Effective Use of ANS Cobol Computer Programming Language

Laurence S. Cohn

ness Data Processing:
dited by
ning & J. Daniel Couger

TP312.4
C 1

P312.4
C1

7660462

Effective Use of ANS COBOL Computer Programming Language

Laurence S. Cohn

E7660462

江苏工学院
图书馆藏书

WILEY-INTERSCIENCE

JOHN WILEY & SONS
New York • London • Sydney • Toronto

Copyright © 1975, by John Wiley & Sons, Inc.

All rights reserved. Published simultaneously in Canada.

No part of this book may be reproduced by any means, nor transmitted, nor translated into a machine language without the written permission of the publisher.

Library of Congress Cataloging in Publication Data:

Cohn, Laurence S 1939-
Effective use of ANS COBOL computer programming language.

(Business data processing, a Wiley series)
1. COBOL (Computer program language) I. Title.
QA76.73.C25C66 001.6′424 75-5584
ISBN 0-471-16436-4

Printed in the United States of America

10 9 8 7 6 5 4 3 2

TP312.4
C1

Effective Use of ANS COBOL Computer Programming Language

Business Data Processing: A WILEY SERIES

Editors: Richard G. Canning
Publisher, Canning Publications, Inc.

J. Daniel Couger
Professor Management and Computer Science
University of Colorado

Personnel Implications for Business Data Processing
Robert A. Dickmann

Design of Sequential File Systems
Thomas R. Gildersleeve

Management Reporting Systems
James M. McKeever in collaboration with Benedict Kruse

A Guide to Packaged Systems
Robert V. Head

Control of the Information System Development Cycle
Robert I. Benjamin

Decision Tables: Theory and Practice
Solomon L. Pollack, Harry T. Hicks, Jr., and William J. Harrison

An Introduction to Data Base Design
John K. Lyon

The Organization of the Data Processing Function
Frederic G. Withington

COBOL Support Packages: Programming and Productivity Aids
Stanley M. Naftaly, Bruce G. Johnson, and Michael C. Cohen

Modular Programming in COBOL
Russell M. Armstrong

Effective Use of ANS COBOL Computer Programming Language
Laurence S. Cohn

试读结束，需要全本PDF请购买 www.ertongbook.com

Preface

This material has been assembled as a supplemental text for programming language students as well as the practicing computer programmer. Prior knowledge of ANS COBOL is not required. However, one must understand COBOL or be concurrently studying COBOL or ANS COBOL when using the text. Although this text is primarily geared to the programmer who is or will be working with IBM's OS or DOS systems, others will also benefit. The material is arranged so that the reader can easily locate the subject of interest through the contents, which brings recommended approaches to problems and potential problems, together with illustrations demonstrating the recommendations.

LAURENCE S. COHN

Lyndhurst, Ohio
December 1974

About the Assembler

Laurence S. Cohn received degrees in economics and accounting from Queens College of the City University of New York and Pace University, where he was also awarded a Certificate of Achievement in Electronic Data Processing.

He is a member of A.C.M., A.S.M., and S.A.M., and has appeared as guest speaker on such topics as "Undertaking Computer Feasibility Studies" and "Computer Capability And Cost Awareness".

Mr. Cohn has assembled, produced, and directed a film short on IBM's OS Job Control Language and assembled and produced "ANS COBOL Advanced Usage," a multi-media education kit currently used in industry and universities within the United States and abroad. In addition, he has assembled management and technical programs for the general public as well as for several Fortune 500 companies.

Illustrations

Figure	Title	Page
1	Sequential Data File Retrieval and Creation (DOS and OS)	48
2-1 to 2-3 3-1 to 3-3	Direct Access Creation (DOS)	49 52
4, 5-1 to 5-3 6	Index Sequential-Random Updating and Retrieval of An Index File (DOS)	55 59
7-1	Index Sequential-Random Updating and Retrieval of an Index File (OS)	60
7-2 to 7-4	Index Sequential-Random Updating of an Indexed File (DOS and OS)	61
8	Fixed Length Records (DOS and OS)	64
9	Variable Length Records (DOS and OS)	65
10	Undefined Length Record (DOS and OS)	66
11	Spanned Records/Blocking (DOS and OS)	67
12-1 to 12-4	Apply Write-only (DOS and OS)	68
13-1 to 13-3	Working Storage (DOS and OS)	72
14, 15	77 Levels (DOS and OS)	75
16	88 Levels (DOS and OS)	76
17	Occurs/Occurs Depending On Clause (DOS and OS)	77

18 External Floating Point (DOS and OS) 78

19 Alphabetic Data Item (Dos and OS) 78

20-1 to 20-2 Numeric Edited (DOS and OS) 79

21 Picture Consideration (DOS and OS) 81

22 Representation of Occurrence Number (Subscript Field) (DOS and OS) 82

23-1 to 23-8 Write Subroutine (Generating Printouts With One Write Statement) (DOS and OS) 83

