: Printer Font Descriptions"). Only the first four parameters need to be set ("X Max", "Y Max", "X Dpi" and "Y Dpi"). These will allow the program to stretch the font if it does not already fit the printer's format.

The font can be loaded as described in section 4.3 ("Load PFM Font"). If a requester appears, informing the user that the selected font has a different format from the current font format, either the "Proceed" or the "Stretch" gadgets of the requester should be selected. The "Proceed" option should be selected only if the current font has a maximum width ("X Max") wider than that of the font being loaded, but the current height is equal to (or slightly higher than) that of the selected font.

At this point, the font loaded by the Personal Fonts Maker has the dimensions required by the printer's download format. It is now necessary to select the correct FFDL sequences which will send the font data to the printer. The sequences are loaded automatically when the font parameters are loaded from a file (section 7.1, "Load Preferences" and 12.6.4, "Parameter Files: Printer Font Descriptions"). If the FFDL sequences in the selected parameter file already match the format of the printer, no changes need to be made to the font description.

To check whether the current font format works properly on the printer being used, there is no better way than to test it. The two "Printer Test" functions (section 4.12) perform a download and a test print. The "Printer Test/Character" function downloads and prints only the current character. This makes it easier to isolate and track down any errors, and will not print too many "garbage" characters if the printer does not interpret the data as a font download.

As described in section 7.3 ("Font Description"), it is very important that the FFDL sequences are set properly for the download to succeed. The following are examples of valid FFDL sequences which could be used to download the font described here.

"Prologue" (section 7.3.5):

Epson "ESC/P" (Epson Standard Code for Printers), NEC Pinwriter:

```
ESC \% (0) ESC \x (1) ESC \: NUL NUL NUL
```

NEC Pinwriter Plus:

```
FS \W (1) ESC \% (0) ESC \x (1) ESC \: NUL NUL NUL
```

IBM "Compatibles":

```
ESC \: NUL NUL NUL
```

IBM Proprinter:

```
ESC \I (3) ESC \= (0) (0)
```

Fujitsu DPL24C/DPL24I:

```
ESC \: NUL (0) (0)
```

The "Prologue" selects a ROM (resident) font and copies its characters to the download memory. The NEC Pinwriter Plus printers require the initial "FS \W (1)" sequence (memory-set) in order to select the 8-bit character encoding mode, rather than the 7-bit mode, which would not allow characters whose code is greater than 127 to be downloaded. On some printers one or more DIP switches, or software parameters (memory/soft switch), must be set before the printer's memory can be used for the downloading of characters. In the above examples, the IBM Proprinter is set in the fixed-width letter quality mode.

"ON Sequence" (section 7.3.6):

Epson "ESC/P", NEC Pinwriter and Pinwriter Plus:

```
ESC \& (0) CNUM CNUM (1) XSIZ (1) VDAT
```

IBM "Compatibles":

```
ESC \= (0) CNUM CNUM (1) XSIZ (1) VDAT
```

IBM Proprinter:

```
ESC \= x(1*48+2) (21) CNUM (0) (0) VIDT REPT((46-LENG(VIDT)),NUL)
```

Fujitsu DPL24C/DPL24I:

```
ESC \& (0x10) CNUM CNUM XSIZ VDAT
```

The "ON Sequence" is repeated once for every "On" character in the font, sending the character's code, size and image data to the printer. Each character is downloaded individually, as one or more "Off" characters could interrupt the sequence of "On" characters (unless "Off" characters are sent as well, as described in sections 7.3.7 and 12.6.4). The "(21)" code in the IBM Proprinter sequence is the identification code for the Letter Quality II font on the IBM 4201/002 printer. Other codes can be

used. The code for a draft font on an IBM 4202 or 4201/002 printer model is "(20)". Since the fixed width mode is used in the IBM Proprinter, a "REPT" instruction has been added to convert the characters of the proportionally spaced font to a fixed width, by adding some trailing "NUL" codes.

"Epilogue" (section 7.3.8):

Epson "ESC/P", NEC Pinwriter and Pinwriter Plus, IBM "Compatibles":

```
ESC \% (1)
```

IBM Proprinter:

```
ESC \I (7)
```

Fujitsu DPL24C/DPL24I:

```
ESC \% (5) (0)
```

The "Epilogue" is sent to the printer at the end of the download, activating the print mode in the downloaded font. The sequence for the IBM Proprinter 4201/002 selects the Letter Quality II downloaded font. This sequence is ignored by the 4202 model. The terminating "(7)" code can be replaced with "(4)", "(5)" (not on the 4202 model) and "(6)", to respectively select the draft, 12 CPI FastFont and Letter Quality downloaded font.

A "Range" (section 7.3.9) from 32 to 126 (decimal codes) should work on most printers. The widest possible range is 0 to 255. The printer's documentation should be read to verify which characters can be downloaded. Another restriction may be imposed by the available memory. Section 12.1 ("TextChars") explains what can be done if the printer's memory is not sufficient to store all the characters in a font.

