MACINTOSH C CARBON

A Hobbyist's Guide to Programming the Macintosh in C

Version 1.0

K. J. Bricknell

CONTENTS

Chapter 1	System Software, PowerPC Run-Time Environment, Managing Memory, and Resources
Chapter 2	The Classic Event Manager — Low Level and Operating System Events
Chapter 3	Menus
Chapter 4	Introduction to Windows
Chapter 5	Carbon and Universal Procedure Pointers
Chapter 6	The Appearance Manager
Chapter 7	Introduction to Controls
Chapter 8	Dialogs and Alerts
Chapter 9	The Finder and the Application
Chapter 10	Apple Events
Chapter 11	QuickDraw Preliminaries
Chapter 12	Drawing With QuickDraw
Chapter 13	Offscreen Graphics Worlds, Pictures, Cursors, and Icons
Chapter 14	More on Controls
Chapter 15	Carbon Printing
Chapter 16	More on Windows
Chapter 17	The Carbon Event Manager
Chapter 18	Files and Navigation Services
Chapter 19	More on Resources, and Core Foundation Preferences
Chapter 20	Carbon Scrap
Chapter 21	Text, TextEdit, Dates, Times, and Numbers
Chapter 22	Lists and Custom List Definition Functions
Chapter 23	Drag and Drop
Chapter 24	Basic Sound and Speech
Chapter 25	Miscellany
Chapter 26	More On Text - Multilingual Text Engine

PREFACE

The Carbon Application Programming Interface

Apple announced the intended introduction of a new operating system, to be known as Mac OS X, at the 1998 World Wide Developers Conference. Mac OS X, the first version of which was released on 24 March 2001, is not just another Mac OS update; it is a completely new operating system complete with "modern" operating system features such as pre-emptive multitasking and protected memory. It features a completely new user interface, called Aqua, whose appearance and behaviour differs significantly from that of the original Mac OS (represented in its latest, and no doubt last, incarnation by Mac OS 9).

Mac OS X runs on G3 and G4 PowerPC machines only, meaning that machines based on PowerPC 604 and 603 microprocessors must necessarily remain with Mac OS 9 and earlier. A large installed base of these latter machines will no doubt remain for many years to come. In addition, it is likely that many owners of machines capable of running Mac OS X will nonetheless remain with Mac OS 9 and earlier. In these circumstances, it was perceived as all but essential that programs written to take advantage of Mac OS X's advanced features also be capable of running on Mac OS 8 and 9 without modification. Fortunately, Apple has devised the means whereby this can be achieved, namely, the Carbon Application Programming Interface (API).

An API is basically a collection of system software routines, two examples being `GetNewCWindow` (which programmers call to create windows) and `FSpOpenDF` (which programmers call to open a file). Mac OS 9 includes a collection of more than 8,000 such system software routines, which are now known collectively as the Classic API. Apple determined that about 2000 of these APIs were incompatible with a modern operating system like Mac OS X. The remaining 6000 or so "clean" APIs were isolated and, together with many additions, included in the Carbon API.

It turns out, therefore, that experienced Macintosh programmers will not need to learn a completely new API in order to program for the completely new operating system that is Mac OS X. Any programmer familiar with the Classic API will be able to transition to the Carbon API with comparatively little effort. In addition, newcomers to Macintosh programming will not be required to learn two APIs in order to write applications intended to run native on both Mac OS X and Mac OS 8 and 9.[1]

A further advantage of Carbon is that existing applications written using the Classic API can be "carbonized" with but a fraction of the effort that would be required to completely rewrite them for Mac OS X using the Cocoa API.

This book, then, is for those who wish to learn to create applications that will, firstly, run on both Mac OS X and Mac OS 8 and 9 and, secondly, take advantage of Mac OS X's unique features when run on that system. It is intended as a reasonably comprehensive entry point to programming the Macintosh using the Carbon API.

[1] That said, there is another API, called Cocoa, that may be used to write programs for Mac OS X. Cocoa's advantage is that it allows applications to be written more quickly that is the case using the Carbon API; however, such applications will not run on Mac OS 8 and 9.

The First Task — Learn the C Language and CodeWarrior IDE

The main assumption made by this book is that you have already learned both the C language and the Metrowerks CodeWarrior IDE (Integrated Development Environment) (http://www.metrowerks.com/platforms/mactools/).[2]

The Macintosh C Carbon Phase

When you have learned the C language and the fundamentals of the CodeWarrior IDE, you are ready to open this book. As you move through this second phase of the journey, you will quickly discover that learning C and the IDE was by far the easiest part!

Essentially, this book covers all of the territory which, in the author's judgement, needs to be covered before you write your first serious application. This includes, for example, how to create and manage all elements of the user interface (menus, windows, controls, dialogs, alerts, lists, drag and drop, etc.), how to perform file input/output, how to print documents, how to draw text and graphics, and so on.

General Structure of Macintosh C Carbon

The general structure of most chapters in this book is the same: first comes the information, then a list of constants, data types and functions relevant to the subject of that chapter, then the source code listing of one or more demonstration programs related to the subject of that chapter, and, finally, line-by-line comments that explain the workings of the source code.

The book itself is supported by the demonstration program package, which contains the CodeWarrior project and source code files, and Resorcerer resource files, for all demonstration programs. The demonstration programs associated with Chapters 18 to 25 exist in two versions:

- The primary version, which utilises the Carbon event model introduced at Chapter 17, is the version whose source code listing and associated comments are included in the book itself.
- The secondary version utilises the Classic event model (see Chapter 2). The explanatory comments for this version are contained in PDF files included in the relevant demonstration program (Classic event model version) folders.

What You Will Need

System Software

For Mac OS 8 and 9, Macintosh C Carbon requires:

- CarbonLib 1.4 or later. (CarbonLib is the extension, located in the Extensions folder in the System folder, that implements the Carbon API on Mac OS 8 and 9.)
- Mac OS 8.6 or later. (CarbonLib 1.4 is not compatible with versions of Mac OS 8 earlier than 8.6.)

For Mac OS X, Version 10.0.3 or later is required. (The so-called frameworks that implement the Carbon API on Mac OS X are an integral part of Mac OS X and are installed when you install that system. Frameworks are like shared libraries on Mac OS 8/9.)

Development System

As previously stated, it is assumed that your development system will be Metrowerks CodeWarrior. CodeWarrior Pro 6 (or later) is itself a Carbon application, and is much to be preferred.

Resource Editor

A resource editor allows you to create resources for programs and files. The resource editor you will need is Resorcerer, a product of Mathemaesthetics (http://www.mathemaesthetics.com/index.html).

[2] CodeWarriorU (http://www.codewarrioru.com/CodeWarriorU) conducts on-line courses in the C language and the CodeWarrior IDE.

References and Other Sources of Information

References enable you to access information relating to the system software, and are quite indispensable. The Carbon Developer Documentation section and the Mac OS 8 and 9 Developer Documentation Function Index (http://developer.apple.com/techpubs/macosx/Carbon/carbon.html and http://developer.apple.com/techpubs/FunctionIndex/FunctionIndex.html) at Apple's WWW site are the most up-to-date sources of information available. You might also wish to subscribe to Apple's Carbon development mailing list (http://www.lists.apple.com/mailman/listinfo/carbon-development).

Programmer's Calculator

You will find a programmer's calculator very useful for converting between decimal, hexadecimal and binary values, the shareware program CalcWorks (http://sitelink.net/jbrochusoftware) being ideal for that purpose.

Icon Editor

Versions of Resorcerer earlier than Version 2.4 did not facilitate the creation of 32-bit icons or the large (128 by 128 pixel) thumbnail icons utilised by Mac OS X. If your version of Resorcerer is earlier than Version 2.4, the shareware program Iconographer (http://www.mscape.com/products/iconographer.html) is possibly the best option for the creation of these icons.

