HINT & TIPS

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

Copying more than a discful

Lee Calcraft

If you want to copy more than a discful of files from one machine to another it is tricky to ensure that each subsequent disc starts at exactly the right file, and it is easy to omit the odd file or directory between each discful saved. But you can quite easily automate the process on RISC OS 3.

Suppose you have dragged a large number of objects, or a large application or directory to the target floppy. After a while you will get the disc full message. Resist the temptation to respond to the message with the mouse, and simply remove the floppy. Then copy its contents to the destination machine by dragging as normal.

Now simply insert the floppy back into the source machine and delete its contents, either by dragging to your Dustbin, or by using the Delete option on the Filer menu. When the disc is empty you can finally respond to the earlier error message by clicking on Retry. As if by magic, the copying will continue from where it left off, recreating any nested directories as necessary. When you come to transfer this discful to the destination machine, do exactly the same drag as you did the first time, and all files will be correctly merged - and of course, you can repeat the process over a number of discs, providing that no file is larger than the disc capacity of the discs you are using.

Numbered tickets in Ovation

D.P. Brooks

Good Impressions (RISC User 7:8) showed how to generate sequentially numbered tickets using Impression. The same method can be used with Ovation with one or two modifications. The Basic program remains the same, although it is not important that the high number is the same as the last ticket number, provided it exceeds it.

Prepare a new document and select the master page, on which you should create the ticket frames. Insert into each frame a small text frame, just big enough for one line of text, where the number is to go. Select the link tool and link each number frame. The master page may now be saved as a stylesheet if required.

Return to the main document and insert a page, which should now contain the set of ticket frames. Enter the text into the first, and use copy and paste to fill the other frames. Whenever you generate a new page, link the last number frame on the previous page with the first number frame on the new page. The numbers file created by your Basic program can be dropped into the first number frame at any time, and provided you have got the frame size right the numbers will flow from frame to frame.

Named loads in DeskEdit

Lee Calcraft

The little-known keyboard shortcut Escape-L in DeskEdit puts up a dialogue box to load a named file. This can be very handy if you want to check a file for possible virus infection without displaying it in the Filer, and so giving it the opportunity to infect your machine. Simply insert your floppy into the drive, press F12 and type:

```
    Cat
```

to catalogue the disc, and then get back to the desktop (by pressing Return), and use Escape-L to load any suspicious files.

Run file quirks

James Wright

Commands in an Obey file are executed sequentially, each command being seen only after the previous one has finished. This means, for example, that if you want an application s !Run file to carry out a further command after it has installed the application with a Run command, the subsequent instructions will not be executed until the application is terminated. This may be what you want in some cases (see this month s Just-a-Page for an example), but you will be caught out if you try to do something which is intended to take effect while the application is still around.

It also means that if there are any additional lines after the Run command, whether they contain instructions or not, the !Run file will be left open all the time the application is running. You may discover this by accident when you try to edit the !Run file while the application is installed. A single linefeed at the end of the Run command line is acceptable (though not obligatory), but any other blank lines will leave the file open. Note that if you are unfortunate enough to have two such !Run files open at the same time, there is a nasty bug in RISC OS 3.1 which causes a crash if you try to quit one of the applications. This appears to have been fixed in RISC OS 3.5.

Rich text format with Wordz

Atle Mjelde B rdholt

A little known feature of Colton s Wordz (from version 1.05 onwards) is its ability to load and save RTF (Rich Text Format) files. These have an Acorn approved filetype (&C32), and enable you to transfer documents complete with information on styles, margins, tabs, fonts etc. to and from many PC and Mac applications, as well as any other Acorn-based applications that may offer the facility.

Automating pinboard saves

Lee Calcraft

One thing which I find really annoying about the pinboard supplied with RISC OS 3 is that every time you add an icon to it, you have to go through a considerable rigmarole to save the results (no marks for the team which did not have time to improve things in this respect on the Risc PC!).

To ensure that a newly pinned icon will remain pinned when you next boot your machine you have to resave the desktop boot file. And unless you reboot your machine before you do this (so losing any newly pinned icons that you wanted to save!), your boot file will contain all kinds of junk that you really do not want.

However, there is a way to make the process almost painless. The idea is to use the pinboard Save option to save the latest pinboard and backdrop information, and to simply edit your desktop boot file so that it automatically runs this at boot-up. Once you have done this, you can permanently save any changes in your pinboard in a matter of seconds by selecting the Save option from the pinboard menu, and dragging the file (already named Pinboard) to the nominated directory. This should be somewhere that can be easily accessed. On the Risc PC I save it directly inside !Boot, but anywhere will do.

