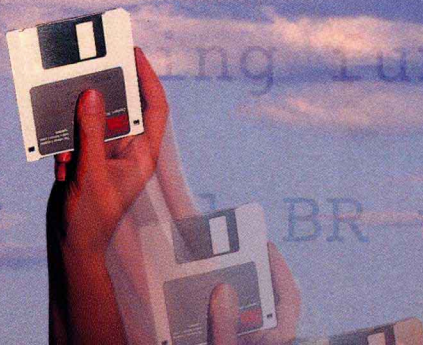
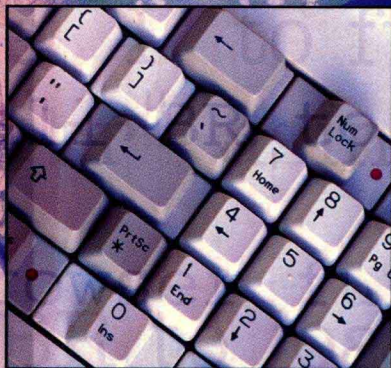
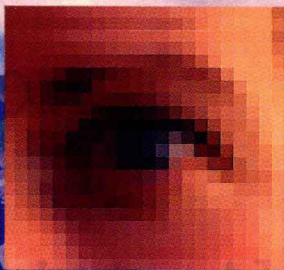


where COBOL meets personal computing

```
move in-buff-len to in-b  
add 1 to in-buff-ptr.  
process-by-the-byte.  
perform get-byte  
if next-byte = "A"  
    move "B" to next-by  
end-if  
if not-eof  
    perform put-byte  
end-if.  
get-byte.
```

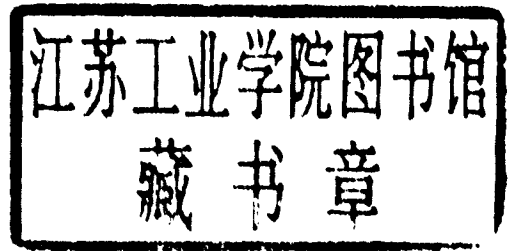


MICRO FOCUS®

Personal COBOL™ 2.0 for DOS

Micro Focus Personal COBOL™ 2.0
for DOS

Programmer's Guide



A Micro Focus Publication

MICRO FOCUS GROUP
Palo Alto/Newbury/Philadelphia/Tokyo/Munich/Paris/Barcelona

Copyright© 1993 Micro Focus Ltd.
All Rights Reserved.

Permission to copy the text or software contained herein may be granted by Micro Focus on a case by case basis. For more information or to request permission, contact Micro Focus Publishing, ATTN: Permissions, P.O. Box 52160, Palo Alto, California 94303, U.S.A. No part of this book may be copied by any means without written permission except in the case of brief quotations for use in education or in articles and reviews.

Micro Focus has made every effort to ensure that this book is correct and accurate, but reserves the right to make changes without notice at its sole discretion at any time. If you find errors, or believe there are omissions, please submit your comments in writing to Micro Focus Publishing.

The software described herein is supplied under a license which may be found at the back of this book. The software may be used or copied only in accordance with the terms of such license. See the *Sample Programs* section on page 1-8 for information about copying and using the sample programs provided with Personal COBOL.

In particular, any warranty of fitness of Micro Focus software products for any particular purpose is expressly excluded and in no event will Micro Focus be liable for any consequential loss.

ISBN: 1-56928-000-2

Micro Focus® and Animator® are registered trademarks of Micro Focus Limited. Micro Focus Personal COBOL™ is a trademark of Micro Focus Limited. Other registered trademarks and trademarks are acknowledged as the property of their respective owners.

Acknowledgements

This Personal COBOL Programmer's Guide contains material based on ANS X3.23-1985, the American National Standard for COBOL. That publication requests that documents using material from it should contain the following acknowledgment.

COBOL is an industry language and is not the property of any company or group of companies, or of any organization or group of organizations.

No warranty, expressed or implied, is made by any contributor or by the CODASYL Programming Language Committee as to the accuracy and functioning of the programming system and language. Moreover, no responsibility is assumed by any contributor, or by the committee, in connection therewith.

The authors and copyright holders of the copyright material used herein:

FLOW-MATIC (trademark for Sperry Rand Corporation) Programming the Univac (R) I and II, Data Automation Systems copyrighted 1958, 1959, by Sperry Rand Corporation; IBM Commercial Translator Form No.F28-8013, copyrighted 1959 by IBM; FACT, DSI27A5260-2760, copyrighted 1960 by Minneapolis-Honeywell.

have specifically authorized the use of this material in whole or in part in the COBOL specifications. Such authorization extends to the reproduction and use of COBOL specifications in programming manuals or similar publications.

