Chapter 1

From Mac to Windows

Hi! Remember me? Back in 1984, I wrote a book called *Macintosh Revealed* that was the first widely available text on how to program Apple's Macintosh computer. In fact, for a time it was the *only* book available on Macintosh programming: Apple's own comprehensive manual on the subject, *Inside Macintosh,* didn't appear in a commercially published edition until nearly a year later. In the years since, *Macintosh Revealed* has gone through two editions, three publishers, and expanded from its original two volumes to a four-volume series. (I've lost count of how many volumes *Inside Macintosh* runs these days!) It's old and out of date by now, and left behind by the evolving Macintosh technology, but I'm amazed at how many people still come up to me, at trade shows and seminars, to tell me they got their start in Macintosh programming by reading my books. It's a great feeling to think that I may have helped the Macintosh grow into the great computer it is today.

All of us who were Macintosh devotees back in those early days knew that we were onto something revolutionary—a whole new way of interacting with computers. Our straitlaced friends with their stodgy old IBM clones could scoff at our cute little "toy" computer, but we knew better. Keyboards and command lines and 80-by-24 character screens were destined to become a thing of the past; the future belonged to mice and bitmaps, windows and menus, multiple fonts and character styles, pointing and clicking, cutting and pasting. Our clonehead friends were already obsolete; they just didn't know it yet.

Even when Microsoft introduced its Windows operating system for IBM-compatibles, we weren't impressed. Those early versions of Windows were sluggish, unresponsive, and limited by the relatively low resolution of the available screens. Because they were built on top of the cumbersome DOS operating system, they inherited all of its annoying limitations as well—such as a file system featuring interminable directory paths leading to local file names that couldn't be longer than eight characters with a three-character extension. Besides, the Windows user interface was such a transparent knock-off of the Macintosh. Why settle for a cheap imitation when you could have the real thing?

Well, a funny thing happened on the way to the future. Somewhere along the line, millions of users began flocking to Windows instead of the Macintosh. As the Windows user base grew, software developers began focusing more of their efforts on that platform and less on the Macintosh. Those of us who were diehard Macintosh loyalists still knew we had the best system going, but we could feel the tide turning

against us. As time went on, it was becoming harder and harder to stay active and credible in the personal-computer industry without some level of Windows literacy.

So I had just about decided it was time to swallow my pride and try to learn a little about Windows programming when, by a fortuitous stroke of timing, my phone rang one day last fall. The caller introduced himself as James Plamondon from Microsoft, product evangelist for Windows 95, the new version of Windows due out Real Soon Now. James said he was familiar with my Macintosh books from his days as a Macintosh registered developer and was wondering if I'd be interested in doing a book on Windows programming for the Macintosh audience. I told him I was probably not the right person for the job: I was basically a Macintosh kind of guy who had made a career of staying as far as possible from any computer with three letters in its name or the name of its operating system. I'd written some documentation once for a client who insisted I do the work on a PC-compatible machine, but it had taken a powerful effort of will not to return the machine at the end of the contract with a brick through the screen. I hadn't laid eyes on Windows in years, and as far as DOS was concerned, I didn't know an **`AUTOEXEC.BAT`** from a cricket bat.

James waited patiently while I finished my tirade, then explained with equanimity that Windows had come a long way since I last saw it, that the new version had finally broken free of its dependence on DOS, that it was perfectly possible nowadays to run Windows without ever going near a **`CONFIG.SYS`**, and that for the purposes he had in mind, my almost total ignorance of anything to do with Windows was a Good Thing. He wanted someone with a Macintosh background, he said, to sit down and learn Windows for the first time and then share the experience with the Macintosh community. Well, here was as good an excuse as any to take the Windows plunge. James can be a very persuasive guy, but the fact is he didn't need to do much convincing to get me to do the book.

In the interest of full disclosure, let's be clear about something right from the start: *Microsoft is paying me to write this book.* But I also want to be clear that they are exercising no editorial control over its content. I told James when I agreed to this project that I was not interested in writing Microsoft marketing literature. My deal with Microsoft was that I spend some time learning about the system and then write about it from a Macintosh point of view. I insisted on complete freedom to criticize as well as praise, to compare Windows to the Macintosh both positively and negatively. Any fault I might find with Windows is published here with Microsoft's full knowledge and consent. By the same token, anything I say in Windows' favor is there because *I really mean it,* not because Microsoft is paying the freight. The opinions expressed in this book are candid, unbiased, and entirely my own. Neither James nor I would have it any other way.

This book is what's known in publishing circles as a "work for hire." As the author, I get no royalties from sales of the book, just a flat fee for writing it. That's an important point, because it helps to ensure my editorial independence. Because I've

already made every nickel I'm ever going to see from this project before the book ever hits the presses, I have no financial stake in its commercial success or that of Windows 95 itself. I don't care how many copies Microsoft sells or doesn't sell, either of the book or of the system. I have no incentive to try to sell you anything. I am *not* interested in persuading anyone to abandon the Macintosh platform and switch to Windows development. I'm certainly not about to do that myself; I love the Macintosh too much, and always will. I take it as given that if you're reading this book, it's because you've already decided, for reasons of your own, to learn how to write Windows programs. If that's not the case, you have my permission to leave the book in the store and go spend the money on a nice dinner instead. No harm, no foul.