24-1 to 24-4 Reducing Subscript's Use (DOS and OS) 91

25 Indexing (DOS and OS) 95

26 Deblocking (DOS and OS) 96

27 Reserve No Alternate (DOS and OS) 97

28-1 to 28-3 Same Area Clause/Same Record Area (DOS and OS) 98

29 Segmentation (DOS and OS) 101

30 Condition Tests (DOS and OS) 102

31 Literal Instead of A Subscript (DOS and OS) 103

32 SEEK Statement (DOS and OS) 103

33 START Statement (DOS and OS) 104

34 Apply Core-Index 105

35-1 to 35-4 Accumulating Totals (DOS and OS) 106

36 Issuing An Array's Value (DOS and OS) 110

37-1 to 37-2 Binary Search (DOS and OS) 111

38 Direct Table Access (DOS and OS) 113

39 On Size Error (DOS and OS) 114

40 Comparison (DOS and OS) 115

41 Editing Card Input (DOS and OS) 116

42-1 to 42-7 Use After Error in the Declaratives Section (DOS and OS) 117

43	Return-Code (DOS and OS)	123
44-1 to 44-2	Debug Card (DOS and OS)	124
45	Blow Up No Map-Pack Instruction	126
46	Clearing A Group of Pack Fields	127
47	Examine Statement	128
48	A Sign Removal (DOS and OS)	129
49	Move All Literal (DOS and OS)	130
50-1 to 50-2	The Corresponding Option in Move And Substract (DOS and OS)	131
51	Copy Statement Options (DOS and OS)	133
52-1 to 52-2	Replacing Option of the Copy Statement	134
53-1 to 53-4	Subprogram Linkage Statements	136
54	GO BACK Statement (DOS and OS)	140
55	EXIT Program Statement	141
56	Perform Statement Formats (DOS and OS)	142
57	Perform Varying Logic Flowchart (DOS and OS)	143
58-1 to 58-7	Perform Statements (DOS and OS)	144
59-1 to 59-8	Internal Sort (DOS and OS)	151
59-9	Internal Sort With No Exits (DOS and OS)	159
60	File Description For Report (DOS and OS)	160
61-1 to 61-8	File Description (DOS and OS)	161
62	Generated Report	169
63	Print-Switch (DOS and OS)	171
64	Sort Feature Registers (DOS and OS)	172
65-1 to 65-2	Special Names Paragraph	173
66	Figurative Constants (DOS and OS)	175
67	Compound Conditions (DOS and OS)	176

68 Implied Subjects and Implied Relational Operators (DOS and OS) 177

69 Comments (DOS and OS) 178

试读结束，需要全本PDF请购买 www.ertongbook.com

Contents

1. FILE ORGANIZATION 1

Data Organization 1

• Sequential Data Organization-File Retrieval and Creation • Direct Data Organization-Access Creation • Index Sequential Organization-Creating Index File • Random Updating And Retrieval Of An Indexed File

Record Organization 3

• Fixed Length Records • Variable Length Records • Undefined Length Records • Spanned Records • Blocking • Apply Write-only

2. STORAGE HANDLING 6

Organizing Working Storage 6

• Working Storage • 77 Levels • 01 - 49 Level • 66 Level • 88 Level • OCCURS Clause • OCCURS DEPENDING ON Clause • REDEFINES Clause • Internal Decimals • External Decimal • Binary Items • Floating Point Short • Floating Point Long • External Floating Point • Alphabetic Data Item • Numeric-Edited

Coding Clues For Core Efficiency 12

• Use of Packed Form COMPUTATIONAL - 3 • PICTURE Consideration • Numeric Fields • Representation Of Occurrence Number (Subscript Fields) • WRITE Sub-

routine • Reducing Subscript's Use • Indexing • Subroutine

Quick Ways Of Reducing Core Size 15

• Deblocking • RESERVE NO ALTERNATE • SAME AREA Clause • Same Record Area • REDEFINES Clause • Segmentation
SECTION
Priority Numbers
SEGMENT-LIMIT

3. ACHIEVING SPEED OF EXECUTION 18

Consideration In Coding 18

• Unprintable Characters • Efficient Coding • Condition Tests • Literal Instead Of A Subscript-Name • Same Record Area

Consideration In Data Division 19

• Blocking Factor • RESERVE NO ALTERNATE • SEEK Statement • START Statement • APPLY CORE-INDEX • Accumulating Totals

Table Handling 20

• Issuing An Array's Value • Binary Search • Direct Table Access • Table Lookup

4. DEFENSIVE PROGRAMMING 22

• ON SIZE ERROR • Initialization • Comparison • Editing Card Input • USE AFTER ERROR In The DECLARATIVES SECTION • RETURN-CODE

