
How and When the Finder Launches Your Application

This section provides a brief summary of how the **Finder**-using the previously described resources-starts up your application whenever the user requests the **Finder** to launch your application or to open or print a document supported by your application.

The simplest scenarios under which the **Finder** launches your application occur when the user double-clicks your application icon or selects it and chooses Open from the **Finder**'s File menu. In these cases, the **Finder** calls the **Process Manager** to start your application. The **Process Manager** creates a partition of memory for your application, loads your code into this partition, and sets up the stack, heap, and A5 world for your application. The **Process Manager** returns control to the **Finder**, which then relinquishes control to your application. (See the **Process Manager** for more information.) Your application then performs the tasks necessary to open itself-such as opening an untitled document window, for example.

When the user requests the **Finder** to open or print a document supported by your application, the **Finder** calls the **Process Manager** and launches your application in the same way, except that the **Finder** also sets up the information your application needs to open or print the document and passes this information to your application. This information includes a list of files to open or print. Applications that do not support high-level events use the **CountAppFiles**, **GetAppFiles**, and **ClrAppFiles** procedures or the **GetAppParms** procedure to get this information. In System 7.0, applications that support high-level events receive this information through Apple events.

The user can request the **Finder** to open documents created by your application by double-clicking one of their icons, and the user can request the **Finder** to open or print documents by selecting one or more icons and choosing Open or Print from the **Finder**'s File menu. The **Finder** reads the creator field of each selected file to find the document's creator. Typically (as described in Finder Information in the Volume Catalog), your application sets the four-character string specified in its signature resource as the creator of its documents. The **Finder** searches for the application whose signature matches each document's creator. If the document's creator matches your application's signature, the **Finder** calls the **Process Manager**, launches your application, and then passes your application the name of the selected document or selected multiple documents. Your application should then open the documents in titled windows or print them, as appropriate.

If the user tries to open documents created by your application and your application is missing, the **Finder** displays an alert box telling the user that your application is missing. The **Finder** displays the name of your application in this alert box if you provide your documents with a string resource (of resource ID -16396) containing your application's name. (See **Messages When the Finder Can not Find Your Application**)

Sometimes when your application is already running, the user might double-click a document created by your application. If your application supports high-level events, the **Finder** sends your application the Open Documents event. If your application does not support high-level events, the **Process Manager** accommodates the user by simulating a mouse-down event that calls your application's menu command

for opening files. The **Process Manager** accomplishes this by first looking for a File menu with an item named Open. Since some applications do not have a File menu and since others use a command with a different name, the **Process Manager** then looks in the application's resource fork for 'mstr' and 'mst#' resources with resource IDs 102 and 103. An 'mstr' resource has the same format as an 'STR' resource. An 'mstr' resource with resource ID 102 should contain the name of the menu containing the Open command. An 'mstr' resource with resource ID 103 should contain the name of the menu item containing the Open command. An 'mst#' resource has the same format as an 'STR#' resource. An 'mst#' resource with resource ID 102 should contain the name or names of the menu or menus containing the Open command. An 'mst#' resource with resource ID 103 should contain the name or names of the menu item or items containing the Open command.

The user can also request the **Finder** to launch your application by dragging one icon or several icons to your application's icon. The **Finder** determines whether to launch your application by comparing the document's file type (which is stored in the volume catalog) against the list of your application's supported file types. The **Finder** compiles this list from the 'FREF' resources you create for your application; the **Finder** stores this list in the desktop database. If the document's file type appears in the 'FREF' list for your application, the **Finder** calls the **Process Manager**, launches your application, and passes it the name of the selected document or selected multiple documents. Your application should then open the documents in titled windows.

If your application supports the Open Documents event, you can also specify disks, folders, and a wildcard file type for all other files in your 'FREF' resources so that users can launch your application by dragging their icons to your application icon. After the **Finder** uses the **Process Manager** to launch an application that supports high-level events, the **Finder** sends the application an Open Documents event, which includes a list of alias records for desktop objects that the application should open. It is up to your application to open disks, folders, or all possible file types in a manner appropriate to the needs of the user.

To support stationery, your application should specify the isStationeryAware constant in its 'SIZE' resource and always check the isStationery bit of a document passed to it by the **Finder**. If the isStationery bit is set for a file that the user wants to open, your application should copy the contents of the stationery pad into a new document and open the document in an untitled window.

System 7.0 allows users to create aliases, which are desktop objects that represent other files, directories, or volumes. If the user opens an alias that represents a document created by your application, the **Finder** resolves the alias for you; that is, it passes your application the name and location of the document itself, not the alias. (Aliases are described in the next section.)