F6. Finder Resources

 Most programmers using FaceIt make use of a "temporary" res file that contains program-specific resources.   Near the end of a programming project, these program resources are moved from the temporary file to the program file (by copying and pasting using ResEdit or other resource editor).
   This is also a good time to add the additional Finder-related resources necessary for support of file icons, MultiFinder and System 7 features, and application file balloon help.   The reasons for adding these Finder-related resources last are (1) most Finder-related resources are not recognized by the Finder unless they are part of the program file itself (so it doesn't do you any good to put them in the temporary res file), and (2) once the Finder is made aware of the file icon resources, editing of these is made complicated by the need to keep the Finder's "Desktop" file updated.
   The addition of Finder-related resources to your finished program file is important if your program is to behave like other Macintosh applications.   The first step in installing Finder resources is to copy all of the resources from the file "Finder.Rsrc" (found in the "Finder Stuff" folder) to your program file:
 BNDL 128
 FREF 128,129,130
 ICN# 128,129,130
 mstr 100,101,102,103
 SIZE -1
 hfdr -5696
 STR 1001
The remainder of this topic describes the purpose of these resources and how to customize them to fit your needs.

Adding File Icons   (BNDL, FREF, ICN# resources)

The BNDL, FREF, and ICN# resources are the minimum resource types needed to display application and document file icons.   These resources can be edited as a group using ResEdit's "BNDL" editor which associates icons with file types.   The "Signature" should be set to your program's creator type (a 4-character string that uniquely identifies the program file and all document files created by it). When the BNDL editor is closed, it adds a new resource of this type with res ID = 0 to the resource file.
   If the "Extended View" of the BNDL editor is displayed, then a "© String" can be entered.   This string is typically used to supply version and copyright information, and is displayed in the "Version" field of the Finder's info dialog.   Alternatively, you can create two vers-type resources that supply more complete information about the program.   The vers resources should have res IDs 1 and 2, where vers 1 supplies the string for version and copyright info, and vers 2 supplies a string that describes a set of files that work together (System files, for example, have version numbers that can differ from the overall System release number).
   Double-clicking on an icon in the BNDL editor opens the "Icon Family" editor to edit the family of icons associated with a particular file type.   As you will see, ICN# icons are just one of several types of icons that can be supplied for each file type.   The other icons are used on color screens and when displaying smaller versions of the icon.   The ICN# icon, however, is the only icon used by System 6.0x.
   Finally, before quitting ResEdit, let the Finder know what you have done by checking the "Has BNDL" check box in the file's File Info window (opened from the "Get File/Folder Info" menu item in ResEdit), and entering the file's creator name that you chose when editing the BNDL.   When you return to the Finder you should find that your program file and document files are now displaying their new icons.
   If you get lost, take a look at how icons were added to the "fDemoLP+" example program, which was set up to display the default icons from the "Finder.Rsrc" file.
   NOTE: Changes made to existing icons do not appear on the desktop until the Finder's "Desktop" file (which contains a copy of your icons) is rebuilt.   To do this, hold the option and command keys down at startup, when inserting a disk, or when quitting to the Finder.   A nasty side effect of doing this is that it will destroy all Finder "Get Info" comments for all files (a problem that Apple had promised to fix in System 7 but did not).

Opening From Finder (mstr resources)

When the user saves files from within your program, window- or control-driving modules assign your

creator name and an appropriate file type to the document file so that the Finder knows what icon to display for the document file, and what program to launch when the user attempts to open or print the document from the desktop.   The technical details of supporting opening and printing such files "from the Finder" for all versions of the System (≥ 6.0.5) have become complex over time, but FaceIt's support for this feature hides such details and works the same across all System versions.

◦ Automatic Help

   When the user attempts to open or print a document from the Finder, FaceIt translates this attempt into a message returned from DoLoop that will be the same under all System versions:

   uMenuID = 1100
   uMenuItem = 512
   uResult = 1 (open) or 2 (print)
   uString = file type
   uName = file name
   uI2 = working directory reference number
   & current directory set to that containing the file

One such event is returned for each file being opened/printed.   The program can then do whatever is appropriate to open or print the file.   The fDemoXY example program contains an example of handling such events to support opening TEXT files from the Finder.

◦ Technical Details

   FaceIt's support for opening and printing files from the Finder hides from the program the complex interactions with the Finder that occur in different ways under different versions of the System.   In some cases, it is necessary for the programmer to know something about these details.

