

Demonstration Program Folders and Files

The demonstration programs were written using Metrowerks CodeWarrior for Mac OS, Professional Edition, Version 6.2. Resource files were created using Resorcerer 2.2 and 2.4.

Document files utilised by certain demonstration programs are located in the folder **Mac C Carbon Demo Documents**, which is located in the folder **Mac C Carbon Demos**. The first word in each document file title is the name of the demonstration program which the file supports.

For Mac OS 8/9, you should simply copy the folder **Mac C Carbon Demos** to a convenient location on your hard drive. For Mac OS X, you should copy the folder titled **Mac C Carbon Demos** to the Applications folder and the folder titled **Mac C Carbon Demo Documents** to the relevant user's Documents folder.

The **Menus2** demonstration program incorporates a demonstration of Apple Help, including the methodology used to create an item in the Mac OS 8/9 Help menu. The Apple Guide file titled **Menus Guide**, which will cause a "Menus Help" item to be created in the Mac OS 8/9 Help menu, should be retained in the folder Menus2. That folder also contains an alias of the folder **Menus Help**. The alias should be placed:

- In the Help folder in the System Folder on Mac OS 8/9.
- In the user's Help folder (~/Library/Documentation/Help) on Mac OS X.

You will then be able to access the help content by choosing Menus Help from the Menus2 program's Help menu.

Note that the prefix file, which appears by default in the Prefix File edit text item in the Settings dialog box (Language Settings/C/C++ Language section) has been deleted in all CodeWarrior projects.

Running the Programs

Before running a demonstration program, you should read the top section of the source code comments in the relevant chapter in the book (or, for the post-Chapter 17 Classic event model versions, in the PDF files located in the relevant demo program folder). For most programs, this explains what to do, what to expect, and what to note.

All of the demonstration programs may be run from within CodeWarrior with the exception of the program AppleEvents (Chapter 10). By its nature, this

program must be run as a built (that is, double-clickable) application. The demonstration program Files (Chapter 18) may be run within CodeWarrior, although certain aspects of the programs can only be explored by running it as a built application.

Note that the human interface guidelines for Mac OS 8/9 are inconsistent with those for Mac OS X when it comes to such matters as, for example, the required spacing between controls in dialogs. The demonstration programs observe the Mac OS X (Aqua) guidelines where these inconsistencies occur.

IDE and System Preparation

If the version of CodeWarrior you are using includes a version of the Universal Headers earlier than Version 3.4 and/or the version of CarbonLib in your Mac OS 8/9 Extensions folder is earlier than 1.4 (or does not exist), you should download the CarbonLib SDK (Software Development Kit) Version 1.4GM (or later) from <http://developer.apple.com/carbon/> and:

- Replace the existing folder **CIncludes** in Metrowerks CodeWarrior / Mac OS Support / Universal / Interfaces with the one in the Carbon Support folder in the SDK folder.
- Copy the extension CarbonLib from the Implementation Libraries folder in the SDK folder to the Mac OS 8/9 Extensions folder (replacing the existing version if it exists) and restart your computer.
- Copy the two stub libraries **CarbonLib** and **CarbonFrameWorkLib** from the CarbonLib folder in the SDK folder to Metrowerks CodeWarrior / Mac OS Support / Universal / Libraries / StubLibraries, replacing the existing versions if they exist.

Known Issues on Mac OS 8/9 (CarbonLib 1.4)

As of CarbonLib Version 1.4, the only known issue affecting the demonstration programs has to do with the system-managed Window menu in applications using the Carbon event model. In applications which use the Classic event model (e.g., the demonstration programs Windows2 and MLTETextEditor), the second item in the Window menu toggles between "Collapse Window" and "Uncollapse Window". However, in applications which use the Carbon event model (e.g., the demonstration program CarbonEvents2, "Collapse Window" and "Uncollapse Window" are two separate menu items.

Known Issues on Mac OS X 10.0.4

As of Mac OS X 10.0.4, known issues are as follows:

- In Carbon applications, there is a problem with window-modal dialogs and alerts (sheets) causing applications to occasionally crash when the sheet is invoked. This affects the demonstration programs DialogsAndAlerts, CarbonEvents2, and Files. Also, in Carbon applications, sheets are not translucent. (It is understood that both of these issues will be resolved in a future Mac OS X release.)
- There is no support for Help tags for menus.
- In the demonstration programs AppleEvents, Files, and MLTETextEditor, the version information in the 'plst' resource is not used by the Finder. Instead the Finder uses the information in the 'vers' resource. (It is understood that this will not occur on Mac OS X 10.1.)
- The titles of pop-up menu button controls do not dim when the controls are activated.
- In the demonstration program GWorldPicCurslcn, the function GetPictInfo, called in the function doPicture, returns incorrect information regarding the number of rectangles, ovals, strings, etc.
- In the demonstration program Controls3, scrolling of the auto-scrolling text box is very slow. This is because this control does not yet have a built-in timer.
- In the demonstration programs CarbonEvents2 and MonoTextEdit, the calls to QuitAppModalLoopForWindow will not re-enable the menus disabled by the associated calls to RunAppModalLoopForWindow. It is expected that this problem will not occur on Mac OS X 10.1.x. For versions of Mac OS X earlier than 10.1.x, you can eliminate the problem by making the following changes:
 - In the demonstration program CarbonEvents2, in the source code file Dialogs.c, move the call to DisposeWindow from immediately after the call to QuitAppModalLoopForWindow in the function dialogEventHandler to immediately after the call to RunAppModalLoopForWindow in the function doMovableModalDialog.
 - In the demonstration program MonoTextEdit (Carbon event model version), in the source code file HelpDialog.c, move the calls to doCloseHelp and doDisposeDescriptors from immediately after the call to QuitAppModalLoopForWindow in the function

helpWindowEventHandler to immediately after the call to RunAppModalLoopForWindow in the function doHelp.

- In the demonstration program MLTETextEditor, scrolling of text is jerky and erratic.

Notes on 'plist' Resources

Of necessity, all demonstration programs contain a 'plist' resource. The following notes are additional to the information on 'plist' resources contained in the book.

Resorcerer Version 2.4 is provided in two versions:

- Resorcerer 2.4 PPC, the "Classic" version which you should use if you are running Mac OS 8/9.
- Resorcerer 2.4 OSX, the "Carbonised" version which you should use if you are running Mac OS X, and which you may also use, with certain limitations, if you are running on Mac OS 8/9.

Resorcerer 2.4 PPC has a bug which prevents you from opening 'plist' resources. Accordingly, to open and edit 'plist' resources, use Resorcerer 2.4 OSX.

As stated at Chapter 10, you can create a complete 'plist' resource by simply typing the information property list in a text editor and copying and pasting the listing into an empty 'plist' resource using Resorcerer. To check the construction of the listing, you can save it to a text editor file and, on Mac OS X, start the Terminal application and use the "pl" utility to check the listing by entering:

```
pl <name of file>
```

As an alternative method of creating and editing an information property list, you can use the PropertyListEditor application located in Mac OS X's Developer folder.

When you make changes to a 'plist' resource, you may need to perform the Mac OS X equivalent of rebuilding the desktop database in order for the changes to be recognized by the Desktop. To do this, open the user's folder and, in the directory ~/Library/Preferences, delete the files LSApplications, LSClaimedTypes, and LSSchemes. Then log out and log back in.