To set up this method, you need only to remove all the commands in your desktop boot file to do with the pinboard (i.e. all those that begin Pin.. , or Backdrop). Fortunately they are all grouped together. Then simply replace them with a single command to run your new pinboard file. For example:

```
     Run ADFS::HardDisc4.$.Pinboard
```

The one snag to this method is that if you ever resave your desktop boot file, you will need to replace the pinboard commands again with a single call to run your pinboard file. However, this is generally best avoided in any case. If you do want to make changes to your desktop boot file it is usually easier to save the new file with a different name, and then simply edit your present boot file using bits from the new one.

Faster installation of printer manager

Dr. M. Richard

If you only ever use one particular type of printer, the printer manager application will load significantly faster if you either delete or move any unused subdirectories from within the !Printers application directory. The required subdirectories for the various printer types are as follows:

dp	for dot-matrix printers
lj	for LaserJet printers
ps	for PostScript printers

Remove those that are not required, but do keep a backup copy in case you need them at a later date.

Foiling the practical joker

Sam Chamberlain

I don t know about you, but I hate it when school pupils with a basic knowledge of the *Configure command decide to put several fake hard discs on the icon bar! There is a simple solution to this, and to other nasties such as wiping discs or unplugging modules, as the following program demonstrates:

```
     10 REM Basic Protection
     20 REM Not Public Domain
     30 *Set Alias$Configure "Echo Unable to config
ure"
     40 *Set Alias$Modules "Echo No such thing!"
     50 *Set Alias$Unplug "Echo Already unplugged!"
     60 *Set Alias$Wipe "Echo NO!"
```

After running this program, if anyone tries to use the Configure, Modules, Unplug or Wipe commands, all they will get is the corresponding message and nothing else. The program can be run from a boot file when the machine starts up. Experienced programmers could find a way around this, but it s probably not the experienced programmers you wish to protect against.

Using non-Latin1 characters

Govind Kharbanda

In Write-Back (RISC User 6:9) Richard Hallas wrote about the inability of RISC OS 3 to allow the user access to non-Latin1 characters in desktop applications, such as r caron (as in Dvorak) and c circumflex, as used in Esperanto. These, as he says, are actually present in RISC OS 3 fonts.

Until recently I used to obtain these in Edit by using the system font and altering the alphabet. For c circumflex for instance, you could use either of the following commands from the command line:

```
     *Alphabet Latin3
     *Country Esperanto
```

and for r caron:

```
     *Alphabet Latin2
```

The command does not seem to work from a task window, however. The full range of characters for each alphabet can be seen by using the Chars application in the Apps directory.

A c circumflex (in Latin3) can now be obtained by typing Alt-a, while an r caron can be obtained in Latin2 by typing Alt-o.

If you use an outline font in Edit, however, only the Latin1 set is used, regardless of the configured alphabet. Fortunately I have found a solution, which involves setting the system variable Edit$Options . One of the optional parameters to this variable can be used to set the default display font, including the required alphabet. First of all, press F12 to get to the command line, then type:

```
     *Show Edit$Options
```

If the variable is not already declared on your computer you will just get a blank line, or alternatively the current definition will be displayed, which will look something like this:

```
     Edit$Options : f7 b0
```

There may be a number of other parameters. The parameter for setting the font is n followed by \F<font name>\E<alphabet>, e.g.:

```
     n\FTrinity.Medium\ELatin1
```

If this parameter already exists you should alter it to the required alphabet; if it doesn t exist you should add it after any other parameters already there. This is done using Set, as in the following example:

```
     Set Edit$Options f7 b0 n\FTrinity.Medium\
 ELatin3
```

Make sure that you retain all the existing parameters in the command.

Now when you next load a file into Edit it will be displayed in the specified font and alphabet. Unfortunately you can only use one font and alphabet at any one time in any Edit document.

The same principle can be used to access non-Latin1 characters in your own programs. For example, you can declare a font in Basic with the following command:

```
     SYS "Font_FindFont",,"Trinity.Medium
 \ELatin3",size%,size%,0,0 TO font%
```

More on path variables...

Alan Wrigley

In Technical Queries (RISC User 6:9) we described the difference between a ...$Dir variable and a ...$Path one, by explaining that the latter creates a kind of pseudo filing system. This enables you to access a path specified, for example, in a variable App$Path, simply by prefixing files with App:.

An important feature of path variables that was not mentioned, however, is that they can be set up to point to multiple paths, merely by separating the pathnames with a comma. For example, if you look at the RISC OS 3 variable Edit$Path, you will see:

```
     Resources:$.Apps.!Edit.,Resources:$.Resources.
 Edit.
```

This enables Edit to look for its resources in two separate places using the same path prefix.

... and font path variables

Alan Wrigley

The variable Font$Path, which is used by the font manager to locate any fonts in your machine, is itself a pseudo filing system for the very reason given in the previous hint. You can take advantage of this to get a quick listing of all the fonts known to the font manager. The normal way to do this is to use the FontCat command, but this simply lists all the fonts and weights one to a line, making it difficult to see the font names at a glance if you have a lot of them. If, however, you use: