0008

COLLABORATORS

	TITLE : 0008		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		August 8, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

0008

1.1   Personal Fonts Maker - 3. PFM: The Screen

```
              3.              PFM: The Screen

   The main window of the Personal Fonts Maker, the quick character
selection window and the file requester are among the most used graphical
environments of the program. This chapter describes the functions which
can be accessed from these windows. Other windows and requesters,
associated with specific parameters or functions, are described in
chapters 4 to 7, 10 and 11.


3.1           The Character Editing Box

   The character editing box occupies the central part of the main screen
of the Personal Fonts Maker. It always displays a magnified image of the
current character.

   When the mouse pointer is over the box, the mouse can be used to edit
the displayed character. When the left button is pressed, a dot is painted
with the foreground colour. Pressing the right mouse button, the dot under
the pointer is cleared. Sections
              3.17
               and
              3.18
               explain how the brush can
be used to edit elements larger than a single dot.

   The image of the mouse pointer changes from the standard arrow into a
cross when the pointer is over the character editing box. As described in
sections 1.9.2 and 1.9.8, the mouse buttons can be used to select menu
items only when the pointer has the default arrow image. When the image is
a cross, the pointer is over the box, while two arrows indicate that the
reference points can be moved (section
              3.21
              ).

   After another program has been used, causing a window other than that
of the Personal Fonts Maker to be activated, the first "click" of the left
mouse button does not modify anything in the character editing box.
```

Instead, it activates the window of the Personal Fonts Maker, informing the program that it is being used again.

   The width and the height of the character editing box are calculated from the "X Max", "X Dpi", "Y Max" and "Y Dpi" variables, described in sections 2.6.23 to 2.6.26 and 7.3.1 to 7.3.4. The size of the box can vary depending on the video mode, whether it be NTSC, PAL or another. Since the displayed dots in the box are made up of screen pixels, the horizontal and vertical sizes of the dots will be rounded to the closest number of video pixels. In all cases, the program keeps the aspect of the characters displayed in the box as close as possible to the ratio defined by the user through the "X Dpi" and "Y Dpi" variables.

   The minimum work size of the box is one single dot (1 column, 1 row). The maximum size of the box is 255 x 255 dots, but it can be smaller depending on the selected display size (sections
                3.13
                 and
                3.14
) and the
screen display mode. In practice other factors (most notably the screen size) limit the maximum size of the box, therefore 255 by 255 is a theorical limit. These minimum and maximum values are the same as the limits for the size of the characters in a font. Therefore the box cannot be made larger than the limits specified by the "X Max" and "Y Max" font parameters, described in sections 7.3.1 and 7.3.2.

   As described in section 7.6, different types of dots can be displayed, with or without a grid separating the dots. The colours of the grid and the dot depend on the status of the current character (section
                3.10
). If
the character is "On", the grid will be displayed in dark green and the dots will be black (or any other colour specified for the dots), while if the character is marked as "Off", both the grid and the dots will be light green (which looks like grey in the green environment).

3.2   The "Font" Gadgets

   The Personal Fonts Maker makes it possible to work with two different fonts. Both fonts can reside in the program's memory at the same time. As described in section 2.6, each font can have its own environment of font parameters. This means that each font can have its own, independent, values for parameters like display size, FFDL sequences and densities. Each font can have a maximum of 257 characters, with codes ranging from 0 to 256. The two font environments are conventionally called Font 1 (one) and Font 2 (two). Data can be exchanged between the two fonts even if the font parameters differ (more in sections
                3.11
                ,
                3.12
                ,
                3.17
                ,
                3.18
                 and

7.10.3).

   Two gadgets can be used to switch from one font environment to the
other. At the top left of the main program screen there is a list of
parameters which can be modified. The "Font" parameter, followed by the
two gadgets, is the first in this list. The user can either click with the
mouse on the "1/2" symbol gadget, or type the number (1 or 2) in the
string gadget. Section 1.9.6 explains how to use string gadgets. The
string gadget must be selected with the mouse, and its content deleted
with the <BS> or <Del> key before the new number can be typed from the
keyboard. After the number has been typed, the <Return> (or <Enter>) key
must be pressed. A number other than 1 or 2 will be automatically
corrected to the closest valid number. It is definitely easier and faster
to change font environment by selecting the "1/2" gadget.

   If the Amiga has not enough free RAM the Personal Fonts Maker may not
be able to allocate memory for a second font environment. In this case, an
error message is displayed, and the second font environment cannot be
selected. Appendix G ("PFM Program Messages") lists and explains all
error messages.

3.3   The "Character #" Gadgets

   There are several ways to select the character which is to appear
and/or be edited by the user in the character editing box. Section
                3.22
                describes the "Quick character selection". Another method is ↩
                    explained
here.

   Among the parameters which appear on the top left of the main screen,
the text "Char. #" introduces three gadgets which can be used to select
the character to be edited. The first gadget is an integer gadget, which
is a string gadget that can contain only numbers (section 1.9.6). The two
"left arrow" and "right arrow" gadgets can be selected to decrease or
increase the value in the string gadget, respectively. The latter two
gadgets are very useful to access a character which has a code close to
the one of the character currently being edited.

   The numerical value displayed in the string gadget can range from 0 to
256. This means that 257 different characters can be accessed. As
described in section 2.8, the code 256 is reserved for the undefined
character, which is generally used by fonts stored in the Amiga format.
The same section explains the meaning of the displayed numerical code,
which depends on the selected character set.

   The default image of the current character, according to the selected
character set, is displayed in the gadget described in section
                3.7
                  .

3.4   The "X Size" Gadgets

   The horizontal size of the character grid can be controlled by these
gadgets. The grid can be made smaller or larger. The minimum size is 1

(one column of dots, while the grid can be as large as defined by the "X Max" parameter, described in section 7.3.1.

