

MACINTOSH PASCAL

A Hobbyist's Guide to Programming the Mac OS in Pascal

Version 2.1

Koryn Grant and K. J. Bricknell

MACINTOSH PASCAL:
***A Hobbyist's Guide to Programming the Mac OS in
Pascal***

Version 2.1

©1999, K. J. Bricknell

Portions of **Macintosh Pascal: A Hobbyist's Guide to Programming the Mac OS in Pascal** were adapted from the Inside Macintosh series of books and **Develop** magazine, © Apple Computer, Inc. All rights reserved. Used with the permission of Apple Computer, Inc.

Apple, the Apple logo, LaserWriter, and Macintosh are trademarks of Apple Computer Inc., registered in the United States and other countries.

Finder and QuickDraw are trademarks of Apple Computer Inc.

Metrowerks is a registered trademark of Metrowerks Inc.
CodeWarrior is a trademark of Metrowerks, Inc..

PostScript is a trademark of Adobe Systems incorporated, which may be registered in certain jurisdictions.

No warranty or representation is made, either express or implied, with respect to this manual, its quality, accuracy, or fitness for a particular purpose. As a result, this manual is distributed "as is", and you, the distributee, are assuming the entire risk as to its quality and accuracy. In no event will the author be liable for direct, indirect, special, incidental, or consequential damages resulting from any defect or inaccuracy in this manual.

CONTENTS

	<i>Preface</i>
1	<i>System Software, Memory, and Resources</i>
2	<i>Low-Level and Operating System Events</i>
3	<i>Menus</i>
4	<i>Windows</i>
5	<i>Microprocessor Matters</i>
6	<i>The Appearance Manager</i>
7	<i>Introduction to Controls</i>
8	<i>Dialogs and Alerts</i>
9	<i>Finder Interface</i>
10	<i>Apple Events</i>
11	<i>QuickDraw Preliminaries</i>
12	<i>Drawing With QuickDraw</i>
13	<i>Offscreen Graphics Worlds, Pictures, Cursors, and Icons</i>
14	<i>More on Controls</i>
15	<i>Printing</i>
16	<i>Files</i>
16B	<i>More on Files — Navigation Services</i>
17	<i>More on Resources</i>
18	<i>Scrap</i>
19	<i>Text and TextEdit</i>
20	<i>Lists and Custom List Definition Functions</i>
21	<i>Floating Windows</i>
22	<i>Sound</i>
23	<i>Miscellany</i>

PREFACE

Macintosh Pascal: A Hobbyist's Guide to Programming the Mac OS in Pascal

This book relies heavily on information contained in the principal volumes of the Addison-Wesley publication **Inside Macintosh**. Some demonstration programs include Koryn's translations into Pascal of KJ's translations into C of Pascal code examples in that publication. In addition, parts of Chapters 21 and 22 rely on information contained in Issues No 11, 15, and 17 of **develop** (The Apple Technical Journal). Apple Computer, Inc, which holds the copyright to those publications, has kindly consented to the authors distributing Macintosh Pascal on the Internet as a free publication.

Origin

Macintosh Pascal: A Hobbyist's Guide to Programming the Mac OS in Pascal is a translation by Koryn Grant of the publication **Macintosh C: A Hobbyist's Guide to Programming the Mac OS in C** by K. J. Bricknell

Purpose and Evolution

Macintosh Pascal represents an attempt to provide a reasonably comprehensive entry point to Macintosh programming for the beginning hobbyist.

Versions 1.1 of Macintosh C and Macintosh Pascal were published on the Internet in early 1997. A few months after the release of Mac OS 8.0, a decision was taken to upgrade these two publications to accommodate the new features and changes ushered in by Mac OS 8.0 and, more particularly, an important new component of the system software known as the Appearance Manager. After some discussion, the authors concluded that producing a new version that addressed Mac OS 8.0 and the Appearance Manager while continuing to look backward to System 7 minus the Appearance Manager would not be a good idea.¹ Their reasoning was that the cumbersome mishmash of alternative text, screen-shots, source code, source code files, resource files, etc., that would have resulted from any attempt to address both worlds in the one book would have rapidly exhausted the patience of the reader — not to mention the sanity of the writer! Their guess was that the majority of hobbyists would be quite happy to learn to program exclusively for Mac OS 8 (or System 7 plus the Appearance Manager), would not need

