

CARBON PROGRAMMING
UPDATES AND CORRECTIONS — JANUARY 2002

CONTENTS

Errata — First Printing	2
Demonstration Program Bug Fixes and Additions	9

ERRATA — FIRST PRINTING

If your copy of Carbon Programming is from the first printing, the following corrections are relevant.

Chapter 1, Page 22

The table on this page has been corrupted. This is the version supplied by the author:

Type	ID	Attributes			Handle
		Preload	Lock	Purgeable	
MENU	128	•			123C
WIND	128			•	NULL
PICT	128			•	NULL
PICT	129			•	NULL

Chapter 3, Page 96

The note appearing on this page should not be there. (It is a duplication of the note appearing on page 105.)

Chapter 4, Fig 4

One window (bottom right) has been deleted from Fig 4. This is the version supplied by the author:



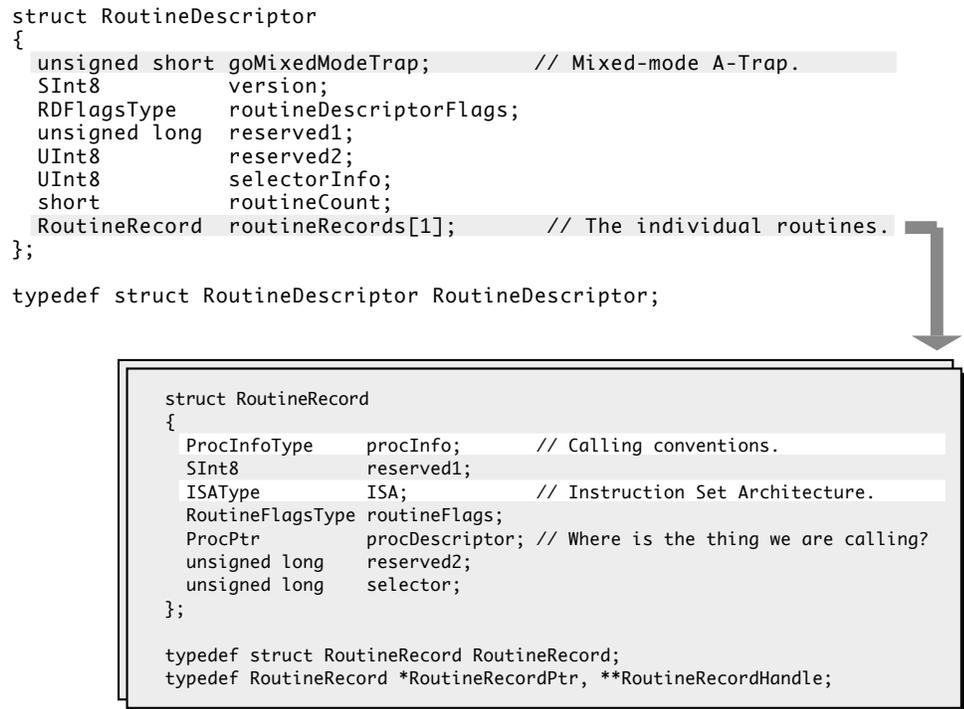
FIG 4 - WINDOW TYPES FOR FLOATING WINDOWS (TITLE BAR AT TOP)

Chapter 5

This figure was supplied by the author, but is missing from Chapter 5. It should have appeared at the top of page 215.

```
struct RoutineDescriptor
{
    unsigned short goMixedModeTrap; // Mixed-mode A-Trap.
    SInt8 version;
    RFlagsType routineDescriptorFlags;
    unsigned long reserved1;
    UInt8 reserved2;
    UInt8 selectorInfo;
    short routineCount;
    RoutineRecord routineRecords[1]; // The individual routines.
};

typedef struct RoutineDescriptor RoutineDescriptor;
```



