SC21-7667-4

File No. S34-28

IBM System/34
RPG II
Reference Manual

Program Number 5726-RG1

SC21-7667-4

File No. S34-28

IBM System/34
RPG II
Reference Manual

Program Number 5726-RG1

Fifth Edition (January 1982)

This is a major revision of, and obsoletes, SC21-7667-3 and technical newsletter SN21-8136. Chapter 10 has been reorganized to place the operation codes in alphabetical order. New operation code diagrams have also been added. Other changes or additions to the text and illustrations are indicated by a vertical line to the left of the change or addition.

This edition applies to release 8, modification 0 of IBM System/34 RPG II Program Product (Program 5726-RG1), and to all subsequent releases and modifications until otherwise indicated in new editions or technical newsletters. Changes are periodically made to the information herein; changes will be reported in technical newsletters or in new editions of this publication.

It is possible that this material may contain reference to, or information about, IBM products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that IBM intends to announce such IBM products, programming, or services in your country. (For example, ideographic support is available only in Far East countries.)

This publication contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

Use this publication only for the purposes stated in the *Preface*.

Publications are not stocked at the address below. Requests for copies of IBM publications and for technical information about the system should be made to your IBM representative or to the branch office serving your locality.

This publication could contain technical inaccuracies or typographical errors. Use the Reader's Comment Form at the back of this publication to make comments about this publication. If the form has been removed, address your comments to IBM Corporation, Publications, Department 532, Rochester, Minnesota 55901. Comments become the property of IBM.

IBM may use and distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever. You may, of course, continue to use the information you supply.

Preface

PURPOSE OF THE MANUAL

The RPG II reference manual aids the programmer in writing RPG II programs for the IBM System/34. The manual describes the program documentation the programmer needs to write, test, and maintain RPG II programs. Using this manual, the programmer can:

- Use the detailed reference material to code, compile, and debug RPG II application programs
- Use the detailed reference material to implement RPG II functions such as work station support, data structuring, and display format generation
- Use the information as a guide to the RPG II program cycle
- Achieve more efficient use of disk and main storage and more efficient program execution

MAJOR TOPICS

The following chapter descriptions summarize the major topics discussed in this manual:

- Chapter 1 describes the RPG II program cycle, briefly describes the indicators that can be used to change the program cycle, and provides an overview of the RPG II specifications.
- Chapters 2 through 9 provide a column-by-column description of the RPG II specifications in the order they are used:
  - Control
  - File description
  - Extension
  - Line counter
  - Telecommunications
  - Input
  - Calculation
  - Output

  Each column description includes a list of possible entries, a general discussion of column use, considerations for all possible entries, a specific discussion of each entry, and, where pertinent, charts and examples.
- Chapter 10 provides a detailed explanation of the operation codes used on the calculation specifications.
- Chapter 11 describes how records are selected for processing in multifile processing and how match field values are assigned.
- Chapter 12 describes the special considerations for using the CONSOLE device in an RPG II program. When the CONSOLE device is specified in an RPG II program, the operator can enter data from a display station directly into an executing RPG II program.
- Chapter 13 describes the special considerations for using a WORKSTN (display station) device in an RPG II program. When a WORKSTN device is specified, an RPG II program can receive input data from, and write output data to, one or more display stations. Sample programs that use the WORKSTN device are included in Chapter 13.
- Chapter 14 describes how to create, define, and load tables and arrays for an RPG II program.
- Chapter 15 describes the RPG II auto report function, which can reduce the coding required for similar programs and which can produce formatted reports.
- Chapter 16 presents programming considerations that can help the programmer save storage and improve performance.
- Chapter 17 presents a detailed description of the RPG II program logic cycle.
- Chapter 18 describes the command statements required to compile and execute an RPG II program.
- Chapter 19 contains sample RPG II programs. These include the specifications and the printed output for the programs. Chapter 13 contains sample programs for the WORKSTN device.
- Chapter 20 describes the support provided for the processing of ideographic data.

- Appendix A includes summary charts for the specification sheets, for the operation codes, for the indicators, and for the display screen format S and D specifications.
- Appendix B contains the RPG and auto report printed messages that are generated by the compiler.
- The glossary provides a list of RPG II terms and their definitions.

SYSTEM REQUIREMENTS

See the *IBM System/34 Planning Guide*, GC21-5154, for the system requirements for the System/34 RPG II Compiler.

The System/34 RPG II compiler provides ideographic support when used with the ideographic version of the SSP and the ideographic hardware devices that version supports.

RELATED PUBLICATIONS

- *IBM System/34 Introduction*, GC21-5153
- *IBM System/34 Planning Guide*, GC21-5154
- *IBM System/34 System Support Reference Manual*, SC21-5155
- *IBM System/34 Concepts and Design Guide*, SC21-7742
- *IBM System/34 Source Entry Utility Reference Manual*, SC21-7657
- *IBM System/34 Data Communications Reference Manual*, SC21-7703
- *IBM Introduction to RPG II*, GC21-7514
- *IBM System/34 Installation and Modification Reference Manual: Program Products and Physical Setup*, SC21-7689
- *IBM System/34 Operator's Guide*, SC21-5158
- *IBM System/34 Displayed Messages Guide*, SC21-5159
- *IBM System/34 Screen Design Aid Programmer's Guide and Reference Manual*, SC21-7716
- *IBM System/34 Master Index*, SC21-7739
- *IBM System/34 Interactive Communications Feature Reference Manual*, SC21-7751

The *System/34 Introduction* includes a *Publications Summary* that contains a brief description of the contents of each of the System/34 system publications.

RPG II Coding and Debugging Material

- *RPG Control and File Description Specifications*, GX21-9092
- *RPG Extension and Line Counter Specifications*, GX21-9091
- *RPG Telecommunications Specifications*, GX21-9116
- *RPG Input Specifications*, GX21-9094
- *RPG Calculation Specifications*, GX21-9093
- *RPG Output Specifications*, GX21-9090
- *RPG Auto Report Specifications*, GX21-9139
- *RPG Indicator Summary*, GX21-9095
- *RPG Debugging Template*, GX21-9129
- *Translation Table and Alternate Collating Sequence Coding Sheet*, GX21-9096
- *System/34 Display Screen Format Specifications*, GX21-9253
- *IBM 5251 Display Station Keyboard Template Assignment Sheet and Display Screen Layout Sheet*, GX21-9271

Contents

CHAPTER 1. INTRODUCTION 1-1
RPG II Functions 1-1
Steps in Using RPG II 1-1
Overview of Specification Sheets 1-4
Control Specifications 1-4
File Description Specifications 1-4
Extension Specifications 1-4
Line Counter Specifications 1-4
Telecommunications Specifications 1-4
Input Specifications 1-5
Calculation Specifications 1-5
Output Specifications 1-5
How RPG II Works 1-5
RPG II Program Cycle 1-5
Indicators 1-9
Common Entries 1-12
Columns 1-2 (Page) 1-12
Columns 3-5 (Line) 1-13
Column 6 (Form Type) 1-13
Column 7 (Comments) 1-13
Columns 7-12 (/EJECT) 1-13
Columns 7-12 (/TITLE) 1-13
Columns 7-14 (/SPACE) 1-14
Columns 75-80 (Program Identification) 1-14

CHAPTER 2. CONTROL SPECIFICATIONS 2-1
Columns 1-2 (Page) 2-1
Columns 3-5 (Line) 2-1
Column 6 (Form Type) 2-1
Columns 7-9 (Size To Compile) 2-2
Column 10 (Object Output) 2-2
Column 11 (Listing Options) 2-2
Columns 12-14 (Size to Execute) 2-2
Column 12 2-2
Columns 13 and 14 2-2
Column 15 (Debug) 2-3
Columns 16-17 2-3
Column 18 (Currency Symbol) 2-3
Columns 19-20 (Date Option) 2-3
Column 19 (Date Format) 2-3
Column 20 (Date Edit) 2-3
Column 21 (Inverted Print) 2-4
Columns 22-25 2-4
Column 26 (Alternate Collating Sequence) 2-4
Columns 27-36 2-9
Column 37 (Inquiry) 2-9
File Sharing 2-10
Inline Inquiry Subroutine (SUBR95) 2-10
Columns 38-40 2-11
Column 41 (1P Forms Position) 2-11
Column 42 2-11
Column 43 (File Translation) 2-11
Column 44 2-14
Column 45 (Nonprint Characters) 2-14
Columns 46-47 2-14
Column 48 (Shared I/O) 2-14
Columns 49-51 2-14
Columns 52-53 (Number of Formats) 2-14
Columns 54-56 2-15
Column 57 (Transparent Literal) 2-15
Columns 58-74 2-15
Columns 75-80 (Program Identification) 2-15

CHAPTER 3. FILE DESCRIPTION SPECIFICATIONS . 3-1
Columns 1-2 (Page) 3-1
Columns 3-5 (Line) 3-1
Column 6 (Form Type) 3-1
Columns 7-14 (Filename) 3-2
Column 15 (File Type) 3-2
Input Files 3-2
Output Files 3-2
Update Files 3-2
Combined Files 3-4
Column 16 (File Designation) 3-4
Primary Files 3-4
Secondary Files 3-4
Chained Files 3-5
Record Address Files 3-5
Table or Array Files 3-5
Demand Files 3-5
Column 17 (End of File) 3-6
Column 18 (Sequence) 3-6
Column 19 (File Format) 3-6
Columns 20-23 (Block Length) 3-7
Columns 24-27 (Record Length) 3-7
Column 28 (Mode of Processing) 3-8
Consecutive 3-9
Sequential by Key 3-9
Sequential Within Limits 3-10
Random by Relative Record Number or Key 3-11
Random by Addrout File 3-17
Shared File Considerations 3-17
Columns 29-30 (Length of Key Field or Record Address Field) 3-17
Column 31 (Record Address Type) 3-19
Column 32 (File Organization or Additional Input/Output Area) 3-20
File Organization 3-20
Additional Input/Output Area 3-22
Addrout Files 3-28
Columns 33-34 (Overflow Indicator) 3-28
Automatic Overflow 3-28
Overflow Indicator Specified 3-28
Columns 35-38 (Key Field Starting Location) 3-29
Column 39 (Extension Code) 3-29
Columns 40-46 (Device) 3-29
BSCA 3-30
CONSOLE 3-30
CRT 3-32
DISK 3-33
KEYBORD 3-33
PRINTER 3-33
WORKSTN 3-33
SPECIAL Device Support 3-33
Reading and Updating a Work Station Utility Transaction File (SUBR22) 3-37

System Input Subroutine for SPECIAL Device (SUBR01) 3-45
1255 MICR Subroutine (SUBR08 and SUBR25) 3-45
Columns 47-52 3-45
Column 53 (Continuation Lines–K) 3-45
Columns 54-59 3-46
Name of Label Exit 3-46
Continuation Line Option for SPECIAL Device 3-46
Continuation Line Options for WORKSTN File 3-47
Continuation Line Keyword Option 3-49
Columns 60-65 (Storage Index) 3-49
Column 66 (File Addition) 3-51
Adding Records to an Indexed File (A) 3-51
Adding Records to Delete-Capable Direct and Sequential Files 3-52
Extending a Sequential File 3-52
Loading Records in an Unordered Sequence (U) 3-56
File Addition Functions for Indexed Files 3-56
File Addition Functions for Sequential Files (File Extension) 3-56
Random File Addition Functions for Delete-Capable Direct and Sequential Files 3-56
Column 67 3-56
Columns 68-69 3-56
Column 70 3-56
Columns 71-72 (File Condition) 3-57
Columns 73-74 3-57
Columns 75-80 (Program Identification) 3-57
File Description Charts 3-57
How to Use the Charts 3-57

