

New Technical Notes

Macintosh



®

Developer Support

MPW Library Q&As

Platforms & Tools

M.PT.MPWLibrary.Q&As

Revised by: Developer Support Center

October 1992

Written by: Developer Support Center

October 1990

This Technical Note contains a collection of Q&As relating to a specific topic—questions you've sent the Developer Support Center (DSC) along with answers from the DSC engineers. While DSC engineers have checked the Q&A content for accuracy, the Q&A Technical Notes don't have the editing and organization of other Technical Notes. The Q&A function is to get new technical information and updates to you quickly, saving the polish for when the information migrates into reference manuals.

Q&As are now included with Technical Notes to make access to technical updates easier for you. If you have comments or suggestions about Q&A content or distribution, please let us know by sending an AppleLink to DEVFEEDBACK. Apple Partners may send technical questions about Q&A content to DEVSUPPORT for resolution.

New Q&As and Q&As revised this month are marked with a bar in the side margin.

Getting MPW to recognize a file created using “open”

Written: 5/3/89

Last reviewed: 8/1/92

I'm using the C standard library routines to read and write text files. But when I create a file using `open()`, MPW doesn't recognize it as a TEXT file. What's going on?

When `open()` creates a file, it leaves the `fdType` and `fdCreator` blank, since it doesn't know the format of the file you are going to write (it's possible to write binary data using the standard C library).

To change the file type and creator of a file created by `open()`, use `SetFInfo`, as shown in the example below:

```
#include <stdio.h>
#include <fcntl.h>
#include <Files.h>
```

```
int main(void)
```

```
{  
  int fd;  
  short vol, err;  
  FInfo fInfo;
```

```
fd = open("junk.out",O_CREAT | O_RDWR | O_TRUNC);
if (fd < 0) {
    printf("Error creating file\n");
    return(1);
}
(void) close(fd);
err = GetVol((StringPtr) NULL,&vol);
if (err != 0) {
    printf("Error %d getting current volume\n",err);
    return(1);
}
err = GetFInfo((StringPtr) "\pjunk.out", vol, &fInfo);
if (err != 0) {
    printf("Error %d getting file info\n",err);
    return(1);
}
fInfo.fdType = 'TEXT';
fInfo.fdCreator = 'MPS ';
err = SetFInfo((StringPtr) "\pjunk.out", vol, &fInfo);
if (err != 0) {
    printf("Error %d setting file info\n",err);
    return(1);
}
return(0);
}
```

By the way, you are generally much better off if you use the File Manager's routines to manipulate your files. The performance will be much better.

X-Refs:

File Manager

Macintosh Technical Note "Mixing HFS and C File I/O"

SIOW libraries run in 32-bit mode

Written: 7/30/91

Last reviewed: 8/1/92

Comments in the SIOW.r file indicate that the Standard I/O Window (SIOW) software is not 32-bit clean. Are there any known bugs that crop up if the current software is run in 32-bit mode anyway?

—

The "not32BitCompatible" setting in the SIZE resource is ignored, more or less. Originally, it was going to trigger an alert like "Be careful - this application may crash!" if the system was running in 32-bit mode; thankfully, this alert was removed before System 7.0 was shipped (though A/UX still has it).

As far as the SIOW libraries are concerned, there isn't anything about them that restricts their use to 24-bit environments. Other than the problems discussed in the release notes that come with MPW 3.2 (which includes SIOW), Apple doesn't know of any bugs with SIOW.