Universal Headers, CarbonLib, and Stub Libraries

The demonstration programs assume the use of Version 3.4.1 or later of the Universal Headers[3] and, as previously stated, CarbonLib 1.4 or later.

If the version of CodeWarrior you are using includes an earlier version of the Universal Headers 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 SDK.
- Copy the extension CarbonLib from the SDK to the Mac OS 8/9 Extensions folder, replacing the existing version if it exists.
- Copy the two stub libraries CarbonLib and CarbonFrameWorkLib from the SDK to Metrowerks CodeWarrior / Mac OS Support / Universal / Libraries / StubLibraries, replacing the existing versions if they exist.

Human Interface Guidelines - Mac OS 8/9

Useful additions to your library when you get further down the track would be the publications:

- Macintosh Human Interface Guidelines and Mac OS 8 Human Interface Guidelines (http://developer.apple.com/techpubs/mac/HIGuidelines/HIGuidelines-2.html).
- Inside Mac OS X — Aqua Human Interface Guidelines (http://developer.apple.com/techpubs/macosx/macosx.html).

[3] The Universal Headers were introduced at the same time as the Power Macintosh. Amongst other things, they enabled programmers to write (Classic API) source code capable of being compiled as either 680x0 code or PowerPC code — hence the term “Universal”.

Demonstration Programs

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 Demo Documents to the relevant user's Documents folder.

Before running a demonstration program, you should read the top section of the source code comments in the relevant chapter (or, for the post-Chapter 17 Classic event model versions, in the previously-mentioned PDF files). For most programs, this explains what to do, what to expect, and what to note. Note that the comments sections occasionally contain additional information incidental to the main purpose of the demonstration.

As far as is possible, each demonstration program avoids making calls to system software routines that are only explained in a later chapter. However, achieving that ideal has not been possible in the demonstration programs associated with the earlier chapters. For example, the demonstration program associated with Chapter 1 must, of necessity, make calls to system software routines relating to windows (the subject of Chapter 4) and drawing in a graphics port (the subject of Chapter 12). Where this occurs, you should simply accept, on faith, that the associated source code does as is stated in the demonstration program comments section. The important thing is to concentrate on that part of the source code pertaining to the subject of the chapter with which the program is associated.

Terminology

All but the later volumes of the official Mac OS reference work (Inside Macintosh) are Pascal-oriented, reflecting the fact that system software routines were originally designed to be called from a Pascal program. Because of this historically cosy relationship between Pascal and the Mac OS 8/9 system software, the C programmer will often be confronted by Pascal terminology. For example:

- The Pascal terms "procedure" and "record" are sometimes used in C-oriented programming books and other publications in a context where the C programmer would ordinarily expect the terms "function" and "structure".[4]
- The names of many system software data structures end in `Rec` (for record) rather than, say, `Struc` (for structure).
- The names of certain fields in many system software data structures end in `Proc` (for procedure) rather than, say, `Func` (for function).

As a reflection of the fact that the later additions to Apple's Inside Macintosh series of publications are C-oriented rather than Pascal-oriented, this book uses C terminology exclusively. Hence "structure" is used rather than "record" and "function" is used rather than "procedure".

Early Days

This version of Macintosh C Carbon was finalised during the early days of both Mac OS X and the Mac OS 8/9 Carbon implementation. At that time, both systems contained bugs which impacted on some of the demonstration programs. Known issues in this regard, for which the author begs the reader's indulgence, are listed in the ReadMe file in the demonstration program package.

K. J. Bricknell, AM
Canberra
Australia
kjbricknell@ozemail.com.au

[4] In C, a routine which returns a value and a routine which does not return anything are both called **functions**. In Pascal, a routine which returns a value is also called a **function**; however, a routine which does not return anything is called a **procedure**.