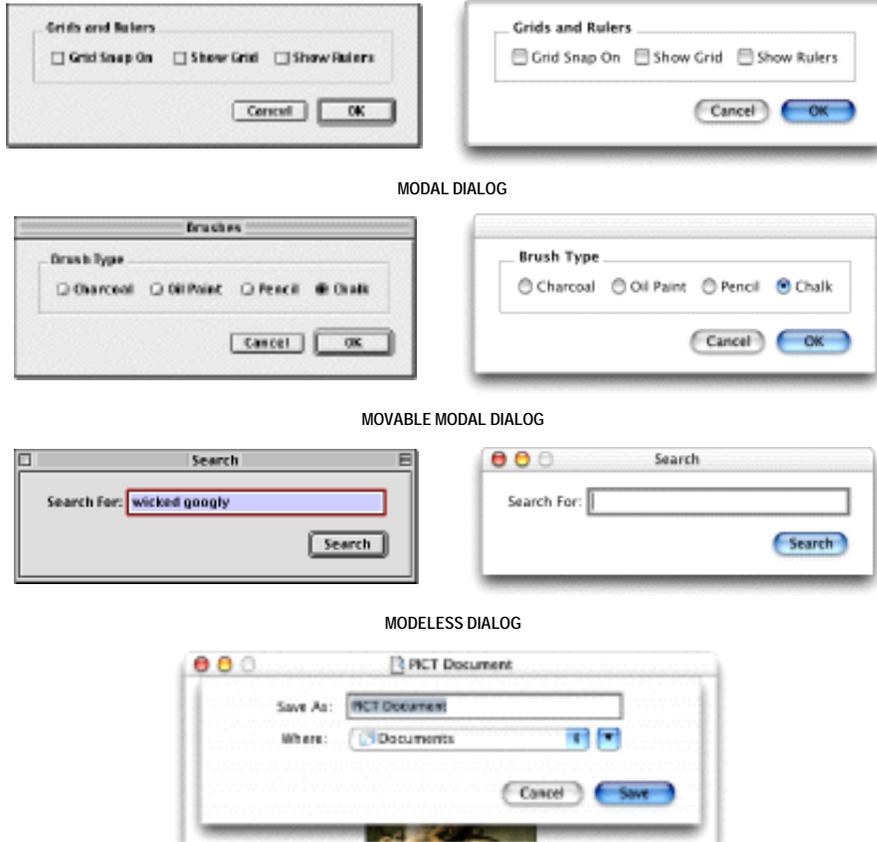
```
struct RoutineRecord
{
    ProcInfoType procInfo; // Calling conventions.
    SInt8 reserved1;
    ISAType ISA; // Instruction Set Architecture.
    RoutineFlagsType routineFlags;
    ProcPtr procDescriptor; // Where is the thing we are calling?
    unsigned long reserved2;
    unsigned long selector;
};

typedef struct RoutineRecord RoutineRecord;
typedef RoutineRecord *RoutineRecordPtr, **RoutineRecordHandle;
```

FIG 1 - THE ROUTINE DESCRIPTOR STRUCTURE AND A ROUTINE STRUCTURE

Chapter 8, Fig 2

The bottom (sheet) dialog and figure title is missing. This is the figure supplied by the author:



WINDOW-MODAL (SHEET) DIALOG

FIG 2 - TYPES OF DIALOGS

Chapter 11, Fig 7

The author supplied the correct Fig 7. This was replaced in the book by a duplication of Fig 11. Here is the correct Fig 7 as supplied by the author:

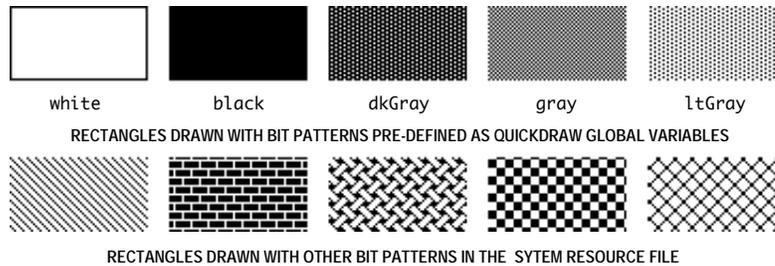


FIG 7 - RECTANGLES DRAWN USING BIT PATTERNS IN THE SYSTEM RESOURCE FILE

Chapter 12, Fig 5

This figure has been corrupted. This is the figure as supplied by the author:

