HINTS & TIPS

Keep sending in your hints on anything relevant to the Archimedes range. And remember, we pay for every hint we publish.

OBEY FILE ERRORS

Lee Calcraft

If an error box appears on the Desktop which does not specify which task it comes from, it is likely that this is caused by an error in an Obey file, and in particular the !Run file of an application which has just been launched.

It would be very helpful if future versions of the OS could display the name of the file in such cases - Acorn please note!

When such errors occur, the Obey file stops executing at the line in error, and it can be hard to sort out what is going on. For example, you might get:

```
Too long
```

as the entire error message.

One way to track down such errors is to insert the line:

```
ERROR "Program halted"
```

halfway down your Obey file, and run your application. If you see the Program halted message in an error box, you know that the Obey file error lies after this line. Then simply move the ERROR line around until you have tracked down the bug.

LOST APPLICATION SPRITES

Graham Crow

Some Applications do not include a line such as:

```
IconSprites <Obey$Dir>.!Sprites
```

in their !Run file. The result is that if such applications are dragged onto certain applications, such as Menon the popular application launcher, their icon fails to appear on the icon bar. To remedy this, simply add the above line to the !Run file.

WORK AREA IN Edit AND DeskEdit

Lee Calcraft

The Edit$Options code to set the width of the work area in an Edit window is:

```
Ann
```

where nn is the number of characters required. Thus to set a window width of 60 characters use:

```
*Set Edit$Options A60
```

All windows opened after that point will have the new maximum work area width.

DeskEdit$Options works in the same way, and you may like to add the Ann command to the DeskEdit$Options setting in DeskEdit s !Run file.

AUTOMATIC HI-RES RESOURCE ICONS

M.J.Ebourne

A little-known feature of RISC OS 3 appears to be that placing copies of both a !Sprites and !Sprites22 file in Resources:$.Resources.Wimp using a resource utility such as ResourceFS (RISC User 6:5 magazine disc), you should find that the Wimp will automatically use the correct set of icons (i.e. low or high resolution) for the mode you are in. The same is true for Sprites23 files too.

This should make programs such as !ResoIcons, which performs a *IconSprites every time a mode change occurs (sometimes taking a noticeable amount of time to do so) somewhat redundant. This is why most disc-based applications have all three !Sprites, !Sprites22 and !Sprites23 files inside them. The Filer will then display the correct hi-res or low-res icons for the mode you are in.

RISC User DESKTOP DIARY UPDATE

Mike King

Mike Ironmonger s Diary application (RISC User 2:9) has, with the changes in RISC OS, developed a nasty habit of generating a variety of errors when other applications are installed alongside it on the icon bar. There is a simple solution to this problem. You should change line 3270 of the program from:

```
     DIM block% 90,menu% 400
```

to:

```
     DIM block% 150,menu% 400
```

This fix should extend the life of this very useful application.

REPLACEMENT FOR VDU 2

Graham Crow

SYS OS_PrintChar is a useful call. It sends a character to the printer stream and may be used directly, with no need for VDU 2 or writing to the file printer:. Here are a couple of procedures which make use of this call:

```
     DEF PROCprintstring(str$)
     LOCAL J%,c$
     FOR J%=1 TO LENstr$
     c$=MID$(str$,J%,1)
     SYS "OS_PrintChar",ASC c$
     NEXT
     ENDPROC
     :
     DEF PROCprintfile(fsp$)
     F%=OPENUP(fsp$)
     IF F%=0 THEN ERROR 255,"Debug File not
found"
     WHILE NOT EOF#F%
     b%=BGET#F%
     IF b%=&A SYS "OS_PrintChar",&D
     SYS "OS_PrintChar",b%
     ENDWHILE
     CLOSE #F%:F%=0
     ENDPROC
```

The first procedure sends a string to the printer and the second spools the same printer output to a file. The call is fully documented in the RISC OS 2 PRM on page 188 and in the RISC OS 3 PRM on 1-521.

HALTING A SOUND SAMPLE

Dave Patten

If you are using sampled sounds which are installed as modules, and which are then played by using the SOUND command, the duration of sound output totally ignores the duration parameter in the command. Thus for example if you have a sample which plays for 30 seconds, it will still play for the full 30 seconds even when called with:

```
     SOUND 1,-15,100,1
```

A simple way to stop the sound at will is to assign the sound channel to voice zero - for example:

```
     *ChannelVoice 1 0
```

will silence the sound on channel 1.