The characters in a proportional font (see sections 2.3 and 8.14) can have different widths. The letter 'I', for example, generally occupies less horizontal space than the 'M' character of the same font. This means that the box containing the 'I' letter can be narrower, and therefore the bitmap of this letter can occupy less memory than that of a larger letter.

This parameter does not need to include additional spaces at the left or right of the character. These spaces are useful to indicate how much free space is to appear around a character when it is printed surrounded by other characters, but it would be a waste of memory to define these spaces in a graphical way, leaving empty columns at the sides of the dots which make up the character. The "Space" and "Kerning" parameters (sections
              3.5
               and
              3.6
) should be used for this purpose.

If the font is not proportionally spaced, then all characters will have the same width. The "X Size" parameter will be constant for all characters in the font. In such a font the characters are usually centred at the middle of the editing box. A simple macro (see chapter 6 and sections 12.5 and 13.2) which increases the "X Size" parameter of all characters can be used to transform a proportionally spaced font into a font where all characters have the same width.

The gadgets work in a manner similar to those of the other parameters grouped in this part of the screen. The user can type in a value in the string gadget, or use the two gadgets at the right of the displayed value to decrease or increase the value itself.

The "X Size", like the "Space" and "Kerning" string gadgets, also accepts the '−' (minus) and the '+' (plus) character instead of a number. When the user writes the '−' character in the gadget and presses <Return>, the program replaces the sign with the width of the last marked brush (section
              3.18
). If the '+' sign is written instead, that character will be replaced by the width of the brush stretched to the current font height. The brush itself is not stretched, but the value is calculated as if the brush had been stretched according to the selected stretch mode (section 7.10, "Stretch"). If no brush has been defined, the gadget is filled in with the previous value. One technical note: if the brush is loaded from a file, the "xAspect" and "yAspect" fields of the IFF BMHD structure must be set properly, otherwise '+' is converted to "X Max" (section 7.3.1). The Personal Fonts Maker sets these fields correctly when it saves a brush.

The '−' and '+' options can be extremely useful if they are used from a macro (see chapter 6). For example, if the user wants to record an event like "Set X Size to the brush width plus 3" it is sufficient to type a '−' in the string gadget, and then press the "right arrow" gadget three times. This must be done while the "record macro" mode is activated, as explained

in chapter 6.

   Any modifications to the value of this parameter can be "undone" as
described in section
                3.9
                .

   This parameter is also called "Character Width" in the documentation of
some printers.

   The associated FFDL variable ("XSIZ") is described in section
2.7.2.22.

3.5   The "Space" Gadgets

   The "Space" parameter defines the logical horizontal size of the
character, starting from the beginning of the bitmap. The value associated
with this parameter is usually slightly higher than the "X Size"
parameter. The "Space" parameter indicates by how many dots the printer
head has to be moved to the right after the graphical data associated with
the character is printed. This parameter affects the placement of the
following character, rather than the position of the character it refers
to.

   For example, if the character 'A' has an "X Size" of 32, a "Space" of
34 will indicate that there have to be two (34 - 32 = 2) empty columns of
dots at the right of the letter 'A' when it is printed followed by other
text. The character will therefore occupy 34 columns of dots horizontally.
The bitmap associated with the character, however, will only occupy memory
for the 32 columns of dots which make up the graphical part of the letter.
The same result could be achieved by defining an "X Size" of 34 and an
equal "Space", but the two empty columns of the larger bitmap would be
wasted. It is always preferable to set "X Size" to the "real" horizontal
size of the bitmap of the character, and set a higher "Space" to define a
proper distance from the following characters in the printed text.

   The range of valid values for this parameter is from 0 to 255. The
gadgets can be used to set the parameter exactly as described for "X Size"
in section
                3.4
                . If the '-' or '+' characters are written instead of a
number in the integer gadget, they will be replaced by the width of the
last defined brush, or by the width of the brush stretched to the maximum
height, respectively.

   Some printer handbooks call this parameter "Delta X".

   The associated FFDL variable ("SPCE") is described in section
2.7.2.16.

3.6   The "Kerning" Gadgets

   The "Kerning" parameter determines the starting position of the
character when it is printed. The parameter specifies an offset from the
position immediately following the last character which was printed or

displayed.

For example, if the string "AB" is printed, where 'A' has an "X Size"
of 32 and a "Space" of 34, and 'B' has a "Kerning" value of 1, there will
be three (34 - 32 + 1 = 3) empty columns of dots between the letters 'A'
and 'B'. This shows how the "Kerning" and "Space" parameters can be used
to control the space which must separate a character from the others, at
the left and the right of the character, respectively.

The range of valid values for this parameter is from -128 to 127. If
the "Kerning" parameter is set to a value smaller than 0, and the "Space"
character of the previously printed (or displayed) character is the same
as or smaller than the "X Size", then the bitmaps of the two characters
will overlap (if the output device supports this).

The meaning of the word "kerning" used in this documentation is
inherited from the Amiga environment. Amiga fonts do not have
two-dimensional kerning tables like those used in other systems.

The "Kerning" and "Space" parameters do not affect the bitmap of the
character. Instead, they are used to control the horizontal spacing
between the characters when several characters are grouped together to
print, display or output text. These two parameters are stored with each
character when the font is saved in either the PFM or Amiga (only if the
Amiga font is proportionally spaced) format (see section 2.4). The use of
these two parameters is optional when fonts are output through FFDL
sequences, as in this case the format depends on the requirements of the
target system.

If a '-' (minus) character is written instead of a number in the integer
gadget, it is replaced with the width of the last defined brush, while a
'+' character is replaced with the width of the brush stretched to the
maximum font height.

This parameter is also called "Left Offset" and "Character Origin" in
the documentation of some printers.

The associated FFDL variable ("KERN") is described in section
2.7.2.10.

3.7   The "Default" Gadget

The "Default" gadget always contains the standard image defined by the
character set data for the character currently being edited. If no
character set has been loaded, or the current character has no equivalent
in the current character set, no image is displayed. Section 2.8 deals in
more detail with character sets.