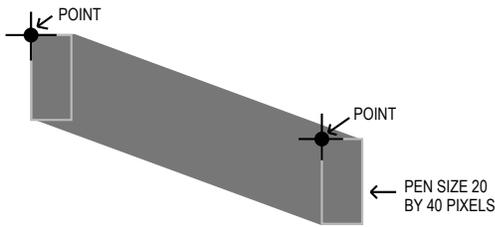


FIG 5 - A LINE DRAWN BY LineTo OR Line

Chapter 12, Fig 6

This figure has been corrupted. This is the figure as supplied by the author:

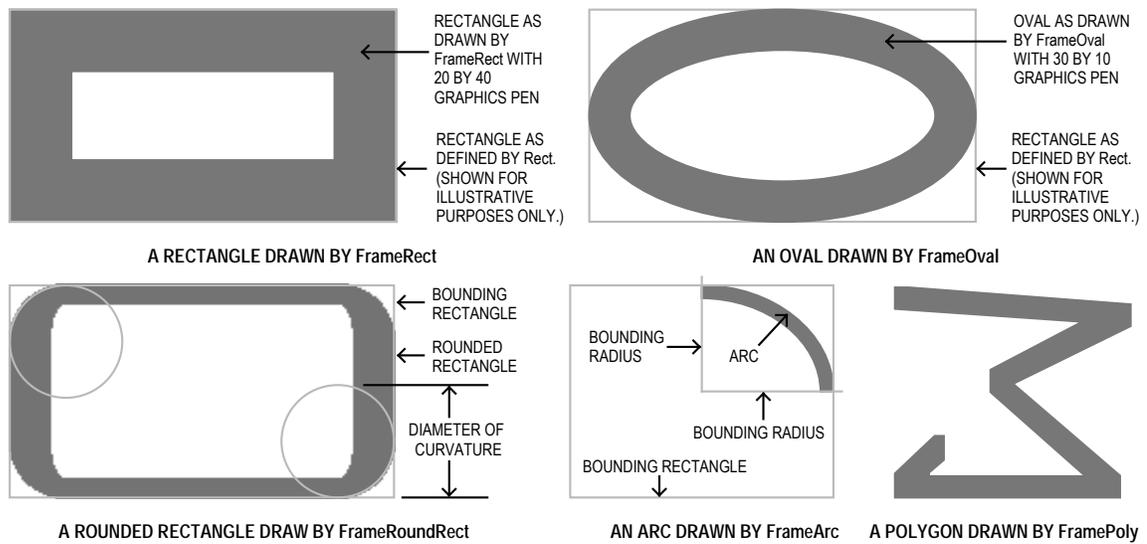


FIG 6 - FRAMED SHAPES DRAWN WITH QUICKDRAW FRAMED SHAPE DRAWING FUNCTIONS

Chapter 12, Fig 7

This figure has been truncated at left. This is the figure as supplied by the author:

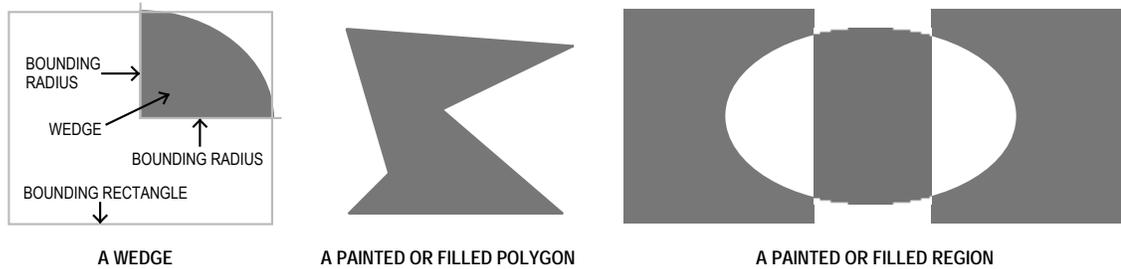


FIG 7 - A WEDGE, A PAINTED OR FILLED POLYGON, AND A PAINTED OR FILLED REGION

Chapter 18, Fig 4

This figure has been corrupted. This is the figure as supplied by the author:

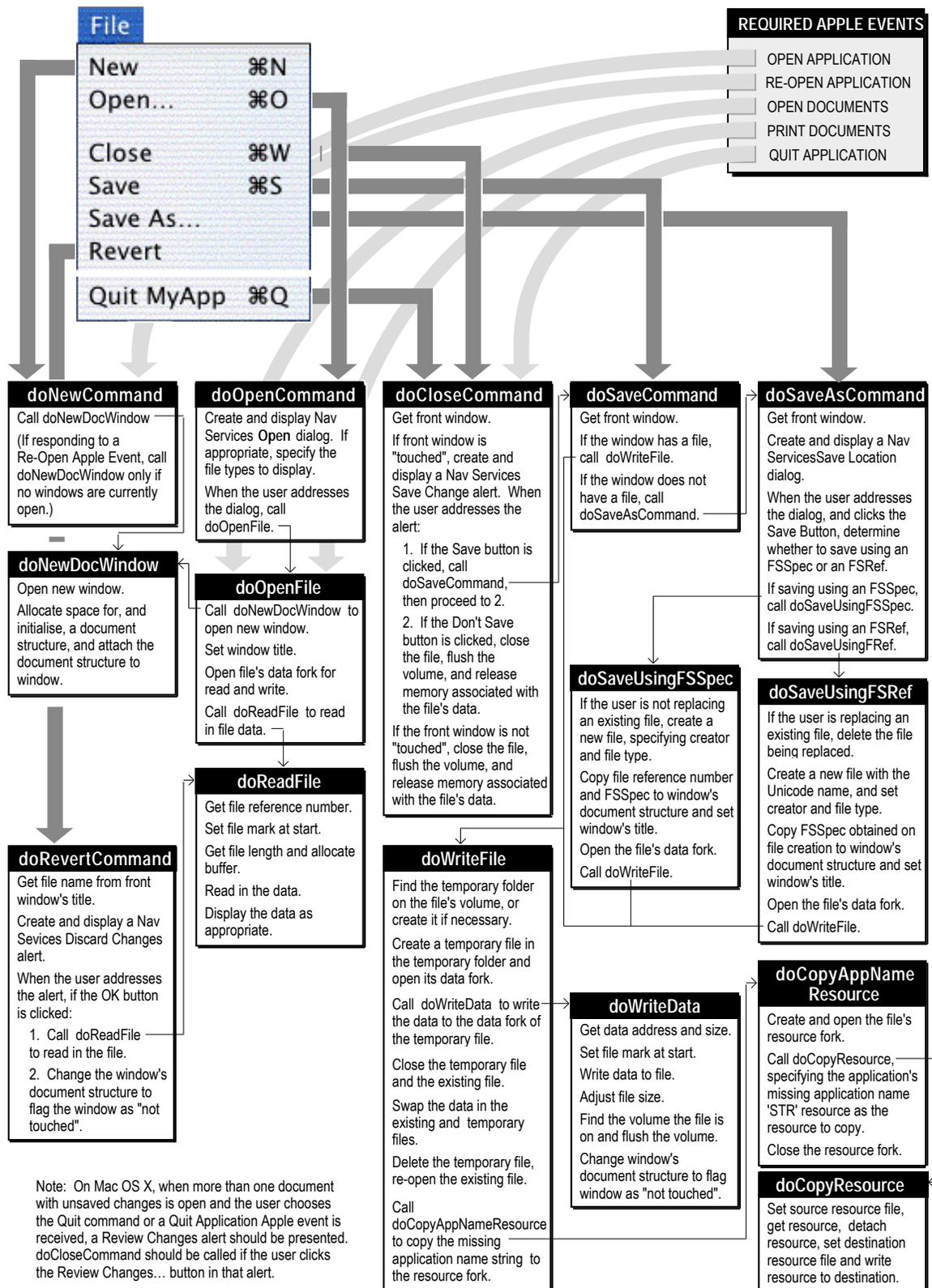


FIG 4 - GENERAL FILE MENU, QUIT ITEM, AND REQUIRED APPLE EVENTS HANDLING STRATEGY

Chapter 24, Fig 1, Fig 2, and Fig 3

Although referred to in the text, all supplied figures were omitted from this chapter. Here are the figures as supplied by the author:

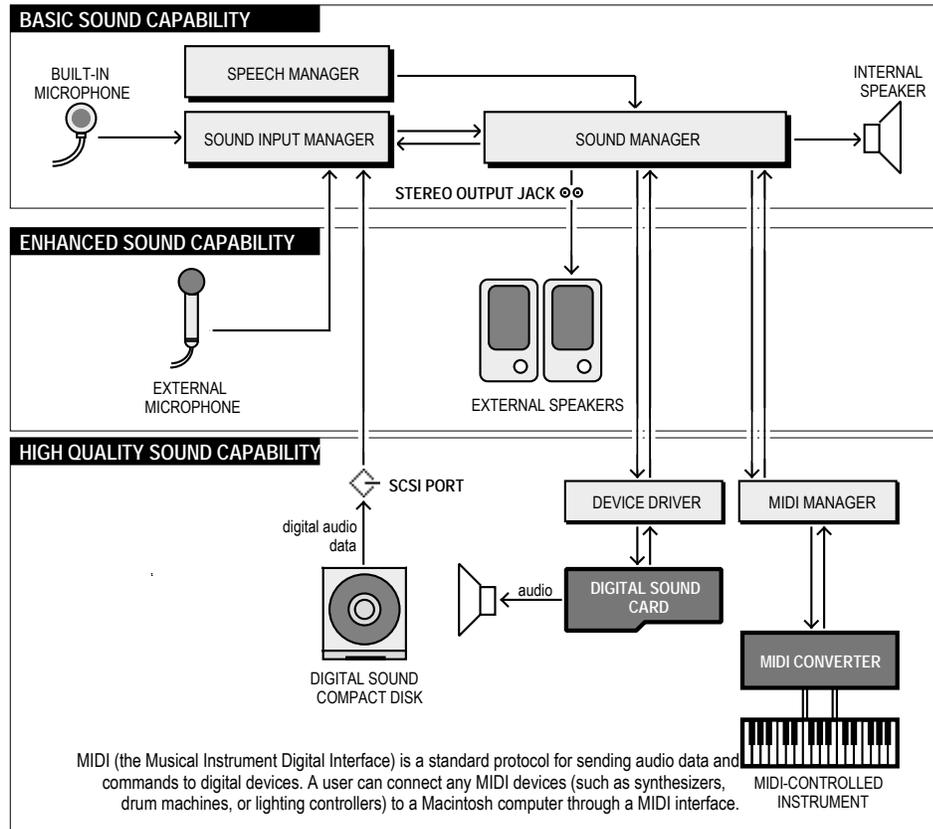


FIG 1 - SOUND CAPABILITIES OF MACINTOSH COMPUTERS

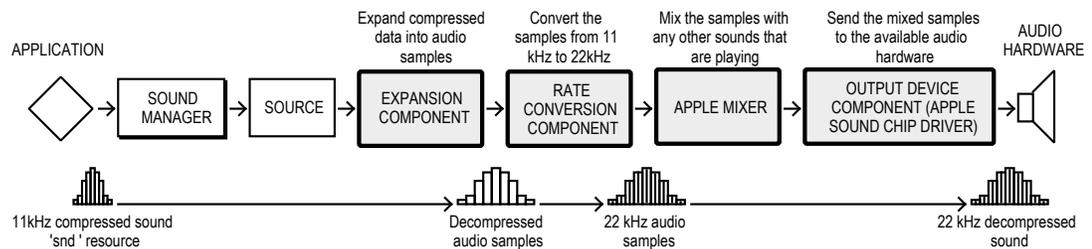


FIG 2 - A TYPICAL SOUND COMPONENT CHAIN



FIG 3 - SOUND RECORDING DIALOG — MAC OS 8/9

General

In addition to the above, there are a great many words (constant, data type, and function names, etc.) in the Demonstration Program Comments sections of all chapters which should have appeared in the monospaced font.

The author apologizes to purchasers for the errors in the first printing of Carbon Programming. These errors arose because the author was not availed of the opportunity to review and correct a pre-printing copy before the presses rolled.

DEMONSTRATION PROGRAM BUG FIXES AND ADDITIONS

Demonstration Program Menus1

When a sub-menu item is chosen in the (hierarchical) Font menu, the method used to determine the item number of the menu item to which the sub-menu is attached is to walk the menu items looking for a match between the font name returned by `GetFontName` and each menu item's text. (See the `for` loop and the call to `EqualString`.) When a match is found, that menu item is assigned a "dash" mark.

This method breaks on Mac OS X when certain Mac OS X fonts (e.g., American TypeWriter) are chosen. The reason for this is that, in the case of certain Mac OS X fonts, the menu item text does not necessarily equate to the font name returned by `GetFontName`. The consequence of this is that the "dash" mark will not be drawn against the relevant item.

The following replacement for the `doFontMenu` function finds the menu item by walking the menu items and comparing the menu reference obtained at the first line of the `doFontMenu` function with the menu reference of each item's sub-menu, as obtained by the call to `GetMenuItemHierarchicalMenu`. When a match is found, that menu item is assigned the "dash" mark.

```
void doFontMenu(MenuID menuID,MenuItemIndex menuItem)
{
    MenuRef      menuRef, fontMenuRef, subMenuRef;
    OSStatus     osError;
    FMFontFamily fontFamily;
    FMFontStyle  fontStyle;
    Str255       fontName, styleName;
    SInt16       a, numberOfFontMenuItems;

    menuRef = GetMenuRef(menuID);

    osError = GetFontFamilyFromMenuSelection(menuRef,menuItem,&fontFamily,&fontStyle);
    if(osError == noErr)
    {
        TextFont(fontFamily);
        TextFace(fontStyle);

        if(menuID == mFont)
        {
            CheckMenuItem(menuRef,gCurrentFontMenuItem, false);
            gCurrentFontMenuItem = menuItem;
            CheckMenuItem(menuRef,gCurrentFontMenuItem, true);

            if(gCurrentFontSubMenuRef != NULL)
                CheckMenuItem(gCurrentFontSubMenuRef,gCurrentFontSubMenuItem, false);

            GetMenuItemText(menuRef,menuItem, fontName);
            drawItemString(fontName);
        }
        else
        {
            if(gCurrentFontSubMenuRef != NULL)
                CheckMenuItem(gCurrentFontSubMenuRef,gCurrentFontSubMenuItem, false);
            gCurrentFontSubMenuRef = menuRef;
            gCurrentFontSubMenuItem = menuItem;
            CheckMenuItem(gCurrentFontSubMenuRef,gCurrentFontSubMenuItem, true);

            fontMenuRef = GetMenuRef(mFont);
            CheckMenuItem(fontMenuRef,gCurrentFontMenuItem, false);
        }
    }
}
```

```

numberOfFontMenuItems = CountMenuItem(fontMenuRef);
for(a=1;a<=numberOfFontMenuItems;a++)
{
    GetMenuItemHierarchicalMenu(fontMenuRef,a,&subMenuRef);           //// MAIN CHANGE

    if(subMenuRef == menuRef)
    {
        gCurrentFontMenuItem = a;
        break;
    }
}

SetItemMark(fontMenuRef,gCurrentFontMenuItem,'-');

GetMenuItemText(fontMenuRef,a,fontName);
drawItemString(fontName);
GetMenuItemText(menuRef,menuItem,styleName);
DrawString("\p ");
DrawString(styleName);
}
}
else
    ExitToShell();
}

```

Demonstration Program PreQuickDraw

On Mac OS X 10.0 and later, if "Set Pixel Depth 8 (256 Colours)" is chosen from the Demonstration menu, subsequent choices of the other menu items in the Demonstration menu will not change the pixel depth. Accordingly:

- Add this global variable:

```
Boolean gRunningOnX = false;           //// ADD THIS LINE
```

- In the main function, add the indicated lines to disable the "Set Pixel Depth 8 (256 Colours)" menu item if the program is running on Mac OS X:

```

Gestalt(gestaltMenuMgrAttr,&response);
if(response & gestaltMenuMgrAquaLayoutMask)
{
    menuRef = GetMenuItem(mFile);
    if(menuRef != NULL)
    {
        DeleteMenuItem(menuRef,iQuit);
        DeleteMenuItem(menuRef,iQuit - 1);
        DisableMenuItem(menuRef,0);
    }

    menununRef = GetMenuItem(mDemonstration);           //// ADD THIS LINE
    DisableMenuItem(menuRef,iSetDepth8);               //// ADD THIS LINE

    gRunningOnX = true;                                 //// ADD THIS LINE
}

```

On Mac OS X 10.0 and later, this section in the function doDisplayInformation is irrelevant and should be deleted:

```

MoveTo(10,320);
DrawString("\pOn Mac OS X, drag window after pixel depth and screen resolution changes to");
DrawString("\p ensure that");
MoveTo(10,333);
DrawString("\pbytes per row, pixel image bytes, and colour values are updated.");

