

0053

COLLABORATORS

	<i>TITLE :</i> 0053		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		January 16, 2023	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	0053	1
1.1	Personal Fonts Maker - 6. PFM: The Macro Menu	1

Chapter 1

0053

1.1 Personal Fonts Maker - 6. PFM: The Macro Menu

6. PFM: The Macro Menu

Font design often involves a lot of repetitive work. If, for example, an existing font is to be used as a point of departure to create an italic variation of the whole font, it would be faster to show the program how to italicize one character, and then let the program repeat the sequence on the desired number of characters. Another operation which font designers often have to repeat is to copy letters from the standard 7-bit ASCII set range to several more positions in the font, where those same characters will have to appear with diacritical marks. Once the basic characters are copied, it is often sufficient to add some standard signs on the top of the character. The most annoying part of this work is the selection of all the source and destination codes of the characters to be copied. If this could be done only once, just to "teach" the program what to copy and where to paste it, the subsequent jobs would be less repetitive for the designer. The Personal Fonts Maker's powerful macro functions help the user in these situations and many more.

A macro is a program interpreted and executed by the Personal Fonts Maker. All functions which can be executed from a macro can also be selected manually by the user, using menus, gadgets, the mouse and the keyboard. The main purpose of macros is to automate a sequence of Personal Fonts Maker functions. As described in section 12.5, the Personal Fonts Maker comes with different pre-defined macros. Defining a new macro is as easy as showing the program the sequence of operations to do. Experienced users can view the macro program listings, or even use a text editor to write or modify macros. This is usually not necessary.

Each macro is identified by a name and an associated letter on the keyboard. Up to 26 different macros (as the number of letters from 'A' to 'Z') can be stored in the program's memory at the same time. Macros are shared by both font environments. A macro can be saved to and loaded from disk. When a macro is defined or loaded, it is associated with the letter on the keyboard which will be used as a shortcut to execute that macro. When the program is started, it loads all macros specified in the default parameter files (sections 2.6, 2.6.16 and

6.1
).

The following sections explain how to load, save, define, delete, examine and execute macros. Section

6.6

("Examine Macro") contains a detailed description of the program functions which can be accessed from within a macro.

6.1 Load Macro

This function loads a macro which was previously stored. The standard file requester (section 3.23) is used to select the file containing the macro. The file name will remain associated with that particular macro when it is stored in the program's memory.

Before the file requester, a special requester, used to specify the key which is to be linked with the macro being loaded, is displayed. This requester is called the macro requester. Most functions described in this chapter use the macro requester. A list box, similar to that of the file requester, contains the names of all the macros currently in memory. There are 26 elements in the list. The elements are gadgets which can be selected with the mouse. Each element begins with a letter from 'A' to 'Z'. The names are ordered after the keyboard letter associated with each macro. The text in those elements which already contain a macro is displayed in dark green. All other (empty) elements contain only the keyboard letter, in light green. Both kinds of elements can be selected. If an element which already contains a macro is selected, the new macro will replace the previously existing macro. Once an element (gadget) is selected with the mouse, it remains highlighted until a new element is selected.

At the bottom of the macro requester are the usual "Proceed" and "Cancel" gadgets. "Proceed" will not work (it will cause a short screen flash instead) if no element in the list box was selected, i.e. if no keycap letter was defined. A double click on a gadget in the list box is equivalent to the selection of both the associated element and "Proceed". Section 7.11.3 ("Double Click") explains how to activate or disable this option. "Cancel" always aborts the current operation. Other functions use slightly different versions of the macro requester, containing one or more additional gadgets.

When the macro requester is used to link a key with a macro it is important that the selected letter can easily be associated with that particular macro. For example, it would be easy to remember that the key <O> is associated with the "Outline" macro. The letters associated with the macros do not need to be selected starting from 'A'. It does not matter whether the only two macros are associated with the keys <K> and <Q>, as long as these letters mean something to the user. Of course, a macro can also be selected with the mouse, from a list displaying all the names of the macros, rather than pressing a single key. This is described in sections

6.7

("Execute Macro") and

6.8

("Controlled Execution").

If the program encounters an error in the macro which is being loaded, an error message is displayed and the operation is cancelled. The content of the selected macro is cleared from the program's memory. Appendix G explains the meaning of the error messages. Section

6.6

("Examine Macro")

and appendix L ("Macros Command Consistency Table") list the rules which are most often violated, resulting in an error.

6.2 Save Macro

As described in the introduction to this chapter, as well as in section

6.3

("Start Record"), up to 26 different macros can exist at the same time in the Personal Fonts Maker's memory. Any of these macros can be stored to a file separately.

The macro requester, identical to the requester described in section

6.1

("Load Macro"), allows the user to choose the macro to be stored.

But, unlike the "Load Macro" function, only an element of the list box containing a macro can be selected. The program will not let the user select an empty element. It would not make sense to store... nothing.

The macro requester is followed by the file requester. A warning message can be displayed if a file with the same name already exists (section 7.11.4 "Confirm Overwriting"). The function can be aborted selecting the "Cancel" gadget of either the macro requester, the file requester, or the requester containing the warning message.

The macro is saved as a plain ASCII text file, as described in sections

6.3

("Start Record") and

6.6

("Examine Macro"). In fact, such a file can

also be created by the user with a word processor or a text editor, as long as the commands contained in the macro file follow the rules explained in section

6.6

.

The file requester contains the name originally assigned to the macro as the default file name. This name can, however, be modified by the user. The macro file is "freed" from the link to a keycap letter. When the macro is loaded again, either with the "Load Macro" function, or from a parameter file (sections 2.6, 2.6.16 and 7.1), it is again associated with one of the 26 letters.

As described in sections 1.9.7 and 7.8 ("Icons"), the Personal Fonts Maker can associate a Workbench icon with the macro file.

6.3 Start Record

This function starts the automatic macro record mode of the Personal Fonts Maker.