The image displayed in the gadget is not a reduced image of the
character displayed in the editing box. It is the standard image for the
character in the set having the code of the edited character.

If, for example, the "IBM_Usa2.set" (the default character set) is
used, and the letter 'A' is edited and modified into a letter 'B', the
gadget will still display a letter 'A'. The font, however, will contain
two adjacent 'B's, having codes 65 and 66. Even if all characters in the

font were hand-drawn to look like 'B's, the "Default" gadget would still
display the standard image associated with each code defined for the
character set.

   The "Edit Character Set" function (section 4.11) can be used to design
a character set having the same (or – at worst – stretched) characters as
a font being edited. If that character set is then loaded when the font
from which the character images were taken is edited, the "Default" gadget
will always display a reduced image of the character in the character
editing box. This must be repeated every time the font is modified.
However, more direct and powerful functions, like those described in
sections
                3.13
                 ("Smaller"),
                3.15
                 ("The 'Dynamic View' Gadget") and 4.12
("Printer Test"), can be used to see the reduced image of the characters
in a font.

   When the user selects the gadget with the mouse, the "Quick Character
Selection" mode is activated. This causes a special box to be displayed,
the function of which is described in section
                3.22
                 .

3.8   The "CLR" Gadget

   When this gadget is selected, the content of the character editing box
is cleared (CLR comes from "CLeaR"). The character bitmap is reset. All
coloured dots in the box will disappear.

   Section
                3.9
                 describes how to restore the content of the editing box
after the selection of the "CLR" function.

3.9   The "UNDO" Gadget

   This function nullifies the effect of the last character editing or
resizing operation. If the last operation has caused an undesired effect,
it is sufficient to select "UNDO" to restore the previous image and size
of the current character. "UNDO" itself can be undone, selecting the
gadget more than once.

   "UNDO" can restore the character data if it is called after one of the
following functions:

   "X Size"   (section
                3.4
)
   "CLR"   (section
                3.8
)
   "UNDO"   (section
                3.9

```
                   )
    "Paste from Buffer"        (section
              3.12
              )
    "Draw with Pencil/Brush"  (section
              3.17
              )
    "Shift"                    (section
              3.19
              )
    "Flip"                     (section
              3.20
              ).
```

3.10   The "ON/OFF" Gadget

   With the Personal Fonts Maker the user can decide which characters in a
font should be used, and which should be excluded from the font. The
number of characters in a font is determined by the user. As explained in
section 2.8, there may be up to 257 different characters in a font.

   The "ON/OFF" gadget can be selected to determine whether some graphical
character data should be associated with a particular character set code
or not. When the gadget is selected with the mouse, it toggles from the ON
position to OFF, and vice versa. The editing box, the grid and the dots
inside it have different colours for "On" and "Off" characters. If the
character is "On", the grid is displayed in dark green and the dots are
black (or any other colour specified for the dots), while both the grid
and the dots of "Off" characters are light green.

   When the gadget is in the ON position, the character in the editing box
can be modified. The user can paint with the mouse, change the size of the
bitmap and do any necessary changes only if the character is "On".

   If a character is not to be used in the font being designed, it should
be marked as "Off". When the gadget is toggled from the ON position to
OFF, the data in the editing box is not cleared. However, the colour of
the box and its contents change to inform the user that the character
cannot be modified, unless it is switched "On" again. "Off" characters are
not saved when the font is saved, but they are preserved in the program's
memory along with all other characters. This makes it possible to set an
"Off" character back "On" without losing the character data. However, if
the font is stored and loaded again, "Off" characters are lost.

   The undefined character (section 2.8), which is the last character in
the font, and has code 256, is handled in a slightly different way by the
program. Even if the character is "Off", it is saved when the font is
saved as an Amiga font (sections 2.4 and 4.6). An Amiga font must always
contain graphical data for the undefined character. Therefore, the
undefined character is always stored as if it was "On". If the font is
stored in the standard PFM font format, the character is stored only if it
is "On".

   An "Off" character cannot be modified. If the user tries to draw in the
editing box, the screen flashes and an error message is displayed. Other
"passive" functions, which do not modify the character data, work even if

the character is "Off". For example, the character can be copied to the
internal buffer (section

        3.11

        ), or a brush can be defined (section

        3.18

). However, the brush cannot be selected using the right mouse ←
          button
("Cut" mode), as this would clear part of the character.

  "Off" characters are not output by the "Printer Test" (section 4.12).
If the size or the proportions of the characters are modified through the
"Font Description" requester (section 7.3), "Off" characters are cleared,
rather than being cut or stretched like "On" characters.

   As "Off" characters are not output or saved, they reduce the total
amount of memory occupied by the font data. This can be especially useful
when memory limitation becomes a problem. The buffer of some printers, for
example, is insufficient to store 256 different characters in
letter-quality mode. The characters effectively necessary to print a text,
however, are much less than 256. A short letter generally does not use
more than 50 different characters, including capitals, numbers and
punctuation signs. The "Quick Character Selection" function (section

        3.22

) and the "Font Statistics" requester (section 4.14) give the user
control and overview of the entire font at once. The "TextChars" program,
described in section 12.1, analyzes an ASCII text file and creates a
macro which switches to the OFF position all characters which are not used
in that file.

3.11   The "Copy to Buffer" Gadget

   The Personal Fonts Maker has an internal character buffer. An entire
character can be copied from and pasted to the editing box from the
buffer. This is useful when a character has to be moved or copied from one
position to a different one, in the same font or to another font. The
buffer is shared by the two font environments. The buffer memory is
independent from the brush memory.

   When the "Copy To Buffer" gadget is selected, a copy of the character
in the editing box is stored into the buffer. This includes the
character's image and the "X Size", "Space" and "Kerning" parameters. The
data can be recalled at any time using the "Paste from Buffer" function.
The data is not cleared until new data is copied to the buffer, or the
program is terminated.