Once the font format and the FFDL sequences have proven to work properly, the entire font can be downloaded. The "Write Font Data" functions (section 4.13) can be used to send the font either directly to the printer, or to a file. Section 13.5 ("Interacting with a Word Processor") explains how the downloaded font can be used from a word processor.

13.5 Interacting with a Word Processor

Word processors can print text in two ways: in text mode or in graphics mode. When the text mode is selected, the program sends a stream of characters and control codes to the driver, which in turn sends the data to the printer. In this mode, the images of the characters to be printed must already be in the printer's memory, either in the ROM, or in the download area. If a font has been downloaded, it can be used by a program printing in text mode.

When a program prints in graphics mode, it outputs the entire text as an image. This means that each individual dot which makes up the resulting

text must be sent by the word processor. Since such a large amount of data has to be sent (and translated by the driver), this technique is much slower than the text mode. Also, the (poorer) quality of a text printed in graphics mode is usually not comparable with the letter quality text modes built into modern printers. To use a font designed with the Personal Fonts Maker with software which prints the text in graphics mode, the font must be saved in the Amiga format (section 4.6) and loaded as an Amiga font by the program which has to print it. This section explains how to use a downloaded font printing in text mode.

Section 13.4 ("Downloading a Font to the Printer") explains how to create the font download data and send it to the printer or to a file. The same data which is normally sent to the printer to print with a downloaded font (section 4.13.2, "Write Font Data/Printer") can be sent to a file (section 4.13.1, "Write Font Data/File"), from where it can be used whenever necessary.

Section 12.2 ("PrintRawFiles") explains how to send the font download data stored in a file to the printer. To download such a font, it is sufficient to double click on the file icon. This will automatically execute the PrintRawFiles program. PrintRawFiles can also appear in the Startup-Sequence, from where it could be used to automatically download a font to the printer whenever the system is booted.

With most word processors it is possible to download a file without having to use PrintRawFiles. The "Print File" function of Cloanto's Personal Write word processor, for example, allows the user to send any file to the printer, without modifying the file's content (or allowing the printer driver to do so). The "Printer Control File" of the same word processor allows the user to select a file which is to be automatically sent to the printer before each print.

Whatever method is used to send the printer download data contained in a file to the printer, it is important to remember that the data must be received by the printer "as is". The Personal Fonts Maker already writes the font download data in the printer's format. Any further processing, by a printer driver or any other software, is likely to make the file unrecognizable by the printer.

Once the font has been sent to the printer, either by the Personal Fonts Maker, by PrintRawFiles or by another program, it can be used to print the text. The FFDL sequence associated with the "Epilogue" parameter usually contains the control sequences which instruct the printer to use the downloaded font. If this is not the case, such a command must be sent to the printer by the word processors. Most programs cannot do this, as this function is not supported by the Amiga printer drivers. For this reason, it is important to activate the font by placing the appropriate control codes in the "Epilogue" sequence. Once this command is received by the printer, the program printing the text should not send a command selecting a different font, as this would restore the printer's default ROM font (or another soft font). If the "Printer Test" function (section 4.12) of the Personal Fonts Maker seems to work properly, downloading and printing a font, but the same font cannot be used by a word processor, it is likely that the word processor automatically selects a different font at the beginning of the print. If this is the case, the parameters of the

word processors should be set so that the program does not automatically set a font of the printer before printing the text.

Once a font has been downloaded, the new font can also be selected from the printer's control panel. Some printers can even be programmed to use the downloaded font as the default font at power-up. The solution suggested before, i.e. sending the appropriate font activation codes to select the downloaded font, is the one which works most efficiently on the majority of printers.

Section 14.2 ("Problems with Printers") describes some problems which may be encountered when working with a printer.

13.6 Creating a Word Processor Font Size Table

The Personal Fonts Maker package contains an interesting parameter file, called "SizeTable.prf" (section 12.6.3). The commands of the Font Format Description Language contained in that file are capable of creating a size table indicting the width of each character in the current font. This application, which may seem unusual to those who have used the FFDL only to create font files, is only one of the many examples of the power and flexibility of the Font Format Description Language.

Many word processors can use different fonts to print the text. If the font used to print the text is not the same as the font loaded in the word processor's memory, the program will not be able to format the printed text correctly, unless all the characters have the same width, or the printer justifies the text. Some programs allow the user to specify the width of every character in the font used by the printer. Once this table is complete, the word processor "knows" about the new font, even if it is a downloaded font, and can justify, center and align the text properly. This does not need to be done if the printer can format the text automatically, and the word processor supports this function.

The "SizeTable.prf" can be loaded, as any other parameter file, with the "Load Preferences" function (section 7.1). One of the "Write Font Data" functions (section 4.13) can be used to print the table, or send it to a file, from where it can be loaded, viewed and printed with a word processor.