```

On Mac OS X 10.0 and later, the color table aspects of the demonstration are not relevant. Accordingly, add the indicated lines to this section in the function `doDisplayInformation`:

```
// ..... if any entries in colour table, draw the colours

MoveTo(250,20);
DrawString("\pColour table:");

entries = (*colorTableHdl)->ctSize;

if(gRunningOnX)                                     //// ADD THIS LINE
{
    MoveTo(290,130);                                //// ADD THIS LINE
    DrawString("\pNot relevant on Mac OS X");        //// ADD THIS LINE
}                                                    //// ADD THIS LINE
else                                                //// ADD THIS LINE
{                                                    //// ADD THIS LINE
    if(entries < 2)
    {
        MoveTo(260,100);
        DrawString("\pOnly one (dummy) entry in the colour");
        MoveTo(260,115);
        DrawString("\ptable. To cause the colour table to be");
        MoveTo(260,130);
        DrawString("\pbuilt, set the monitor to bit depth 8");
        MoveTo(260,145);
        DrawString("\p(256 colours), causing it to act like ");
        MoveTo(260,160);
        DrawString("\pan indexed device.");
    }
}                                                    //// ADD THIS LINE
```

Demonstration Program Files (Carbon Events Version)

To avoid potential crashes on Mac OS X, the function `doSynchroniseFiles` should not be called during the application quitting sequence. Accordingly, in the source code file `Files.c`, in the function `doIdle`, add the indicated line:

```
void doIdle(void)
{
    if(GetWindowKind(FrontWindow()) == kApplicationWindowKind)
        if(!gQuittingApplication)                //// ADD THIS LINE
            doSynchroniseFiles();
}
}
```

Aqua Human Interface Guidelines state that all windows should come forward when the Review Changes button in a Review Changes alert is clicked. Accordingly, in the source code file `Files.c`, in the function `quitAppEventHandler`, add this local variable:

```
ProcessSerialNumber processSerialNumber;        //// ADD THIS LINE
```

and also add the indicated lines:

```
// ..... if more than one touched window, create Review Changes alert, handle button clicks

else if(touchedWindowsCount > 1)
{
    itemHit = doReviewChangesAlert(touchedWindowsCount);

    if(itemHit == kAlertStdAlertOKButton)
    {
        GetFrontProcess(&processSerialNumber);    //// ADD THIS LINE
        SetFrontProcess(&processSerialNumber);    //// ADD THIS LINE

        gQuittingApplication = true;
    }
}
```

On Mac OS 8/9 only, when the "Choose a Folder Dialog" item is chosen in the Demonstration menu, a folder is selected and the Choose button in the dialog is clicked, the folder name, volume reference number and parent directory ID are *not* drawn in the frontmost window. Accordingly, in the source code file ChooseAFolderDialog.c:

- Add this global variable:

```
extern Boolean gRunningOnX;          //// ADD THIS LINE
```

- In the function doChooseAFolderDialog, add the indicated lines:

```
if(osError == noErr)
{
    if(gRunningOnX)                //// ADD THIS LINE
    {                               //// ADD THIS LINE
        if((osError = GetSheetWindowParent(FrontWindow(),&parentWindowRef)) == noErr)
            windowRef = parentWindowRef;
        else
            windowRef = FrontWindow();
    }                               //// ADD THIS LINE
    else                             //// ADD THIS LINE
        windowRef = FrontWindow();  //// ADD THIS LINE

    GetIndString(message,rMiscStrings,sChooseAFolder);
}
```

Demonstration Program Files (Classic Events Version)

On Mac OS X, the application will crash during the quitting application sequence when Save Changes sheets are present. To eliminate this, prevent the function doSynchroniseFiles from being called when the quitting application sequence is in progress. In the source code file Files.c, in the function eventLoop, add the indicated lines:

```
while(!gDone)
{
    if(WaitNextEvent(everyEvent,&eventStructure,180,NULL))
        doEvents(&eventStructure);
    else
    {
        if(!gQuittingApplication)  //// ADD THIS LINE
            doSynchroniseFiles();  //// ADD THIS LINE
    }                               //// ADD THIS LINE

    if(gRunningOnX && gQuittingApplication)
```

On Mac OS 8/9 only, when the "Choose a Folder Dialog" item is chosen in the Demonstration menu, a folder is chosen and the Choose button in the dialog is clicked, the folder name, volume reference number, and parent directory ID are not drawn in the frontmost window. Accordingly, in the source code file ChooseAFolderDialog.c, make the same changes indicated for the Carbon Events version of this demonstration program (see above).