3.12   The "Paste from Buffer" Gadget

   When this gadget is selected, the character data previously stored in
the buffer is applied to the current character. If the editing box already
contains a character image, this is replaced by the new data.
"Transparent" dots of the new data will clear "coloured" dots in the same
position as the old character, unless the "OR" mode is active (section

                         3.16
). The data in the buffer is never cleared by this operation, and ←
                           can
be used again.

    The "X Size", "Space" and "Kerning" parameters (sections
                         3.4
                         ,
                         3.5
                          and

                         3.6
) of the current character are automatically updated by the paste
function. If the character which is copied to the buffer comes from a font
having different sizes or dot densities than the destination font, the
Personal Fonts Maker can automatically stretch the image to fit the new
editing box perfectly. Section 7.10 describes the different stretch
options of the program. This is one of the features which distinguish the
buffer from the brush. When data is exchanged through fonts using the
brush, it is never stretched.

3.13   The "Smaller" Gadget

    This gadget can be used to make the character editing box smaller.

    The amount by which the box is made smaller depends on the quantity and
size of the individual dots which make up the character. Each time the
gadget is selected, the dot's height is reduced by one video pixel. The
width of the character dots (in screen pixels) is modified so that its
proportion with the dot height is as close as possible to the one
specified by the user (sections 7.3.3 and 7.3.4). The operation fails if
any of the two dimensions would become smaller than two video pixels as a
result of the resizing. An error message is displayed in this case.

    This operation is also executed automatically whenever the character
editing box is too wide to fit in the screen.

3.14   The "Larger" Gadget

    This function is the opposite of "Smaller", described in section
                         3.13
                         .
The "Larger" gadget can be used to make the character editing box larger.

    As for the reverse procedure, this function acts on the dot units which
make up the character editing box. When the gadget is selected, the height
of the dots which make up the character editing box is increased by one
video pixel. The width of the dots is increased so that the overall X/Y
ratio is as close as possible to the proportions specified by the user
(sections 7.3.3 and 7.3.4). If, as a result of the operation, the
editing box would not fit in the program screen, the operation fails and
an error message is displayed.

    Neither "Larger" nor "Smaller" modify the character bitmap or any other
information associated with the font or the character itself. The only

purpose of these two functions is to change the degree to which the
character editing box is magnified.

3.15   The "Dynamic View" Gadget

   This gadget controls the activation of the dynamic view mode. When this
function is enabled, it is possible to verify the aspect of the current
font as if it were being used as an Amiga screen font. Text can be typed
by the user in a special screen, as if it were typed in a big string
gadget (section 1.9.6). The resolution of the screen is determined by the
font density (XDPI, YDPI) and ratio (XDPI/YDPI), as explained in sections

            3.24
              ("The Font Requester") and 7.3.3 ("X Dpi").

   The resolution of the screen is automatically adapted whenever the font
density settings are changed (either manually or by loading a parameter
file). The program tries to maintain the screen display ratio as close as
possible to the font ratio. If the ratio is between 0.75 and 1.33, the
resolution of the screen is determined by the dot density. A density of 50
dpi or higher activates the high resolution interlace mode.

   The dynamic view mode is most useful when designing Amiga screen fonts.
The text which is displayed in the dynamic view screen is immediately
updated to reflect any changes to the current font. The font is displayed
as it might appear after an "Export Amiga Font" operation (section 4.6).
Only characters of the Amiga character set can be displayed. The encoding
vector is used to remap the characters into the Amiga character set
(section 2.8, "Character Sets").

   This function has not been designed in order to emulate the printer's
output. The basic video picture elements (pixels) cannot be adapted to fit
the printer's dot size, shape, placement and density. This can be achieved
using the "Printer Test" functions (section 4.12).

   When the "Dynamic View" gadget is first selected, a new screen is
opened. If a dynamic view screen is already open (but not the frontmost
screen), it is moved to the front. The gadget can be selected again to
close the screen.

   Text can be typed in the screen as in a standard Amiga string gadget
(section 1.9.6). A flashing cursor indicates that the dynamic view screen
is active. Appendix I lists the special keys which can be used to move
the cursor and edit the text. The <Help> key can be used to make the
character under the cursor the current character (i.e. display it in the
character editing box). The main screen can be activated again by pressing
the <Return> key or hitting the right mouse button.

   If the "Fixed Pitch" attribute is set (section 8.14), and all
characters have the same "X Size" (section
            3.4
            ), the value of the "Space"
and "Kerning" parameters (sections
            3.5
             and
            3.6

) is ignored. The status
of the "Right to Left" attribute (section 8.15) determines the direction
of text display and editing.

3.16   The "OR" Gadget

   This gadget changes the status of the "OR" option whenever it is
selected. If the "OR" mode is not active, the selection of the gadget
causes it to become active, and vice versa. In the "OR" mode, the gadget
remains highlighted and looks like a push-button in the "pressed"
position.

   The following functions are influenced by the status of the "OR"
gadget:

     "Paste from Buffer"   (section
                    3.12
)
     "Draw with Brush"   (section
                    3.17
)
     "Shift"   (section
                    3.19
)
     "Flip"   (section
                    3.20
).

   In the "OR" mode, the graphic functions listed above cause all the
graphical data written to the character editing box to overwrite the
existing data. This is called the "OR" mode, as the image bitplane is the
result of the bitwise OR operation (section 2.7.3.3) of the bits in the
original editing box and the bits of the brush or the character buffer, or
the image resulting from a shift or flip operation (sections
                    3.19
                     and

                    3.20
).

   In practice, "coloured" dots are copied to the destination bitplane,
while "transparent" dots in the source bitplane are ignored. If the right
mouse button is used to write in the box (erase mode), coloured dots of
the source bitplane clear the corresponding dots in the destination
bitplane.

