Shared Files and Folders Allow multiple, simultaneous user of applications

Normally, applications on an AppleShare server volume can only be used by one user at a time. When an application is launched, the Finder checks the **shared bit** and the **file attributes**. The **file attributes** are used to determine if the the file is locked, etc., and the **shared bit** is used to determine if the launch can proceed even if the file (application) is already in use (locked bit set, etc.).

Part of the reason why applications are not automatically sharable is based on the design of the **Resource Manager**. The **Resource Manager** is the foundation of the **Font Manager**, the **Segment Loader** and a good portion of the Operating System, as well as database-like tasks. Even so, it was never designed to perform multi-user database tasks.

If more than one user opened the same resource file at the same time, and one of them had write access to the file and modified the resource, it would invalidate the other user's **resource map**. This could cause a crash, and possibly corrupt files on disk. If both users had write access to the file and wrote to the file, an invalid resource file is almost a certainty. Although you can tell the Resource Manager to write out an updated **resource map**, there is no way for another user to get a fresh copy of the map in memory if the file changes.

What this all means is that you should not set the **shared bit** in your application unless you know that it will never write to itself on disk. Apple has several rules to make sure your application is multi-user friendly.

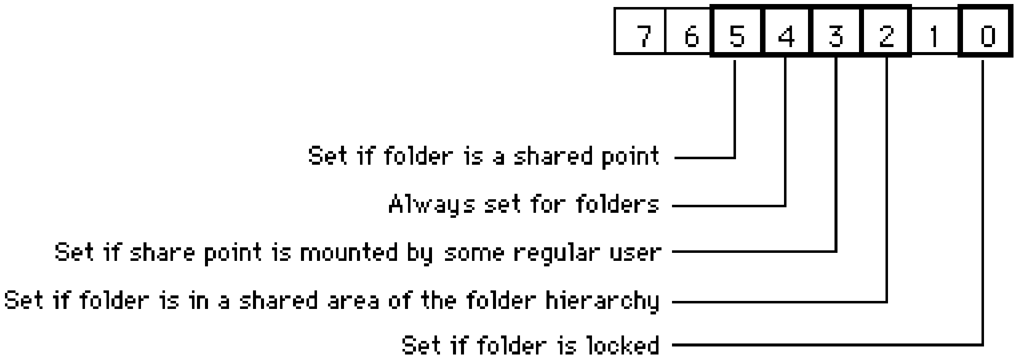
- Make sure your application does not create temporary files with fixed names in a fixed place (such as the directory containing the application).

- Make sure your application is aware that it may be in a multi-user environment. For instance, make sure that it works correctly if a volume containing an existing document is on a locked volume. Make sure to check all the result codes from **File Manager** calls.

By setting the **shared bit**, you are promising that your application will work 100% correctly if it is launched by more than one user.

There are several differences to note between AppleShare 2.0.1 and File Sharing. File Sharing allows both folders and volumes to be shared over a network. This is done via the Finder's Sharing dialog. Access privileges for the shared folders can be set by the local user.

To allow applications to see shared folders, bit definitions have been added to the ioFlAttrib.bitmap returned by **PBGetCatInfo** when information is being sought about a folder. The figure below shows ioFlAttrib for folders as returned by **PBGetCatInfo** under the System 7 File Manager.


7
6
5
4
3
2
1
0
Set if folder is a shared point
Always set for folders
Set if share point is mounted by some regular user
Set if folder is in a shared area of the folder hierarchy
Set if folder is locked

ioFlAttrib for a Folder

Note: These bits are READ-ONLY for folders. Do not try to set them with **PBSetCatInfo**. As noted in Inside Macintosh, Volume VI, **PBCatSearch** searches only on bits 0 and 4. The additional bits returned in ioFlAttrib by **PBGetCatInfo** cannot be used by **PBCatSearch**.

File Sharing supports a user access privilege called blank access privileges. A folder with blank user access privileges ignores the other access privilege bits and uses the access privileges of its parent. This is the default mode for folders in a shared area. Volumes that support blank access privileges have the bHasBlankAccessPrivileges bit set in the vMAttrib field in the GetVolParmsInfoBuffer structure returned by **PBHGetVolParms**.

Folders with blank access privileges can be identified by calling **PBHGetDirAccess**. Use **PBHSetDirAccess** to set blank access privileges for a folder.

Note: Only the blank access privileges bit in the high byte of ioACAccess may be set while calling **PBHSetDirAccess**. The blank access privilege bit is not returned in the ioACUser field by the **PBGetCatInfo** routine.