Before the actual recording begins, the macro requester is displayed to allow the user to select a single character and a name to be associated with the macro to be recorded. The macro requester used by this function has one additional string gadget, where the user can write the name to be assigned to the macro. The name is selected automatically if an element in the list box which already has an associated macro is chosen. In this case, the new macro will replace the macro that was previously assigned to the same key. If no name is selected, either from the list box, or through the string gadget, the program will assign the default "Unnamed_x.mcr" name, where the 'x' suffix is replaced by a letter in the range 'A' to 'Z'.

While the macro record mode is active, an 'M'-sign ('M' like "Macro") is displayed under the usual mouse pointer. This is to remind the user that all actions are being recorded. There is no limit to the number of actions which can be recorded in a single macro, apart from the available system memory.

The "recordable" actions and commands include all gadgets of the main program screen, any modifications to the content of the character editing box, and the brush editing functions. The program will also record the functions executed through their keyboard shortcuts. The menus which cannot be selected (because the associated functions cannot be recorded) are "ghosted", as described in section 1.9.8 ("Menus").

When the "ON/OFF" gadget, or another two-state gadget is selected, the program records the new state, rather than the transition to the opposite state. For example, if the program records the selection of the "ON/OFF" gadget which activates the current character, but the macro is later "re-played" when the current character is already "On", the character status will remain the same.

When an editing operation in the character editing box is recorded, either in the "Pencil" or in the "Brush" mode, the absolute coordinates of the operation are recorded. This means that when an action in the character editing box is recorded, it will later be re-played in exactly the same position in the box. The position, defined as the number of horizontal and vertical dots from the top-left of the box, will not vary if the size of the box is different from the size of the box in which the macro was recorded.

When a brush is defined during the macro record mode and the mouse pointer is either at the right or/and below the limit of the character editing box when the mouse button is released, the final position will be recorded as "rightmost" or/and "lowest-possible". In this case, it is possible to record a position in the character editing box relative to the right, or/and bottom, rather than an absolute coordinate from the top left corner. This is useful when a sequence like "Copy the entire character, reverse it, and paste it back" is to be recorded. Such a sequence would not work properly on proportionally spaced fonts, as the sizes of some characters would inevitably be different from the size of the character on which the macro was recorded. If the normal absolute position is recorded in such a case, the macro will reverse only part of any character larger

than the character used to record the macro. The method just described solves this problem. Similarly, a reference point can be set to the bottom-most position by moving the mouse pointer below the reference point zone (section 3.21).

There is another way of defining a position in the character editing box in a macro. It allows the user to freely define relative-positions and positions to be specified by the user when the macro is run. This technique, however, requires the macro to be written or modified in its text format, rather than recorded through this function. The text format is described in depth in section

6.6

.

The first instructions in a macro are generated automatically by the program when this function is called. These instructions contain information on the initial status of the "OR", the "Pencil/Brush", the "Define Brush" and the "Stretch" modes (sections 3.16, 3.17, 3.18 and 5.8.1) and the position of the brush handle (section 5.9). This information, placed at the beginning of a macro, ensures that a macro will produce the same results when it is executed as when it was originally recorded.

The macro record mode can also be started by pressing a key from <A> to <Z> while <Shift> and <Ctrl> are held down. This sequence replaces this function, and does not cause the initial macro requester to be displayed. The user cannot specify a name for the macro being recorded. The default "Unnamed_x.mcr" name is assigned to the macro. The key pressed by the user is assigned to the macro and appended to the macro name. The name of the macro can later be changed with the "Examine Macro" function (section

6.6

), or when the macro is saved.

This function can be recalled by pressing the <M> (capital 'M') key on the keyboard. This will produce the same effect as the selection of the "Start Record" menu item. This is different from pressing <Shift>, <Ctrl> and a letter-key, which does not display the macro requester.

The macro record mode is terminated by selecting the "End Record" function, as described in section

6.4

.

Sections 12.5 ("Predefined Macros") and 13.2 ("A Simple Macro") contain additional information on macros for beginners.

6.4 End Record

This function terminates the macro record mode. The 'M' sign is removed from the mouse pointer image. The macro which has just been recorded is immediately available to be executed, saved or examined.

6.5 Delete Macro

This function allows the user to remove one or more macros from the list of the macros currently associated to the 26 keyboard keys from <A> to <Z>.

A variation of the macro requester employed by the "Save Macro" function (section

6.2

) is used to select the macros to be deleted. Unlike that requester, however, more than one macro can be selected with the requester displayed by this function. The names of the selected macros remain highlighted until they are selected again with the mouse. A double-click on an element in the list box causes that element to be highlighted and return to the normal state, rather than cause the requester to disappear.

The memory occupied by the deleted macros is freed. Usually, this is not a lot of memory. An average macro does not occupy more than 1 Kbyte of RAM. If the macros were saved to a file, those files are not deleted.

It is not necessary to use this function to "free" a key before a new macro (loaded or recorded) is associated to the same key. This is automatically done when a new macro is assigned to a key already associated with a macro.

6.6 Examine Macro

The Personal Fonts Maker stores macros in a text format which the user can easily read. A macro program consists of a series of lines of text, each containing one or more commands. The macros can either be examined by the user using a word processor (or a text editor) to load a macro file saved by the Personal Fonts Maker, or viewed using this function. This function does not allow the user to modify a macro. A word processor like Personal Write can be used to do this.

The first part of this section describes the use of this function, while the remaining part explains the format used by the Personal Fonts Maker to store macros, and the rules which must be observed in order to write a correct macro. The subsections at the end of this section describe in detail each program function which can be called from within a macro.

This function uses two requesters. The first requester is a macro requester similar to the one used by the "Load Macro" function (section

6.1

). The second requester displays the content of the macro selected using the first requester.

The macro requester is used to choose the macro to be analyzed. Once a macro is selected with the mouse, its name is displayed in the string gadget below the list box, where it can be modified. The <Return> key must be pressed to confirm a newly entered name. Once <Return> is pressed, the new name is associated with the macro and displayed in the list box. If <Return> is not pressed, but another macro (or the same macro) is selected, its name is again displayed in the string gadget.

Once a macro has been selected with the mouse, it can be examined by

pressing the "Proceed" gadget at the bottom of the requester, or double-clicking on the macro name in the list box. If the "Cancel" gadget is selected instead, the second requester is not displayed, and the "Examine Macro" operation terminates.