   For example, when the "OR" mode is active, successive "Shift"
operations (section
                    3.19
) cause the character in the editing box to
become bolder and bolder. The "Flip" function can be used to create
symmetries.

3.17   The "Pencil/Brush" Gadget

The mouse can be used to draw in the character editing box either in
"Pencil" or in "Brush" mode. In the "Pencil" mode, a single, coloured, dot
is used as the unit for paint operations. If the "Brush" mode is selected,
the previously defined (section
                3.18
) or loaded (section 5.1) brush is
used to "write" into the editing box.

If the mouse pointer is over the character editing box and the user
presses the left mouse button, the specified graphic data (a dot or a
brush) is written in the box under the mouse pointer. If the right mouse
button is pressed instead, the data under the mouse pointer will be
erased where there are dots set in the selected graphic data. This is
called the "erase" mode.

Normally ("erase" and "OR" modes off), if the brush is used to write,
the entire brush replaces the data under the mouse pointer. The entire
area under the brush is cleared before the coloured dots of the brush are
copied to the editing box. This means that transparent dots in the brush
will create a "hole" in the destination area.

If the "erase" mode is selected (right mouse button), the area will be
cleared and coloured dots in the brush will not be copied to the
destination area, as they erase dots in the box that have just been
cleared by the entire brush. In this way all the dots in the editing box
which are under the brush will be cleared.

If the "OR" mode is active, the area under the brush is not cleared
before the dots from the brush modify the destination area in the box.
This means that only the coloured dots of the brush affect the data in the
character editing box.

If both the "OR" and the "erase" modes are active, the bits which are
set in the bitplanes of the brush (those which make up the coloured dots)
will clear the corresponding bits in the bitmap of the character editing
box.

In the "Pencil" mode, the program behaves as if a brush consisting of
one coloured dot (a 1 by 1 brush) was used. However, if a brush was
defined, it is not cancelled by the selection of the "Pencil" mode. The
selection of the "Brush" mode causes the same brush to be available.

The same brush data can be shared between fonts of different formats.
Unlike the data stored in the character buffer (section
                3.12
                ), however, a
brush is never stretched or adapted automatically by the Personal Fonts
Maker, even if the X/Y proportions of the font from which the brush was
defined and the destination font differ. The rows and columns of dots
which were copied from the original font are pasted to the same number of
dots in the destination font.

When the user positions the brush by using the mouse, the outline of
the brush is displayed in the character editing box. Those parts of the
brush which are not over the box are not displayed, nor are they pasted if
a mouse button is pressed. Section 5.9 explains how to select the part of
the brush which is to be linked with the pointer (e.g. centre or edge of

the brush).

3.18   The "Define Brush" Gadget

    When this gadget is selected, the "Brush Definition" mode begins. The
gadget remains highlighted until a brush is defined using the mouse. If
the gadget is selected again while it is highlighted, the brush definition
is cancelled.

    The brush definition mode is active until the user selects one corner
of the brush rectangle in the editing box, presses a mouse button, moves
the mouse pointer to the opposite corner, and releases the mouse button.
The "Draw with Brush" mode is automatically activated after a brush has
been defined, so that the user can work with the brush immediately.

    During the brush definition, the mouse pointer is used to select the
dots which mark the beginning and end of the brush. Since the brush is
defined as a rectangle, the two positions represent opposite corners of
the rectangle. When the mouse pointer is over the character editing box
and a mouse button is pressed (and held down), a green border is drawn
around the dot under the pointer. This is to highlight the rectangle which
is being selected. One corner of the rectangle will remain linked with the
initial position. The other corner will follow the mouse pointer. When the
mouse is moved (with the button still pressed), the rectangle will change
shape accordingly. When the mouse button is finally released, the
definition of the brush is complete. A copy of the selected area is stored
as "the brush".

    The brush can be defined either with the left or with the right mouse
button. If the left button is used, the brush is simply copied from the
selected region of the editing box. The data in the box is not modified,
therefore this operation can be performed even when the current character
is "OFF" (section
                3.10
). If the right mouse button is used instead, the
brush is "cut" from the character editing box. The selected region in the
box is cleared.

    Once a brush is defined, it remains in the program's memory until
another brush is defined or loaded, or the program is terminated. Chapter
5 explains how it is possible to load a brush, save it, modify it, or
change the position of the brush handle.

    A single '−' (minus) or '+' (plus) sign can be written instead of a
number in the string gadgets of the "X Size", "Space" and "Kerning"
parameters (sections
                3.4
                ,
                3.5
                 and
                3.6
). The program automatically
replaces the '−' sign with the width of the brush, while the '+' sign is
replaced with the width of the brush stretched to the maximum font height.
If no brush has been defined, the gadget is filled in with the previous
value.

3.19   The Four "Shift" Gadgets

   The "Shift" gadgets are used to shift (move) the content of the
character editing box towards the specified direction. Each gadget has an
arrow symbol which indicates the direction towards which the image is
moved when that gadget is selected. Whenever a gadget is selected, the
dots in the box are moved by one position.

   The dots at the margins of the box are never lost as a result of a move
operation. The character editing box can be imagined as a torus. The dots
which are moved out of the box on one side come back in from the opposite
side.

   If the "OR" mode is selected (section
              3.16
              ), only the coloured dots
are moved. The coloured dots can be "smeared" all over the character
editing box. This can be useful to create different bold levels of a
character. If, for example, the "Shift Right" gadget is selected when the
"OR" mode is active, the character displayed in the editing box will
become bold. If the character has some coloured dots on the rightmost
column of the box, the "X Size" could be increased (section
              3.4
) before
the move operation to avoid the appearance of undesired dots in the
leftmost column.

3.20   The Two "Flip" Gadgets

   The data in the character editing box can be "flipped" horizontally and
vertically. The two "Flip" gadgets, on the right of the four "Shift"
gadgets, can be used to produce a mirror image of the character in the
editing box, or turn it upside-down. The symbols on the gadgets indicate
the direction of the rotation.