CHAPTER 4. EXTENSION SPECIFICATIONS 4-1
Columns 1-2 (Page) 4-2
Columns 3-5 (Line) 4-2
Column 6 (Form Type) 4-2
Columns 7-10 4-2
Columns 11-18 (From Filename) 4-2
Columns 19-26 (To Filename) 4-2
Columns 27-32 (Table or Array Name) 4-3
Table Name 4-3
Array Name 4-3
Columns 33-35 (Number of Entries Per Record) 4-3
Columns 36-39 (Number of Entries Per Table or Array) 4-5
Columns 40-42 (Length of Entry) 4-6
Column 43 (Packed or Binary Field) 4-6
Column 44 (Decimal Positions) 4-6
Column 45 (Sequence) 4-8
Columns 46-57 4-8
Columns 58-74 (Comments) 4-10
Columns 75-80 (Program Identification) 4-10

CHAPTER 5. LINE COUNTER SPECIFICATIONS 5-1
Columns 1-2 (Page) 5-1
Columns 3-5 (Line) 5-1
Column 6 (Form Type) 5-1
Columns 7-14 (Filename) 5-2
Columns 15-17 (Line Number–Number of Lines Per Page) 5-2
Columns 18-19 (Form Length) 5-2
Columns 20-22 (Line Number–Overflow Line) 5-2
Columns 23-24 (Overflow Line) 5-2
Columns 25-74 5-2
Columns 75-80 (Program Identification) 5-2

CHAPTER 6. TELECOMMUNICATIONS SPECIFICATIONS 6-1
Columns 1-2 (Page) 6-2
Columns 3-5 (Line) 6-2
Column 6 (Form Type) 6-2
Columns 7-14 (Filename) 6-2
Column 15 (Configuration) 6-2
Column 16 (Type of Station) 6-2
Column 17 (Type of Control) 6-2
Column 18 (Type of Code) 6-2
Column 19 (Transparency) 6-3
Column 20 (Switched) 6-3
Columns 21-31 6-3
Column 32 (Location of Identification–This Station) 6-3
Columns 33-39 (Identification–This Station) 6-3
Column 40 (Location of Identification–Remote Station) 6-3
Columns 41-47 (Identification–Remote Station) 6-4
Columns 48-51 6-4
Column 52 (ITB) 6-4
Columns 53-54 (Permanent Error Indicator) 6-4
Columns 55-57 (Wait Time) 6-5
Columns 58-59 (Record Available Indicator) 6-5
Column 60 (Last File) 6-6
Columns 61-62 (Polling Characters) 6-6
Columns 63-64 (Addressing Characters) 6-6
Columns 65-74 6-6
Columns 75-80 (Program Identification) 6-6
File Description Specifications for BSCA Files 6-6
Columns 7-14 (Filename) 6-6
Column 15 (File Type) 6-6
Column 16 (File Designation) 6-7
Column 17 (End of File) 6-7
Column 19 (File Format) 6-7
Columns 20-23 (Block Length) 6-7
Columns 24-27 (Record Length) 6-7
Column 32 (File Organization or Addition I/O Area) 6-7
Columns 40-46 (Device) 6-7

CHAPTER 7. INPUT SPECIFICATIONS 7-1
FILE AND RECORD-TYPE IDENTIFICATION ENTRIES 7-2
Columns 1-2 (Page) 7-2
Columns 3-5 (Line) 7-2
Column 6 (Form Type) 7-2
Columns 7-14 (Filename) 7-2
Columns 14-16 7-2
Columns 15-16 (Sequence) 7-2
Assigning Sequence Numbers 7-2
Column 17 (Number) 7-6
Column 18 (Option) 7-6
Display Station Local Data Area 7-6
Columns 19-20 (Record Identifying Indicator, **,DS) 7-7
Record Identifying Indicators 7-7
Look-Ahead 7-10
Data Structures 7-17
Columns 21-41 (Record Identification Codes) 7-22
Position (Columns 21-24, 28-31, and 35-38) 7-23
Not (N) (Columns 25, 32, and 39) 7-23
C/Z/D (Columns 26, 33, and 40) 7-23
Character (Columns 27, 34, and 41) 7-23
AND Relationship 7-24
OR Relationship 7-24
Column 42 7-24

- FIELD DESCRIPTION ENTRIES 7-26
- Column 43 (Packed or Binary Field) 7-26
  - Zoned Decimal Format (Blank) 7-26
  - Packed Decimal Format (P) 7-27
  - Binary Format (B) 7-29
- Columns 44-51 (Field Location) 7-30
- Column 52 (Decimal Positions) 7-31
- Columns 53-58 (Field Name) 7-31
  - Field Names 7-31
  - Special Words (PAGE, PAGE1-PAGE7) 7-32
- Columns 59-60 (Control Level) 7-34
  - Assigning Control Level Indicators 7-34
  - Split Control Fields 7-36
- Columns 61-62 (Matching Fields) 7-36
  - Match Fields 7-36
  - Sequence Checking 7-39
- Columns 63-64 (Field Record Relation) 7-39
  - Record Identifying Indicators (01-99) 7-43
  - Control Level (L1-L9), Matching Record (MR), and External (U1-U8) Indicators 7-43
  - Halt Indicators (H1-H9) 7-43
- Columns 65-70 (Field Indicators) 7-43
  - Assigning Indicators in Columns 65-70 7-45
  - Numeric Indicators (01-99) 7-45
  - Halt Indicators (H1-H9) 7-45
- Columns 71-74 7-45
- Columns 75-80 (Program Identification) 7-45

- **CHAPTER 8. CALCULATION SPECIFICATIONS 8-1**
- Columns 1-2 (Page) 8-2
- Columns 3-5 (Lines) 8-2
- Column 6 (Form Type) 8-2
- Columns 7-8 (Control Level) 8-2
  - Control Level Indicators (L0, L1-L9) 8-3
  - Last Record Indicator (LR) 8-3
  - Subroutine Lines (SR) 8-6
- AND/OR Lines (AN, OR) 8-6
- Columns 9-17 (Indicators) 8-6
  - Field Indicators (01-99) 8-6
  - Command Key Indicators (KA-KN, KP-KY) 8-6
  - Record Identifying Indicators (01-99) 8-6
  - Resulting Indicators (01-99) 8-11
  - Control Level Indicators (L1-L9) 8-11
  - Last Record Indicator (LR) 8-11
  - Matching Record Indicator (MR) 8-11
  - Halt Indicators (H1-H9) 8-11
  - External Indicators (U1-U8) 8-11
  - Overflow Indicators (OA-OG, OV) 8-11
  - Relationship Between Columns 7-8 and Columns 9-17 8-11
- Columns 18-27 (Factor 1) and Columns 33-42 (Factor 2) 8-12
  - Literals 8-12
- Columns 28-32 (Operation) 8-19
- Columns 31-32 8-19
- Columns 43-48 (Result Field) 8-20
  - Erase 8-20
  - Field Name, Table Name, Array Name, Array Element, or Data Structure 8-20
- Columns 49-51 (Field Length) 8-21
- Column 52 (Decimal Positions) 8-22
- Column 53 (Half Adjust) 8-22
- Columns 54-59 (Resulting Indicators) 8-23
  - Test Results 8-23
  - Allowing Command Keys to be Pressed (SET) 8-25
  - Setting Indicators (SETON, SETOF) 8-25
- Columns 60-74 (Comments) 8-25
- Columns 75-80 (Program Identification) 8-25

- **CHAPTER 9. OUTPUT SPECIFICATIONS 9-1**
- Columns 1-2 (Page) 9-2
- Columns 3-5 (Line) 9-2
- Column 6 (Form Type) 9-2
- Columns 7-14 (Filename) 9-2
- Columns 14-16 (AND/OR) 9-2
- Column 15 (Type) 9-2
  - Heading Records (H) 9-2
  - Detail Records (D) 9-2
  - Total Records (T) 9-2
  - Exception Records (E) 9-2
- Columns 16-18 (ADD/DEL) 9-4
  - ADD 9-4
  - DEL 9-4
- Column 16 (Fetch Overflow or Release) 9-5
  - Fetch Overflow 9-5
  - Release 9-5
- Columns 17-22 (Spacing and Skipping) 9-6
  - Column 17 (Space Before) 9-6
  - Column 18 (Space After) 9-6
  - Columns 19-20 (Skip Before) 9-6
  - Columns 21-22 (Skip After) 9-6
- Columns 23-31 (Output Indicators) 9-8
  - AND and OR Lines 9-8
  - Command Key Indicators (KA-KN, KP-KY) 9-11
  - Overflow Indicators (OA-OG, OV) 9-11
  - First Page Indicator (1P) 9-13
  - Halt Indicators (H1-H9) 9-13
  - External Indicators (U1-U8) 9-13
- Columns 32-37 (Field Name) 9-13
  - Field Names 9-13
  - Special Words 9-17
- Column 38 (Edit Codes) 9-21
- Column 39 (Blank After) 9-22
- Columns 40-43 (End Position in Output Record) 9-22
- Column 44 (Packed or Binary Field) 9-26
- Columns 45-70 (Constant or Edit Word) 9-26
  - Constants 9-26
  - Format Name 9-28
  - Edit Codes 9-28
  - Edit Words 9-28
- Columns 71-74 9-37
- Columns 75-80 (Program Identification) 9-37

CHAPTER 10. OPERATION CODES 10-1
Arithmetic Operations 10-4
Move Operations 10-4
Move Zone Operations 10-5
Compare and Testing Operations 10-5
Bit Operations 10-5
SETON and SETOF Operations 10-5
Branching Within RPG II 10-6
Subroutine Operations 10-6
Linking to External Subroutines 10-6
WORKSTN Operations 10-6
Programmed Control of Input and Output 10-6
ALPHABETIZED OPERATION CODES 10-7
ACQ (Acquire) 10-7
ADD (Add) 10-7
BEGSR (Begin Subroutine) 10-7
BITOF (Set Bits Off) 10-8
BITON (Set Bits On) 10-8
CHAIN (Chain) 10-11
COMP (Compare) 10-13
DEBUG (Debug) 10-15
Records Written for DEBUG 10-15
DIV (Divide) 10-16
ENDSR (End Subroutine) 10-16
EXCPT (Exception Output) 10-16
EXIT (Exit to an External Subroutine) 10-18
EXSR (Execute Subroutine) 10-19
Coding Subroutines 10-19
FORCE (Force) 10-19
GOTO (Branch To) 10-22
KEY (Key) 10-25
LOKUP (Lookup) 10-30
LOKUP With One Table 10-31
LOKUP With Two Tables 10-31
Referencing the Table Element
Found in a LOKUP Operation 10-32
Using the LOKUP Operation with Arrays 10-32
MHHZO (Move High to High Zone) 10-38
MHLZO (Move High to Low Zone) 10-38
MLHZO (Move Low to High Zone) 10-38
MLLZO (Move Low to Low Zone) 10-38
MOVE (Move) 10-38
MOVEA (Move Array) 10-40
MOVEL (Move Left) 10-43
MULT (Multiply) 10-45
MVR (Move Remainder) 10-45
NEXT (Next) 10-46
POST (Post) 10-46
READ (Read a Record) 10-47
REL (Release) 10-47
RLABL (RPG II LABEL) 10-48
Message Retrieve Subroutine (SUBR23) 10-53
SET (Set) 10-54
Using SET Operations in Subroutines 10-58
User Message Member 10-58
Special Combinations of the SET and
KEY Operations 10-60
SETLL (Set Lower Limit) 10-60
SETOF (Set Off) 10-62
SETON (Set On) 10-62
SHTDN (Shut Down) 10-62
SORTA (Sort an Array) 10-63
SQRT (Square Root) 10-64
SUB (Subtract) 10-64
TAG (Tag) 10-64
TESTB (Test Bit) 10-67
TESTZ (Test Zone) 10-69
TIME (Time of Day) 10-69
XFOOT (Summing the Elements of an Array) 10-69
Z-ADD (Zero and Add) 10-69
Z-SUB (Zero and Subtract) 10-69

CHAPTER 11. MULTIFILE PROCESSING 11-1
No Match Fields 11-1
Match Fields 11-1
Assigning Match Field Values (M1-M9) 11-1
Processing Matching Records 11-3

CHAPTER 12. CONSOLE FILE CONSIDERATIONS . 12-1
Restrictions For Using a CONSOLE File 12-1
When Used With KEYBORD/CRT Device 12-1
Automatic Generation of Display Screen Formats . . . 12-1
Display Format 12-2
Prompt Format 12-5
Altering the Display Screen Format 12-5
Specifications For a CONSOLE File 12-5
File Description Specifications 12-5
Input Specifications 12-6
Erasing the CONSOLE File Buffer 12-7

CHAPTER 13. WORKSTN FILE CONSIDERATIONS AND SAMPLE PROGRAMS 13-1
Program Attributes 13-1
SRT (Single Requestor Terminal) Program 13-2
MRT (Multiple Requestor Terminal) Program 13-3
Attaching a Device to a Program 13-4
WORKSTN File Input Processing 13-4
End-of-File Considerations 13-8
Writing a Program with a WORKSTN File 13-9
Creating a Display Screen Format 13-9
Restrictions for Using a WORKSTN File 13-13
Command Keys 13-14
Control Specifications 13-14
File Description Specifications 13-14
Input Specifications 13-17
Calculation Specifications 13-19
Output Specifications 13-21
WORKSTN Exception/Error Handling 13-21
Specifications for the INFDS Data Structure . . . 13-25
Specifications for the INFSR Subroutine 13-29

Special Display Format Considerations 13-30
Overriding Fields in a Format 13-30
Read Under Format 13-31
Processing the Duplicate Character Value (Hex 1C) 13-31
Enabling/Disabling Command Keys 13-32
Enabling/Disabling Function Control Keys 13-32
IBM-Written Subroutines SUBR20 and SUBR21 . . . 13-32
Setting and Restoring External Indicators (SUBR20) 13-32
Reading and Writing the Display Station Local Data Area (SUBR21) 13-34
WORKSTN Sample Programs 13-34
Sample Program ITMINQ 13-34
Sample Program INQPUT 13-42
Sample Program INQRY2 13-46
Saving Fields (SAVDS) or Indicators (IND) for a WORKSTN File 13-49
Using MICs with a WORKSTN File 13-49
Debugging WORKSTN Programs 13-50

CHAPTER 14. TABLES AND ARRAYS 14-1
Rules For Creating Table or Array Input Records . . . 14-3
Defining Tables and Arrays 14-3
Loading Tables and Arrays 14-4
Compile-Time Tables and Arrays 14-4
Preexecution-Time Tables and Arrays 14-4
Execution-Time Arrays 14-4
Searching Tables and Arrays 14-7
Referencing Arrays 14-7
Array Name and Index 14-7
Referencing an Array in Calculations 14-8
Modifying Contents of Tables and Arrays 14-9
Adding Entries to Short Tables or Arrays 14-9
Table and Array Output 14-10
Editing Entire Arrays 14-10
Example of Using Tables 14-12
File Description Specifications 14-12
Extension Specifications 14-12
Input Specifications 14-12
Calculation Specifications 14-12
Examples of Using Arrays 14-14
Building an Array Using Field Indexes 14-14
Building an Array Using Fixed Indexes 14-14
Calculating Totals with Arrays 14-14
Using Arrays to Format Field Output 14-14
Printing Array Elements 14-20

CHAPTER 15. AUTO REPORT FUNCTION 15-1
Auto Report Generated Specifications 15-1
Format of the Generated Specifications 15-1
Order of Generated Specifications 15-5
Option Specifications 15-6
*AUTO Specifications 15-8
*AUTO Page Heading Specifications 15-8
Record Description Specifications 15-10
Field Description Specifications 15-11
*AUTO Output Specifications 15-12
Record Description Specifications 15-12
Field Description (Blank or B in Column 39) 15-13
Field Description (A in Column 39) 15-15
Field Description (C in Column 39) 15-18
Field Description (1-9 or R in Column 39) 15-20
Group Printing 15-21
/COPY Statement Specifications 15-22
Modifying Copied Specifications 15-26
Report Format 15-30
Spacing and Skipping 15-30
Placement of Headings and Fields 15-30
Generated Specifications 15-34
Generated Calculations 15-34
Generated Output Specifications 15-34
Programming Aids 15-39
Examples of Using Auto Report 15-44

CHAPTER 16. PROGRAMMING CONSIDERATIONS . 16-1
Storage Saving Techniques 16-1
Overlay Process 16-1
General Storage Saving Techniques 16-4
Performance Improvement Techniques 16-8
Coding Techniques for Calculation Operations 16-8

CHAPTER 17. DETAILED RPG II PROGRAM LOGIC . 17-1

CHAPTER 18. COMPILING AND EXECUTING RPG II PROGRAMS 18-1
Compiling the RPG II Source Program 18-1
RPG Command Statement 18-1
RPGX Command Statement 18-4
RPGR Command Statement 18-4
Compiling the Auto Report Source Program 18-8
AUTO Command Statement 18-8
Auto Report Halts 18-10
Cross-Reference Listing 18-10
Listing Format 18-10
Running the Object Program 18-11
RPG II Halt Procedures 18-11

CHAPTER 19. SAMPLE PROGRAMS 19-1
Sample Program 1 (SAMPL1) 19-1
Control Specifications 19-1
File Description Specifications 19-1
Input Specifications 19-1
Calculation Specifications 19-1
Output Specifications 19-1
Sample Program 2 (SAMPL2) 19-4
Control Specifications 19-4
File Description Specifications 19-4
Input Specifications 19-4
Calculation Specifications 19-4
Output Specifications 19-4
Sample Program 3 (SAMPL3) 19-4
Control Specifications 19-4
File Description Specifications 19-4
Input Specifications 19-10
Calculation Specifications 19-10
Output Specifications 19-10
Sample Program 4 (SAMPL4) 19-10
Control Specifications 19-10
File Description Specifications 19-10
Input Specifications 19-17
Calculation Specifications 19-17
Output Specifications 19-17
Sample Program 5 (SAMPL5) 19-17
Control Specifications 19-17
File Description Specifications 19-17
Input Specifications 19-17
Calculation Specifications 19-17
Output Specifications 19-17
Sample Auto Report Program (EXAUT2) 19-21
Control Specifications 19-21
/COPY Statements 19-21
Calculation Specifications 19-28
*AUTO Specifications 19-28
Sample WORKSTN File Program (ORD) 19-33
Expanding Sample WORKSTN File Program ORD 19-51

CHAPTER 20. IDEOGRAPHIC SUPPORT 20-1
Specifying Ideographic Data 20-1
Ideographic Constants 20-1
Ideographic Literals 20-2
Ideographic Fields, Tables, and Arrays 20-2
Ideographic Comments 20-3
Processing Considerations 20-3
Move Ideographic Data with Deletion of Control Characters (SUBR40) 20-4
Move Ideographic Data with Addition of Control Characters (SUBR41) 20-5
Ideographic Device Support 20-6
Messages 20-6

APPENDIX A. SUMMARY CHARTS A-1
Control Specifications A-1
File Description Specifications A-4
Extension Specifications A-9
Line Counter Specifications A-11
Telecommunications Specifications A-12
Input Specifications A-16
Calculation Specifications A-19
Output Specifications A-22
Option Specifications (Auto Report) A-25
Operation Codes A-26
Valid Indicators A-28
Summary of Indicators A-29
Display Screen Format Specifications A-33
S Specifications A-33
D Specifications A-37

APPENDIX B. PRINTED MESSAGES B-1
RPG II Messages B-1
RPG II Displayed Messages B-54
RPG II Auto Report Messages B-55
RPG II Linkage Editor Messages B-66

APPENDIX C. WORKSTN RETURN CODES C-1

GLOSSARY G-1

INDEX X-1

Chapter 1. Introduction

RPG II FUNCTIONS

RPG II consists of a symbolic programming language and a compiler program. The language is commercially oriented and specifically designed for writing application programs to produce reports that meet common business data processing requirements. The compiler diagnoses the source program, translates the source program into an executable object program, and supplies the logic that is the framework for the RPG II program.

STEPS IN USING RPG II

To use RPG II to prepare a report, you must follow the general steps shown in Figure 1-1. The numbers in the following text refer to the numbers on the figure:

1 You must analyze the report requirements to determine the format of the input files and the layout of the finished report. For example, determine what fields in the input records are to be used, what calculations are to take place, where the data comes from, where the data is to be located in the output records, and how many and what kind of totals must be accumulated.

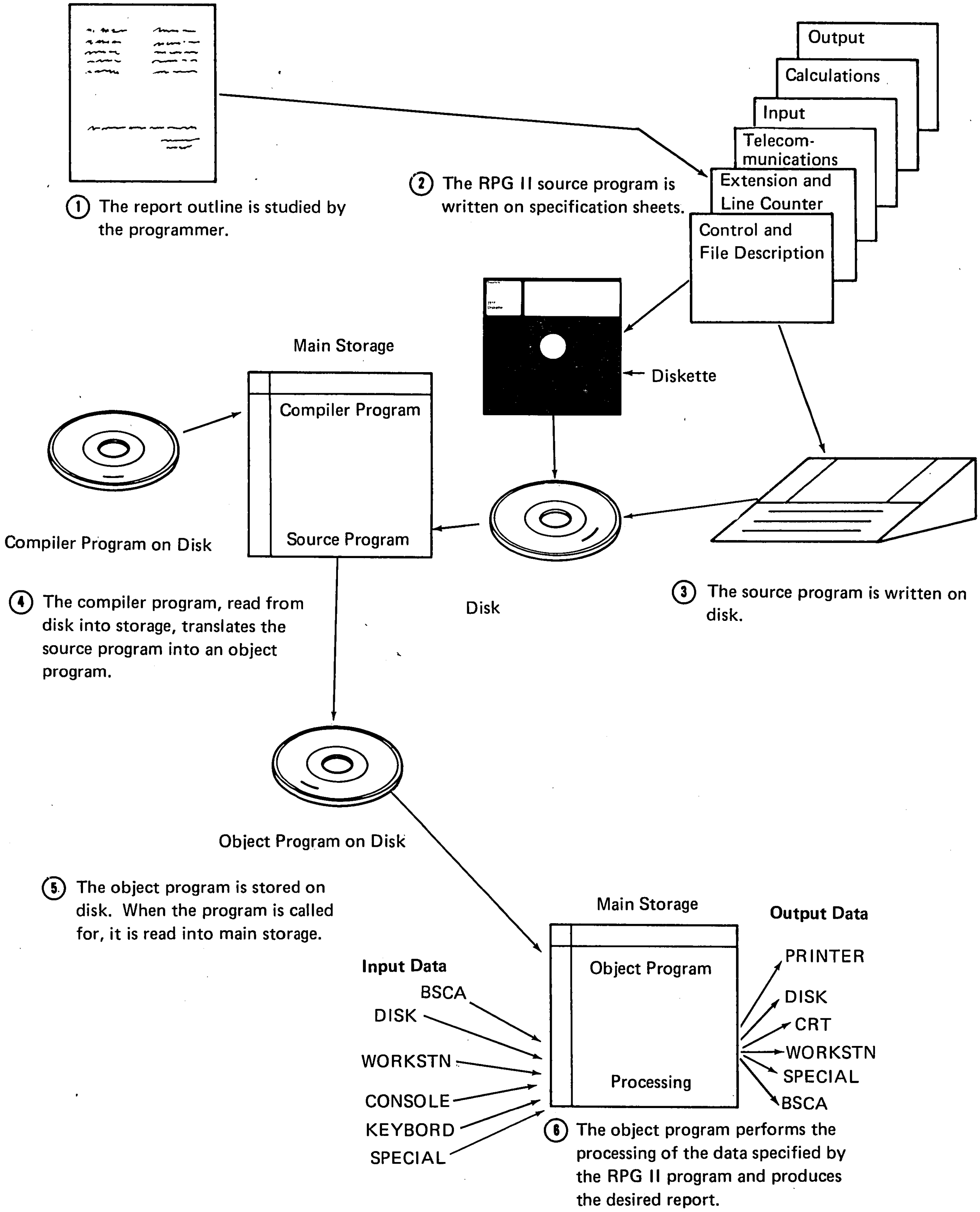
2 You must then provide the RPG II compiler with information about these requirements by coding the RPG II specification sheets. The specification sheets are designed so that one specification line represents one statement in the source program. The following specification sheets are used to code an RPG II program. For a brief description of the information contained on each specification sheet, see *Overview of Specification Sheets* in this chapter.
   a. Control and file description specifications
   b. Extension and line counter specifications
   c. Telecommunications specifications
   d. Input specifications
   e. Calculation specifications
   f. Output specifications

3 After the RPG II program has been coded on the specification sheets, the source program must be placed in a library on disk in the order shown in Figure 1-2. The source program can be placed on disk in one of the following ways:
   a. Use the source entry utility (SEU) program of System/34 Utilities program product to enter the source program from a display station. See the *SEU Reference Manual* for information about the source entry utility.
   b. Use the reader-to-library copy function of the utility program $MAINT and enter the source program from a display station. See the *System Support Reference Manual* for more information.
   c. If the source program is on diskette in basic data exchange format, use the TOLIBR command to move the source program to a library on disk. See the *System Support Reference Manual* for more information.

4 After the source program is entered into the system, the operator can run the RPG II compiler by entering the required procedure or command statements from the display station or by placing the job on the input job queue (see the *System Support Reference Manual*). The compiler processes the source program under the control of the system support program product.

5 If the compiler does not find any terminal errors in the program, the object program is produced and stored on disk. The object program contains all the machine instructions required to prepare the report.

6 The operator can now execute the object program by entering the required procedure or command statements from the display station or by placing the job on the input job queue (see the *System Support Reference Manual*). (See Chapter 18 for examples of the command statements.) The object program is read into the system and executed, and the report is produced.



(1) The report outline is studied by the programmer.

(2) The RPG II source program is written on specification sheets.

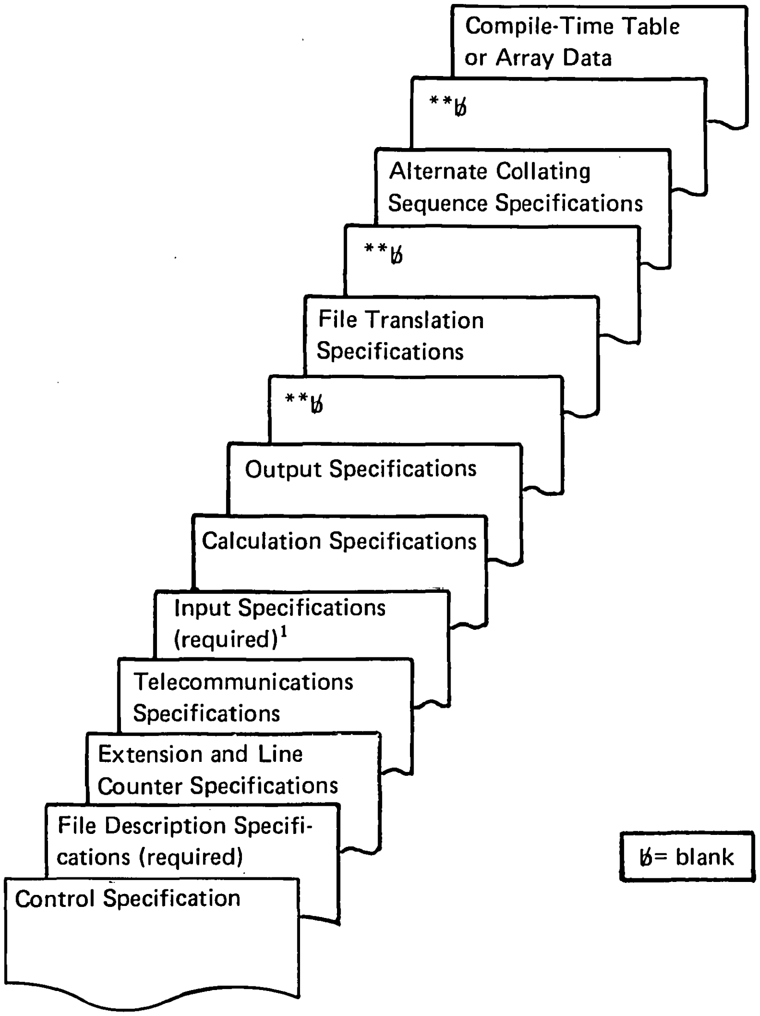
(3) The source program is written on disk.

(4) The compiler program, read from disk into storage, translates the source program into an object program.

(5) The object program is stored on disk. When the program is called for, it is read into main storage.

(6) The object program performs the processing of the data specified by the RPG II program and produces the desired report.

Figure 1-1. Preparation of a Report Using RPG II


Compile-Time Table or Array Data
**ƀ
Alternate Collating Sequence Specifications
**ƀ
File Translation Specifications
**ƀ
Output Specifications
Calculation Specifications
Input Specifications (required)[1]
Telecommunications Specifications
Extension and Line Counter Specifications
File Description Specifications (required)
Control Specification
ƀ= blank

[1] If the KEY operation is specified as the only input, input specifications are not required.

Figure 1-2. Required Order of Specifications for the RPG II Source Program

OVERVIEW OF SPECIFICATION SHEETS

Use the RPG II specification sheets to write your RPG II program. The specification sheets are designed in columns so that the information is presented in the format required by the RPG II compiler. Each specification line represents one statement in the source program. Each specification sheet and the information it provides to the compiler are briefly described in the following text.

Control Specifications

The control specifications provide the compiler with information about the program and describe the system you are using. The information includes:

- Storage size needed to execute the program
- Date format for the program
- Whether special RPG II functions such as alternate collating sequence or file translation are to be used
- Whether disk files will share a single input/output area
- Number of formats in the display screen format load member for a WORKSTN file

For a detailed description of the control specifications, see Chapter 2.

File Description Specifications

The file description specifications describe all files (for example, input files, output files, and combined files) that are used by the object program. The information for each file includes:

- Name of the file
- How the file is used
- Size of records in the file
- Input or output device used for the file
- Whether the file is conditioned by an external indicator

For a detailed description of the file description specifications, see Chapter 3.

Extension Specifications

The extension specifications describe all record address files, table files, and array files used in the program. The information includes:

- Name of the file
- Number of entries in a table or array input record
- Number of entries in a table or array
- Length of the table or array entry

For a detailed description of the extension specifications, see Chapter 4.

Line Counter Specifications

Line counter specifications indicate at what line overflow occurs and the length of the form used for each printer file in the program. Information for each printer file includes:

- Number of lines per page (length of form)
- Overflow line

For a detailed description of the line counter specifications, see Chapter 5.

Telecommunications Specifications

The telecommunications specifications describe each BSCA file used in the program. The information includes:

- Name of the file
- Description of the network used
- Type of station
- Type of control
- Type of code used
- Station identification

For a detailed description of the telecommunications specifications, see Chapter 6.

Input Specifications

The input specifications describe the records and fields in the input files used by the program. The information for each record includes:

- Name of the file
- Sequence of record types
- Indications for record identifying indicators, data structures, look-ahead fields, record identification codes, type of numeric fields, match fields, control level fields, field record relation, field indicators
- Location of fields in the record
- Name of field

For a detailed description of the input specifications, see Chapter 7.

Calculation Specifications

Calculation specifications describe the calculations to be performed on the data and the order in which they are to be performed. Calculation specifications can also be used to control certain input and output operations. The information includes:

- Control level and conditioning indicators for the operation specified
- The fields or constants to be used in the operation
- The operation to be performed
- Resulting indicators that are set after the operation is performed

For a detailed description of the calculation specifications, see Chapter 8.

Output Specifications

The output specifications describe the records and fields in the output files and the conditions under which output operations are performed. The information includes:

- Name of the file
- Type of record to be written
- Spacing and skipping instructions for printer and CRT files
- Output indicators that specify when the record is to be written
- Name of the field and location in the output record
- Editing specifications
- Constants
- Format name for the WORKSTN file

For a detailed description of the output specifications, see Chapter 9.

HOW RPG II WORKS

Each object program that the RPG II compiler generates goes through the same general cycle of operations. The phrase *program cycle* refers to the operations that are performed for each record read.

RPG II Program Cycle

The program cycle involves three basic logic steps:

- Reading information (input)
- Performing calculations (processing)
- Recording results (output)

Within each program cycle, these basic logic steps can be divided into numerous substeps in which you as the programmer can assign indicators to control when calculation and output operations occur.

According to RPG II program logic, calculation and output operations are performed at two different times in a cycle: total time and detail time (see Figure 1-3). First, the program performs total calculation operations (those conditioned by a control level indicator in columns 7 and 8 of the calculation specifications) and total output operations (those specified by a T in column 15 of the output specifications). Second, the program performs all detail calculation operations (those not conditioned by a control level indicator in columns 7 and 8 of the calculation specifications) and all detail output operations (those specified by a D or H in column 15 of the output specifications).

Total calculation and total output operations are generally performed on data accumulated for a group of related records that form a control group. A control group is a set of records all having the same information in a control field. Each time a record is read, the program checks the information in the control field to determine whether it differs from the control field information on the previous record. When the information differs from the previous record's control field, a control break occurs, indicating that all records from a particular group have been read and a new group is starting. When all records from a control group have been read, the program performs total calculation and output operations using the information accumulated from all records in that group. Information contained in the record that starts the new control group is not used in the total operations.

Detail calculation and detail output operations are generally performed for each record read if all conditioning indicators are satisfied. If either of the following conditions is met, detail calculation and output operations are performed:

- All total calculation and total output operations are completed, but the last record is not processed.
- No total operations are to be done (the information in the control field has not changed).

Figure 1-4 shows the specific steps in the general flow of the RPG II program cycle. A program cycle begins with step 1 and continues through step 11, then begins again with step 1. Steps 7 and 8 are known as total time, and steps 1 and 11 are known as detail time. The following statements describe the steps shown in Figures 1-3 and 1-4.

Step 1. If the conditioning indicators are satisfied, RPG performs the heading or detail output (those lines having an H or D in column 15 of the output specifications).

Step 2. RPG turns off all control level and record identifying indicators.

Step 3. RPG reads a record and turns on the appropriate record identifying indicator.

Step 4. RPG determines whether a control break occurred. (A control break occurs when the control field of the record just read differs from the control field of the previous record.)

Step 5. If a control break has occurred, RPG turns on the proper control level indicator and all lower control level indicators except L0, which is always on.

Step 6. If this is the first cycle, RPG goes to step 9.

Step 7. RPG performs total calculation operations (those conditioned by control level indicators in columns 7 and 8 of the calculation specifications) if the appropriate control level indicators are on.

Step 8. RPG performs total output operations (those lines having a T in column 15 of the output specifications) according to output specifications.

Step 9. RPG determines whether the LR indicator is on. If it is, all records have been processed, and the program ends.

Step 10. RPG makes data from the record read at the beginning of the cycle (step 3) available for use in detail calculation and output operations.

Step 11. RPG performs all detail calculation operations (those not conditioned by control level indicators in columns 7 and 8 of the calculation specifications) on the data from the record read at the beginning of the cycle.

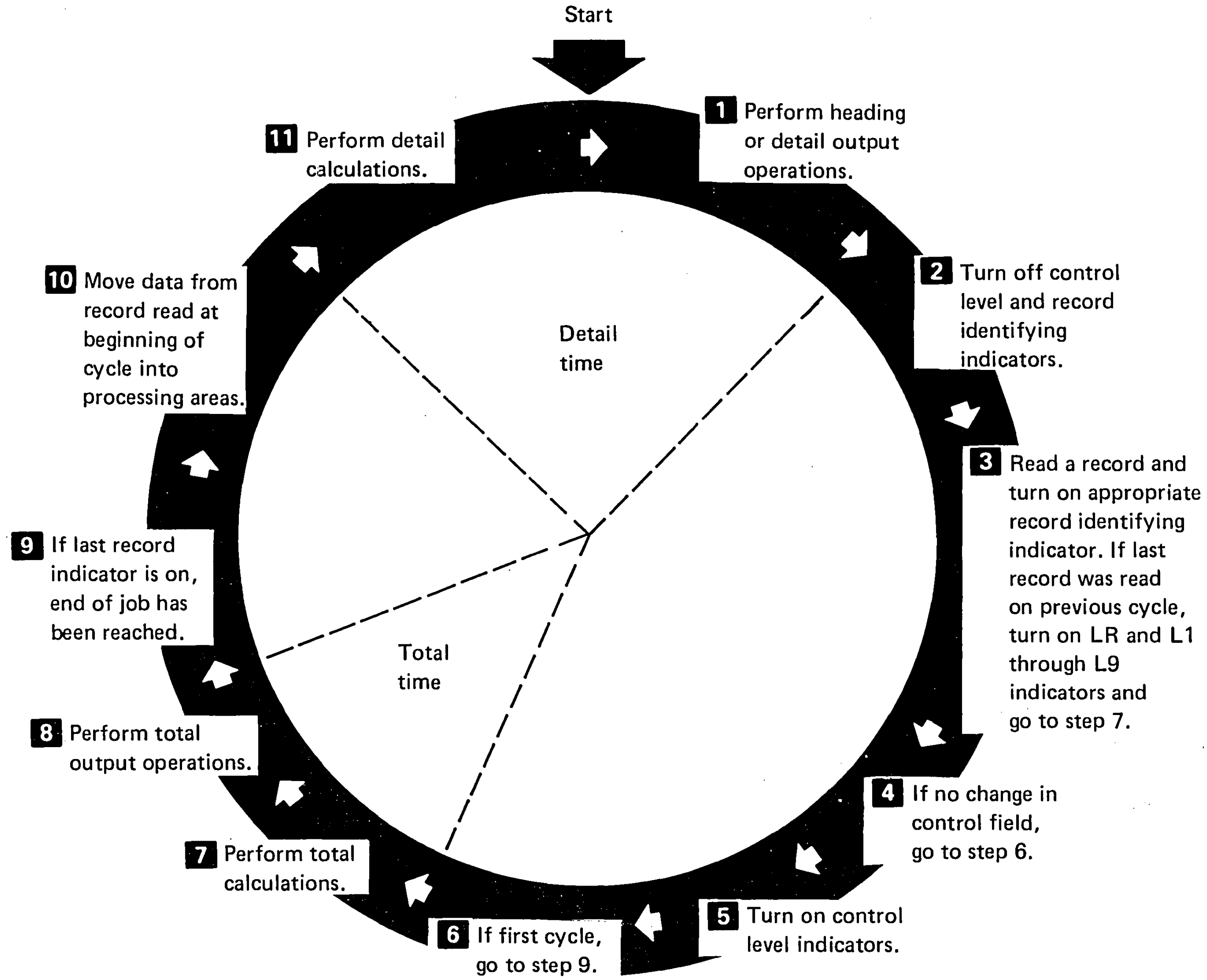

Start
1 Perform heading or detail output operations.
2 Turn off control level and record identifying indicators.
3 Read a record and turn on appropriate record identifying indicator. If last record was read on previous cycle, turn on LR and L1 through L9 indicators and go to step 7.
4 If no change in control field, go to step 6.
5 Turn on control level indicators.
6 If first cycle, go to step 9.
7 Perform total calculations.
8 Perform total output operations.
9 If last record indicator is on, end of job has been reached.
10 Move data from record read at beginning of cycle into processing areas.
11 Perform detail calculations.
Detail time
Total time

Figure 1-3. Steps in RPG II Logic, Showing Total Time and Detail Time

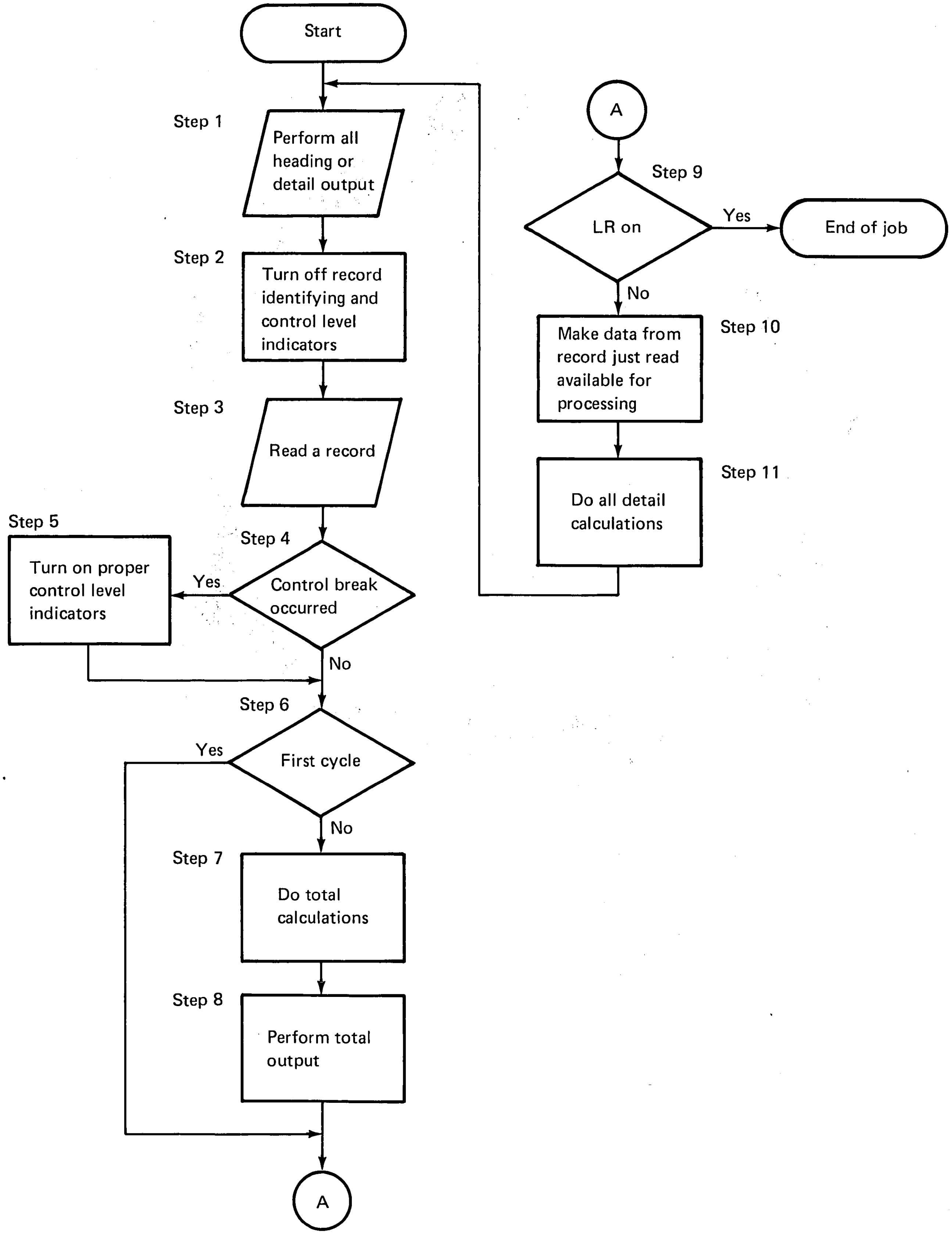

Start
Step 1
Perform all heading or detail output
Step 2
Turn off record identifying and control level indicators
Step 3
Read a record
Step 4
Control break occurred
Yes
No
Step 5
Turn on proper control level indicators
Step 6
First cycle
Yes
No
Step 7
Do total calculations
Step 8
Perform total output
A
A
Step 9
LR on
Yes
End of job
No
Step 10
Make data from record just read available for processing
Step 11
Do all detail calculations

Figure 1-4. RPG II Program Logic Cycle

The first and last cycles of a program differ somewhat from the other cycles. Before the first record is read in the first cycle, the program prints lines conditioned by the 1P (first page) indicator and also performs any heading or detail output operations having no conditioning indicators or all negative conditioning indicators. Heading lines printed before the first record is read might consist of constant or page heading information or fields for reserved words, such as PAGE and UDATE. In addition, the program bypasses total calculations and total output steps.

During the last program cycle, when no more records are available, the LR (last record) indicator turns on, automatically causing all control level indicators to turn on. The program performs the total calculations and total output, and the program ends.

For a detailed discussion of RPG II program logic, including the first and last cycles of a program, see Chapter 17, *Detailed RPG II Program Logic.*

Indicators

RPG II logic is built around indicators. Because the logic is set up to test the status of various indicators at specific times, the status (on or off) of indicators can be used to affect the sequence of a program's operations.

Usually indicators are set on or off by conditions in the program itself. However, you can also set certain indicators with the SETON, SETOF, and SET operation codes (see Chapter 10, *Operation Codes*). At the start of each program, all indicators are off except the 1P (first page) indicator, L0 indicator, and any external (U1 through U8) indicators that have been set on. At the beginning of each program cycle, only record identifying indicators (01 through 99, L1 through L9, and LR) and control level indicators (L1 through L9) are turned off; all other indicators remain unchanged. For WORKSTN files, command key indicators are turned off just before data from the record just read is made available for processing; all command key indicators are turned off except the one corresponding to the command key used.

Figure 1-5 shows the valid RPG II indicators and the specifications on which they can be used to condition operations. The following text briefly describes each type of indicator and where it can be specified. For a complete description of an indicator and how it can be used, see the appropriate column description in the applicable chapter of this manual.

01-99 (Field, Record Identifying, Resulting, and Conditioning Indicators)

You can assign any numbers from 01 to 99 to indicate:

- The type of record read. See *Columns 19-20 (Record Identifying Indicator,**, DS)* in Chapter 7, *Input Specifications.*
- The status (plus, minus, zero/blank) of an input field. See *Columns 65-70 (Field Indicators)* in Chapter 7, *Input Specifications.*
- The results of a calculation operation. See *Columns 54-59 (Resulting Indicators)* in Chapter 8, *Calculation Specifications.*

Any indicators that you assign in these columns or that you set on or off with a SETON or SETOF operation code can also:

- Establish field record relations. See *Columns 63-64 (Field Record Relation)* in Chapter 7, *Input Specifications.*
- Condition calculation operations. See *Columns 9-17 (Indicators)* in Chapter 8, *Calculation Specifications.*
- Condition output operations. See *Columns 23-31 (Output Indicators)* in Chapter 9, *Output Specifications.*

Indicators reflect only one condition at a time. When one indicator is used to reflect two or more conditions, it is always set to reflect the condition of the last operation performed.

If any indicator from 01 to 99 is set on or off by the operation code SETON or SETOF, it remains on or off until an instruction in a specification line containing that same indicator is performed. The indicator is then set to reflect the condition that results from the operation performed.

H1-H9 (Halt) Indicators

Use any halt indicators, H1 through H9, to:

- Stop the program when an unacceptable condition occurs.
- Condition calculation or output operations that are not to be performed when an unacceptable condition occurs. This conditioning is necessary; otherwise, all calculation and detail output operations are still performed (before processing stops) for the record that caused the unacceptable condition.
- Establish field record relations. See *Columns 63-64 (Field Record Relation)* in Chapter 7, *Input Specifications.*

Indicators	File Description Specifications		Input Specifications				Calculation Specifications			Output Specifications
	Overflow (33-34)	File Conditioning (71-72)	Record Identifying[1] (19-20)	Control Level (59-60)	Field Record Relation[1] (63-64)	Field (65-70)	Control Level (7-8)	Conditioning (9-17)	Resulting (54-59)	Conditioning (23-31)
01-99			X		X	X		X	X	X
H1-H9			X		X	X		X	X	X
1P										X[3]
MR					X[2]			X		X
OA-OG, OV	X							X	X	X[4]
L0							X			X
L1-L9			X	X	X[2]		X	X	X	X
LR			X				X	X	X	X
U1-U8		X[5]			X			X	X	X
KA-KN, KP-KY								X	X[6]	X

Note: X denotes the indicators that can be used.

[1] Not valid on look-ahead fields.
[2] When field named is not a match field or a control field.
[3] Only for detail or heading lines.
[4] Cannot condition an exception line, but can condition fields within the exception record.
[5] Not valid for table input files.
[6] Valid for SET, KEY, and SETOF operations only.

Figure 1-5. Valid Indicators

1P (First Page) Indicator

Use the 1P indicator to condition lines that are to be printed on only the first page. These lines are usually heading lines. Data printed for the heading lines is usually specified as constants in columns 45 through 70 of the output specifications.

All lines conditioned by the 1P indicator are printed before the first record from the input file is processed. Therefore, do not use the 1P indicator to condition output fields that require data from input records because the output will be meaningless. The 1P indicator cannot be used with a WORKSTN file or to condition calculation operations.

The 1P indicator is on at the beginning of the program and turns off after all lines conditioned by it are printed.

MR (Matching Record) Indicator

Use the MR indicator to condition calculation and output operations that are to be performed only when records match. The MR indicator turns on when a primary file record matches any secondary file record on the basis of the match field indicated by M1 through M9. The MR indicator is always set to reflect the match or nonmatch condition before any detail calculation operations are performed. If all primary file records match all secondary file records, the MR indicator is always on. (For a discussion of matching records, see Chapter 11, *Multifile Processing*.)

OA-OG, OV (Overflow) Indicators

Use overflow indicators for printer files primarily to condition the printing of heading lines. To use an overflow indicator in columns 23 through 31 of the output specifications, you must assign an overflow indicator to each printer file on the file description specifications in columns 33 and 34. This same indicator must then be used to condition all lines that are to be written to the associated printer only when overflow occurs.

L0 Indicator

You do not need to assign the L0 indicator because it is always on. Therefore, it is normally used only in columns 7 and 8 of the calculation specifications to specify that the calculation be done at total time. The L0 indicator cannot be set off with the SETOF operation code.

L1-L9 (Control Level) Indicators

Control level (L1 through L9) indicators signal when a change in a control field has occurred. Therefore, you can use them to condition operations that are to be performed only when all records with the same information in the control field have been read. You can also use them to condition total printing (last record of a control group) or to condition detail printing (first record in a control group). Control level indicators always turn on after the first record of a control group is read. Control level indicators can be used on input (columns 19 and 20, 59 and 60, and 63 and 64), calculation (columns 7 and 8, 9 through 17, and 54 through 59), and output (columns 23 through 31) specifications.

LR (Last Record) Indicator

Use the LR indicator to condition all operations that are to be done only at the end of the job. For all primary or secondary files (except KEYBORD), the LR indicator normally turns on when the last record is detected. No record identifying indicators will be on while last record processing is performed for these files. When LR turns on, all other control level indicators (L1 through L9) also turn on.

For KEYBORD files and demand files, the LR indicator must be turned on at the appropriate time in calculation specifications. Record identifying indicators can be on while last record processing is performed for these files. When LR is turned on in detail calculations, all other control level indicators turn on at the beginning of the next cycle. LR and the record identifying indicators are both on throughout the remainder of the detail cycle, but the record identifying indicators are turned off before LR total time.

Do not specify an L0 through L9 indicator in an OR relationship with an LR indicator because the specified operations will be done twice at LR time.

All total lines conditioned by LR are performed last. The program ends after all total records are written.

U1-U8 (External) Indicators

The external indicators U1 through U8 are normally set prior to processing by an operation control language (OCL) statement (SWITCH). Their setting can be changed during processing, allowing the program to alter the status of these indicators.

The external indicators can be used:

- To determine whether a file is to be used for a job (see *Columns 71-72* in Chapter 3, *File Description Specifications*)

- To condition calculation operations

- To condition output operations

- As field record relation indicators

For a discussion of external indicators when used with a WORKSTN file, see Chapter 13, *WORKSTN File Considerations and Sample Programs.*

KA-KN, KP-KY (Command Key) Indicators

Assign command key indicators to specify what command keys the operator can press for a SET operation (see *SET* in Chapter 10, *Operation Codes*).

All 24 command keys can be used for a WORKSTN file. You can use the command key indicators to condition calculation and output operations. To document the use of the command keys for the operator, you can use the template assignment form on the *IBM 5251 Display Station Keyboard Template Assignment Sheet and Display Screen Layout Sheet.*

Twenty-four command keys are designated for the top row of the keyboard. In the lowercase position, key 1 corresponds to command key indicator KA, key 2 to KB, ... - (minus) to KK, and = (equal) to KL. In the uppercase position, key | corresponds to command key indicator KM, @ to KN, ... and + to KY. Any of the command key indicators that can be used in a SET or SETOF operation or for a WORKSTN file can then be used to:

- Condition calculation operations in columns 9 through 17 of the calculation specifications

- Condition output operations in columns 23 through 31 of the output specifications

When the command key indicators are used as conditioning indicators in the preceding columns, they are turned on and off in the following manner:

- They are turned on when the appropriate command key is pressed for a SET operation.

- They are turned off when the specified command key is not pressed for the SET operation or when a SETOF operation is performed.

When RPG makes the data read from a display station attached to a WORKSTN file available for processing, all command key indicators are set off. If a command key was pressed when the data was read into the program, the corresponding command key indicator is set on.

COMMON ENTRIES

Columns 1 through 7 and 75 through 80 are common to all RPG specification sheets. The entries that can be made in these columns are described in the following text.

Columns 1-2 (Page)

Entry	Explanation
Blank	No page number is used
01-99	Page number

Use columns 1 and 2 in the upper right corner of each sheet to number the specifications sheets, in ascending order, for your job. You can use more than one of each type of sheet, but keep all sheets of the same type together. When all specifications sheets are filled out, arrange them in the order shown in Figure 1-2.

Columns 3-5 (Line)

Entry	Explanation
Blank	No line number is used.
Any numbers	Line numbers.

Use columns 3 through 5 to number the lines on each page. Columns 3 and 4 are preprinted on each sheet so, in most cases, line numbering is already done. For instance, the calculation specifications sheet is prenumbered for lines 01 through 20. Below the prenumbered lines you can add or insert up to five more specifications (see Figure 1-6).

Page and line numbers are optional entries and are not required to successfully compile an RPG II program. Columns 1 through 5 are checked for ascending sequence, and RPG II prints an S in the left margin of the RPG II listing for any statement that is out of sequence. If you use SEU to enter the source program, you can request that SEU serialize the statements.

The control specifications line is always line 01. Any other lines on the sheets can be skipped. The line numbers used need not be consecutive, but should be in ascending order.

Column 6 (Form Type)

Entry	Explanation
H	Control (header) specifications
F	File description specifications
E	Extension specifications
L	Line counter specifications
T	Telecommunications specifications
I	Input specifications
C	Calculation specifications
O	Output specifications

Column 6 contains a preprinted letter on all sheets. The letter identifies the type of specifications for each line of coding. The H in column 6 of the control specifications stands for header or control record. The control specification must always be the first record in the RPG II source program (see Figure 1-2).

Column 7 (Comments)

Entry	Explanation
*	Comment line

Use an asterisk in column 7 to identify the line as a comment line. Use comments throughout your program to help document the purpose of a certain section of coding. Any character can be used in a comment line. Comments are not instructions to the RPG II program; they serve only as a means of documenting your program.

Columns 7-12 (/EJECT)

Entry	Explanation
/EJECT	The specifications following this entry are to begin on a new page of the compiler listing.

The /EJECT specification is not printed on the compiler listing.

Columns 7-12 (/TITLE)

Entry	Explanation
/TITLE	The heading information (such as a title or security classification) that follows the /TITLE entry appears at the top of each page of the compiler listing. The heading information is entered in columns 14 through 74.

A program can contain more than one /TITLE statement. Each /TITLE statement provides heading information for the compiler listing until the next /TITLE statement is encountered. To print on the first page of the compiler listing, a /TITLE statement must be the first statement encountered. Information specified by the /TITLE statement is printed in addition to compiler heading information.

The /TITLE statement causes an eject to the next page before the title is printed. The /TITLE statement is not printed on the compiler listing.

Columns 7-14 (/SPACE)

Entry	Explanation
/SPACEƀn	Line spacing occurs at this point in the compiler listing. Valid entries for n are 1 to 3. If n is not specified, 1 is assumed.

One blank (ƀ) must precede the value you specify for n. The value you specify for n indicates the number of blank lines to be spaced before the next specification line is printed. If n is greater than the number of lines remaining on the current page, the next specification line is printed on a new page. If just /SPACE is specified, one line is spaced.

/SPACE is not printed on the compiler listing; it is replaced by the appropriate line spacing. The spacing indicated by /SPACE is in addition to the two blank lines that occur between specification types.

Columns 75-80 (Program Identification)

Entry	Explanation
Blank	RPGOBJ is the program identification assigned.
Any valid program name	Program identification. The first character must be alphabetic but cannot be #, $, or @. The remaining characters must be alphameric with no imbedded blanks. No special character can be used.

Control Specifications

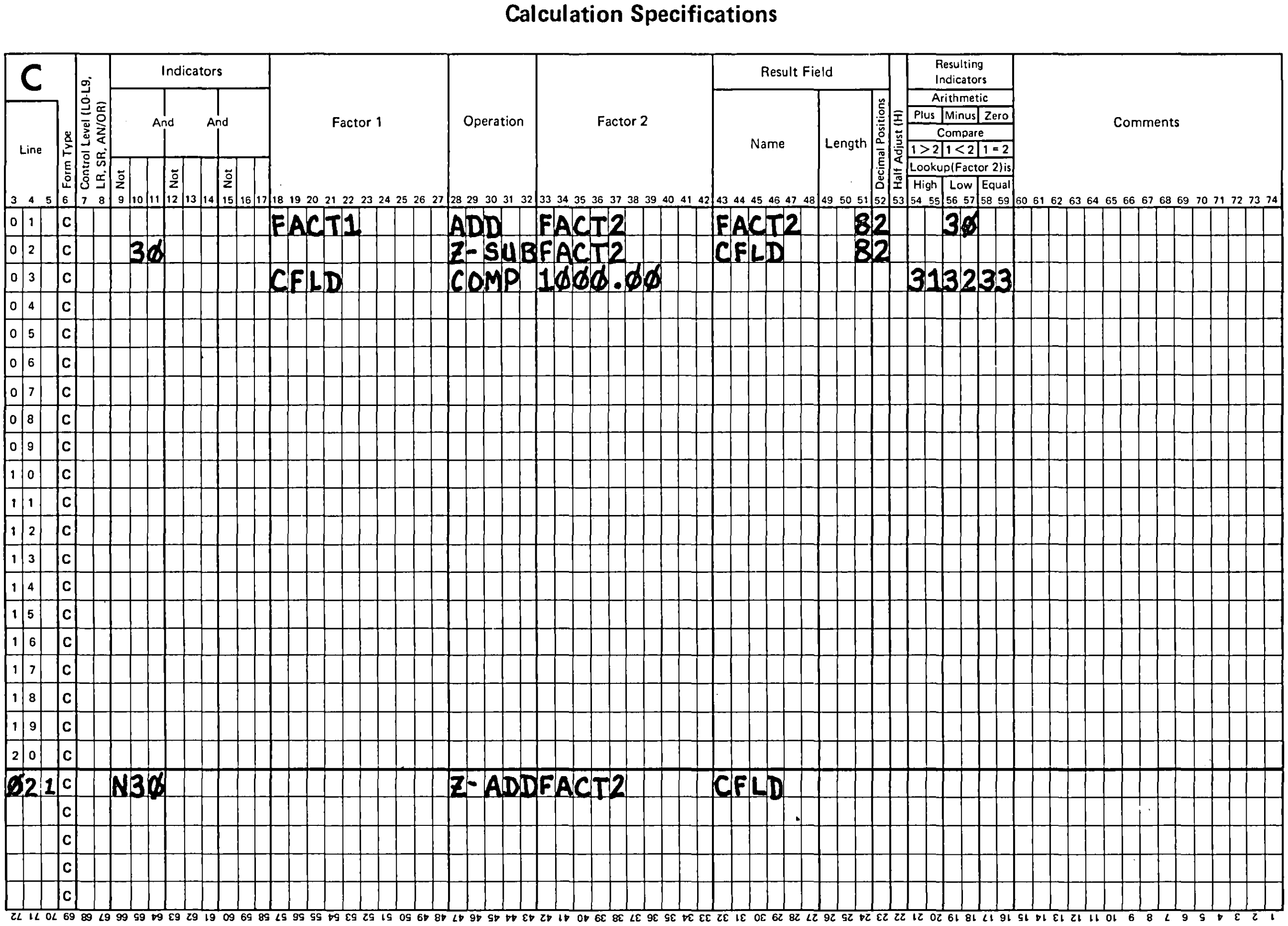
Columns 75 through 80 of the control specifications are used by the compiler to name the object program and to identify each record in the object program. See *Columns 75-80 (Program Identification)* in Chapter 2 for a complete description of how the compiler uses this entry on the control specifications.

All Other Source Program Specifications

Columns 75 through 80 of all source program specifications, except the control specifications, can contain any characters. These columns can contain the program name used in the control specifications, or they can contain any other characters to identify a certain portion of the program. These entries are ignored by the compiler but appear in the source program listing.

Note: To be compatible with other RPG systems, the specifications sheets show only 80 positions for each statement. However, each statement in an RPG II source program can contain up to 96 characters. Columns 81 through 96 are available for comments.

To code a line that must be inserted between two completed lines, enter the preceding line number in columns 3 and 4 of the line following the last line with a printed line number. Then enter a number from 1 to 9 in column 5. A maximum of nine lines can be inserted between two specification lines.



Calculation Specifications

C Line (3-5)	Form Type (6)	Control Level (L0-L9, LR, SR, AN/OR) (7-8)	Indicators: Not, And, Not, And, Not (9-17)	Factor 1 (18-27)	Operation (28-32)	Factor 2 (33-42)	Result Field Name (43-48)	Length (49-51)	Decimal Positions (52)	Half Adjust (H) (53)	Resulting Indicators: Plus / 1>2 / High (54-55)	Minus / 1<2 / Low (56-57)	Zero / 1=2 / Equal (58-59)	Comments (60-74)
01	C			FACT1	ADD	FACT2	FACT2	8	2			3Ø		
02	C		3Ø		Z-SUB	FACT2	CFLD	8	2					
03	C			CFLD	COMP	1ØØØ.ØØ					31	32	33	
04	C													
05	C													
06	C													
07	C													
08	C													
09	C													
10	C													
11	C													
12	C													
13	C													
14	C													
15	C													
16	C													
17	C													
18	C													
19	C													
20	C													
Ø21	C		N3Ø		Z-ADD	FACT2	CFLD							
	C													
	C													
	C													
	C													

Figure 1-6. Insertion of Coding Lines

Control specifications provide special information about your program and describe the computer system to the RPG II compiler. One control specifications statement is required for every program. If the control specifications statement is omitted from the source program, a blank control specification is assumed by the compiler. See the individual column descriptions for the meaning of blank entries.

Write the control specifications on the first line of the RPG Control and File Description Specifications sheet (see Figure 2-1). The control specifications should always be the first statement in the source program.

COLUMNS 1-2 (PAGE)

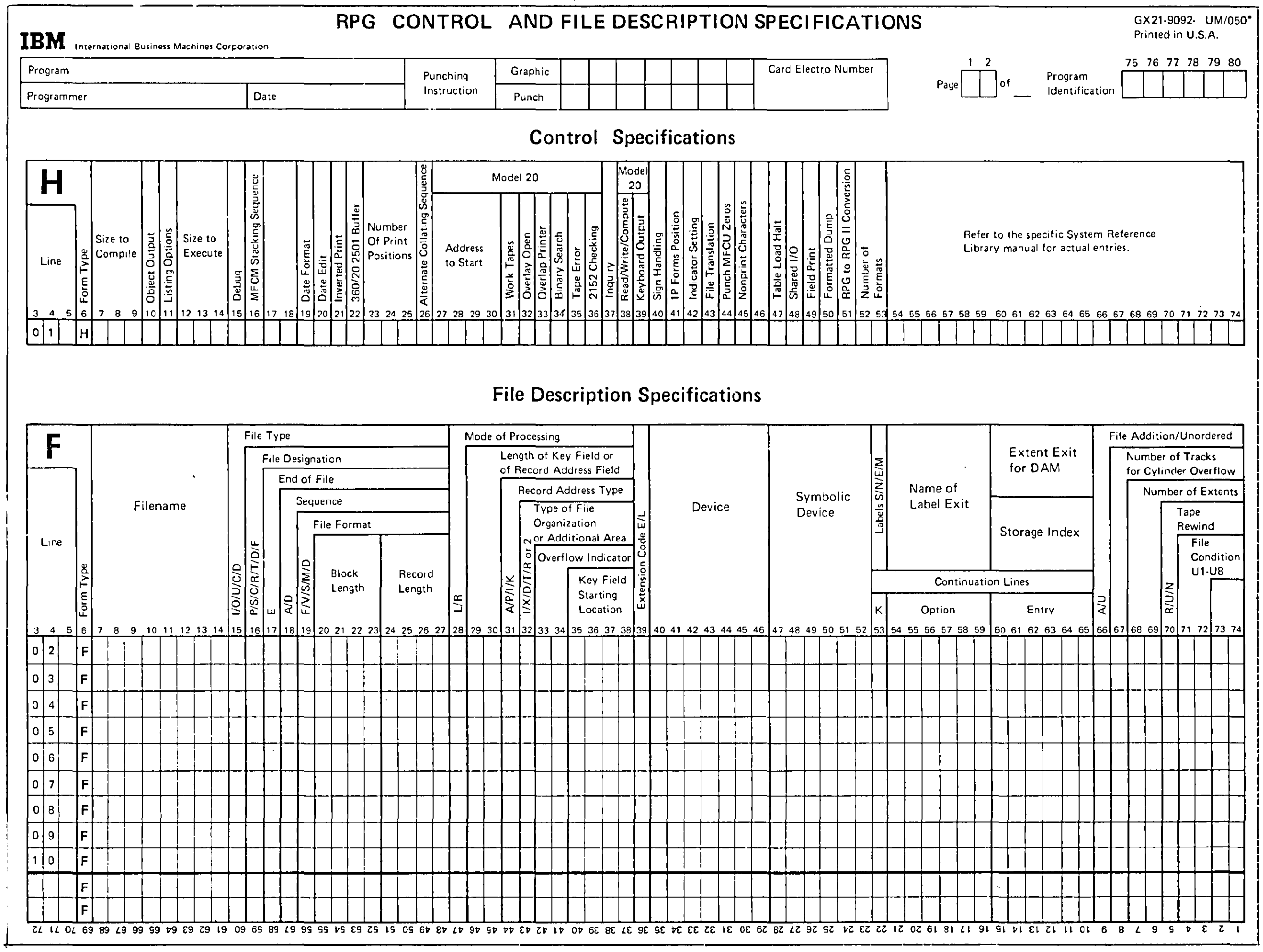
See *Common Entries* in Chapter 1.

COLUMNS 3-5 (LINE)

See *Common Entries* in Chapter 1.

COLUMN 6 (FORM TYPE)

An H must appear in column 6 to identify this line as the control specifications (or header) statement.