¹ Most, though not all, of the new features introduced by Mac OS 8.0 were provided by the Appearance Manager, the earlier versions of which were delivered as an extension. Appearance Version 1.0 shipped with, and supported, Mac OS 8.0 only. However, as of Version 1.0.1, the extension works with System 7.1 through Mac OS 8.1 on Macintoshes and Power Macintoshes. As of Mac OS 8.5, the Appearance Manager was included in the System file. This was Version 1.1. Mac OS 8.0 and 8.1 run only on Power Macintoshes (PowerPC processors) and Macintoshes with Motorola 68040 processors. Mac OS 8.5 runs only on Power Macintoshes.

their programs to be backward-compatible to System 7 minus the Appearance Manager, and thus would not want to have their attention continually diverted to older and (to them) irrelevant ways and means.

Version 2.0 of Macintosh C, which was published in mid-1998, and which brought that publication up-to-date with Mac OS 8.0, was intended for that particular audience. The new version of Macintosh Pascal was held over until it could be upgraded to Version 2.1. Versions 2.1 of Macintosh C and Macintosh Pascal bring both publications up-to-date with Mac OS 8.5 less Window Manager 2².

For those who, for one reason or another, need to stay exclusively in the System 7 minus Appearance Manager world, or who need to write programs that are backwards-compatible with the old ways and means, Version 1.2 (Frozen) remains available. Version 1.2 (Frozen) is essentially Version 1.1 with an updated preface and the demonstration program files updated for Metrowerks CodeWarrior Pro 3 and the latest Universal Interfaces.

The First Task — Learn the Pascal Language

The main assumption made by Macintosh Pascal is that you have already learned the Pascal language. Accordingly, if you do not already know Pascal, learning that language will be your first task.

There are many books available that teach Pascal, and an excellent list of these can be found at <http://www.pascal-central.com/macbooks.html>.

As you are learning Pascal, do not spend too much time on the subject of console input/output, since this has limited application in the world of the graphical user interface. In addition, you can afford to gloss over file input/output at this stage, since Macintosh Pascal examples utilise Macintosh system software routines, rather than the standard Pascal file input/output routines, to effect file input/output. (Indeed, it is entirely possible that you will never need to use the standard Pascal file input/output routines.)

The Macintosh Pascal Phase

When you have learned the Pascal language, you are ready to open Macintosh Pascal. As you move through this second phase of the journey, you will quickly discover that learning Pascal was by far the easiest part!

Essentially, Macintosh Pascal covers all of the territory which, in the judgement of the authors, 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, etc.), how to ensure that your application observes the house rules of the Macintosh graphical user interface and cooperative multitasking environment, how to perform file input/output, how to print files, how to draw text and graphics, and so on.

Considerable thought has been given to the sequence in which each topic is introduced, the content of most chapters relying to some extent on a full understanding of what has gone before. Accordingly, you should note that Macintosh Pascal is not intended to be a randomly-accessed reference work; rather, it should be regarded as more in the nature of a course of study in which each chapter should be worked through in sequence.

² Shortly after the release of Mac OS 8.5, Apple advised of certain bugs in that part of Window Manager 2 concerned with floating windows support and recommended that the associated Application Programming Interfaces (APIs) not be used until the fault is rectified, possibly in Mac OS 8.6. In light of this situation, the authors have decided to disregard Window Manager 2 in this version of Macintosh Pascal.

General Structure of Macintosh Pascal

The general structure of most chapters of Macintosh Pascal 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 which explain the workings of the source code.

The book itself is supported by the CodeWarrior project and source code files, and Resorcerer resource files, for all demonstration programs.

What You Will Need

Development System

Apart from Macintosh Pascal you will, of course, require a development system. Macintosh Pascal assumes that that system will be Metrowerks CodeWarrior.

The Metrowerks product **Discover Programming For Macintosh** includes a Pascal compiler that produces code that will run on 680x0-based Macintoshes and (in emulation) on the PowerPC-based Power Macintosh. Since all Macintosh Pascal demonstration programs are capable of being compiled as either 680x0 or PowerPC code, and since the project files are "multi-target" (PowerPC *and* 680x0), the Discover Programming For Macintosh package will be quite adequate for your purposes.

The significantly more expensive Metrowerks product **CodeWarrior Pro**³, adds, amongst other things, a compiler capable of producing code which will run native on PowerPC-based Macintoshes.