   If the "OR" mode is active, only the coloured dots of each half-image
are copied to the other half, producing symmetrical images.

3.21   Reference Points

   As described in section 2.3, four reference points can be freely
positioned by the user to mark certain vertical positions of the character
editing box. These (horizontal) lines of reference help the designer of a
font to maintain a coherent relationship from one character to another.
Four arrow-symbols on the left of the character editing box indicate the
current position of the points.

   The reference points should be positioned at the height of the cap
line, the mean line, the baseline and the underline position, whereas the
ascender line and the descender line are naturally marked by the top and
bottom lines of the character editing box. Some macros and parameter files
depend on these standard settings.

Section 2.3 describes the typographical usage of these lines. The user designing a new font, or just adding some characters to an existing one, can check at a glance whether the image in the character editing box fits within the design rules of the font.

To move the reference points, the mouse pointer must be positioned over the reference point zone, which is a vertical strip immediately on the left of the character editing box, where the triangles associated with the reference points are displayed. When the mouse pointer is in this zone, the image of the pointer is changed into a couple of converging arrows, indicating that the Personal Fonts Maker is ready to accept user input regarding the reference points.

When the pointer is in the reference point zone, and while the left mouse button is pressed, the reference point mode is active. In this mode, the top left part of the screen is used to display the vertical position of each of the four reference points (numbered 1 to 4), and the current position associated with the mouse pointer. A horizontal line, crossing the entire character editing box, highlights the line of reference which could be associated with the current mouse position.

As reference points show up between horizontal rows of dots, rather than on the rows, the number of possible positions for the reference points is equal to the "Y Max" font variable (section 7.3.2) plus one. The numbering of the positions starts from the top of the editing box, with position 0 (zero).

Each of the four reference points has an associated function key, from <F1> to <F4>. While the reference point mode is active, one or more function keys can be pressed to move one or more reference points to the highlighted position. More than one key can be pressed at the same time, and the mouse pointer can be moved while the keys are held down. The triangular arrows associated with the reference points are moved as the mouse moves. Each arrow has a different colour, but when two or more arrows occupy the same position, only one arrow can be seen in that position.

The parameters containing the positions of the reference points can be addressed from the FFDL, as described in section 2.7.2.14. The FFDL variables associated with the reference points are named REF1, REF2, REF3 and REF4.

3.22   Quick Character Selection

When the "Default" gadget of the main window of the Personal Fonts Maker is selected (section
                3.7
                ), the quick character selection mode is
activated. This causes a special box to be displayed.

The quick character selection box is displayed instead of the character editing box. The images and the data displayed in the "Character #", "Default" and "ON/OFF" gadgets are updated by the program, but the "Character #" and "ON/OFF" gadgets cannot be selected. The "Mode" and "Cancel" gadgets, which replace the space normally reserved for the "X Size", "Space" and "Kerning" gadgets, are the only gadgets which can be

selected.

   All the default images of the characters in the current set are
displayed in the box. The characters are displayed in vertical columns,
each containing ten images. When the mouse pointer is moved over one of
the images, that image becomes highlighted. If the pointer is not over one
of the images, the image associated with the current character, which was
displayed in the character editing box before the quick character
selection mode was activated, is highlighted instead. The data displayed
in the "Character #", "Default" and "ON/OFF" gadgets always refers to the
highlighted character.

   The "Mode" gadget can be used to switch between "Mode 1" and "Mode 2"
of the quick character selection. In "Mode 2", all characters which are
"Off" (section
                3.10
                ), are marked as such by an "OFF" symbol, drawn over
the character image. This makes all "Off" characters recognizeable at a
glance. When the left mouse button is pressed over a character image, the
quick character selection mode terminates, and the highlighted character
becomes the current character. The selection of the "Cancel" gadget also
causes the termination of the quick character selection mode, but the
current character and the "On/Off" status of all characters remain the
same as before the activation of this mode. Pressing the <Esc> key is
equivalent to selecting the "Cancel" gadget.

   If the left mouse button is pressed when the pointer is over one of the
character images, the quick character selection mode terminates and that
(highlighted) character becomes the current character. Whenever the
pointer is over a character image, and the right mouse button is pressed,
the status of that character is switched from "On" to "Off", and vice
versa (section
                3.10
                , "The 'ON/OFF' Gadget"). If "Mode 2" is selected, the
"OFF" symbol over the character is set or cleared in accordance with the
new status of the character. The "OFF" graphical symbol is placed over the
characters which are off only if "Mode 2" is selected, while the "ON/OFF"
gadget is always updated in both mode 1 and mode 2.

   Like all other types of character selection, the quick character
selection can be recorded in a macro (see chapter
                3
). Any modifications
to the "On/Off" status of characters, however, are not recorded, unless
the gadget on the main screen is used.

3.23   The File Requester

   The file requester is one of the most used requesters of the Personal
Fonts Maker and Printer Driver Modifier. More than a dozen functions of
both programs use the file requester described here. The window containing
the file requester has a title indicating the function which displayed the
requester.

   The requester is called "file requester" because it allows the user to
work with file names (section 1.4). When the user selects a function

which needs to write data in a new file, update a file or load data from
an existing file, the standard file requester is displayed. In general,
the file requester is displayed whenever the user must define which file
is going to be the target of an input/output operation. Sections 1.4,
1.5 and 1.12 should be carefully read and understood before this section
is read.

To identify a file so that it can be accessed, the user must specify
the name of the volume or device on which the file is stored, either
physical or logical (sections 1.4 and 1.12, e.g. "PFM", "DF0" or
"PFM_ASSIGNED"), the remaining path (i.e. the drawers the file is in) and
the file name itself. The names can be written using the keyboard or
selected with the mouse. The file requester contains different gadgets,
each of which is described in detail in the following paragraphs.

