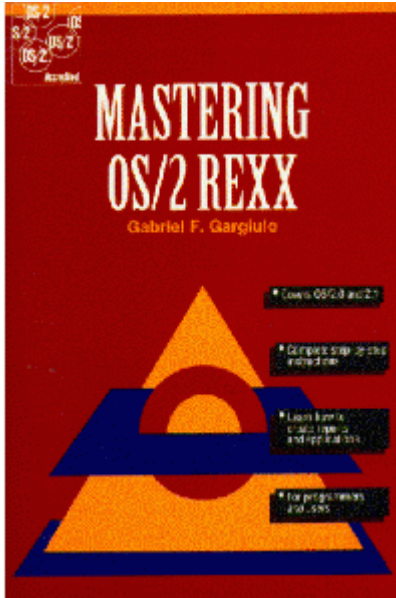


**Mastering OS/2 REXX**Written by [Carsten Whimster](#)[FEEDBACK](#)[SEARCH](#)[TOP](#)[BACKWARD](#)[FORWARD](#)**Introduction**

In The Codesmith's Library, I focus on development books and materials. I write this column from the point of view of an intermediate programmer. Pick up whichever book strikes your fancy, and join the growing group of people following our programming columns.

Please send me your comments and thoughts so that I can make this column what you want it to be. I read and respond to all mail.

Mastering OS/2 REXX is an introductory REXX programming book. It assumes little or no previous programming knowledge.

**Errata**

I have been working a bit more on my OS/2 WWW page, but I need a more permanent place to put it, as I have to pay for the current location separately, so you won't see it advertised any more, until I find a new spot.

It has been quiet with respect to mail from readers again. It would be nice if more readers would take the time to send in their thoughts and comments, both positive and negative, but then we are all busy.

Watcom C/C++ 10.0 is here. It looks really good, but I haven't had the time to look closer yet. It came in a package the size of a hard-cover book, containing only the CD-ROM (in case you hadn't guessed that I chose this format), a "Getting Started" booklet, and some flyers. The complete documentation package can be had for around \$100 as an update (to 9.5), or \$150 for the whole deal. I decided to live with the on-line version for now, as the price for the documentation is the same whether you buy it right away or not.

**Mastering OS/2 REXX**

Mastering OS/2 REXX is a how-to book on REXX programming for rank novices. It does a lot of hand-holding along the way, and is very unintimidating. It is aimed squarely at people who have either never programmed, or have programmed very little. It introduces new concepts with great patience, and has many little sample programs to illustrate key concepts and functionality. The following are the chapters:

1. What You Can Do With REXX
2. Where Does REXX Fit in on OS/2?
3. How to Create and Execute a REXX Program
4. REXX Syntax: The Rules of the Language
5. The Basics: Some Simple REXX Verbs
6. IF - the Conditional
7. String Manipulation: The Parse Instruction
8. Debugging
9. Trapping Errors
10. Math
11. SELECT: The Case Structure
12. Executing OS/2 Commands
13. Built-in Functions
14. User-Written Functions/Subroutines
15. Looping
16. Queueing Data with REXX
17. Compound Variables
18. Reading and Writing Files
19. The INTERPRET Instruction
20. Converting from Batch Files
  - Appendix A: Problems and Solutions
  - Appendix B: OS/2 Commands
  - Appendix C: REXX Instructions
  - Appendix D: REXX Functions
  - Appendix E: REXX Reserved Variables

The first two chapters are very introductory in nature, and only describe briefly what can be done with REXX, and how REXX is accessed on OS/2.

Chapter 3 introduces REXX properly, and eventually introduces the first little snippet of REXX code, but first it describes in (excrutiating, for those of us familiar with EPM and WPS) detail what shadows are, how to make a directory for your programs, how to start the editor (EPM), a brief tutorial on EPM, how to run your REXX programs on the commandline, and how to use PMREXX to run your REXX programs in the WPS environment. It also shows how to tell where your program is being executed, from within your program. Tokenizing of a program is demonstrated as well. Although this all sounds terribly slow for the experienced user, it surprises you a little from time to time with unexpected nuggets of obscure information.

Chapter 3 was only a brief demonstration, and in chapter 4 the syntax and some of REXX's peculiarities are explained. Numbers, literals, and variables are introduced, and some of the more basic keywords of REXX as well. Chapter 5 introduces SAY, PULL, ARG, ECHO, and EXIT. By the way, each chapter is fairly short (with a couple of exceptions), and at the end there is a question section, with the answers at the back of the book. This really helps the novice to test his or her newly gained knowledge at an early stage, and reinforces the material well.