The table displays a value for each "On" character in the font. The value includes the width of the character image, plus the additional margins to the left and the right of the character. This is the sum of the "Space" and "Kerning" parameters, which is displayed (or printed). "X Size" is not considered. The existing FFDL sequences can, of course, be viewed and modified by the user to create a different kind of table. For example, "X Size" may be used instead of the sum of "Space" and "Kerning".

At the end of each line of the table a CR+LF (Carriage Return + Line Feed) combination is output. This works well on most printers, and is also handled by a majority of word processors, should the data be output to a file. The FFDL sequences can be modified to output only a LF (Line Feed, decimal value 10) code if a file following the Amiga default rules for plain text formats is to be created.

The following is a sample table, generated by the FFDL sequences contained in the "SizeTable.prf" file, relative to the "Pegasus_24" font:

PFM Size Table

	0	1	2	3	4	5	6	7	8	9
0	--	--	--	--	--	--	--	--	--	--
10	--	--	--	--	--	--	--	--	--	--
20	--	21	--	--	--	--	--	--	--	--
30	--	--	24	7	17	25	23	24	25	8
40	14	14	22	22	8	20	8	23	17	12
50	23	20	21	18	18	19	18	18	8	8
60	18	20	18	22	27	31	27	29	31	32
70	35	29	31	27	29	31	31	34	33	29
80	32	32	29	28	31	29	23	32	29	28
90	28	11	23	11	17	28	8	24	21	21
100	23	21	20	25	20	11	24	24	11	31
110	23	23	23	23	21	22	17	23	23	32
120	24	22	24	16	5	16	22	--	29	23
130	21	24	24	24	24	19	21	21	21	17
140	15	11	27	27	31	29	36	21	22	21
150	23	23	22	27	27	21	33	35	39	25
160	24	14	21	23	23	33	18	20	22	--
170	--	32	30	7	29	29	--	--	--	--
180	--	--	--	--	--	--	--	--	--	--
190	--	--	--	--	--	--	--	--	--	--
200	--	--	--	--	--	--	--	--	--	--
210	--	--	--	--	--	--	--	--	--	--
220	--	--	--	--	--	21	--	--	--	--
230	--	--	--	--	--	--	--	--	--	--
240	--	--	--	--	--	--	--	--	16	--
250	--	--	--	--	--	--				

13.7 Patching a Printer Driver

As described in chapter 9, existing Amiga printer drivers can sometimes be improved if they are modified with the Printer Driver Modifier. Section 9.2 ("Problems with Standard Drivers") describes the complex control sequences used by the Epson X, Epson Q, NEC Pinwriter and other printer drivers to print a simple '½' (one half) character. Rather than overlapping a '1' (one), a '-' and a '2' sign moving the printer head back and forth, it is possible to use the '½' character which is already built into most printers. This example shows how to replace the existing sequence with the code associated with the '½' character in the IBM PC set, shared by the printers mentioned above.

Printer drivers are stored in the "printers" drawer, inside the "devs" drawer of the Workbench and Extras disks. The Extras disk usually contains all printer drivers. If the Printer Driver Modifier is used for the first time, it is better to use a copy of the disk on which the drivers are stored before this example is repeated. This will prevent any undesired

changes being made to the drivers.

To load the Printer Driver Modifier it is sufficient to double click its Workbench icon. Next, the printer driver must be loaded. To do this, the "Load Driver" function (section 10.1) must be selected. In the file requester displayed by this function, the file containing the driver can be chosen from the names displayed in the list box. Among those names, "EpsonQ", "EpsonX" and "Nec_Pinwriter" would all work fine for this example. The driver is loaded immediately after the "Proceed" gadget of the file requester is selected.

To access the sequence associated with the '½' character, the "Section/Character" option must be selected (section 10.7.2). The "Number" string gadget must be activated, the value contained in the gadget deleted with the <BS> and <Del> keys, and the keys <1>, <8>, <9> and <Return> typed. This will display the control sequence associated with the character whose Amiga code is 189, which is '½', in the editing gadget (section 9.4.2).

The editing gadget should be selected with the mouse, and its entire content deleted (pressing <X> while <Amiga> is held down, or using the <BS> and <Del> keys). The FFDL constant "(171)" (without quotes, but with the parentheses) should be written in the editing gadget. If the printer driver is saved, the decimal code 171 will be sent to the printer whenever a program using that printer driver prints a '½' character.

To save the printer driver, the "Save Driver" function (section 10.2) must be used. It is always a good practice to have a copy of the driver being modified available, in case the result of the changes needs to be reversed. Also, the modified driver should be saved with a name different from that of the original driver. In this way, the Printer Driver Modifier will be able to identify it as a variant of an original driver and issue the "Internal Name Mismatch" message (more in appendix H) when that driver is loaded again.