CREATING EVEN MORE AIM IMAGES USING ChangeFSI

Phillip Coffey

As Peter Schmidt suggested (Hints & Tips RISC User 6:9), using ChangeFSI to convert RISC OS Sprites to the AIM format for manipulation by the AIM image processing software is not particularly straightforward. You can use the command line format used by Peter Schmidt, or you can use the following method from either the command line or a Task Window, inserting the correct paths at the appropriate points:

```
     *DIR <ChangeFSI_Path>
     *ChangeFSI -Help
     ChangeFSI interactive input (read FSIinfo for
     command line details)
     Source file: <SourceFileNamePath>
     Destination file: <DestinationFileNamePath>
     Output mode: AIM
      (X)Scale (e.g. 2:1): =
     Y Scale: =
     Info on picture?
     Invert picture colours?
     Compute histogram of input?
     Apply histogram equalisation?
     Expand input dynamic range?
     Sharpen picture by (e.g. 12 for harsh, 24 for
     soft)
     Rotate?
```

In response to the last seven prompts, just press Return in each case. This isn t, alas, the end of the story. When you load the output file into AIM it will report that no header information has been found. To remedy this, use the header file trui+, which can be found inside the Images directory inside AIM. You should rename this file to the same name as your output file with a + appended. The information inside the header file does not need to be altered, but AIM will now happily accept your image.

TASK WINDOW PROBLEMS

Andrew Brooks

If you have any programs which used to work with RISC OS 2 Task Windows, but do not display carriage returns properly with RISC OS 3, then simply add this alias to the start of their !Run files:

```
     Set Alias$ShellCLI_Task %TaskWindow GOS -
ctrl-name TaskWindow -wimpslot 256K -task &%%0-
txt &%%1
```

Note that this is a single line.

QUICKER PRINTER SETUPS

David Spencer

The RISC OS 3 Printers application has a very handy, but not well documented, short cut for setting up printer configurations. Quite simply, Shift-clicking Select over a particular printer s icon bar icon will open the configuration dialogue box for that printer, whilst Shift-clicking Adjust will open the connections dialogue box. This is much easier and quicker than using a menu to bring up the printer control list and then another menu to open the setup boxes. Incidentally, always remember to click on OK in the setup boxes, rather than just closing them, otherwise any changes will be lost.

FILE OUTPUT USING OS_GBPB

Graham Crow

A useful SYS call often overlooked is SYS OS_GBPB with R0=1 or R0=2. This writes bytes to an open file at the specified file pointer (R0=1) or the current one (R0=2). R1 contains the file handle, R2 a buffer to hold the data, and R3 the number of bytes to write. The file is automatically extended if necessary. This can easily be written as a procedure, for example:

```
     DIM buffer% &100
     fsp$="RAM::RamDisc0.$.MyFile":REM or
whatever
     F%=OPENOUT(fsp$)
     IF F%=0 THEN ERROR 255,"Could not Open
File)
     separator$=CHR$(&A):REM linefeed
     PROCwritebytes(F%,"The quality of mercy..."
     +separator$,buffer%)
     CLOSE #F%:F%=0
     OSCLI("SetType "+fsp$+" &FFF")
     END
     :
     DEFPROCwritebytes(file%,str$,buf%)
     REM write bytes to open file, extending as
     necessary
     LOCAL len%
     len%=LENstr$:$(buf%)=str$
     SYS"OS_GBPB",2,file%,buf%,len%
     ENDPROC
```

The call is fully documented in the RISC OS 2 PRM on page 872 and in the RISC OS 3 PRM on 2-65.

FASTER FONT CATALOGUING

M.J.Ebourne

An interesting feature of the new Font Manager is buried deep within the RISC OS 3 PRM. It concerns the use of message files. I m sure many readers have a sizable collection of fonts on their hard discs, as I do. This can be somewhat of a burden, since it can take an annoyingly long time to catalogue them each time your system re-boots. A method of reducing this delay is to include a Messages1 file inside your !Fonts directory (the 1 is the territory number for the UK, change this for other countries). The message file provides a translation table between physical font identifiers and local font names. This message file should contain lines of the form :

```
     Font_<fontidentifier>:<localfontname>
```

for example:

```
     Font_Trinity.Medium.Italic:Trinity.Medium
     Italic
```

I have omitted the full stop between Medium and Italic in order to make the font list look nicer in the new style font menus (e.g. Edit and Impression). The real upshot of this is that if you provide a messages file, the Font Manager does not scan the fonts directory, thus reducing boot-up time. An added bonus is that if the <localfontname> is followed by an asterisk (i.e. *) it will be taken as the default for that font family if the font is clicked on from the main font list.

FaxPack STATUS DISPLAY

Lee Calcraft

You may not have spotted that the large Fax setup window generated by the FPdriver application actually contains a dynamic status display. This gives call progress information, and then details about file transfer - all of which can be very handy, especially if you are getting line errors, or are sending large