

Technical Note TB42

Finder Icon Positioning and File Initialization

CONTENTS

[The Original Positioning Mechanism
\(or what happens if you do nothing\)](#)

[Why Some Applications Try to Go Behind the Finder's Back](#)

[The Magic Icon Placement Solution](#)

[The Scriptable Finder](#)

[Conclusion](#)

[References](#)

[Downloadables](#)

This Technical Note discusses how your application can tell the Finder where to position a newly created file or folder.

[Jun 01 1995]

The Original Positioning Mechanism (or what happens if you do nothing)

The Finder uses the File Manager records reserved for the Finder's use, `FInfo` and `DInfo`, to keep track of the position (location) of files and folders appearing in windows displayed in view "by icon" or in view "by small icon." Whenever the File Manager creates a new file or directory, the `FInfo` or `DInfo` records are cleared. That sets the position to (0, 0) and clears the `hasBeenInited` Finder flag.

Whenever the Finder sees a file or directory with a cleared `hasBeenInited` Finder flag, it positions the file or directory in the first empty space in the window or on the desktop, and sets the `hasBeenInited` Finder flag. At that time, if a file's `hasBundle` Finder flag is set, the Finder also installs the bundle information from the file's bundle resource and all of its bundled resources into either the volume's desktop database or Desktop file, and installs the information needed by the Translation Manager, including the file's 'open' and 'kind' resource data.

[Back to top](#)

Why Some Applications Try to Go Behind the Finder's Back

Some applications want to set the position of new files or folders themselves. For example, an archiver might want extracted files to have the same position that the original file did before it was placed into the archive. In order to do this with a file, they must put the position in the `fdLocation` field of the `FInfo` record. They must also set the `hasBeenInited` Finder flag in the `fdFlags` field of the `FInfo` record; otherwise, the Finder will just reposition the file the next time it sees it, no matter what value is stored in the `fdLocation` field.

The problem With Setting the `hasBeenInited` Finder Flag

The problem with setting the `hasBeenInited` Finder flag is that it prevents the Finder from installing bundle and Translation Manager information into the desktop database. To fix this problem, some applications have been known to try to add information to the desktop database themselves, and wind up making a mess of the disk. That's why the Desktop Manager of *Inside Macintosh: More Macintosh Toolbox* includes the following warning:

Warning:

Although routines that set information in and get information from the desktop database are described in this section, you should never use these routines to change, add to, or remove any information from the desktop database. Manipulating the desktop database is likely to wreak havoc on your users' systems.

[Back to top](#)

The Magic Icon Placement Solution

Starting with System 7 Pro (System 7.1.1), the Finder supports magic icon placement. Magic icon placement allows an application to specify where a newly created file or folder with a cleared `hasBeenInited` Finder flag will be placed when the Finder initializes it. If the `hasBeenInited` Finder flag is clear, and the file or folder's position is somewhere inside the "magic rectangle" centered at (-20,000, -20,000), then the Finder will add 20,000 to the file or folder's horizontal

and vertical position, and place it there. The magic rectangle is (-24,000, -24,000, -16,000, -16,000).

Warning:

Don't put files whose `hasBeenInited` Finder flag is set into the "magic rectangle" or the Finder really will position them there!