On-Line Reference

An on-line reference enables you to quickly and easily access information relating to the system software, and is thus quite indispensable. You can choose between **THINK Reference** (<http://www.xplain.com/thinkreference/>) and Apple's CD-ROM-based **Macintosh Programming Toolbox Assistant** (<http://www.devdepot.com/descpage.html/PCODE=STBASST>).

THINK Reference is somewhat out-of-date but still quite useful. If you decide on THINK Reference, be aware that the spelling of many of the constants and function names listed therein is now quite out-of-date, and that many new constants, data types and functions have been introduced since the last version of THINK Reference was published. The most up-to-date references in these matters are the Universal Interface files produced by Apple and included in the Metrowerks CodeWarrior packages.⁴

Resource Editor

A resource editor allows you to create resources for programs and files. A copy of Apple's resource editor **ResEdit** is included with the CodeWarrior package; however, Apple ceased developing ResEdit some time ago, and it is simply not up to the task of creating the new resources introduced with Mac OS 8 and the Appearance Manager. The resource editor you will need is **Resorcerer**, which is produced by Mathemæsthetics, Inc (<http://www.mathemaesthetics.com/>).

³ All Macintosh Pascal demonstration programs were written using CodeWarrior Pro 3. Specially-priced academic versions of CodeWarrior Pro are available for students. Information on Metrowerks CodeWarrior products, including system requirements, is available at <http://www.metrowerks.com/>

⁴ The Universal Interfaces were introduced at the same time as the Power Macintosh. Amongst other things, they enable you to write source code capable of being compiled as either 680x0 code or PowerPC code — hence the term "Universal".

Other Tools

Another useful tool is **ZoneRanger**, a dynamic memory inspection tool that allows you to investigate how effectively and efficiently your application uses memory. ZoneRanger is included with the CodeWarrior package.

You will also find a programmer's calculator very useful for converting between decimal, hexadecimal and binary values, the nicely-presented shareware program **CalcWorks** being ideal for that purpose.

Human Interface Guidelines

Useful additions to your library when you get a little further down the track would be the publications **Macintosh Human Interface Guidelines** and **Mac OS 8 Human Interface Guidelines**, both of which are available at http://developer.apple.com/techpubs/mac/user_interface.html.

Universal Headers and Libraries Version 3.2

Version 2.1 of Macintosh Pascal assumes the use of **Version 3.2 or later of the Universal Interfaces and Libraries**. Version 3.2 was not included with CodeWarrior Pro 4 or earlier. You can download Version 3.2 from <http://developer.apple.com/sdk/>. In the folder titled Documentation, you will find a file titled CodeWarrior Users - ReadMe. To install Version 3.2 in CodeWarrior Pro 4 or earlier, simply follow the instructions in that file. (Note that the statement at the top of the file that "the 3.2 Universal Headers come pre-integrated into CodeWarrior Pro4" is incorrect. Version 3.2, in the event, was not included in Pro 4.)

There are also some minor problems with Version 3.2 Universal Interfaces that prevent them from compiling when using Case Sensitive Identifiers. (This option can be found in the Pascal Language preferences panel in CodeWarrior; by default, the demonstration programs for Macintosh Pascal have Case Sensitive Identifiers on). To fix these problems, make the following changes:

- In the Universal Interfaces file "Movies.p", change all occurrences of LONGINTPtr to LongIntPtr.
- In the Universal Interfaces file "fp.p", change all occurrences of decimal to Decimal.
- In the Universal Interfaces file "fp.p", change all occurrences of decform to Decform.

Special Requirements — Chapter 16B Demonstration Program

If you are running Mac OS 8.1 or earlier, the demonstration program associated with Chapter 16B (Files2) requires that you acquire and install a shared library (**Navigation**) not included with Mac OS 8.1 and earlier. Download the Navigation Services SDK (Software Development Kit) from <http://developer.apple.com/sdk/> and copy the shared library to your extensions folder. In addition, if you are using a 680x0 system, replace **OpenTransportLib.68K** in your Extensions folder with the one supplied in the SDK.

Special Requirements — System 7 Users

If you are using System 7, you will need System 7.5.5 or later for the demonstration program associated with Chapter 16B (Files2). Your most important special requirement, however, is the **Appearance extension Version 1.0.3**. Version 1.0.3 of the Appearance extension is included in the Appearance SDK (Software Development Kit), which may be downloaded from <http://developer.apple.com/sdk/>.

