8. Resource Files

FaceWare makes extensive use of "resources" to simplify program code. Resources are independent parts of your program that can often be edited by users to suit their needs or preferences. The more of your program that resides in resources, for example, the easier it will be to convert it for use with other languages. Resources include such things as templates for cursors, menus, windows, and dialogs.

A good way to create and edit common resource types is with the program ResEdit, a copy of which, licensed from Apple, is shipped with all FaceWare modules. A good way to learn to use ResEdit is via the book ResEdit Complete by Peter Alley and Carolyn Strange. Another way is to simply experiment using ResEdit to edit common resource types within copies of the ".Rsrc" files shipped with FaceWare modules. The most common types of resources that you'll use ResEdit to edit are those used with menus, cursors, and controls in ViewIt windows:

STR# - list of Pascal strings (any size)
TEXT - block of plain text (any size)
MENU - template used when creating new menus
PICT - picture (any size)
ICON - icon (32 x 32 bits)
SICN - list of small icons (16 x 16 bits each)
cicn - color icon (any size)
PAT - pattern (8 x 8 bits)
PAT# - list of patterns (8 x 8 bits each)
CURS - cursor (16 x 16 bits)
acur - list of CURS resources (16 x 16 bits each)
clut - list of RGB colors (3 x 2-byte integers each)

SIZE Resource

For proper operation under MultiFinder and System ≥ 7.0, compiled programs must contain a SIZE resource. All of our demo program projects have been configured to include such a SIZE resource. To learn more about SIZE resources, see the "Finder Resources" topic in the FaceIt Guide. As described in that topic, at least 4 of the bit flags in the SIZE resource must be set for proper operation under MultiFinder and System 7:

◦ Accept suspend events
◦ Does activate on FG switch (is MultiFinder-aware)
• 32 Bit Compatible
• High-level event aware

In addition, the partition size should be set to at least 512K, but the optimum value will depend on the modules in use, the size of the program, and other program-specific factors.

Shared vs. Program-Specific

Resources are stored in files on disk. Finished programs typically contain within themselves most of the resources that they require. During development, however, it often makes sense to keep resources in files separate from the main program file so that the program can be recompiled without affecting these resources and without requiring that time be spent recombining them with the program file.

When using FaceIt and ViewIt, we recommend making use of 2 different external resource files during development: the "FaceWare" file containing modules that can be shared between programs, and a file with the extension ".Rsrc" containing resources that are program-specific (such as windows and menus used by that program). The ".Rsrc" files are also referred to as "temporary" or "supplemental" resource files.

Temporary Res Files

A new ".Rsrc" temporary resource file can be created by duplicating and renaming one of the ".Rsrc" files associated with a demo program or the file named "Minimum.Rsrc". The "Minimum.Rsrc" file contains resources typically used by programs using FaceIt. These resources consist of the LoadIt module (required) that is used to load FaceWare modules, plus other resources used by FaceIt (optional):

LoadIt Resources (required)
ALRT 1000

DITL 1000
FCMD 1000
FCNT 1000, 1001
FaceIt-Related Resources (optional)
MENU 1001-1004
where the ALRT and DITL resources are accessed by name so that, if necessary, you can renumber these to avoid conflicts with existing resource ID numbers.

When DoInit (described in next topic) is called at the start of your program, the FaceIt dispatching procedure looks for the LoadIt module (FCMD 1000) that is required to call FaceWare modules. If LoadIt cannot be found in the program file, then it trys to open the temporary resource file whose name has been placed in uName. If this res file cannot be found or does not contain LoadIt, then the FaceIt procedure simply quits back to the Finder. This usually occurs if the ".Rsrc" file is not in the same folder as the program file, or if the file name passed in uName does not exactly match the resource file name.

When finished with the development of a program, the modules used by it in the FaceWare file can be moved to the program file with MoveIt, and resources from its ".Rsrc" file can be moved to the program using ResEdit or other resource editor to produce a stand-alone application.

More About LoadIt

Since FaceWare modules are never linked to program code, something must do the work of finding such modules when they are called, loading them, and jumping to them. This work is done by the LoadIt module. To work properly, LoadIt must be made a part of your program file or other resource file opened only by your program (it cannot be shared). This explains why we recommend placing a copy of LoadIt within the ".Rsrc" files used by programs during development. The addition of LoadIt makes your program "FaceWare compatible" in the sense that it can then call any other FaceWare module.

The FaceWare File

The "FaceWare" file is similar to the Macintosh System file. It contains resources that can be shared by programs and is usually placed in the System folder where it can be found by programs in different folders.

The default FaceWare file shipped with FaceWare modules on the FaceWare Utilities disk (not in shareware version) contains ViewIt, FaceIt, and a basic set of control-drivers. Other modules can be added to this file using MoveIt. All demo programs accompanying FaceWare modules make use of the modules in this file to keep the demos simple and to prevent them from becoming confused with hundreds of lines of code devoted to event handling and window management.

A second, larger version of the FaceWare file is shipped with the ViewIt product (and is part of shareware version). This version contains additional on-line help resources and editing code to support ViewIt's on-line editing. If using ViewIt, you will probably make use of this larger version of the FaceWare file during development. When finished with development, you can then move modules from the smaller version of the FaceWare file to your program file to create a stand-alone application that does not contain editing-related resources.

Finally, note that the size of the FaceWare file has nothing to do with the size of your finished application. This file contains modules that you may never use, resources within modules that may not be needed, and help text that will not be needed in finished programs. Keeping all this stuff in one file is just an efficient way to work with it during development: only the resources used by a program are loaded into memory, and only one copy of each resource is present on disk.

Resource Numbering

To avoid any conflicts between resources used by the main program and FaceWare modules, the range of resource ID numbers from 1100 to 7499 is reserved for modules. Program resources should be numbered less than 1100 or greater than 7499 (we typically use resource IDs in the range of 1000 to 1099 for demo program resources).

The resource ID of the FCMD-type resource accompanying each module is referred to as the "baseID" of the module (1100 = FaceIt, 1200 = ViewIt, etc.). Each module also has a "versID" (version ID) that is equal to 10x the module's version number (version 2.2 = versID 22). Thus each FaceWare module can be uniquely identified by specifying its baseID and versID.