   First, if using System 7 and the "High level event aware" bit is set in the SIZE resource (described below), then the Finder posts separate Apple events for each file that the user attempts to open or print from the Finder.   FaceIt does the work of intercepting these events and translates them into simpler pseudo-menu events.   It also understands the quit and launch Apple events, making it System 7 "friendly" (i.e., it handles all 4 "required" events: Open Application, Open Document, Print Document, and Quit Application).

   Alternatively, if using System 6.0x or the "High level event aware" bit is not set in the SIZE resource, then the Finder uses a different scheme to inform the program when files need to be opened or printed. In this case, launching a program causes the Finder to store information about the launch in a special record that can be examined using the toolbox calls CountAppFiles and GetAppFiles.   FaceIt uses these calls to get info about the files to open or print from the Finder, and then translates this info into the simpler pseudo-menu events.

   If running under 6.0x (or not using AppleEvents) and the user attempts to open or print a file from the Finder when the application is already open, then yet another scheme is used by the Finder to notify the program.   In this case, the Finder searches for mstr-type resources that tell it which program menu items it should execute to open the file.   The default mstr resources from "Finder.Rsrc" are:

   mstr 100 = title of menu containing Quit item ("File")
   mstr 101 = text of item corresponding to Quit ("Quit")
   mstr 102 = title of menu containing Open item ("File")
   mstr 103 = text of item corresponding to Open ("Open...")

where the default text is shown in parentheses.   It doesn't really matter which item is associated with opening files (as long as it is one that is always installed and enabled), since FaceIt intercepts the Finder's attempt to execute the item and posts the correct message to your program.   Due to the existence of some System extensions that also make use of the mstr resources, however, it is best to use an item that is related to opening files.

 What To Do: Other than adding the mstr resources to the program file, the most you should have to do to ensure that pseudo-menu events are returned to your program when users attempt to open or print files from the Finder is to check that the mstr resources are associated with menu items that exist in your main menus.   Once this is done, test opening, printing, and quitting (i.e., restarting) from the Finder with the "High level event aware" flag set and not set, under System 7 and 6.0x.   The only difference you should see is that, under 6.0x, files cannot be printed from the Finder if the application is already open.

Size and Behavior   (SIZE resource)

   The SIZE resource copied from the "Finder.Rsrc" file (or added in some other way) is required for your

program to operate properly under MultiFinder and System 7.   This resource specifies the size of the partition to use, and other Finder-related aspects of program operation.   Due to the now common use of MultiFinder or System 7, most of our demo projects are configured to automatically add a SIZE resource when program files are created.

   When your program is launched under MultiFinder or System 7, the Finder searches for a SIZE resource.   If no SIZE resource is found, then default settings are used that are unlikely to optimize your program's operation.   The default partition size, in particular, will often be too small for programs in which on-line editing of ViewIt windows is to occur.

   When viewing a SIZE resource with ResEdit, the element "Size" is the preferred number of bytes, and the "Min size" is the minimum number of bytes required to run your program under MultiFinder and System 7.   The memory required to run your program will, of course, depend upon the memory used by your program plus the memory used by the FaceWare modules that it calls.   A simple way to determine the minimum partition size is to set aside more than enough memory when running under MultiFinder, and then check the free heap memory via a PeekCt control in a ViewIt window.   Remember that the heap is where new text, pictures, etc., get allocated, so never leave less than 32K or so of free heap memory, preferably much more.

For MultiFinder (or System 7) and FaceIt to interact optimally, the bit flags set in the SIZE resource should include the following (see IM V6 for more info):
 ◦ Accept suspend events
 ◦ Does activate on FG switch (is MultiFinder-aware)
 • 32 Bit Compatible
 • High-level event aware
In addition (as described in the "Background Processing" topic), programs that support background processing and switching should set an additional bit flag:
 ◦ Can background (receives background null events)

Program File Help   (hfdr, STR resources)

   The presence of the hfdr -5696 resource (don't change this ID number!) in a finished application file causes the Finder to use the text in an associated STR resource as the balloon help text whenever the cursor is moved over the application icon and balloon help is enabled.   The last 2 bytes in the hfdr resource contain the ID number of the associated STR resource (in this case 1001).   Thus you can simply change the STR's text to cause the Finder to display information about your application.   The default text we have entered in STR 1001 simply says that the application uses FaceWare modules, so you should change this to text that describes your application.