Preface

Welcome to Personal COBOL

If you are new to COBOL, you are about to learn the most successful programming language of all time. COBOL has been the language of choice for business applications around the world for over 30 years. It has stood the test of time because it is based on sound principles and because it evolves to meet the changing needs of computing. When I learned COBOL 24 years ago, I never could have envisioned how it would evolve. The great thing has been that once I learned the basic principles of COBOL, I had a framework which made it easy to learn new COBOL features and indeed to learn new computing and data processing concepts. Amazingly, more than 95 percent of what I have learned is still of value to me today. Since I fully expect COBOL to be around for at least another 30 years, your investment in COBOL should be just as valuable for you.

COBOL stands for COMmon Business Oriented Language. This name is the key to its success and to understanding what COBOL is all about. So what do “Common”, “Business Oriented”, and “Language” tell us about COBOL?

COBOL is “Common” in the sense that it is common to all computers. In other words, a COBOL program that you write for one computer can later be run on a different make or model of computer. COBOL code is often a major company asset; worldwide the investment in COBOL programs is valued at trillions of dollars. Because of this, COBOL applications need to continue operating as new generations of computers and operating systems come into use. The code you write on your PC with Personal COBOL will run unchanged on other PC’s and Unix machines, and on future machines which are undreamed of today.

COBOL is "Business Oriented" in the sense that it was designed specifically for business applications. Everyday components of office life such as files, customer records and balance sheets are represented in a natural form in COBOL. One example of this is the ability to draw a "Picture" of how you would like a number to appear on a report. For example, you can write:

```
Account-Balance, Picture is $9999.99-
```

to indicate that you want the value to appear as a dollar sign, four digits before the decimal point, and two digits after followed by a minus sign if necessary. All you have to do is move the number to "Account-Balance" and it will automatically be formatted with the decimal point and currency symbol appearing where indicated, and a minus sign appearing if the number is negative. Furthermore, any arithmetic is handled to the precision required by accountants. In addition to its Business Oriented features, COBOL is also a good general purpose language, and Micro Focus has added some extra features to make it a good systems programming language.

The word "Language" in the name of COBOL is appropriate because COBOL is a language rather than a code; it is in fact modeled on the English language. For example, you can write:

```
Multiply hours-worked by rate giving pay
```

This aspect of COBOL has the effect of making it more understandable (or "readable") than most programming languages. It does also mean you have to do more writing than in some programming languages, but the effort is repaid many times over when you or someone else looks at your program in the months and years ahead and tries to work out what it is doing. You might not expect your first programs to have a very long life but many COBOL programs remain in productive use for anywhere from 10 to 20 years.

Micro Focus has set out to build the best tools for developers of business applications. It is the world leader in providing COBOL development environments for DOS, Windows, OS/2 and Unix. Micro Focus has made COBOL programmers more productive by providing tools that are specially tailored to their needs. In fact, Micro Focus Animator was the world's first debugger to work at the source code level. Personal COBOL is a combination of those programmer productivity tools that are particularly well-suited to those who are learning COBOL and writing personal applications.

If you want an example of an application that can be written in COBOL on a PC for use on a PC you need look no further than Personal COBOL itself. The Compiler, Editor, Animator, Screens and Help System are all written in Micro Focus COBOL. These are examples of systems programming rather than typical business applications. They do, however, illustrate COBOL as a common and a readable language. The compiler was written originally in 1976. COBOL's "readability", assisted further by associated tools such as Animator, has permitted many developers over the years to understand, debug and enhance it. This allows it to remain a "living" application on the forefront of technology. The fact that COBOL is "Common" has been used to port the compiler to hundreds of different platforms ranging from 8-bit micros to mainframes.

You might want to start with something less ambitious. But whatever you choose as your first application, I wish you many enjoyable hours with Personal COBOL. Have fun.