The second requester displays the text which makes up the selected macro. A list box, similar to the ones used in the file requester and the macro requester, contains the lines of text. If the macro is too long to fit in the list box, only part of it is displayed, but the text can be scrolled up or down using the knob at the right of the list box. This works in exactly the same way as the list box described in section 3.23 ("The File Requester").

Three gadgets appear at the bottom of the second requester: "Proceed", "Codes" and "Number". When "Proceed" is selected, the requester containing the macro listing is removed, and the macro requester is displayed again. A new macro can be examined by selecting the "Proceed" gadget of the macro requester, as explained above.

When the "Codes" gadget is selected, the list box displays the more compact four-character names of the macro functions. The gadget remains highlighted until it is selected again with the mouse. When the gadget is not highlighted, a more extended (and understandable) name is displayed for each function in the list box. The subsections at the end of this section contain the compact codes of all macro functions. For example, the code for the function "Brush definition mode OFF" is "MKB0" (section

6.6.37
).

When the "Number" gadget is selected, each line displayed in the list box is preceded by a line number, indicating the position of the line in the macro. The first line is line number one. The gadget remains highlighted (and the "Number" mode active) until it is selected again with the mouse.

The extended names of the functions, which are usually similar to the names of the menus associated with the functions, are displayed by default in the list box to make the macro more readable to the user. Four-character macro codes, which are the same codes used to read and write macros stored in a file, and can be processed very efficiently, can be displayed by selecting the "Codes" gadget.

While the extended names displayed in the list box are different from language to language, the four-character codes are language independent. The codes are the same for all languages in which the Personal Fonts Maker is translated. A macro file written - for example - on the German version of the Personal Fonts Maker can be executed on the Italian version, and vice versa.

The following subsections describe the meaning of all the four-character codes. Users wishing to write or modify macro files should adhere to the format described here.

Every line of a macro program contains the code of one or more functions to be executed, each followed by one or more parameters, if required. Macros are processed from the top to the bottom. The first

instruction is the first to be executed, while the last function to be executed is contained in the last program line.

Files containing macros stored by the Personal Fonts Maker follow the same format displayed by the "Examine" requester to display macros in the "Codes" mode. The prefix "PFM MCRO" is added at the beginning of the macro file, immediately before the macro text, to make a file containing a PFM macro immediately recognizable.

A macro file written by the user can contain additional comments and formatting characters, like spaces and empty lines, to make the macro more readable. This information is skipped by the program when the macro is loaded. More than one macro command can appear on the same line. Consecutive commands must be separated by at least one space (ASCII decimal code 32), TAB (ASCII code 9) or Line-Feed (LF, ASCII code 10) character. Comments must begin with a ';' (semicolon, ASCII code 59) and terminate at the end of the line. Spaces and empty lines in excess, as well as comments, are skipped by the program. Macro commands written in lower case letters are automatically converted to the proper format (capital letters). In general, these are the same rules which apply for the parameter files, described in section 2.6. When a macro is saved by the the Personal Fonts Maker, it always follows the rigid internal spacing rules of the program, even if the macro was originally read from a file containing the commands in a more "relaxed" format.

For example, the following macro

```
PFM MCRO ; SetSpcKrn.mcr

BRST 0

; dummy brush
MKB1
LBTD 0 0
Move M 0
LBTU M 0
CrON
; use brush X size to set space
Spce - SPC+
Kern 1 DOTM
```

will be correctly loaded and executed by the program, but will be saved by the Personal Fonts Maker in the following default format:

```
PFM MCRO

BRST 0
MKB1
LBTD 0 0
MOVE M 0
LBTU M 0
CRON
SPCE -
SPC+
KERN 1
DOTM
```

Most syntax errors (e.g. a non-existing macro code) in a macro file will be detected and signalled when the file is loaded. Other errors (e.g. "Brush mode not possible"), encountered during the execution of the macro will cause an error message to be displayed. Appendix G describes the various possible error messages.

Apart from the general syntax rules mentioned above, there are some additional limitations to macros written by the user. For example, a "Left Mouse Button UP" command cannot appear unless a previous command has specified that the mouse should be pressed. This cannot happen if the macro is automatically recorded by the Personal Fonts Maker, but it is not impossible for a user writing a macro to forget such a line.

Both the following subsection and appendix L ("PFM Macros Command Consistency Table") explain how some commands must be preceded or followed by other commands. For example, an LBTD ("Left mouse BuTton Down") command must be followed immediately by one or more optional MOVE instructions and a LBTU ("Left mouse BuTton Up"). SETR ("SEt Reference points") must be followed by a LBTU; only one or more FnKD ("Function Key Down") may appear between SETR and LBTU, but an FnKU ("Function Key Up") must come immediately after a FnKD instruction relative to the same key.

The macro consistency table, in appendix L, lists in a compact format all the conditions which must occur, or are prohibited, when a particular macro command is executed. In general, unless different instructions are given in appendix L and in the following subsections, the mouse buttons must be in the "Up" position, the reference point mode must not be active and the four function keys <F1> to <F4> must not be pressed. Most macro commands can only be executed if all these conditions are met. Exceptions are documented in the sections describing the particular commands.

Some macro commands must be followed by one or two parameters. In general, the parameter is a number. For example, the XSIZ command must be followed by the "X Size" value. The commands used to set parameters associated with the string gadgets in the main screen which accept a '-' (minus) or '+' (plus) sign instead of the number ("X Size", "Space" and "Kerning", sections 3.4 to 3.6), can also be followed by the same '-' or '+' sign.

Unlike parameter files, where variables must be followed by a numerical value introduced by an equal sign, the value(s) must be written immediately after a macro command, separated only by one or more space (or TAB) characters. No equal signs need to be written, as the value is not assigned to a variable, but is the parameter of a command.

The commands which need the position of the pointer in the character editing box to work must be followed by two parameters, indicating respectively the column (X position) and row (Y position) from the top left of the character editing box. For example, "RBTD 0 0" means that the right mouse button must be pressed on the top left corner of the box. A space (or TAB) character must separate the two values.

