

# New Technical Notes

Macintosh



---

Developer Support

## OpenRFPPerm: What your mother never told you.

### Toolbox

Written by: Dave Burnard

April 1988

This note corrects an error in the description of the Resource Manager routine `OpenRFPPerm` found in *Inside Macintosh Volume IV*.

---

On page IV-17 in the Resource Manager chapter of *Inside Macintosh Volume IV* it states,

“`OpenRFPPerm`, like `OpenResFile`, will not open the specified file twice; it simply returns the reference number already assigned to the file. In other words, `OpenRFPPerm` cannot be used to open a second access path to a resource file...”

**This statement is incorrect.** `OpenRFPPerm` behaves exactly as described if an application attempts to open a second access path, **with write access**, to a resource file without MultiFinder. With MultiFinder, however, if the second attempt comes from a different application than the one that originally opened the file `OpenRFPPerm` **will not** return the reference number already assigned to the file, instead it will return `-1` and `ResError` will be set to `opWrErr` (`-49`). In fact in similar circumstances with MultiFinder, `OpenResFile` behaves the same way, returning `-1` and setting `ResError` to `-49`.

Note, however, that with or without MultiFinder, `OpenRFPPerm` **will** create multiple, unique, read-only access paths to a resource file. Using this feature is not safe and should be avoided since if a resource file is opened twice, once with read-write permission and once with read-only permission then two copies of the resource map will exist in memory. If one of the resources in the file is changed and written to disk then the two maps will become inconsistent and the second access path will see what it thinks is a trashed resource file.

If you must use this technique for read-only access, only call `OpenRFPPerm` when your application is ready to read information from the file and close the file immediately afterwards. Otherwise you risk having the resource file change out from under your application.

### Further Reference:

- The Resource Manager