The following functions show how to create a file or folder, and position it using the magic icon positioning technique.

```

OSErr  SetIconLocation(short vRefNum, long dirID, ConstStr255Param name,
                      Point location)
{
    /* (-20,000, -20,000) is the origin of the magic rectangle */
    enum { magicNumber = 20000 };

    OSErr      result;
    CInfoPBRec pb;

    pb.hFileInfo.ioNamePtr = (StringPtr)name;
    pb.hFileInfo.ioVRefNum = vRefNum;
    pb.hFileInfo.ioDirID = dirID;
    pb.hFileInfo.ioFDirIndex = 0; /* use ioNamePtr and ioDirID */
    result = PBGetCatInfoSync(&pb);
    if ( result == noErr )
    {
        /* set the magic location */
        location.v -= magicNumber;
        location.h -= magicNumber;
        pb.hFileInfo.ioFlFndrInfo.fdLocation = location;

        /* make sure the kHasBeenInited flag is clear */
        pb.hFileInfo.ioFlFndrInfo.fdFlags &= ~kHasBeenInited;

        /* save the new information back to disk */
        pb.hFileInfo.ioDirID = dirID;
        result = PBSetCatInfoSync(&pb);
    }
    return ( result );
}

OSErr  HCreateLocate(short vRefNum, long dirID, ConstStr255Param fileName,
                    OSType creator, OSType fileType, Point location)
{
    OSErr      result;

    result = HCreate(vRefNum, dirID, fileName, creator, fileType);
    if ( result == noErr )
        result = SetIconLocation(vRefNum, dirID, fileName, location);

    return ( result );
}

OSErr  HDirCreateLocate(short vRefNum, long parentDirID,
                       ConstStr255Param directoryName, Point location,
                       long *createdDirID)
{
    OSErr      result;

    result = DirCreate(vRefNum, parentDirID, directoryName, createdDirID);
    if ( result == noErr )
        result = SetIconLocation(vRefNum, parentDirID, directoryName, location);

    return ( result );
}

```

Determining if Magic Icon Placement Exists

It doesn't hurt anything to use the magic icon placement technique on a machine running a Finder that doesn't support it. Since the `hasBeenInited` Finder flag is cleared, older Finders will just reposition the file or folder in the ordinary manner. Some programs may wish to set both the `hasBeenInited` Finder flag and the position of the file if magic icon placement is not available. Programs can determine if magic icon placement exists by calling `Gestalt` with the Finder's `gestalt` selector, `gestaltFinderAttr`. If the `gestaltFinderAttr` selector does not exist, or if the attribute bit `gestaltFinderMagicPlacement` is clear, then magic icon placement is not available. The following code shows how to check for magic icon placement:

```
Boolean HasMagicPosition(void)
{
    long  attrib;

    if ( (Gestalt(gestaltFinderAttr, &attrib) == noErr) &&
        ((attrib & gestaltFinderMagicPlacement) != 0) )
        return ( true );
    else
        return ( false );
}
```

[Back to top](#)

The Scriptable Finder

This still leaves a couple of problems that magic icon positioning doesn't solve.

The first problem is the Finder does not initialize files until it sees them, and it does not go looking for them. A file created inside a new folder will not be seen by the Finder until that folder is opened by a user. This means that any bundle information in the file will not be added to the desktop database until some later time. This is a problem for some applications, because the Finder will not be able to launch documents of an application until it has seen the application file.

The second problem is magic icon positioning doesn't let you change the position of files once the Finder has seen them. When the Finder opens a folder, it gets (and sets if needed) the position of all file or folder icons in that folder's window. If you change the position of a file or folder using magic icon positioning, the Finder won't notice your changes until the window is closed and then reopened.

The solution to both of these problems is the scriptable Finder. With the scriptable Finder, you can force the Finder to notice new files, even if they are not in a visible window, and you can tell the Finder to move a file or folder's icon. "Scripting the Finder From Your Application" by Greg Anderson in *develop* issue 20 and the *AppleScript Finder Guide* both contain detailed information describing how you can use the scriptable Finder to perform these tasks and many others.

[Back to top](#)

Conclusion

For programs such as an archiver, the magic positioning technique provides an easy-to-implement solution for positioning newly created or recreated files and folders. However, the scriptable Finder, when available, provides a much wider range of services for manipulating the way files and folders are presented to a user.

[Back to top](#)

References

Inside Macintosh: Macintosh Toolbox Essentials , Chapter 7, "Finder Interface"

Inside Macintosh: More Macintosh Toolbox , Chapter 9, "Desktop Manager"

"Scripting the Finder From Your Application" by Greg Anderson, *develop* issue 20

AppleScript Finder Guide

Apple Event Registry: Standard Suites , on the Reference Library edition of Developer CD Series

[Back to top](#)

Downloadables



Acrobat version of this Note (K).

[Download](#)