

Fonts

A "font" refers to the physical shape of a set of printed characters. This Chapter explains how to exploit the readymade AMOS Professional fonts, how to import new ones, and how to design your own fonts using the ready-made Font Editor program.

Text Fonts

Tile sets of fonts used by commands like PRINT, are known as "text fonts", and each AMOS Professional window can have its own individual set, as required. Styles and special effects for text fonts are controlled by instructions such as TEXT STYLE and WRITING, which are fully explained in Chapter 5.6.

Graphic Text Fonts

Although text fonts are suitable for normal uses, an infinite variety of styles can be achieved by exploiting the much more flexible category of fonts known as "graphic text". Text fonts are positioned by referring to their location in terms of characters, but graphic text can be controlled much more accurately, because it is positioned using x,ycoordinates numbered in pixels. AMOS Professional supports the thousands of graphic fonts, available in commercial packages or in the public domain.

ROM Fonts

There are also alternative fonts built into the Amiga's ROM chips, and these are also available for use by the AMOS Professional system.

All fonts are referred to by an individual index number in a font list. No matter what type of font is to be used, you must first "get" it from wherever it is being stored, and then "set" it, ready for use. Users who are familiar with the AMOS system will find that the AMOS Professional system for handling fonts has been streamlined and improved.

When a SET FONT command is given, the system will do an automatic search to see if the required font is already in memory, and if all is well, the specified font is immediately made available for use. If the required font is not found in the current list of available fonts, it will be loaded, but the next time it is called by SET FONT there will be no need to load it again.

GET FONTS

instruction: create a list of available fonts from System disc

```
Get Fonts
```

The GET FONTS command creates an internal list of all available fonts on the System disc, and it should always be called at least once before any changes in settings are made. In practice, you will probably want to use this instruction at the beginning of a program, so that SET FONT may be used freely at any later point. It is very important to remember that if you are distributing run-only or compiled programs to be used by other people, and these programs make use of alternative fonts, then the required font files must be included.

```
E> Get Fonts
For F=0 To 10
Set Font F : T$="AMOS Professional Font: "+Str$(F)
Text 0,100,T$
Wait Key : C!s
Next F
```

GET DISC FONTS

instruction: create a list of available fonts from current disc

```
Get Disc Fonts
```

This instruction is exactly the same as the GET FONTS command, except that it triggers a search through the "Fonts" folder of your current disc only. If new fonts are to be used, then they must first be copied into this folder.

GET ROM FONTS

instruction: create a list of available ROM fonts

```
Get Rom Fonts
```

As you might expect, this command produces a list of the fonts that are built into the computer's ROM chips. At time of writing, the choice is rather limited:

```
Screen Open 0,640,200,16,Hires
Get Rom Fonts
For A=1 To 10
Set Font A : A$="Hello, I'm "+ Font$(A) : Text 0,100,A$
Wait Key : C!s
Next A
```

FONTS

function: return details of available fonts

```
report$=Font$(font number)
```

This function is used to examine an existing font list and make a report, giving details of the specified font number.

The report is given as a string of 38 characters, holding the following information: the name of the font, its physical

height in pixels and its status set to either Disc or Rom. For example:

```
Get Fonts : Set Font 2  
Print Font$(2)
```

SET FONT

instruction: select font for use by Text command

```
Set Font font number
```

This simple command is used to select the character set to be employed by a TEXT instruction, like this:

```
Get Fonts Set Font 2 : Text 100,100,"AMOS" : Set Font 1: Text  
100,120,"Professional"
```

TEXT

instruction: print graphical text

```
Text x,y,text$
```

This command is used to print text at the specified graphical coordinates. All coordinates are measured relative to

the "baseline" of the current character set, which can be found using the TEXT BASE function, explained next.

Normally, the baseline is the notional line on which all characters sit, and the "tails" of certain characters (like g,j,p,q

and y) drop below this baseline. The next example demonstrates how text can be placed at any pixel position on the screen:

```
Do  
Ink Rnd(15)+1,Rnd(15) : Text Rnd(320)+1,Rnd(198)+1,"AMOS Professional"  
Loop
```

TEXT BASE

function: return the text base of the current character set

```
baseline=Text base
```

This function is used to get the reference position of the current font's baseline, given as the number of pixels between the top of the character, and the point that it will be printed on the screen. It is similar to the hot-spot of an Object.