The first gadget in the requester is the "Path" string gadget. The
selected path (including the device name) is always displayed in the
gadget. The gadget can be selected to be updated with the keyboard, but
this is normally not necessary, as both the volume or device name and the
names of the drawers can be selected with the mouse as explained below.
The <Return> key must be pressed after the text of the string gadget has
been modified with the keyboard.

The format of the text in the "Path" string gadget is the following:
device name + ':' (colon) + drawer names. A '/' (slash) must separate two
drawer names. This means that each drawer name is separated by the name of
the drawer containing it by a '/' sign. The volume or device name should
always be specified and followed by a colon, even if no drawer names are
written. If no volume or device name is selected by the user, the name of
the program's current directory is automatically placed in the string
gadget by the program. The file name itself must not be written in this
gadget, but in the "File" string gadget.

When the requester is displayed for the first time by a function, the
"Path" gadget contains the default path name of that particular function.
Section 1.12 lists all default paths, while section 2.6 explains how to
change the default paths.

At the right of the "Path" gadget is the "Parent" gadget with the '/'
(slash) symbol. When this gadget is selected, the name to the extreme
right in the "Path" string gadget is removed from the path. The "Parent"
gadget can be used to delete drawer names and the volume or device name
without having to use the keyboard. The gadget is called "Parent" because
it can be used to move to the parent directory of the current directory,
which is the directory which contains the currently displayed files.

Below the "Path" and "Parent" gadgets is the list box. The list box can
either contain file and directory names or physical and logical device and
volume names.

All file and directory names found following the specified path can be
viewed in the list box at the centre of the requester. All file and
directory names are inserted in a list. After the complete list has been
read by the program, the names are sorted in alphabetical order. Directory
names are grouped at the bottom of the list, preceded by a "(dir)" prefix.
Not all names which are listed necessarily belong to files or directories
which contain data useful for the Personal Fonts Maker (section 1.5).

Some of the names used by the Amiga operating system are excluded from the list (sections 7.11.2 and 11.7.2).

If the list of names is longer than the screen space reserved for the list box, only part of the file names is displayed. The knob on the right of the list box can be used to shift the displayed portion of the list up or down. The knob is a gadget which can be moved when it is selected. While the left mouse button is pressed, the mouse can be moved up or down. If the mouse and the knob are moved up, the displayed names are moved down, so that some space is freed to display the names which are closer to the top of the list. The opposite happens if the knob is moved down. The beginning of the list is always displayed when the knob is in the topmost position. If the knob is in the lowest possible position, the end of the list is displayed. Intermediate knob positions cause proportionally equivalent parts of the list to be displayed. The knob can also be moved by briefly pressing the left mouse button when the pointer is over the container of the knob (but not over the knob). In this way, the knob is "attracted" towards the mouse pointer whenever the left mouse button is pressed and the names in the list box are scrolled by an entire list box page at the time.

The list box normally displays the names of all files and directories contained in the directory and volume (or device) specified in the "Path" gadget. If the right mouse button is pressed, the list box displays all physical and logical volume and device names (sections 1.4 and 1.12). The Personal Fonts Maker automatically detects all storage devices connected to the computer, the volumes (e.g. disks) mounted in these devices and the logical names assigned to both the volumes and the devices. All these names are displayed in the list box. When one of these names is selected with the left mouse button, the "Path" gadget is updated and the list box is again used to display file and directory names. If possible, only the initial part of a longer path is modified, i.e. any names of drawers are not cleared. This is done if the same path also exists on the newly selected volume or device.

It should be noted that when the list box is used to display volume and device names, the same volume can be accessed by selecting either the volume name or the name of the device containing that volume. If the "Expand Path" option is activated (sections 7.11.1 and 11.7.1), the program will automatically write the correct volume name in the "Path" string gadget. Devices other than standard disk drives are included in the list box. Files saved in devices like "RAM" and "RAD" are lost after the computer is switched off.

Once the name of a file, directory, volume or device is displayed in the list box, it can be selected with the mouse. When the left mouse button is pressed and released, the name displayed under the mouse pointer is selected. If the name of a volume, device or drawer is selected, the "Path" string gadget is automatically updated. If a file name is selected (e.g. the name of a font file), the "File" string gadget containing the file name is updated.

Every gadget of the file requester, including the knob, can be selected while the list box is being updated. During this phase, new names are continuously inserted at the bottom of the list, rather than sorted in alphabetical order. In this way, the order of the displayed names cannot change unexpectedly while the user is about to select a name. To avoid

undesired selections during the display update, the displayed names are
not modified while the left mouse button is pressed, and the selection of
a name is accepted only if the mouse button is released over the
highlighted name. If the mouse button is pressed when the pointer is over
a name, but released when the pointer is not over the same name, no
selection is made.

   The updating of the list box, especially if the list is long and the
device is not extremely fast (e.g. a standard disk drive), can be
annoyingly long. The file requester has a special feature which eliminates
all unnecessary delays. Once a list is complete, it is stored in the
program's internal memory (unless the same list has already been saved
before, and is still up to date) and preserved until the program is
terminated. This is done for the lists of all functions which use the file
requester, if there is enough memory. Whenever a function displays the
file requester, or the user modifies the content of the "Path" gadget, the
program scans through all its stored lists to check if the list associated
with that particular path has been read before. The program also verifies
if other programs have added, modified or deleted any names in a directory
while the associated list was stored. If the program finds a matching and
updated list, this is immediately displayed in the list box. The user does
not have to wait for the file requester to slowly create a list which is
already in the program's memory.

   The "File" string gadget under the list box contains the name of the
file. The content of the string gadget is automatically updated if a file
name is selected with the mouse from the list box.

   The maximum length of the full path (device + drawer names) which can
be written in the "Path" gadget is 128 characters. As explained in section
1.4, the length of file names ("File" gadget) should not exceed 25
characters. The number of names which can be stored in a list and
displayed in the list box is limited only by the available memory.

