

TECHNICAL QUERIES

Alan Wrigley answers your questions on how to find out if a specific task is running, and how to replace the directory folder sprite

Q Dear Sir
The DeskTidy application on the 7:5 disc provided a set of icons for use with directories. Is there a way of replacing my own standard directory icons with customised ones as well?

Roy Nelson

A The Filer always represents a standard directory using the sprite directory, so in order to use individual sprites you have to fool the Filer into thinking that the directory is an application. An application is identified by an exclamation mark at the start of its name, so if for example you have a directory containing DTP resources which you have called DTP, then by renaming it !DTP you will be able to use an individual sprite. In order to tell the Filer which sprite to use, the directory must contain a sprite file called !Sprites, which itself contains a sprite called !dtp.

Having done this, however, you are faced with a further problem - you can no longer double-click on the directory to open it since double-clicking on an application normally causes the application to be run rather than the directory to be opened up. If you do this now you will get an error telling you that the !Run file cannot be found. All you need to overcome this is an Obey file called !Run inside your directory containing (for RISC OS 3) just one line:

```
Filer_OpenDir <Obey$Dir>
```

RISC OS 2 requires a slightly more complex !Run file, as follows:

```
Set Alias$Temp Filer_OpenDir <Obey$Dir>
Temp
Unset Alias$Temp
```

This will now cause the directory to be opened if you double-click on its icon.

Dear Sir

A Is there a way to find out if a particular task, for example Edit, is currently running?

G H Keathley

Assuming that you have RISC OS 3, this is quite easy to achieve, using the SWI call TaskManager_EnumerateTasks. This SWI fills a buffer with information about currently active tasks, including the task handle and the task name. The entry parameters (which I will explain in more detail below) are as follows:

R0 = task number in list
R1 = pointer to word-aligned buffer
R2 = buffer length in bytes

and the exit parameters are:

R0 = next number in list, or <0 if no more
R1 = pointer to first unused word in buffer
R2 = number of unused bytes in buffer

Each entry in the buffer on exit is held in 16 bytes, as follows:

0 task handle
4 pointer to null-terminated task name
8 Wimp slot of task
12 flags

Only the bottom two bits of the flags have any relevance - if bit 0 is set, the task is a module, otherwise it is an application; and if bit 1 is set, the task's slot bar in the Tasks window can be dragged.

Since you don't know how many tasks might be running, the easiest way to use the call is to set up a buffer of 16 bytes (which will obviously hold only one entry at a time) and make the call repeatedly, starting with zero in R0 and using the value returned in R0 for the next call, until either R0 is less than zero, or the string pointed to at buffer+4 is the same as the task name you are looking for (the name pointed to will be the one which is quoted in the Tasks window, e.g. DeskEdit or ADFS Filer). If the first of these conditions is true then the task is not currently running. If you are searching for Edit, the code would look something like this:

```
DIM buff% 15:c%=0
REPEAT
SYS "TaskManager_EnumerateTasks",c%,buff%
TO c%
a$="":ptr%=buff%+4
REPEAT
```



```

a$+=CHR$?ptr%:ptr%+=1:UNTIL ?ptr%<32
UNTIL a$="Edit" OR c%<0
IF a$<>"Edit" ERROR 1,"Edit is not running"

```

You can of course replace the last line with any other action you wish to take, such as setting a flag for example:



caption

```
editrunning%=(a$="Edit")
```