

Accessing a Printer

AMOS Professional offers total access to the Amiga's printer driver. The printer configuration is taken directly from your Preferences settings, allowing printer control with a standard set of "escape codes". Any AMOS Professional screen can also be dumped directly onto paper via your printer.

The printer device

Details of your printer are taken from the settings that have been previously entered from the Workbench Preferences utility. A printer can be connected to the Serial Port or the Parallel Port, and AMOS Professional will choose the appropriate device for all printing operations automatically. The first time that a printer is used, the printer drivers are loaded into memory from your start-up disc. If this is not available, a requester will be displayed enabling you to insert the relevant disc. The printer driver consumes a great deal of memory, and can require up to 50k in order to operate. If available memory is running short, it may be easier to access the printer with the SERIAL or PARALLEL commands instead. Additionally, if multi-tasking is being used, it is important to realise that only one program is allowed to access the printer device at any one time. If a previous program has already grabbed the printer, an error message will be generated when an attempt is made to access the printer from an AMOS Professional program. Such errors can be trapped using a command like this:

```
Trap Printer Open
If ERRTRAP : Print "Cannot open Printer Device!" : Endif
```

PRINTER OPEN

instruction: open the standard printer device for use

```
Printer Open
```

This command opens the printer device using your current preferences

PRINTER CLOSE

instruction: close printer port

```
Printer Close
```

Use this instruction to close the printer port that has been previously set with a PRINTER OPEN command. Note that the memory is not freed for use by your AMOS Professional program immediately. Memory is only returned if the following conditions are met: firstly, that no other program has requested the printer device

during multi-tasking,
as explained above. Secondly, if the system becomes short of memory and requires more.

PRINTER SEND

instruction: send a string to the printer

```
Printer Send text$
```

The PRINTER SEND command sends a string of text to the printer, using multi-tasking. The command does not wait for the text to be printed, and returns immediately to the program. All printing operations are performed "invisibly" in the background. Obviously, if the printer is not ready, the appropriate requester will appear.

Embedded commands

The text string to be printed can contain embedded commands, and these will be converted into the appropriate control sequences for the current printer automatically. This means that all the effects such as underline, bold, italic and subscript can be included in your programs. Most embedded commands begin with an "Escape" character, or Chr\$(27), and they will work equally well on any printer. Provided that your particular printer has been installed using the Preferences utility, the entire system will be completely transparent. Here are the rules of successful printing:

- Each line of text should be terminated by a single Line Feed. Normally this will be Chr\$(10), but an embedded command can also be used like this:

```
LF$=Chr$(27)+"E"  
Printer Send "Greetings" +LF$
```

- The LF +CR settings found in the AMOS_Interpreter configuration menus are ignored!
- Zeros are printed as normal characters.
- The PRINTER SEND command should not be used to output raw data. Such data may contain embedded commands that can be misinterpreted by the printer device.
- Because PRINTER SEND uses multi-tasking, there may be a slight delay while the text string is being printed.
- If any changes are made to a string while it is being output, the final print-out may become corrupted.
- To avoid unwanted "garbage collection", such an operation can be forced before transmission begins of data to be printed, with a line such as X =Free.

Here is a list of the most useful embedded commands (Note that ESC is simply a standard name for the Chr\$(27) character):

| Name | Code | Effect |
|---------|---------|--|
| aRIS | ESCc | hard reset |
| aRIN | ESC#1 | initialise to defaults |
| aIND | ESCD | true line-feed |
| aNEL | ESCE | line-feed. This is to be added after every line! |
| aRI | ESCM | reverse line-feed |
| aSGRO | ESC[Om | normal character set |
| aSGR3 | ESC[3m | Italics on |
| aSGR23 | ESC[23m | Italics off |
| aSGR4 | ESC[4m | underline on |
| aSGR24 | ESC[24m | underline off |
| aSGR1 | ESC[1m | boldface on |
| aSGR22 | ESC[22m | normal pitch |
| aSHORP2 | ESC[2w | Elite on |
| aSHORP1 | ESC[1w | Elite off |
| aSHORP4 | ESC[4w | condensed on |
| aSHORP3 | ESC[3w | condensed off |
| aSHORP6 | ESC[6w | enlarged on |
| aSHORP5 | ESC[5w | enlarged off |
| aDEN6 | ESC[6"z | shadow print on |
| aDEN5 | ESC[5"z | shadow print off |
| aDEN4 | ESC[4"z | double-strike on |
| aDEN3 | ESC[3"z | double-strike off |
| aDEN2 | ESC[2"z | near-letter-quality on |
| aDEN1 | ESC[1"z | near-letter-quality off |
| aSUS2 | ESC[2v | superscript on |
| aSUS1 | ESC[1v | superscript off |
| aSUS | ESC[4v | subscript on |
| aSUS3 | ESC3v | subscript off |

In order to print a text string in Italics and underlined, for example, the following routine could be used:

```
ESC$=Chr$(27)
LF$=ESC$+"E"
Printer Open
Printer Send ESC$+"[3m"+ESC$+"[4m"+"Greetings!" +LF$
Printer Close
```

The state of printer output can be monitored by the PRINTER CHECK and PRINTER ERROR functions, and printing can be abandoned completely using PRINTER ABORT, which are all explained later.

Screen dumps

There are three alternative ways of using the PRINTER DUMP command to perform a screen dump.

PRINTER DUMP

instruction: print the contents of an AMOS Professional screen

```
Printer Dump
Printer Dump x1 ,y1 To x2,y2
Printer Dump x1 ,y1 To x2,y2,px,py,setting
```

Used without any parameters, PRINTER DUMP will perform an entire screen dump in a single operation. If the screen contains complex graphics, this may well take a considerable time to complete.

A selected area of the screen can be transmitted to the printer, retaining the current aspect ratio and screen size. In other words, if only half of the current display is to be printed, it will take up exactly half of the space of a complete print-out.

The section of screen to be dumped is set by giving the top left-hand coordinates followed by the coordinates of the corner diagonally opposite.

The third option allows you to change the size parameters and aspect ratio of the original screen image. This is achieved by including additional parameters after x1,y1 and x2,y2, as follows:

Px and py specify the dimensions of the final print-out, and these values are measured in printer pixels rather than normal screen pixels.

Printer pixels vary in size, depending on the command options that are given by the following settings:

Settings refers to a special command parameter, that is used to tell the printer precisely how to draw the current screen image on paper. Here is a list of these settings, which can be combined using AND as well as OR operations from AMOS Professional.

| Value | Name | Description |
|--------|----------|---|
| \$0001 | MILCOLS | px is in 1/1000" (see Note 1) |
| \$0002 | MILROWS | py is in 1/1000" |
| \$0004 | FULLCOLS | use maximum print width (see Note 2) |
| \$0008 | FULLROWS | use maximum print height |
| \$0010 | FRACCOLS | px is a fraction of FULLCOLS (see Note 3) |
| \$0020 | FRACROWS | py is a fraction of FULLROWS |
| \$0040 | CENTRE | centre the image on the page |
| \$0080 | ASPECT | retain the original aspect ratio |
| \$0100 | DENSITY1 | set resolution (dots per inch) |
| \$0200 | DENSITY2 | next resolution |
| \$0300 | DENSITY3 | next resolution |
| \$0400 | DENSITY4 | next resolution |

| | | |
|--------------|-------------|--------------------|
| \$0400 | DENSITY4 | next resolution |
| Value | Name | Description |
| \$0500 | DENSITY5 | next resolution |
| \$0600 | DENSITY6 | next resolution |
| \$0700 | DENSITY7 | set resolution |
| \$0800 | NOFORM FEED | do not eject paper |
| \$1000 | TRUSTME | do not re-set |
| \$2000 | NOPRINT | do not print |

Note 1. MILCOLS and MILROWS measure the px and py parameters in units of 1/1000th of an inch.

So if

px=10000, the print-out will measure ten inches wide.

Note 2. FULLCOLS and FULLROWS make use of the maximum available width and height of the paper.

Note 3. FRACCOLS and FRACROWS specify the width and height of the print-out as a fraction of the current paper size.