John Triance
Group Vice President
Micro Focus

Table of Contents

Preface - Welcome to Personal COBOL

Part I Getting Started	xix
Chapter 1 Introduction	1-1
This Book	1-1
Glossary of Terms	1-2
Which Version of COBOL Does Personal COBOL Support?	1-4
How You Use the Personal COBOL System	1-4
Other Components of Personal COBOL	1-7
Documentation	1-8
 Chapter 2 Installing	 2-1
Hardware and Software Needed	2-1
Installation	2-2
Verifying the Installation	2-5
More System Details	2-8
 Chapter 3 Sample Session	 3-1
Starting Personal COBOL	3-1
On-line Help System	3-3
Editing TICTAC	3-7
Checking TICTAC	3-11
Animating TICTAC	3-12
Examining Data Items Using Animator	3-17
Running TICTAC	3-19

Using Screens	3-20
Creating an Index Program	3-29

Part II Tools

Chapter 4 Editor	4-1
Overview	4-1
General Editing Features	4-1
COBOL Program Editing Features	4-2
Operation	4-3
Invoking the COBOL Editor	4-3
The Editor Screen	4-3
Exiting from the COBOL Editor	4-8
Functions	4-9
Loading a File	4-9
Entering Text	4-11
Inserting, Deleting and Restoring Text	4-15
Moving Through a File	4-21
Tagging Text Lines	4-25
Finding and Replacing Text	4-26
Moving and Copying Text	4-31
Saving a File	4-35
The Save-file Listboxes	4-35
COBOL Support Functions	4-37
Using Edit Windows	4-40
Printing a File	4-52
Drawing Lines	4-53
Editing Files Identified in Your Program	4-54
The Directory Facility	4-55
Printer Codes	4-56

Chapter 5 Checker (Compiling)	5-1
Invoking the Checker	5-1
Using the Checker	5-1
Specifying Checking Options	5-3
 Chapter 6 Animator	 6-1
Overview	6-1
Executing the Program	6-1
Monitoring and Changing Data Items	6-2
Logic Testing	6-2
Operation	6-3
Invoking Animator	6-3
Screen Description and Operation	6-3
Functions	6-12
Function Access	6-12
Function Descriptions	6-17
Error Messages	6-34
 Chapter 7 Screens	 7-1
Overview	7-1
Operation	7-2
Invoking Screens	7-2
Key Functions in Screens	7-5
The Screens Menu System	7-6
The Main Menu	7-6
The Alt Menu	7-11
The Ctrl Menu	7-16
Screen Section Clauses	7-18
Skeleton Program	7-20

The Screens Index Program	7-21
Error Messages	7-26

Chapter 8 On-line Help and Reference (HYHELP)	8-1
Overview	8-1
Operation	8-1
Invoking HYHELP	8-2
Using HYHELP	8-2
Selecting and Activating	8-4
Topics and Files	8-4
Topic Names	8-5
Hot-spots	8-5
Functions Available in HYHELP	8-6
The On-line Reference	8-14

Part III Reference

Chapter 9 Language Reference	9-1
Overview	9-1
COBOL Syntax	9-1
Personal COBOL Program Source	9-1
Syntax Diagram Conventions	9-2
COBOL Syntax Diagrams	9-3

Chapter 10 Library Routines	10-1
Overview	10-1
Call-by-Name Routines	10-1
Routines by Category	10-2
Descriptions of Routines	10-8

Call-by-Number Routines	10-70
Routines Available	10-70
Descriptions of Routines	10-71
Chapter 11 ADIS	11-1
Overview	11-1
Enhanced ACCEPT/DISPLAY	11-1
Screen Section	11-2
Configuration	11-2
Operation	11-2
Single Field ACCEPT/DISPLAY Statements	11-2
Displaying or Accepting Group Items	11-9
Large Screens	11-13
ADIS Run Time Configuration	11-13
Screen Section	11-20
Screen Section Option Descriptions	11-22
Screen Section ACCEPT and DISPLAY Statements	11-30
Keyboard Handling via ADIS	11-32
Function Key Handling	11-32
Types of Keys on the Keyboard	11-33
The CRT STATUS Clause	11-34
Normal Termination of an ACCEPT	11-35
Using the User Function Keys	11-36
Using the ADIS Keys	11-40
Data Key Handling	11-47
Mouse Handling via ADIS	11-56
Using the Mouse	11-56

Chapter 12 COBOL File Handling	12-1
Overview	12-1
Operation	12-2
COBOL File Naming Conventions	12-3
Filename Extensions used in Personal COBOL	12-4
COBOL Data File Assignment	12-8
Fixed File Assignment	12-8
Dynamic File Assignment	12-9
File Buffering	12-11
File Usage	12-11
Duplicate Keys	12-12
 Chapter 13 Compiler Directives	 13-1
Setting Directives	13-1
Descriptions of Directives	13-3
 Chapter 14 IBM Display Attributes	 14-1
Display Attributes	14-1
 Chapter 15 Run Time System Error Messages	 15-1
Overview	15-1
List of Run-Time Errors	15-2

For Further Reference

Index

Chapter 1

Introduction

With Personal COBOL, you have the best software tool available for learning COBOL programming on PCs. You can also use Personal COBOL to develop personal or small business applications for IBM PCs or computers that are compatible with them.

