

PREFACE

What could be more exciting than learning the cool subfile concepts and techniques provided in the first edition of this book? Learning more in this new edition, of course! Actually, subfile concepts haven't changed much from their inception in 1981, but the RPG language has changed, and changed drastically at that. Therefore, I felt it was time to update my previous book, *Subfiles in RPG IV*, to keep up with the new, hip version of RPG, as well as to take the opportunity to clean, tighten, and spit-shine, where needed. (Don't worry, no real spit was used in the creation of this book.)

The most significant change to the RPG language, and the focus of this edition, is the introduction of free-format RPG in the spring of 2001, with the release of V5R1. Yes, the publication of this book coincides with the tenth anniversary of free-format RPG! Although I use free-format RPG almost exclusively in the examples in this book, my purpose here is not to teach you how to use free-format RPG, nor to compare free format to its RPG predecessors. If you would like to learn more about free-format RPG on its own, Jim Martin's great book, *Free-Format RPG IV* (MC Press, 2005), provides everything you need to know about the latest, and definitely greatest, modification to the RPG language.

All that said, one advantage of free-format RPG is that it is easier to read than the prior versions of the language. So, even if you haven't coded much in free-format RPG, but are seasoned in one of the prior flavors of the language, you should be

able to understand what is going on. That, combined with my excellent program documentation (ahem), should make understanding my code a piece of cake.

There are a few examples in this book that I decided to keep as they were in the previous edition, rather than rewrite for free-format RPG. Specifically, in Chapter 5, I kept the original RPG code instead of prototyping the message API call statements for free-format RPG. In Chapter 8, I also retained the original RPG code around the embedded SQL statements. Because there is no way to code embedded SQL in free-format RPG, I kept the RPG surrounding the embedded SQL statements in column-based RPG, so as not to confuse the issue . . . much like I may have confused the issue here in my explanation!

In addition to converting most of the original programs into free-format RPG, I also cleaned things up a little in this edition. Over the years, some “undocumented features” (errors) have been found in the code and text from the first edition. I took this opportunity to fix those things, and to generally rewrite and tighten some of the code from the first edition. In a few places, I realized that the free-format language enabled me to do things in new ways, so I decided to experiment a little, while keeping the original subfile-related content intact. How could I pass up a chance to use a For loop in RPG? Come on, you would have done the same thing. I also reread the whole book and added further explanations where I thought it necessary. (I just wanted you to know that, so you’d be impressed.)

I made a few other changes in the areas where I refer to the machine or the operating system. The term “OS/400” has been replaced with “IBM i” and “AS/400” with “System i.” To be frank, I do not like the new names. I would have loved to stick with “OS/400” and “AS/400,” but I had to roll with the times; however, if you happen to see one of these beloved terms in this book, do not consider it an error. Instead, know that it is my way of going “old school,” showing you that some habits die hard.

Last, with this edition, you can view and download the complete source for all examples at <http://www.mc-store.com/5104.html>.

INTRODUCTION

You may ask yourself, “Why another book on subfiles? Have subfiles changed since their inception on the System/38, in 1981? Can’t I read the IBM manuals and previously written subfile books and get along just fine?” Of course you can! The basic concepts of subfiles haven’t changed much, and yes, you can read the IBM manuals and previously written materials to learn about subfiles.

“So what’s with the new book?” you might ask. Well, even though the basic concepts of subfiles haven’t changed much over the years, there have been some additions and improvements, including new Data Description Specifications (DDS) keywords and their related implementation techniques. Also, the language surrounding subfiles has changed drastically since those early days. The reconstruction of RPG in the form of RPG IV, along with the introduction of the Integrated Language Environment (ILE), has significantly changed how we create applications and has pushed the IBM midrange development environment closer to more of an object-oriented approach.

What follows from the changes to the system, to RPG, and to subfiles are new design fashions and programming techniques that must be consumed to fully take advantage of the environment, allowing us to create the best possible software solutions. A subset of this understanding is learning how to best employ subfiles in this modern environment. It’s one thing to learn the basic concepts of subfile programming; it’s quite another to learn how to program them in ways that allow them to work in harmony with today’s surroundings. Reading previously written materials may satisfy part of the equation, but those resources won’t teach you about what’s new or about

the most effective and efficient ways to incorporate subfile programming into today's development environment.

About This Book

So yes, this is another book on subfiles. But this one will provide you with the concepts, styles, and advanced topics of subfile programming, using RPG IV and ILE as its media. It will provide easy-to-understand explanations of subfile concepts, a bounty of practical examples, and some advanced techniques seldom seen in previous subfile books.

This book will take you on a journey from the very beginnings of subfile programming all the way to advanced techniques practiced by only those with a solid grasp of subfiles and the programming techniques and features of RPG IV and the ILE. More than simply getting you started with subfiles, it's meant to be a comprehensive resource that's used over and over again as you advance from very basic usage to guru-like practices.

This isn't a textbook—there are no exercises and exams at the end of each chapter to test what you've learned—but it certainly can be used as one. Each chapter builds upon the next so that you start with a solid base and build on that base as you proceed through the book. The purpose of this approach is to provide concepts, explanations, and practical examples you can use as templates for further development. The examples are available for download at <http://www.mc-store.com/5104.html>, so don't worry about having to rekey the code. Some basic knowledge of RPG IV and DDS is assumed. The contents of this book are there for the fine-tuning and enhancements of your skills.

What Does It Mean for Your Career?

Mastering subfiles has often been the defining moment in an RPG programmer's career. You might have been the best RPG II programmer in the world coming off the System/36, but if you didn't know subfiles, you were probably dismissed as an intermediate and told to "learn subfiles." Whether this is justifiable is another issue, but it's a widely accepted fact that until you know subfile programming, you can't say you're an expert.

When I interviewed for new jobs, the big question was always, “Do you know subfiles?” Not until I could confidently answer yes to that query and eagerly await the subfile-related questions that would surely follow did I know I had arrived. Armed with the formidable subfile power, I couldn’t wait to go on interviews. Once hired as a programmer with subfile experience, I was looked at in an entirely new light.

This phenomenon is less apparent now for three reasons. First, subfiles have been around for many, many years now, and programmers entering the IBM i world today are being exposed to subfile programming. Second, the intervening years have swallowed up many a System/36 programmer, so there are mighty few programmers coming from a system where subfiles weren’t available. Third, and most disturbing, is the *perception* that more RPG programmers know subfiles. I’ve worked in shops where this perceived knowledge spread throughout the programming staff like the plague. The problem was only partly the programmers’ doing. They were assigned to programming projects that required subfiles and, using existing subfile programs as templates (as we’ve all done), would merrily code away. Well, simply cloning a subfile program doesn’t make you *knowledgeable* in subfile programming. The other problem was with the code being used as a template. As you’ll see later, there are a variety of subfile types, and each is used in a certain circumstance. Understanding this determination is paramount to writing good subfile programs.

What is still happening today in some shops is that the proper subfile techniques aren’t being used in the correct circumstances. Worse still, there’s some bad code out there, even if it is employing the appropriate technique. If someone clones a subfile program but doesn’t know subfiles, he may not know it’s bad or inappropriate code. Once he uses bad or inappropriate subfile code as a template, some people might assume he knows subfile programming when all he really knows is how to clone bad subfile programs. The proliferation of subfile programs in this manner can be hazardous to your IT shop’s health.

Don’t follow in that vein. Come learn some basic and advanced techniques for yourself so you can stop bad subfile programming in its tracks. And take the time to learn how to use subfiles with other IBM i tools such as recursion, data queues, and embedded SQL to create very powerful and efficient applications.