Normally the commands which require a position in the character editing box are followed by two numbers which indicate an absolute position from the top left corner of the box. One or both values may be written in a different format. A value immediately preceded by a '+' (plus) or '-' (minus) sign is interpreted as being relative to the previous position in

the box. The character 'M' is interpreted as the maximum possible value, i.e. either the rightmost or the lowest possible position. The '?' (question mark) character indicates that value will be determined by the user when the macro is executed. When a '?' is encountered during the execution, the Personal Fonts Maker displays a special pointer with a question mark, asking the user to select a position in the box, as indicated in section

```
6.7
  ("Execute Macro").
```

For example,

```
MOVE 3 1
```

will move (the pencil/brush) to the fourth dot of the second row, while

```
MOVE +2 +4
```

will move two dots to the right, four rows under the previous position, and

```
MOVE ? M
```

will move at the bottom of the column specified by the user with the mouse.

As described in section

```
6.3
  ("Start Record"), the only functions which
```

let non-absolute coordinates be recorded using the Personal Fonts Maker's automatic macro recording function are the definition of the brush and the setting of a reference point. The other functions using non-absolute coordinates are reserved for the more experienced users wishing to write complex macros using a word processor, a text editor or another program capable of generating a macro file.

The following subsections explain the usage of the macro commands. All commands use functions already described in one or more other sections of this guide. Here, some aspects of those functions particularly important when a macro is accessing the function are explained in more depth. The sections are ordered alphabetically after the four-character macro codes.

6.6.1 Macro Commands: AFNT

```
Command Name      : AFNT (Select AlternateFoNT)
Reference         : The "Font" Gadgets
Parameters        : -
More in Sections : 2.6, 3.2
```

This command causes the font environment other than the current one to be used. This is equivalent to selecting the "1/2" symbol gadget (section 3.2).

This function is useful when data has to be copied or moved from one font to a different one.

6.6.2 Macro Commands: BHCE

Command Name : BHCE (Brush Handle CEntre)
 Reference : Brush Handle
 Parameters : -
 More in Sections: 2.6.3, 5.9.3

This command places the brush handle at the centre of the brush.

The command is automatically placed at the beginning of a macro if the handle is in the central position when the macro record mode begins. This ensures that the macro is executed in the same conditions which existed when it was recorded (section 6.3).

6.6.3 Macro Commands: BHFL

Command Name : BHFL (Brush Horizontal FLip)
 Reference : Horizontal Flip
 Parameters : -
 More in Sections: 2.6.3, 5.3

This command causes the "Horizontal Flip" function to be executed.

6.6.4 Macro Commands: BHLL

Command Name : BHLL (Brush Handle Lower Left)
 Reference : Brush Handle
 Parameters : -
 More in Sections: 2.6.3, 5.9.4,
 6.6.2

The BHLL command places the brush handle at the lower left corner of the current brush. ↔

Like BHCE, this command is automatically placed at the beginning of a macro if it corresponds to the initial brush handle position.

6.6.5 Macro Commands: BHLR

Command Name : BHLR (Brush Handle Lower Right)
 Reference : Brush Handle
 Parameters : -
 More in Sections: 2.6.3, 5.9.5,
 6.6.2

This command places the brush handle at the lower right of the brush. ↔

Like BHCE, this is one of the commands which can be automatically placed by the program at the beginning of a macro.

6.6.6 Macro Commands: BHUL

Command Name : BHUL (Brush Handle Upper Left)
 Reference : Brush Handle
 Parameters : -
 More in Sections: 2.6.3, 5.9.1,
 6.6.2

The BHUL command selects the "Upper Left" brush handle option. ←
 Section

5.9.1 describes some features which make this brush handle position particularly interesting for certain macro operations.

The BHUL command is placed automatically at the beginning of a macro recorded with the Personal Fonts Maker if the initial brush handle position is on the top left corner.

6.6.7 Macro Commands: BHUR

Command Name : BHUR (Brush Handle Upper Right)
 Reference : Brush Handle
 Parameters : -
 More in Sections: 2.6.3, 5.9.2,
 6.6.2

This command places the brush handle at the upper right corner ←
 of the

brush. Like BHCE and the other brush handle positioning commands, this command can be placed automatically at the beginning of a macro by the Personal Fonts Maker.

6.6.8 Macro Commands: BITA

Command Name : BITA (Brush ITALicize)
 Reference : Italicize
 Parameters : -
 More in Sections: 2.6.13, 5.6, 7.4

This command causes the current brush to be italicized. Section 5.6 ("Italicize") explains this function in detail.

6.6.9 Macro Commands: BRDH

Command Name : BRDH (BRush Double Horizontal)
 Reference : Double Horizontal
 Parameters : -
 More in Sections: 5.8.6

The BRDH command causes the "Double Horizontal" brush function to be executed.

6.6.10 Macro Commands: BRDO

Command Name : BRDO (BRush Double)
 Reference : Double

Parameters : -
More in Sections: 5.8.5

This command doubles the brush, both vertically and horizontally, as described in section 5.8.5.

6.6.11 Macro Commands: BRDV

Command Name : BRDV (BRush Double Vertical)
Reference : Double Vertical
Parameters : -
More in Sections: 5.8.7

BRDV executes the "Double Vertical" brush function, described in section 5.8.7.

6.6.12 Macro Commands: BRHA

Command Name : BRHA (BRush HALve)
Reference : Halve
Parameters : -
More in Sections: 5.8.2

The BRHA command halves the brush. Section 5.8.2 ("Halve") has more on this function.

6.6.13 Macro Commands: BRHH

Command Name : BRHH (BRush Halve Horizontal)
Reference : Halve Horizontal
Parameters : -
More in Sections: 5.8.3

This command halves the brush horizontally.

6.6.14 Macro Commands: BRHV

Command Name : BRHV (BRush Halve Vertical)
Reference : Halve Vertical
Parameters : -
More in Sections: 5.8.4

As described in section 5.8.4 ("Halve Vertical"), this command causes the height of the brush to be halved.

6.6.15 Macro Commands: BROT

Command Name : BROT (Brush ROTate)
Reference : Rotate 90 Degrees
Parameters : -
More in Sections: 5.5

The BROT command rotates the brush clockwise by 90 degrees.