You can develop and run COBOL programs on DOS (3.3 or later), in a Windows (3.0 or later) DOS window, or in an OS/2 (2.0 or later) DOS session.

This Book

This book is for all programmers using Personal COBOL, whether experienced COBOL programmers or beginners. It will enable you to become familiar with the structure and capabilities of Personal COBOL quickly. This book assumes you are familiar with the operating system you will be using.

The parts of this book are as follows:

Part 1. *Getting Started*

describes the Personal COBOL System and how to install it; introduces you to some of the technology used throughout the system; and guides you through a brief session using the software so that you will soon feel at home with it.

Part 2. *Tools*

describes in detail how to use each tool in the Personal COBOL System. The tools are described in the order in which we expect you will want to use them.

Part 3. *Reference*

covers the technical detail of specific aspects of programming with Personal COBOL.

Glossary of Terms

Here is a brief review of what we mean by some of the technical terms in this book.

ADIS	the module of the Personal COBOL System which provides advanced screen ACCEPT and DISPLAY facilities. Programming for ADIS is described in the <i>ADIS</i> chapter of this book.
Animator	Micro Focus' debugger for COBOL programs. Animator is fully described in the <i>Animator</i> chapter of this book.
ANSI	the American National Standards Institute. One of the standards defined by ANSI is for the COBOL language.
ANSI 85 COBOL	the dialect of the COBOL language defined by the ANSI standard; this dialect is also called ANS 85 COBOL or COBOL 85 in some references. The ANSI 85 standard is rapidly becoming the basis for commercial COBOL programming and COBOL education.
Checker	The Micro Focus COBOL compiler which translates COBOL source code to INT code (see below). The Checker is described in the <i>Checker</i> chapter of this book.
COBOL	the programming language most widely used in business applications. The term COBOL is sometimes used to describe the entire COBOL language and sometimes just a subset or "dialect" of the language such as ANSI 85 COBOL.

INT code	short for 'intermediate code', a compact binary code which is produced by the Checker, and which can be executed by the Personal COBOL System. Micro Focus intermediate code files use the filename extension ".INT".
ISAM	an acronym for 'Indexed Sequential Access Method', the COBOL language which allows a simple program to use advanced file handling. The term ISAM is also sometimes used as the name of the Personal COBOL run-time module that handles creation and manipulation of indexed sequential files.
Micro Focus COBOL	the dialect of the COBOL language used in and supported by Micro Focus' products. It is based on and is a superset of ANSI 85 COBOL. Micro Focus COBOL is the most comprehensive implementation of the COBOL language available for any computer.
Personal COBOL	Micro Focus' integrated programming tool system for learning COBOL programming or developing personal or small business applications.
Pcobrun	the module of Personal COBOL which can be used from DOS to run a Personal COBOL program, after the program has been compiled with the Checker.
Run-time system	the part of Personal COBOL which executes the INT code produced by the Checker. Both Animator and Pcobrun use the run-time system.
Screens	Micro Focus' screen painter for rapid development of interactive screen handling. The Screens tool is described in the <i>Screens</i> chapter of this book.

Which Version of COBOL Does Personal COBOL Support?

The COBOL language in this system gives you powerful facilities over and above those usually found in COBOL. Notably, you can write programs to make full use of your personal computer's screen display, and you can access many operating system functions.

Personal COBOL accepts COBOL source programs conforming to both of the following standards and dialects:

- ANSI 85
- Micro Focus COBOL

How You Use the Personal COBOL System

Personal COBOL is provided as a set of integrated tools. When you start Personal COBOL you enter the Editor main menu. From here you can edit your source or go to other menus where you can invoke the other tools in the system. From any menu you can access the On-line Help System via the **F1=Help** key.

Chapter 3 guides you through a short Sample Session using the Editor and other tools so you can quickly learn the basic steps with Personal COBOL. The Editor and other tools are fully described in Part 2 of this book, and they are briefly introduced here.

Developing a program (in any language) usually involves repeated steps known as the edit-compile-debug cycle. In Personal COBOL this means that you start in the Editor, next you use the Checker, then you use Animator. Because the Personal COBOL tools are integrated, this development cycle is fast and easy. The order in which we discuss the tools here reflects the development cycle.