TEXT LENGTH

function: return the length of a section of graphical text

width=Text Length(text\$)

This function returns the number of pixels that make up the width of the characters in the current font, in a given string. This can vary for the same string, depending on the font in use. Furthermore, there are special fonts which assign different widths for each character in the same character set, known as "proportional" fonts. Here is a simple example:

```
TS="Centred Text"  
L=Text Length(T$)  
Text 160-L/2,100,T$
```

Wiping fonts from memory

As fonts are called, they build up in memory. Valuable memory is consumed, and it may be necessary to wipe fonts, using a line like this:

```
Trap Reserve As Data 10,1000000000
```

This forces AmigaDOS to clear out all unused memory, which will affect the fonts that have been stored. Obviously the huge amount of RAM that has been requested cannot possibly be reserved, even after the fonts have been cleared, and an "out of memory" error will be generated. A TRAP is included to cater for this event.

Assigning fonts

In the original AMOS system, you were obliged to go back to the Amiga Disc Operating System every time that the current font directory needed changing. With AMOS Professional, ill(' ASSIGN instruction solves this problem.

ASSIGN

instruction: assign a name to a file or device

```
Assign "Name:" To "New_Pathname"  
Assign "Name:" To "Device"
```

Supposing that you have an extensive library of fonts installed on a hard disc, as part of your development system, but you are writing programs for users who only have use of the internal floppy drive. You will need to test your programs with a reduced number of fonts, and employ the internal drive instead of your hard disc. This is easily achieved with the following line:

```
Assign "Fonts:" To "Df0:Fonts"
```

Now, every time that GET FONTS or GET DISC FONTS is called, the internal drive will be used instead of your hard disc.

Converting font coordinates

Obviously, with graphic fonts using coordinates measured in pixels, and text fonts positioned by character coordinates, a set of conversion functions between the two systems is vital.

XTEXT

YTEXT

functions: convert graphic coordinates to text coordinates

```
text x-coordinate=Xtext(graphic x-coordinate)
text y-coordinate=Ytext(graphic y-coordinate)
```

These self-explanatory functions convert coordinates from the standard graphical screen coordinates that use pixels, to text coordinates, that are given in character lines and column spacings. The resulting text coordinates are relative to the current window, and if the screen coordinate lies outside of this window, a negative value will be returned.

XGRAPHIC

YGRAPHIC

functions: convert text coordinates to graphic coordinates

```
graphic x-coordinate=Xtext(text x-coordinate)
graphic y-coordinate=Ytext(text y-coordinate)
```

This pair of functions performs the conversion of text format coordinates to graphic format coordinates, and can be used to position text over an area of graphics on the screen.

The AMOS Professional Text Font Editor

No matter how many fonts are available in your collection, there may be occasions when you need to create your own character sets, or edit existing fonts for a specific purpose or special effect. Nothing could be simpler!

The AMOS Professional Text Font Editor is available on your Accessories disc, allowing fast, precise and foolproof loading, editing and saving of text fonts.

```
Load "AMOSPro_Accessories:Font8x8_Editor.AMOS"
```

The Font Editor screen is divided into three practical areas. The left-hand side of the screen is used to display an entire character set. The top right-hand area of the screen displays an enlarged 8 x 8 grid, which holds the individual character to be edited. This is the working area. The bottom right-hand screen contains a simple panel of options.

[LOAD FONT]

Click on this option now, and a file requester will appear, asking you to load a font for editing.

This may be the

Default Font which is available in the APSystem folder, (AMOSPro.Default.Font).

After clicking on your choice of font, the individual characters in the set are displayed on the Font Editor working

screen, and any of these characters can be selected by the mouse, for your attention. As soon as a character is

selected, its enlarged image appears in the editing window at the top right-hand section of the screen.

[CLEAR][SET]

This pair of options are used to affect all of the pixels that make up the current character, by either clearing or

setting them. Alternatively, individual pixels can be cleared or set by clicking on them with any mouse button, to

toggle their current status.

[STORE]

The current character also appears in the [STORE] option panel. When you are satisfied with the edited appearance

of the current character, click on this panel to store it into the current font's character set, before moving to the next operation.

[SAVE FONT]

After your editing session is finished, the new font is saved to disc by clicking on this option, and calling up the

familiar file selector. After saving, you can continue the editing process, or call up a new font to be edited.

[QUIT]

Select this option to return to the AMOS Professional Edit screen.

You are warned not to destroy the default font, otherwise AMOS Professional will be unusable the next time it is initialised!