6.6.16 Macro Commands: BRST

Command Name : BRST (BRush STretch)
Reference : Stretch
Parameters : OnOff (1 or 0)
More in Sections: 5.8.1

This command begins or cancels a brush stretch operation, as explained in section 5.8.1. The BRST command must be followed either by a '1' (one) or by a 0 (zero). "BRST 1" is equivalent to selecting the "Stretch" function which activates the stretch mode. "BRST 0" cancels the stretch mode before the brush is stretched.

The "BRST 1" command must be followed either by a "BRST 0", or by a LBTD ("Left BuTton Down"), one or more optional MOVE commands, and a terminating LBTU ("Left BuTton Up"). The MOVE commands are not considered by the program to determine the final brush size, which is calculated entirely from the coordinates following the LBTD and LBTU commands.

This command is automatically placed at the beginning of a macro recorded with the Personal Fonts Maker to ensure the macro is executed in the same conditions under which it was originally recorded.

6.6.17 Macro Commands: BRVS

Command Name : BRVS (Brush ReVerSe)
Reference : Reverse
Parameters : -
More in Sections: 5.7

This command reverses the brush, executing the "Reverse" brush function, described in section 5.7.

6.6.18 Macro Commands: BSHM

Command Name : BSHM (BruSH Mode)
Reference : The "Pencil/Brush" Gadget
Parameters : -
More in Sections: 3.1, 3.17, 3.18

This command sets the "Brush" mode, as described in section 3.17.

When a macro is recorded by the program, a BSHM command is placed at the beginning of the macro if the recording starts when the "Brush" mode is active. This guarantees that the macro is executed under exactly the same conditions which existed when the macro was recorded.

6.6.19 Macro Commands: BVFL

Command Name : BVFL (Brush Vertical FLip)
Reference : Vertical Flip

Parameters : -
More in Sections: 5.4

This command flips the brush vertically, as described in section 5.4.

6.6.20 Macro Commands: CLRC

Command Name : CLRC (CLear Character)
Reference : The "CLR" Gadget
Parameters : -
More in Sections: 3.8

This command clears the character displayed in the character editing box, as if the "CLR" gadget had been selected by the user.

6.6.21 Macro Commands: COFF

Command Name : COFF (Character OFF)
Reference : The "ON/OFF" Gadget
Parameters : -
More in Sections: 2.7.2.2, 3.10

This command switches the current character to the "Off" position, unless it is already "Off".

6.6.22 Macro Commands: CRON

Command Name : CRON (ChaRacter ON)
Reference : The "ON/OFF" Gadget
Parameters : -
More in Sections: 2.7.2.2, 3.10

The character displayed in the character editing box (the current character) is switched "On" by the CRON command.

6.6.23 Macro Commands: DOTM

Command Name : DOTM (DOT Mode)
Reference : The "Pencil/Brush" Gadget
Parameters : -
More in Sections: 3.1, 3.17, 3.18

This command activates the "Pencil" mode, whereas a single dot is used as the unit for paint operations, as described in section 3.17.

This command is automatically placed at the beginning of a macro recorded with the program if the recording starts in the "Pencil" mode.

6.6.24 Macro Commands: DVW0

Command Name : DVW0 (Dynamic VieW off)

Reference : The "Dynamic View" Gadget
 Parameters : -
 More in Sections: 3.15

This command terminates the dynamic view mode and closes the associated screen.

6.6.25 Macro Commands: DVW1

Command Name : DVW1 (Dynamic View on)
 Reference : The "Dynamic View" Gadget
 Parameters : -
 More in Sections: 3.15

This command activates the dynamic view mode. It opens a new dynamic view screen, or brings an existing screen to the front.

6.6.26 Macro Commands: F1KD to F4KD

Command Name : F1KD to F4KD (Function Key Down)
 Reference : Reference Points
 Parameters : YPos (vertical position)
 More in Sections: 1.10.16, 2.7.2.14, 3.21

This set of four commands, each associated with a function key from <F1> to <F4>, is used when the reference point mode is active (section 3.21). When the program encounters such a command, it behaves as if the associated function key was pressed down.

A numerical parameter indicating the position (from the top, starting from 0) at which the reference point is to be moved must follow the command. No MOVE instructions can follow a FnKD command. While the program is recording a macro and the reference point mode is active, FnKD are recorded only after the function key has already been released and the position of the reference point is certain. In fact, FnKD and the associated FnKU are always recorded at the same time, even if the user presses a function key, moves the pointer, presses a second function key, moves the pointer again, releases the first key, and finally releases the second function key after having dragged the reference point up and down several times. Recording an FnKD only when the position of the reference point is determined makes the macro more readable and its execution faster.

A '?' (question mark) can be written after the command to indicate that the position of the reference point is to be selected by the user when the macro is executed. Section

6.7

("Execute Macro") explains how the position is to be selected. If the 'M' character is used as a parameter instead, it is interpreted as the maximum possible value, which corresponds to the lowest possible row-position.

The commands F1KD to F4KD can only appear after a SETR command. Each FnKD must immediately be followed by an associated FnKU ("Function Key Up") command. A LBTU (Left mouse ButTon Up) command must terminate the

sequence.

6.6.27 Macro Commands: F1KU to F4KU

Command Name : F1KU to F4KU (Function Key Up)
Reference : Reference Points
Parameters : -
More in Sections: 1.10.16, 2.7.2.14, 3.21

These commands are used to indicate that the function keys <F1> to <F4> have been released. An FnKU can only appear immediately after a FnKD command (section 6.6.26).

6.6.28 Macro Commands: FLPH

Command Name : FLPH (FLiP Horizontal)
Reference : The Two "FLIP" Gadgets
Parameters : -
More in Sections: 3.20

FLPH flips the image in the character editing box (not the brush) horizontally, as described in section 3.20.

6.6.29 Macro Commands: FLPV

Command Name : FLPV (FLiP Vertical)
Reference : The Two "FLIP" Gadgets
Parameters : -
More in Sections: 3.20

This function flips the content of the character editing box vertically. Section 3.20 describes the operation in detail.