Px,py are assumed to be between \$0000 and \$FFFF, and the print width is calculated using the formula width

$= \text{FULLCOLS} \times \text{px} / \FFFF whereas height is calculated by the formula $\text{height} = \text{FULLROWS} \times \text{py} / \FFFF .

This means that if px = \$8000 the print width will be $\$8000 / \FFF , which equals half of FULLCOLS.

So the print-out

will take up half the width of the paper.

For example, to dump a 100x100 section of the current screen onto a full page, retaining the correct aspect ratio, the following line would be used:

```
Printer Dump 0,0 To 100,100,0,0,$80 or $8 or $4
```

The next example could be used for printing a low resolution screen to an eight by six inch area:

```
Printer Dump 0,0 To 320,200,8000,6000,$1 or $2
```

Alternatively, part of the current screen could be dumped utilising the maximum available height, but with the width

reduced by one third, as follows:

```
Printer Dump 0,0 To 200,200,$5555,0,$8 or $10
```

You are warned not to attempt to change the current screen during a screen dump operation, otherwise the resultant print-out will become scrambled.

Other printer commands

instruction: print data from an address

```
Printer Out address,length
```

This command is used to print some data directly from the memory location starting at a specified address. The data is not processed in any way, so embedded control sequences will be completely ignored. The PRINTER OUT

instruction should be used to send simple text and graphics only.

The address parameter refers to the first character which is to be output, and length specifies the number of characters to be printed. To send a string, the following type of line would be used:

```
Printer Out Varptr(A$),Len(A$)
```

Similarly to PRINTER SEND, it must be ensured that data remains unchanged during the printing process, otherwise the resultant print-out will become corrupted.

PRINTER ABORT

instruction: stop a printer operation

```
Printer Abort
```

This command halts the current printing operation. If your printer device has a large memory buffer, there may be a delay before the printing ceases.

PRINTER CHECK

function: return the status of the printer

```
status=Printer Check
```

Use the PRINTER CHECK function to return a value of -1 (True) if the printer is available for use, or zero (False) if it is in active mid-operation.

PRINTER ONLINE

function: report if printer is on-line

```
status=Printer Online
```

This useful function provides a simple method of checking if the printer is connected and ready for use. It returns a value of -1 (True) if the printer is on-line, otherwise zero (False) will be given. This function only works with parallel printer devices.

PRINTER ERROR

function: check for an error in printing operation

```
status=Printer Error
```

Use this function to check if the current printing operation is proceeding normally. A value of zero suggests that all is well, but any other value indicates an error.

PRINTER BASE

function: get the address of printer base

```
address=Printer Base
```

The PRINTER BASE function is used to return the address of the i/o structure used to control the printer. It is intended for use by skilled operators only! Poking around the internal device structures is a very dangerous operation indeed!

Other ports and devices

The serial Port is examined in the next chapter, and it is also possible to access the parallel port directly, which provides a number of advantages over the printer device. Please refer to Chapter 10.5 for full details.

The complete control and exploitation of other devices that control hardware as well as internal features of the Amiga is dealt with in Chapter 11.5. This chapter ends with a general instruction and function for dealing with ports.

OPEN PORT

instruction: open a channel to an I/O port

```
Open Port channel number,"PAR:"  
Open Port channel number,"SER:"  
Open Port channel number,"PRT:"
```

The three versions of the OPEN PORT command are shown above, and they are used to open a channel to the Parallel Interface, or the RS232 Port, or the printer chosen in your preferences settings. All standard sequential file commands can be performed as usual, except for commands that are specific to disc operations, such as LOF and POF.

This example would print out ten lines via the device connected to the Amiga's RS232 port:

```
Open Port 1,"SER:"  
For x=0 To 10  
    Print #1 ,"Greetings from AMOS Professional!"  
Next x  
Close 1
```

PORT

function: test readiness of device

```
value=Port(channel number)
```

The PORT function is used to test the status of readiness of the specified channel. If the device is waiting to be read

a value of -1 (True) is returned, otherwise zero (False) is given.