Finder Flags

Flag name	Bit number	Description
isAlias	15	The file is an alias file. Reserved for directories-in which case, set to 0.
isInvisible	14	The file or directory is invisible from the **Finder** and from the **Standard File Package** dialog boxes.
hasBundle	13	The file contains a bundle resource. Reserved for directories-in which case, set to 0.
nameLocked	12	The file or directory can not be renamed from the **Finder**, and the icon cannot be changed.
isStationery	11	The file is a stationery pad. Reserved for directories-in which case, set to 0.
hasCustomIcon	10	The file or directory contains a customized icon.
Reserved	9	Reserved; set to 0.
hasBeenInited	8	The **Finder** has recorded information from the file's bundle resource into the desktop database and given the file or folder a position on the desktop.
hasNoINITS	7	The file contains no 'INIT' resources; set to 0.Reserved for directories; set to 0.
isShared	6	The application is available to multiple users. Defined only for applications; otherwise, set to 0.
requiresSwitchLaunch	5	Unused and reserved in System 7.0; set to 0.
colorReserved	4	Unused and reserved in System 7.0; set to 0.
color	1-3	Three bits of color coding.
isOnDesk	0	Unused and reserved in System 7.0; set to 0.

Masks for two of these bits are available as predefined constants:

fHasBundle	set if file has a bundle
fInvisible	set if icon is invisible

Of these **Finder** flags, the only ones that you might ever want to set are these:

- **isInvisible** This flag specifies that a file is invisible from the **Finder** and from the **Standard File Package** dialog boxes. Making a file invisible is generally not recommended. Not even temporary files need to be invisible because the Temporary Items folder into which they should be written is invisible.

- **hasBundle** This flag specifies that a file has a 'BNDL' resource that associates the file with your own icons. When the **Finder** displays or manipulates a file, it checks the file's hasBundle bit (also called the **bundle bit**). If that bit is not set, the **Finder** displays a default icon for that file type. If the hasBundle bit is set, the **Finder** checks the hasBeenInited bit. If the hasBeenInited bit is set, the **Finder** uses the information in the desktop database to display that file's icon. If the hasBeenInited bit is not set, the **Finder** installs the information from the bundle resource in the desktop database and sets the hasBeenInited bit. Most development environments provide a simple tool for setting the bundle bit when you create your application.

- **nameLocked**. This flag specifies that a file cannot be renamed from the **Finder** and that the file cannot have customized icons assigned to it by users.

- **isStationery**. This flag specifies that a file is a stationery pad. To support stationery pads, your application should check this bit for every document passed to it by either the **Finder** or the **Standard File Package**. (**StandardGetFile** and **CustomGetFile** return this flag in the sfType field of the standard file reply record.) If the isStationery bit is set for a file that a user wants to open, your application should copy the template's contents into a new document and open the document in an untitled window.

- **hasCustomIcon**. This flag specifies that a file has a customized icon. Customized Icons explains how users or your application can use customized icons.

The **Finder** manipulates the fields in the FXInfo, DInfo, and DXInfo records; your application should not have to directly check or set any of these fields. The FXInfo and DXInfo records have been changed slightly with System 7.0.

Previously reserved or unused fields in these two records are now partly used by the byte-length fdScript and frScript fields. These new fields are available for future enhancements of the script display capability of the **Finder**.

Ordinarily, the **Finder** displays the names of all desktop objects in the current system script, which depends on the region-specific configuration of the system. The high bit of the bytes in the fdScript and frScript fields is set by default to 0, which causes the **Finder** to display the filename or directory name in the current system script. If the high bit is set to 1, the **Finder** and the **Standard File Package** display the filename and directory name in the script whose code is recorded in the remaining 7 bits. However, in System 7.0, the **Window Manager** and **Dialog Manager** have not been enhanced to support multiple simultaneous scripts, so the system script is used for displaying filenames and directory names in dialog boxes, window titles, and other user interface elements used by the **Finder**. Therefore, until the system software's script capability is fully implemented, you should still treat these fields as reserved.