6.6.30 Macro Commands: FONT

Command Name : FONT (FONT select)
Reference : The "Font" Gadgets
Parameters : Font
More in Sections: 2.6, 3.2

This command, followed by a numerical parameter (1 or 2) indicating the desired font, switches to the selected font environment, as if that font had been selected with the "Font" string gadget, described in section 3.2.

6.6.31 Macro Commands: KERN

Command Name : KERN (set KERNING)
Reference : The "Kerning" Gadgets
Parameters : Kern

More in Sections: 2.7.2.10, 3.6

The KERN command sets the "Kerning" parameter for the current character, as if it had been set through the "Kerning" string gadget. The desired value must follow the KERN keyword. As described in section 3.6, a '-' (minus) or '+' (plus) sign can be written instead of a number.

6.6.32 Macro Commands: KRN-

Command Name : KRN- (decrease KeRNING)
 Reference : The "Kerning" Gadgets
 Parameters : -
 More in Sections: 2.7.2.10, 3.6

This command decreases the "Kerning" value of the current character by one unit. This is equivalent to selecting the left arrow gadget at the side of the "Kerning" string gadget.

6.6.33 Macro Commands: KRN+

Command Name : KRN+ (increase KeRNING)
 Reference : The "Kerning" Gadgets
 Parameters : -
 More in Sections: 2.7.2.10, 3.6

KRN+ increases the "Kerning" value of the current character by one unit. This is the opposite of KRN-.

6.6.34 Macro Commands: LBTD

Command Name : LBTD (Left BuTton Down)
 Reference : The Character Editing Box, The "Define Brush" Gadget
 Parameters : X Pos, Y Pos
 More in Sections: 3.1, 3.16, 3.17, 3.18

When this command is encountered during the execution of a macro, the program responds as if the left mouse button was pressed (and held down). The command requires two parameters: the horizontal and vertical position of the dot in the character editing box over which the button is pressed. As described in section

6.6

, special formats can be used to indicate relative, maximum or user-defined positions.

LBTD can only be followed by one or more MOVE instructions and a LBTU ("Left BuTton Up") command.

6.6.35 Macro Commands: LBTU

Command Name : LBTU (Left BuTton Up)
 Reference : The Character Editing Box, The "Define Brush" Gadget
 Parameters : X Pos, Y Pos

More in Sections: 3.1, 3.16, 3.17, 3.18

This command terminates a sequence beginning with a SETR ("SET Reference points") or LBTU ("Left BuTton Down"). LBTU, which is the opposite of LBTU, must be followed by two values indicating the position in the box over which the button is released.

The two parameters are ignored if the command is used to terminate the reference point mode. Conversely, the Personal Fonts Maker sets both values to zero when the macro is recorded.

6.6.36 Macro Commands: MEMR

Command Name : MEMR (MEMoRize character)
Reference : The "Copy to Buffer" Gadget
Parameters : -
More in Sections: 3.11, 3.12

This command causes the character displayed in the character editing box to be copied to the character buffer, as described in section 3.11.

6.6.37 Macro Commands: MKB0

Command Name : MKB0 (MaKe-Brush mode off)
Reference : The "Define Brush" Gadget
Parameters : -
More in Sections: 3.1, 3.17, 3.18

This command terminates the brush definition mode, as described in section 3.18. This aborts the brush definition request.

6.6.38 Macro Commands: MKB1

Command Name : MKB1 (MaKe-Brush mode on)
Reference : The "Define Brush" Gadget
Parameters : -
More in Sections: 3.1, 3.17, 3.18

MKB1 activates the brush definition mode (section 3.18). The command must either be followed by MKB0, which aborts the operation, or by a LBTU (or RBTU), one or more MOVE commands and a terminating LBTU (or RBTU). No other commands are allowed between the LBTU (or RBTU) and the LBTU (or RBTU). MKB1 only affects the way a subsequent LBTU (or RBTU) is processed. Any other commands may appear between MKB1 and LBTU (or RBTU or MKB0).

6.6.39 Macro Commands: MOVE

Command Name : MOVE (MOVE cursor)
Reference : The Character Editing Box, The "Define Brush" Gadget
Parameters : X Pos, Y Pos
More in Sections: 3.1, 3.16, 3.17, 3.18

This command emulates a movement of the mouse pointer to the specified

position. Two values must follow the MOVE keyword, to indicate the desired position in the character editing box. As described in section 6.6

different formats of the values make it possible to specify absolute, relative, maximum or user-defined coordinates.

If the coordinates are displayed on the screen title bar (section 7.5), these are updated after every MOVE instruction.

Either the left or the right mouse button can be pressed while MOVE is executed.

6.6.40 Macro Commands: ORM0

Command Name : ORM0 (OR Mode off)
 Reference : The "OR" Gadget
 Parameters : -
 More in Sections: 3.16

If the "OR" mode is active when ORM0 is executed, the "OR" mode terminates.

If the program started recording a macro when the "OR" mode was not active, an ORM0 command is automatically placed at the beginning of the macro.

6.6.41 Macro Commands: ORM1

Command Name : ORM1 (OR Mode on)
 Reference : The "OR" Gadget
 Parameters : -
 More in Sections: 3.16

This command activates the "OR" mode, as described in section 3.16.

This is one of the commands which can be automatically placed at the beginning of a macro recorded with the Personal Fonts Maker to ensure that the macro is executed in the same conditions in which it was recorded.

6.6.42 Macro Commands: RBTd

Command Name : RBTd (Right BuTton Down)
 Reference : The Character Editing Box, The "Define Brush" Gadget
 Parameters : X Pos, Y Pos
 More in Sections: 3.1, 3.16, 3.17, 3.18

This command is equivalent to pressing the right mouse button in the character editing box. This can produce different editing effects, depending on whether the brush or the pencil is used to paint. This command must be followed by the coordinates of the dot position in the character editing box, in the format described in section 6.6

6.6.43 Macro Commands: RBTU

Command Name : RBTU (Right BuTton Up)
Reference : The Character Editing Box, The "Define Brush" Gadget
Parameters : X Pos, Y Pos
More in Sections: 3.1, 3.16, 3.17, 3.18

When this command is executed, the Personal Fonts Maker behaves as if the right mouse button was released. The right mouse button can only be used to edit (i.e. paint or clear) data in the character editing box or to define the brush.