To get the most from this book, you should be

- an experienced Macintosh programmer
- familiar with the Macintosh Operating System and User Interface Toolbox
- acquainted with popular Mac development tools like Apple's Macintosh Programmer's Workshop, Symantec's Think C and Think Pascal, or Metrowerks' Code Warrior
- comfortable with the general concepts and principles of Macintosh-style event-driven programming

The purpose of the book is to show you how to put all that Macintosh knowledge to work in writing application software for the Windows platform. You'll be surprised (or not, depending on how you look at it) to find how much of your existing knowledge remains valid and useful on the Windows side of the fence. In each chapter, we'll consider one aspect of the Windows user interface or programming model—windows, menus, onscreen graphics, mouse and keyboard input, or whatever—and compare the Windows approach with what you already know about the subject from the Macintosh. Each chapter ends with a recurring section titled "The Same Thing...Only Different" that summarizes the similarities and differences between the two platforms.

The first thing I did to begin learning Windows myself was to take my old MiniEdit program, the simple cut-and-paste text editor I developed as a programming example for the *Macintosh Revealed* series, and port it to Windows. The conversion turned out to be remarkably easy. The fundamental structure of the program carried over almost intact from one platform to the other; the biggest difference between the two turned out to be that the Windows version—which I call WiniEdit—was substantially shorter than the Macintosh original. Part of the reason lies in the design of the Windows programming interface, which (as we'll learn later) implicitly handles of a lot of the implementation detail that Macintosh Toolbox programs have to do for themselves. But a large part of it, too, was the fact that the conceptual framework for Windows programming is so similar in many ways to that of the Macintosh.

Where applicable, I use code excerpts from WiniEdit to illustrate various points about Windows programming. It turns out, though—for many of the same reasons that WiniEdit is shorter than its Macintosh counterpart, MiniEdit—that it doesn't cover as much of the subject as MiniEdit does. To choose a couple of examples, MiniEdit had to include a substantial amount of code to support text editing and scrolling in its document windows. In Windows, much more of these mechanisms is implemented automatically by the system itself: there's simply a whole lot less code needed on the application side to support them. So we'll find, as we go along, that we'll spending less time examining the details of the program and more time discussing the subject on a theoretical level. If you really want to revel in all the glorious details, you'll find the complete code of the WiniEdit program in an appendix at the back of the book.

I should also put in a word here about programming languages. Back in the early days of the Macintosh, almost all programming for the machine was done in Pascal. That was the language in which Apple documented all of the Toolbox interfaces in *Inside Macintosh,* and it was the natural choice for me to use in *Macintosh Revealed* as well. Times have changed since then, and today most programmers, on both the Macintosh and Windows, prefer C and its object-oriented extension, C++. So before porting MiniEdit to Windows, I first translated it from Pascal to C and then ported it from there. (Actually, although the program doesn't use any of the language's object-oriented features, I used C++ instead of straight C for the sake of a few syntactic conveniences, like function-style typecasting.) Since the only published version of MiniEdit is written in Pascal, I've included the C version in an appendix for convenient comparison.

Let me stress that this book is intended as an *introduction* to Windows for Macintosh programmers. Its purpose is to give you an overview of Windows programming from the Macintosh point of view. I does *not* attempt to cover the subject in any great depth or detail, first because there's just too much ground to cover for a book this size, and second because I'm still just learning this stuff myself and I don't *know* all there is to say about it. Luckily, I can recommend a couple of other good books that you can use to put more flesh on the bare bones.

The definitive manual on all aspects of Windows programming is the *Win32 Programmer's Reference,* published by Microsoft Press. Where I might just mention the name of a function, message, or data structure in passing, the *Programmer's Reference* describes it comprehensively and exhaustively. It is *the* essential reference for all Windows programmers and an indispensable companion to the book you're reading now.

Most Windows programmers cut their eye teeth on *Programming Windows,* by Charles Petzold, also from Microsoft Press (or on one of its many variant editions—*Programming Windows 3.1, Programming Windows 95,* and so forth). Petzold has done for Windows programming what I tried to do for the Macintosh: a readable, informative introduction to the basic ideas and principles that every programmer

needs to understand. It would be presumptuous of me to call Charles the Steve Chernicoff of Windows programming, but I would be honored to be considered the Charles Petzold of the Macintosh.

Another excellent source of information from Microsoft Press is *Advanced Win32 Programming,* by Jeffrey Richter. This is a more technical treatment of the lower-level recesses of the Windows system, such as processes, memory management, and the file system, in far greater depth than I could possibly have attempted here. I've found Richter's book indispensable in understanding these aspects of the system myself.

Finally, let me say that I welcome any and all comments, criticisms, errata, quibbles, or bug reports you might have on any aspect of this book. You can reach me by electronic mail at **chernico@ccnet.com**, or by snail mail in care of the publisher. I hope you find this book useful in making *your* move from Macintosh to Windows.