5. DEBUGGING METHODS 25

• Tracing Logical Flow • Program Runs, Arithmetic Wrong • DEBUG Card • Handling Blowups
Common System Error Messages
Use Of Map
BlowUp No Map

6. PROGRAMMING TECHNIQUES 29

• Clearing A Group Of Packed Fields • EXAMINE Statement • A SIGN REMOVAL • MOVE ALL Literal • The CORRESPONDING Option In MOVE, ADD and SUBTRACT • Subroutines • COPY Statement and BASIS Card

COPY Statement Options
Replacing Option Of The COPY Statement
BASIS Card
INSERT and DELETE Card

• SUBPROGRAM LINKAGE Statements

CALL Statement
ENTRY Statement
GOBACK Statement
EXIT PROGRAM Statement

• PERFORM Statement - Formats • PERFORM Varying Logic Flow Chart • PERFORM Statements

7. SPECIAL FEATURES 34

INTERNAL SORT 34
REPORT FEATURES 35
Data Division Consideration 35

• File Description For Report • Report Description • Generated Report

Procedure Division Statements 37

• INITIATE • GENERATE • TERMINATE

SPECIAL REGISTERS 38
Report Feature Registers 38

• PAGE-COUNTER • LINE-COUNTER • PRINT-SWITCH

Sort Feature Registers 39

• SORT-FILE-SIZE • SORT-CORE-SIZE • SORT-MODE-SIZE • SORT-RETURN

System Common Registers 40

• *TALLY* • *CURRENT-DATE* • *COM-REG* • *NSTD-REELS* • *TIME-OF-DAY* • *RETURN-CODE*

SPECIAL NAMES PARAGRAPH 41
OTHER SPECIAL FEATURES 42

• *USA Figurative Constants* • *Compound Conditions* • *Implied Subjects and Implied Relational Operators* • *COMMENTS*
NOTE
Asterisk In CC-7
• *Access To System Features*
CURRENT-DATE
TIME-OF-DAY
RETURN-CODE
LABEL-RETURN
COM-REG
UPSI Switches

8. ILLUSTRATIONS 47

CHAPTER 1

File Organization

DATA ORGANIZATION

There are three different ways in which to organize a file:

- Sequential data organization
- Direct data organization
- Indexed data organization

Sequential Data Organization

Sequential organization is the most common way to organize a file; the file is created in the same record order as its inputs. It is a faster way to process a file, when most of the records in the file have to be processed. An example of retrieving a sequential file is in Figure 1, line 010, (the SELECT statement) and line 070 (the RETRIEVAL statement).

The statement defining the creation of a sequential file is on line 020 (the SELECT statement), and line 090 is the WRITE statement. The records are read in physical order and are written in the order in which they were inputted.

Direct Data Organization

When direct data organization is used, one specifies the position of the logical record by supplying the ACTUAL KEY, which is defined in the environment division and referred to in the working storage area.

The ACTUAL KEY is constructed from two components:

Track Identifier

Where the record is going to be located, as in Figure 2-2, lines 090 through 130 in DOS, and in Figure 3-2, line 110 in OS.

Record Identifier

Record ID which makes this record unique, as in Figure 2-2, line 140 in DOS, and in Figure 3-2, line 120 in OS.

In DOS, the track identifiers of the ACTUAL KEY supplies the drive, cylinder, head, and record of the desired record (Figure 2-2, lines 140, 110, 120, 130) or the relative track of the desired record. In OS the track identifier component of the ACTUAL KEY is the relative track number.

Under DOS one cannot create a random file sequentially. When sequential retrieval of a random file occurs, the records will be read in the order in which they are on the file, that is, first track records first, second track records next, and so forth, but that is not necessarily the logical sequence of the file.

Index Sequential Organization

A file that is organized in index sequential format must reside on mass storage devices such as 2311, 2314, or 2321.

An index sequential file may only be created in the sequential mode. The key for the file—RECORD KEY—must be specified as in Figure 4 or 5-1, line 100, and it must reside within the physical record as in Figure 5-2, line 020.

To retrieve or update an indexed file randomly, NOMINAL KEY, Figure 6 or 7-1, line 080, and RECORD KEY, line 090, must be specified as well as defined as in Figure 7-3, line 100, and Figure 7-2, line 020. The NOMINAL KEY contains the value of the RECORD KEY of the record desired. The NOMINAL KEY must be specified when it is desirable to start processing the file with a record other than the first. To do that successfully, a START statement must be issued before the first READ statement.

In order to create an index sequential file successfully, the records should enter the system in key order, while checking for and avoiding duplicates, as in Figure 5-3, line 030.

The system keeps an index of the keys. This index is called the cylinder index. On a large volume file, to reduce the amount of lookup time in the cylinder index one can set up a master index, with the en-

试读结束，需要全本PDF请购买 www.ertongbook.com