Demonstration Program DateTimeNumbers (Carbon Events Version)

Two mouse clicks, rather than one, are required to change the active edit text control. Accordingly, in the function `windowEventHandler`, add the indicated line:

```
case kEventClassMouse: // event class mouse
    ConvertEventRefToEventRecord(eventRef,&eventRecord);
    switch(eventKind)
    {
        case kEventMouseDown:
            if(IsDialogEvent(&eventRecord))
            {
                if(DialogSelect(&eventRecord,&gDialogRef,&itemHit))
                {
                    if(itemHit == iButtonEnter)
                    {
                        doAcceptNewRecord();
                        doClearAllFields();
                    }
                    else if(itemHit == iButtonClear)
                        doClearAllFields();

                    result = noErr; ///// ADD THIS LINE
                }
            }
        doAdjustCursor(GetDialogWindow(gDialogRef));
        break;
    }
    break;
```

Demonstration Program Lists (Carbon Events and Classic Events Versions)

When these programs are compiled using CodeWarrior Pro 7 (but, curiously, not Pro 6), the custom list definition function is never called, as a result of which the icons fail to display in the list that utilises the custom list definition function.

In the source code file `WindowList.c`, in the function `doCreateIconList`, add the indicated line:

```
listRect.right = listRect.right - kScrollBarWidth;

listDefSpec.defType = kListDefProcPtr; ///// ADD THIS LINE
listDefSpec.u.userProc = listDefinitionFunctionUPP;
```

Demonstration Program Miscellany (Carbon Events and Classic Events Versions)

Mac OS X 10.1 supports help tags for menu items. Accordingly, to assign help tags to the items in the Demonstration menu:

- Add these five strings to the existing strings in the 'STR#' resource with ID 129:

```
Notification demonstration
Determinate progress indicator demonstration
Colour Picker demonstration
Multiple monitors drawing demonstration
Help tags demonstration
```

- In the header file `Miscellany.h`, add this function prototype:

```
void doHelpTagMenu (void); ///// ADD THIS LINE
```

- In the source code file `Miscellany.c`, in the function `main`, add this line after the call to the function `doHelpTagWindow`:

```
doHelpTagMenu(); ///// ADD THIS LINE
```

- Add this function at the bottom of the source code file HelpTag.c:

```
// ..... doHelpTagMenu

void doHelpTagMenu(void)
{
    Rect          hotRect;
    HMHelpContentRec helpContent;
    MenuRef       menuRef;

    memset(&helpContent,0,sizeof(helpContent));
    HMSetTagDelay(500);

    helpContent.version = kMacHelpVersion;
    helpContent.tagSide = kHMOutsideRightCenterAligned;
    menuRef = GetMenuRef(mDemonstration);

    helpContent.content[kHMMinimumContentIndex].contentType = kHMStringResContent;
    helpContent.content[kHMMinimumContentIndex].u.tagStringRes.hmmResID = 129;
    helpContent.content[kHMMinimumContentIndex].u.tagStringRes.hmmIndex = 5;
    HMSetMenuItemHelpContent(menuRef,1,&helpContent);
    helpContent.content[kHMMinimumContentIndex].u.tagStringRes.hmmIndex = 6;
    HMSetMenuItemHelpContent(menuRef,2,&helpContent);
    helpContent.content[kHMMinimumContentIndex].u.tagStringRes.hmmIndex = 7;
    HMSetMenuItemHelpContent(menuRef,3,&helpContent);
    helpContent.content[kHMMinimumContentIndex].u.tagStringRes.hmmIndex = 8;
    HMSetMenuItemHelpContent(menuRef,4,&helpContent);
    helpContent.content[kHMMinimumContentIndex].u.tagStringRes.hmmIndex = 9;
    HMSetMenuItemHelpContent(menuRef,5,&helpContent);
}
```