   To confirm the selection of a file and remove the requester, the
"Proceed" gadget must be selected, or the <Return> key can be pressed when
the cursor is in the "File" string gadget, or double-clicking (quickly
pressing the left mouse button twice) on a file name in the list box. The
double-click function can be disabled, as described in sections 7.11.3
and 11.7.3.

   If the "Cancel" gadget is selected, the function which displayed the
file requester is aborted. Any changes in the gadgets of the file
requester are undone.

   An optional safety check (section 7.11.4) causes a warning message to
be displayed if the function which displayed the file requester would
overwrite some existing data. Other messages are relative to the function
which displayed the file requester. Appendixes G and H list all
messages of the Personal Fonts Maker and the Printer Driver Modifier.
Section 14.1 contains important information on how to handle errors which
may occur during read/write operations to peripheral storage units.

   Section 7.8 explains how to choose whether Workbench icons should be
associated with files being saved or not. Different kinds of icons can be
selected.

The Amiga operating system may perform another disk write operation some seconds after a file has been stored. For this reason, a disk should not be ejected from the drive (nor should the computer be switched off) for some seconds after the disk drive activity indicator light goes off.

3.24   The Font Requester

The "Import Amiga Font", "Export Amiga Font" and "Delete Amiga Font" functions (sections 4.5, 4.6 and 4.16.2) use a special version of the file requester (section
                3.23
                ), called the font requester. This is
necessary because fonts are stored in a more complex format than other files. For example, fonts can only be read from and saved to a directory to which the logical name "Fonts" has been assigned. For each typeface, a drawer must be created in the "Fonts" directory just mentioned, and an information file describing the fonts (sizes and styles) of that typeface must be created. All this is done automatically by the Personal Fonts Maker. The font requester frees the user from the burden of having to remember how the Amiga operating system handles font files.

The font requester allows the user to type in the "Fonts" (same as "Path" in the file requester) string gadget the physical or logical name of the volume or device to which the "Fonts" logical device name is to be re-assigned. The list box only displays the names of fonts, followed by the vertical size in dots of the font. The names of the default system fonts (e.g. "topaz 8 of" and "topaz 9 oetf") appear even if the specified "Fonts" directory does not contain any font files, as these fonts are always available and usually stored in ROM. The "Parent" gadget cannot be used, since directory names are not necessary.

An Amiga font file is identified not only by the typeface name, but also by the font size and style attributes. Fonts in different sizes and/or with different attributes, but with the same typeface name can exist.

During a load operation, the font name can be selected with the mouse from the list box. This automatically causes the desired size and style to be selected as well. It is also possible to manually enter this information in the string gadget on the right. This can be particularly useful if no font bitmap file exists for a particular font size and style. If the requested font size does not exist, the Amiga operating system either scales the font to the desired format (in version 2.0 and beyond, only if the "Scaled Font" attribute is specified), or loads the closest existing font. The result of scaling a font to a new format is usually better if the point of departure is a vector font. When on the contrary font data is written, the size and style are selected by the program, depending on the font attribute flags (chapter 8), the size and the display ratio of the current font.

Font attributes are appended to the number indicating the size of the font, following a widespread Amiga standard. One lower case character is associated with each of the Amiga font attributes. The characters are:

    CHARACTER   ATTRIBUTE

```
        u           Underlined
        b           Bold
        i           Italic
        e           Expanded
        r           Right to Left
        f           Fixed Pitch
        t           Tall Dots
        w           Wide Dots
        c           Colour Font
        o           ROM Font
        s           Scaled Font
        v           Vector Font
```

   The "Tall Dots" flag is set automatically by the Personal Fonts Maker
if the display ratio (XDPI/YDPI) is higher than 1.33. Conversely, "Wide
Dots" is set if the ratio is lower than 0.75. The same criterion is
adopted to determine the format of the dynamic view screen (sections
                3.15
                and 7.3.3).

   The "Colour Font" flag is read by the Personal Fonts Maker, but never
set, since the program saves single-bitplane fonts. A "ROM Font" is one
that comes bundled with the system (usually in ROM), like "Topaz".

   It may sometimes happen that no custom-designed font bitmap exactly
matches a particular font name, style and size. The Amiga font handling
routines can algorithmically modify an existing font in order to build a
font with the desired format. A font generated by such a procedure is
flagged with the "Scaled Font" attribute.

   As explained in section 2.1 ("From Gutenberg's Type to Computer
Fonts"), the Amiga operating system (version 2.0 and beyond) supports
vector fonts. The "Vector Font" flag is used to immediately recognize
vector font files.

   All other flags can be controlled through the "Attributes" menu,
described in chapter 8.

   Internally, when a font is saved, the Personal Fonts Maker first checks
if another font of the same typeface exists. If not, a directory with the
name of the typeface and a file describing the content of that directory
are created. Then, the font data is written to a file in the new
directory. The name of the file containing the font consists of a number,
indicating the size of the font, followed by the font attribute
characters. For example, when the Personal Fonts Maker stores the
"Andromeda_24" font as an Amiga font, it first creates a directory called
"Andromeda" and a font contents file named "Andromeda.font". Then, the
font data is written to the "24t" file in the "Andromeda" directory. "24t"
means that that file describes a font of size 24, designed for "Tall Dots"
representation.

   When the Personal Fonts Maker adds or modifies some Amiga font data, it
automatically updates the description of the fonts available for that
typeface, which is stored in a separate file, as explained above. Some
other font editing programs do not update this information file, so it
sometimes happens that the font information file does not describe the
fonts which are really available. To avoid this, the Commodore "FixFonts"

program should be run whenever a program other than the Personal Fonts
Maker is used to modify some font files. "FixFonts" must also be used if
any font files are renamed, moved, deleted or copied manually using the
Amiga Shell or CLI.