This command must follow RBTU ("Right BuTton Down"). One or more MOVE commands can appear between RBTU and RBTU.

6.6.44 Macro Commands: RCAL

Command Name : RCAL (ReCALl character)
Reference : The "Paste From Buffer" Gadget
Parameters : -
More in Sections: 3.11, 3.12

This command is equivalent to selecting the "Paste From Buffer" gadget, described in section 3.12.

6.6.45 Macro Commands: SELC

Command Name : SELC (SElect Character)
Reference : The "Character #" Gadgets, Quick Character Selection
Parameters : Chr
More in Sections: 2.7.2.4, 3.3, 3.22

This command, followed by the code of the character to be selected, makes the specified character the current character, and displays it in the character editing box.

This command can be recorded either by writing a value in the "Character #" string gadget, or selecting a character with the quick character selection table.

6.6.46 Macro Commands: SEL-

Command Name : SEL- (SElect previous character)
Reference : The "Character #" Gadgets
Parameters : -
More in Sections: 3.3

This command is equivalent to the left arrow gadget of the "Character #" parameter. The character which comes before the one currently displayed becomes the current character (if such a character, either "On" or "Off", exists).

6.6.47 Macro Commands: SEL+

Command Name : SEL+ (SElect next character)
 Reference : The "Character #" Gadgets
 Parameters : -
 More in Sections: 3.3

This command is equivalent to the right arrow gadget of the "Character #" parameter. This is the reverse operation of SEL-.

6.6.48 Macro Commands: SETR

Command Name : SETR (SET Reference points)
 Reference : Reference Points
 Parameters : -
 More in Sections: 2.7.2.14, 3.21

This command activates the reference point mode. The reference point mode must be terminated by a LBTU ("Left BuTton Up") command.

Only one or more FnKD ("Function Key Down", section 6.6.26) may appear between SETR and LBTU, and only if an FnKU ("Function Key Up") comes immediately after the FnKD instruction relative to the same key.

6.6.49 Macro Commands: SHFD

Command Name : SHFD (SHiFt character Down)
 Reference : The Four "Shift" Gadgets
 Parameters : -
 More in Sections: 3.16, 3.19

Like all shift macro commands, SHFD is equivalent to selecting one of the four "Shift" gadgets. This command shifts the image of the current character down by one position.

6.6.50 Macro Commands: SHFL

Command Name : SHFL (SHiFt character Left)
 Reference : The Four "Shift" Gadgets
 Parameters : -
 More in Sections: 3.16, 3.19

This command moves the character image in the editing box to the left.

6.6.51 Macro Commands: SHFR

Command Name : SHFR (SHiFt character Right)
 Reference : The Four "Shift" Gadgets
 Parameters : -
 More in Sections: 3.16, 3.19

SHFR shifts the character displayed in the character editing box to the right by one dot-position.

6.6.52 Macro Commands: SHFU

Command Name : SHFU (SHiFt character Up)
Reference : The Four "Shift" Gadgets
Parameters : -
More in Sections: 3.16, 3.19

SHFU is equivalent to selecting the "Shift Up" gadget.

6.6.53 Macro Commands: SPCE

Command Name : SPCE (set character SPaCe)
Reference : The "Space" Gadgets
Parameters : Spc
More in Sections: 2.7.2.16, 3.5

This command sets the "Space" parameter of the current character. SPCE must be followed by a numerical value or a '-' (minus) or '+' (plus) sign.

6.6.54 Macro Commands: SPC-

Command Name : SPC- (decrease character SPaCe)
Reference : The "Space" Gadgets
Parameters : -
More in Sections: 2.7.2.16, 3.5

This command is equivalent to selecting the left "Space" arrow gadget, which decreases the value of the current character's "Space" parameter by one.

6.6.55 Macro Commands: SPC+

Command Name : SPC+ (increase character SPaCe)
Reference : The "Space" Gadgets
Parameters : -
More in Sections: 2.7.2.16, 3.5

SPC+ increases the "Space" of the current character by one unit. This command is the opposite of SPC-.

6.6.56 Macro Commands: UNDO

Command Name : UNDO (UNDO last operation)
Reference : The "Undo" Gadget
Parameters : -
More in Sections: 3.4, 3.8, 3.9, 3.12, 3.17, 3.19, 3.20

This command is equivalent to selecting the "UNDO" gadget, described in section 3.9.

6.6.57 Macro Commands: XSIZ

Command Name : XSIZ (set character X SIZE)
Reference : The "X Size" Gadgets
Parameters : Xsiz
More in Sections: 2.6.24, 2.7.2.21, 2.7.2.22, 3.4, 7.3.1

This command is used to set the "X Size" parameter of the current character. The command must be followed by the same value (a number or a '-' or a '+' sign) which would be written in the "X Size" string gadget.

6.6.58 Macro Commands: XSZ-

Command Name : XSZ- (decrease character X SiZe)
Reference : The "X Size" Gadgets
Parameters : -
More in Sections: 2.7.2.22, 3.4, 7.3.1

This command decreases the value displayed in the "X Size" string gadget by one (if possible).

6.6.59 Macro Commands: XSZ+

Command Name : XSZ+ (increase character X SiZe)
Reference : The "X Size" Gadgets
Parameters : -
More in Sections: 2.7.2.22, 3.4, 7.3.1

This command (opposite of XSZ-) increases the value of the current character's "X Size" by one.

6.6.60 Macro Commands: ZOM-

Command Name : ZOM- (decrease ZOoM)
Reference : The "Smaller" Gadget
Parameters : -
More in Sections: 3.1, 3.13, 3.14

This command is equivalent to selecting the "Smaller" gadget, described in section 3.13.

6.6.61 Macro Commands: ZOM+

Command Name : ZOM+ (increase ZOoM)
Reference : The "Larger" Gadget
Parameters : -
More in Sections: 3.1, 3.13, 3.14

This command is equivalent to the selection of the "Larger" gadget,

described in section 3.14.

6.7 Execute Macro

This function starts the execution of a macro. The macro must already be present in the program's memory, after having been either recorded or loaded. The macro is executed only once.