Chapter 6 explains the various uses of IF, including using NOP, comparisons, fuzzy comparisons, and boolean operators.

Chapter 7 is more difficult, and introduces the PARSE instruction. This chapter is not as easy as what came before, and Gargiulo is very careful not to lose his reader. Many of the numerous permutations of the PARSE instruction are demonstrated. I suspect Gargiulo has extensive experience teaching REXX, something which is supported by his claim to offer several REXX-based courses on various operating systems.

Chapter 8 gets a little deeper, and introduces debugging. Both interactive debugging, and tracing is shown. In the same vein, chapter 9 continues the trouble-shooting thread into the trapping of errors. Many types of errors are explained, along with the trapping techniques required to handle them.

Chapter 10 leaves trouble-shooting behind, and introduces math under REXX. REXX is particularly flexible in math, and you can have any precision you desire. Precision and rounding are explained, and how to display numbers.

Just a quick note here: many of the preceding chapters were very short, and although every new topic is accompanied by an example, it would be nice to see some more elaborate examples from time to time. This would enable the novice programmer to get a better feel for what the various constructs and instructions are used for in the real world, and not just the typical half-page programs used more or less through-out the book.

Chapter 11 is dedicated to the SELECT structure, and again, this chapter is relatively short, although four examples were used to demonstrate the use. This all just goes to show that this book is for novice programmers, but doesn't teach programming, only REXX.

Executing OS/2 commands is the topic of chapter 12, and REXX's built-in functions are in chapter 13. Passing parameters is explained in more detail, and how to do this with OS/2 commands is explained. A good-sized list of available REXX functions is given. The list is well organized into categories to help dissipate the unavoidable confusion when confronted with a slew of new functions in one go. Many of these functions will have to be tried, or looked up in the on-line reference before the meaning becomes clear, but after the list, the more important ones are explained in more detail. The logical follow-up to chapter 13 is given in chapter 14, User-written Functions/Subroutines. This chapter shows how to use the RESULT variable, parameter passing and parsing, the RETURN keyword, labels, how to avoid sharing variables, and then compares internal and external REXX functions. Some pointers are given on style too, something which the book in general only shows through the snippets.

Chapter 15 shows the many types of loops available in REXX, and tries to explain where to use the different kinds. This is explained mainly through showing the constraints of each kind, and the higher level theory of the use of the different kinds is skipped over. Again, style and actual programming isn't really taught that much here, only REXX. From time to time there is a little something, but not much.

Chapter 16 surprised me a bit. Queueing is not a basic concept, yet Gargiulo teaches it with the same thoroughness that he teaches the simpler concepts. Simple process-to-process communication is explained, but not the need for it. The novice must be scratching his or her head at this point, saying "ok, but why?". The answer: look elsewhere for a why, this is a REXX how-to.

Compound variables are treated rather late in chapter 17, but the possible real-world use is skipped again.

File input and output is introduced in chapter 18, including reading from and writing to COM ports, the keyboard, printers, and the STD file streams, STDIN, STDOUT, and STDERR.

Chapter 19 demonstrates the INTERPRET instruction in three pages (now you know why many of the preceding paragraphs don't seem to say much more than the title of the chapters).

Finally, the book is wrapped up in chapter 20 by explaining how to convert from batch files to REXX programs, and when this is desirable.

## Summary

This book is very slow-moving, and is not aimed at anyone reasonably familiar with any modern programming language. It is very tutorial in nature, and introduces only the most basic concepts most of the time. The various looping constructs and so on are all explained as if you have never seen them before in any language. The verbal language used by Gargiulo is also very tutorial, as opposed to informational. This leaves me wondering why he skipped the high-level explanations of programming. Usually, introductory books aim to teach programming, and then choose a language to do it. It almost seems as if this book aims only to teach REXX, and not programming. I cannot imagine what kind of person would know how to program, but not be familiar with any actual language? In any case, the book teaches REXX very thoroughly, and it does hand-hold you through some slightly intermediate concepts. I have no problem recommending this book to bright would-be programmers, but I might hesitate a little before recommending it to someone whose computer knowledge is a little more sparse.

I give it a B.

Mastering OS/2 REXX, Gargiulo

- Wiley-QED. ISBN 0-471-51901-4, \$US40, \$50CAN, UK18.99.
- Intended audience: Beginning Rexx Programmers
- Mark: B

## Reviewed Books

Our [Bookstore](#) has reviews and links to Amazon for all books we have reviewed in the past, as well as several books waiting to be reviewed, and others which we recommend.

FEEDBACK SEARCH TOP BACKWARD FORWARD