Demonstration Programs

All of the demonstration programs may be run from within CodeWarrior with the exception of the program that accompanies Chapter 10 — Apple Events. By its nature, this program should be run as a built (that is, double-clickable) application. The demonstration programs at Chapter 16A — Files and 16B — More on Files — Navigation Services may be run within CodeWarrior, although certain aspects of the programs can only be explored by running them as built applications.

You should read the top section of the source code comments in each chapter before running each program. For most programs, this explains what to do, what to expect, and what to note.

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

The latest volumes of the official Mac OS reference work (Inside Macintosh), and Apple's recent developer TechNotes are C-oriented, reflecting Apple's policy of moving away from Pascal and towards C/C++ for their internal development. Because of this, the Pascal programmer will often be confronted by C terminology. For example:

- The C term *function* is often used to mean routine, regardless of whether the routine in question returns a result (a true Pascal *function*) or not (a Pascal *procedure*).
- The names of some newer system software data structures may end in `struc` (for *structure*) rather than, say, `Rec` (for *record*).

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 (and Koryn's personal preference!), Macintosh Pascal Version 2.1 attempts to use a language-neutral description, referring interchangeably to routines as *routines*, *functions* or *procedures*, and to records as *records* or *data structures*.

There are a few other terms (or, rather, words) in this book which, depending on your country of residence, may seem only vaguely familiar. Bear in mind that this book was originally written in Australia and Pascalised in New Zealand, civilised lands where spelling conventions equate with those of the country that invented the language. Hence the word *colour* is generally spelled with a *u*. That said, the *u* has been removed where appropriate — for example, when reference is made to a component of the system software known, officially, as Color QuickDraw. In this way, and at the risk of being accused of inconsistency, the authors seek to offend nobody.

Future of the Mac OS

In early 1997, when Apple announced the intended introduction of a completely new operating system codenamed **Rhapsody**, the days of the Mac OS appeared to be numbered. All that has now changed. As announced by Apple at the 1998 World Wide Developers Conference, the new light on the horizon is now **Mac OS X** (pronounced "Mac OS Ten"), which is probably best described as an offspring of both Mac OS 8 and Rhapsody.

Mac OS X, which is due in the second half of 1999, will not be just another operating-system update; it will be, to all intents and purposes, a new operating system with "modern" operating system features such as pre-emptive multitasking and protected memory. That said, Mac OS X will take its look and feel from today's Mac OS, and will provide a level of compatibility with current Macintosh applications that Rhapsody itself could not offer.

All this is excellent news for the Macintosh programmer, including the beginners for whom this book is intended, the reason being that the vast bulk of what you learn about programming the Mac OS today will remain applicable in the Mac OS X era.

The Mac OS of today includes a collection of more than 8,000 system software functions (usually called **application programming interfaces (APIs)**) that programmers use to do such things as create a window or open a file. Apple has determined that about 2000 of these APIs are incompatible with a modern operating system like Mac OS X. The remaining 6000 or so "clean" APIs, together with a few modifications and additions, have been isolated and codenamed **Carbon**. Carbon represents the core set of APIs that will be used to build Mac OS X applications.

The remaining good news is that Carbon applications will run on Mac OS 8 as well. (At present, it is expected that Mac OS X will run on G3 Power Macintoshes only.) Because Carbon supports both the Mac OS 8 and Mac OS X runtime environments, you will not need to produce separate versions of your application's source code for each of these environments.

All Macintosh Pascal demonstration programs have been analysed by Apple's Carbon-compatibility facility. Of the APIs used in those programs, 92% are supported by Carbon, 2% are supported but will be modified in some way, 4% are supported but not recommended, 0.3% are currently being evaluated (and may or may not be included in Carbon), and 1.5% are not supported. In those circumstances, it is anticipated that the future task of rendering these programs fully Carbon-compatible will involve nothing more than a few hour's work, the upshot being that you may approach this book in the certain knowledge that the greater part of the information contained therein will remain valid well into the future.

Towards Version 2.2

Comments and suggestions on Macintosh Pascal may be addressed to: koryn@clear.net.nz

Dr Koryn Grant, PhD
AM
Palmerston North
New Zealand
April 1999

K. J. Bricknell,
Canberra
Australia
April 1999