The macro requester (section 6.1) is used to select the macro to be executed. The macro starts immediately after the "Proceed" gadget of the requester is selected or the macro name in the list box is double-clicked.

The mouse pointer image changes to an 'M' symbol to indicate that the macro is running. If the step mode is active (section 6.10), the pointer image will blink while the program is waiting for the user to press <Space> or <Esc>.

As described in section 6.6 ("Examine Macro"), it is possible to write a macro containing commands which require the user to select a position in the character editing box.

The execution can be aborted by the user selecting a new function (which will be executed) through the menus, the gadgets or the keyboard. The "Stop Execution" command, described in section 6.9, interrupts the execution without side effects. Some keys have a particular function if the step mode is active, as described in section 6.10.

The character displayed in the editing box when the macro execution command was given remains the current character of the macro, unless the macro contains one or more commands which select a different character or the alternate font. Unless specified otherwise by the macro (through a SELC, SEL-, SEL+, AFNT or FONT command) all editing commands contained in the macro modify the character displayed in the character editing box when the macro execution began. If another character (or font) is selected by the macro, or if any of the displayed values or the character image are modified, the character editing box and the parameters displayed on the screen are modified accordingly, so that the user can follow the execution. The coordinates on the title bar (section 7.5) are also continuously updated.

Some macro commands are followed by coordinates indicating a position in the character editing box. It may sometimes happen that, when a macro was recorded, the character editing box was larger (i.e. it had more columns and/or rows) than the box containing the current character. It is therefore possible that some coordinates exceed the current limits,

falling out of the character editing box. In such a case, the program behaves exactly as if the function was called manually by the user, instead of from a macro. This means that in some cases values which are too high or too low will be replaced by the closest valid value, or with a default value. In other cases, a message will be displayed. If the program does not work as expected it can be very useful to activate the step mode, as described in section

6.10

.

As already explained in section

6.6

, a coordinate can be left

undefined in the macro by writing a '?' (question mark) instead of a numerical value after the command. When such a command is encountered during the execution of a macro, the mouse pointer image turns into an arrow and a question mark. This indicates that the user must select a position in the character editing box (or in the reference point zone) and press the left mouse button when the arrow is over the desired position (e.g. a dot in the character editing box). If only one of a pair of coordinates is to be defined at run time by the user, only that value will be affected by the position specified by the user, while the other value will remain as specified after the macro command. If both coordinates (i.e. the vertical and horizontal position) of the same command are to be defined, the program will stop only once to ask the user to specify a position. In this case, both the vertical and the horizontal position specified by the user will be considered. As most other functions of the Personal Fonts Maker, this is more difficult to describe than to actually use.

A macro can be selected and executed pressing the key associated with that macro while the <Ctrl> key is held down. In that case, no requester is displayed and the execution of the macro begins immediately.

6.8 Controlled Execution

This function is similar to "Execute Macro", described in section

6.7

.

In addition to that function, the "Controlled Execution" allows the user to specify a range of characters on which the specified macro is repeated. For each character included in the range, the macro is repeated and acts as if that character was the current character.

The macro requester contains some special gadgets in addition to the requester described in section

6.1

. The first line under the list box

indicates which macro has been selected. Below (in green), the status of the characters on which the macro is to be repeated. This can be one of three texts: "Every Character", "ON Characters" or "OFF Characters". The status can be changed by clicking on the text with the mouse. This is possible because the status line, i.e. the line printed in green rather than in black, is actually a gadget. The third additional line in this "extended" macro requester contains two string gadgets, used to set the first and the last character on which the macro must be repeated.

For example, a macro transforming the current character into an italic character could be repeated on a group of characters in the font (if not the entire font), italicizing more than one character. The "Controlled Execution" function could be used to specify the codes of the first and the last character to be italicized. Only "On" characters would need to be processed. Supposing that the macro associated to the key <A> had to be repeated for every "On" character in the font, the lines under the list box would read: "Repeat macro A on ON characters from # 0 to # 256".

This function has all the features of "Execute Macro", described in section

6.7

. In addition, "Controlled Execution" can execute a macro more than once. Every character in the current font whose code is in the range specified by the user is examined. If the status of that character ("On" or "Off") meets the requirements set by the user ("Every Character", "ON Characters" or "OFF Characters"), it is made the current character and displayed in the character editing box. The macro is then executed exactly as if the "Execute Macro" function was used.

If a macro changes font environment during its execution, and terminates without restoring the initial font environment, the Personal Fonts Maker automatically goes back to the original font environment, so that when the macro is executed again on the following character, the same initial conditions will be met. The order of the characters on which the macro is executed does not change if the macro makes another character the current character.

If the "Joined Fonts" mode is selected, as described in section 7.9, both font environments will have a current character with the same code. This means that when this function selects the successive character as the current character, the same will be done for the other font environment. This also happens if a macro being executed contains commands which select new characters or "jump" from one font environment to the other.

After the macro has been repeated on all the selected characters, the function terminates, restoring the current character which was set before the start of the macro execution.

The execution of the macro can be "stepped" or interrupted as if it was started with the "Execute Macro" function (section

6.7

).

6.9 Stop Execution

This command stops the execution of a macro. If the macro was started from the "Controlled Execution" function, the macro is stopped and the repetition of the macro on other characters is aborted.

6.10 Step Execution

The macro step mode can be activated by selecting the "Yes" subitem of the "Step Execution" menu item with the mouse. The mode is terminated by

selecting "No".

This option does not start a macro. Instead, it affects the behaviour of the macros which are started with the "Execute Macro" or "Controlled Execution" functions after the selection (or termination) of the step mode.

When a macro is executed in the step mode, each command in the macro is displayed on the title bar on the top of the screen before it is executed. The Personal Fonts Maker waits until the user presses <Space> to process that command. While the program is "waiting", the mouse pointer blinks to remind the user that the step mode is activated.

The information on the title bar is displayed in the same format which was last selected in the "Examine Macro" function (section 6.6) using the "Codes" and "Number" gadgets.

The macro execution can be aborted as usual, for example by pressing <Esc> instead of the space bar.