IBM International Business Machines Corporation

RPG CONTROL AND FILE DESCRIPTION SPECIFICATIONS

GX21-9092- UM/050*
Printed in U.S.A.

Program | Punching Instruction | Graphic | Card Electro Number | Page 1 2 of | Program Identification 75 76 77 78 79 80
Programmer | Date | Punch

Control Specifications

H

Line 3 4 5 | Form Type 6 | Size to Compile 7 8 9 | Object Output 10 | Listing Options 11 | Size to Execute 12 13 14 | Debug 15 | MFCM Stacking Sequence 16 | 17 18 | Date Format 19 | Date Edit 20 | Inverted Print 21 | 360/20 2501 Buffer 22 | Number Of Print Positions 23 24 25 | Alternate Collating Sequence 26 | Model 20: Address to Start 27 28 29 30 | Work Tapes 31 | Overlay Open 32 | Overlap Printer 33 | Binary Search 34 | Tape Error 35 | 2152 Checking 36 | Inquiry 37 | Model 20: Read/Write/Compute 38 | Keyboard Output 39 | Sign Handling 40 | 1P Forms Position 41 | Indicator Setting 42 | File Translation 43 | Punch MFCU Zeros 44 | Nonprint Characters 45 | 46 | Table Load Halt 47 | Shared I/O 48 | Field Print 49 | Formatted Dump 50 | RPG to RPG II Conversion 51 | Number of Formats 52 53 | Refer to the specific System Reference Library manual for actual entries. 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74

0 1 H

File Description Specifications

F

Line 3 4 5 | Form Type 6 | Filename 7 8 9 10 11 12 13 14 | File Type: I/O/U/C/D 15 | File Designation: P/S/C/R/T/D/F 16 | End of File: E 17 | Sequence: A/D 18 | File Format: F/V/S/M/D 19 | Block Length 20 21 22 23 | Record Length 24 25 26 27 | Mode of Processing: L/R 28 | Length of Key Field or of Record Address Field 29 30 | Record Address Type: A/P/I/K 31 | Type of File Organization or Additional Area: I/X/D/T/R or 2 32 | Overflow Indicator 33 34 | Key Field Starting Location 35 36 37 38 | Extension Code E/L 39 | Device 40 41 42 43 44 45 46 | Symbolic Device 47 48 49 50 51 52 | Labels S/N/E/M 53 | Name of Label Exit 54 55 56 57 58 59 | Extent Exit for DAM / Storage Index 60 61 62 63 64 65 | File Addition/Unordered: A/U 66 | Number of Tracks for Cylinder Overflow 67 | Number of Extents 68 69 | Tape Rewind: R/U/N 70 | File Condition U1-U8 71 72 | 73 74

Continuation Lines: K 53 | Option 54 55 56 57 58 59 | Entry 60 61 62 63 64 65

0 2 F
0 3 F
0 4 F
0 5 F
0 6 F
0 7 F
0 8 F
0 9 F
1 0 F
F
F

Figure 2-1. RPG Control and File Description Specifications

COLUMNS 7-9 (SIZE TO COMPILE)

Columns 7 through 9 are not used. Leave them blank. Any entry in these columns is ignored by the compiler. The program is compiled in the available storage specified by the REGION OCL statement. If no region size is specified, the size to compile defaults to 14K.

COLUMN 10 (OBJECT OUTPUT)

Entry	Explanation
Blank	The system halts only when severe (terminal) errors are found.
D	The system halts for both warning errors and severe errors. The operator can continue the job after a halt occurs for a warning error.

Use column 10 to indicate whether the system is to halt for warning errors.

The compiler produces the object program if no severe (terminal) errors are detected in the source statements. This object program is written in the library specified by the OUTLIB parameter on the COMPILE OCL statement. If the OUTLIB parameter is not specified, the object program is written to the system library. The object program remains in the specified library until it is deleted by the programmer. Each object program in a library must be assigned a unique program name. See *Columns 75-80* in this chapter for a detailed discussion of naming the program.

COLUMN 11 (LISTING OPTIONS)

Entry	Explanation
Blank	The object program is produced if no severe errors are found, and an RPG II listing is printed.
B	The object program is produced if no severe errors are found, but no program listing is printed. Use this entry to produce an object program for which you already have a listing.
P	The object program is produced if no severe errors are found, and a partial program listing is printed.

Use column 11 to specify the program listing option to be used when your program is compiled. If any severe errors are found during compilation and if column 11 contains a blank or P, the listing is completed and the system halts.

The RPG II listing consists of the source program listing, table and array information, indicator usage information, the relative location of fields and their attributes, unreferenced field names, diagnostics, and a main storage usage map. The main storage usage map lists the identification, the start address, and the size of each uniquely identifiable segment of code in the object program, defines the amount of main storage required for execution, and lists the number of library sectors required for the object program.

The partial listing includes the source program, indicator usage information, and diagnostics. Excluded from this listing are table and array information, field information, and main storage usage map.

COLUMNS 12-14 (SIZE TO EXECUTE)

Column 12

Entry	Explanation
Blank or 0	The entry in columns 13 and 14 determines the size to execute.
Q, H, or T	The entry in columns 13 and 14 is rounded up to the next even number.

Columns 13 and 14

Entry	Explanation
Blank	The main storage available for object program execution defaults to the region size specified. If no region size is specified, the size to execute defaults to 14K.
02-64	Enter the main storage available for object program execution in a multiple of 2K bytes (K = 1,024 bytes).

Use columns 12 through 14 to specify the amount of main storage to be available for object program execution. The maximum amount of storage you can specify depends on the system size. If column 12 contains a Q, H, or T or if columns 13 and 14 contain an odd number, RPG II rounds the entry in columns 13 and 14 up to the next even number. For example, an entry of Q04 or 005 is rounded up to 006.

The compiled object program occupies up to the amount of main storage specified in columns 13 and 14. The actual amount of storage the program occupies after it is compiled appears on the RPG listing.

COLUMN 15 (DEBUG)

Entry	Explanation
Blank	DEBUG operation is not performed.
1	DEBUG operation is performed.

Use column 15 to indicate whether the DEBUG operation is to be performed. To perform a DEBUG operation:

- A 1 must appear in column 15 when the source program is compiled.
- The DEBUG operation code must be used in the calculation specifications.

See *DEBUG Operation* in Chapter 10 for more information.

COLUMNS 16-17

Columns 16 and 17 are not used. Leave them blank.

COLUMN 18 (CURRENCY SYMBOL)

Entry	Explanation
Blank	Defaults to $ as the currency symbol
Any other character	This character is used as the currency symbol. If a fixed or floating currency symbol is desired, this symbol must be appropriately coded in edit words or in conjunction with the appropriate edit codes. Any character may be specified as the currency symbol except the following characters which have a special meaning in edit codes or words: 0 (zero) * (asterisk) , (comma) & (ampersand) . (decimal point) – (minus sign) C (letter C) R (letter R)

COLUMNS 19-20 (DATE OPTION)

Column 19 (Date Format)

Entry	Explanation
Blank	Defaults to month/day/year if column 21 is blank. Defaults to day/month/year if column 21 contains a D, I, or J.
M	Month/day/year.
D	Day/month/year.
Y	Year/month/day.

Use column 19 to specify the date format for UDATE. The date format specified in column 19 should be the same format as the program date. For example, if column 19 is blank, the input (the program date) must be mm/dd/yy if column 21 is blank or dd/mm/yy if column 21 contains a D. If you specify the date in mm/dd/yy format and the program date in the system is in dd/mm/yy format, RPG will work with an invalid date.

If data containing the UDATE field is transmitted to, or used by, another system, the UDATE format must be yy/mm/dd.

For a description of the program date, see the *System Support Reference Manual.*

Column 20 (Date Edit)

Entry	Explanation
Blank	(1) A slash (/) is assumed when column 21 contains a blank or D and column 19 is blank, or when column 19 contains M; (2) a period (.) is assumed when column 21 contains I or J and column 19 is blank, or when column 19 contains D or Y.
&	A blank separates the date field.
Any other characters	The character entered separates the edited date field.

Use column 20 to specify the type of edited output that appears for the Y edit code, which is specified on the output specifications. For an example of how the entries in columns 19 through 21 affect the editing of date fields, see *Column 38 (Edit Codes)* in Chapter 9, *Output Specifications.*

COLUMN 21 (INVERTED PRINT)

Entry	Explanation
Blank	Decimal periods are used for numeric literals and editing. UDATE format is mmddyy if column 19 is blank. If columns 19 and 20 are blank, a slash (/) is used for the Y edit code.
I	Decimal commas are used for numeric literals and editing. UDATE format is ddmmyy if column 19 is blank. If columns 19 and 20 are blank, a period (.) is used for the Y edit code.
J	J is the same as I except zero is written to the left of the decimal comma when the field contains a fraction. Nondecimal edited fields print with a zero in the low-order (units) position.
D	D is the same as blank except the UDATE format is ddmmyy if column 19 is blank.

Use column 21 to specify the constants to be used with RPG II edit codes that are entered on the output specifications. Decimal period means that numbers are edited with a period before the fraction (183.55) and with a comma denoting thousands (1,435). Decimal comma means that numbers are edited with a comma before the fraction (183,55) and with a period denoting thousands (1.435).

For information on how the entries in column 21 are used to format numeric data, see *Column 38 (Edit Codes)* in Chapter 9, *Output Specifications.*

COLUMNS 22-25

Columns 22 through 25 are not used. Leave them blank.

COLUMN 26 (ALTERNATE COLLATING SEQUENCE)

Entry	Explanation
Blank	Normal collating sequence is used.
S	Alternate collating sequence is used.

Use column 26 only to alter the normal collating sequence for alphameric compare operations, sequence checking, or match fields.

Normal Collating Sequence

Each of the alphabetic, numeric, and special characters used by System/34 is represented in the computer by a hexadecimal value (see Figure 2-2). The order in which these characters appear in the computer is based on this hexadecimal value and is known as the normal collating sequence. For an RPG II program, the system uses this collating sequence only in alphameric compare operations, in sequence checking, and for match fields. You can alter this collating sequence for use in these three types of operations. For example, you can insert a character between 2 consecutive characters, take a character out of sequence, or make 2 characters equal.

Specifying an Alternate Collating Sequence

To specify that an alternate collating sequence is to be used, enter an S in column 26. Describe each character whose collating sequence is to be changed on the Translation Table and Alternate Collating Sequence Coding Sheet (see Figure 2-3). Then transcribe the sequence changes into the correct record format for entry into the system. These records, called the alternate collating sequence table records, must be entered after the RPG II specifications and, if used, the file translation table records. The alternate collating sequence table is a special table that requires no file description or extension specifications and is printed with the compiled program.

Filling Out the Alternate Collating Sequence Coding Sheet

Specify each change in the collating sequence in the *Replaced By* column on the alternate collating sequence coding sheet. In the *Graphic* column find the character whose normal sequence is to be changed, and enter the hexadecimal value of the replacing character in the *Replaced By* column.

For example, if a blank is to be replaced by (or considered the same as) a zero, enter F0 (the hexadecimal value of zero) in the *Replaced By* column next to the hexadecimal value of a blank (40) (see Figure 2-4). Thus, whenever the system reads a blank and uses it in an alphameric compare, match field, or sequence checking operation, the blank is considered the same as a zero.

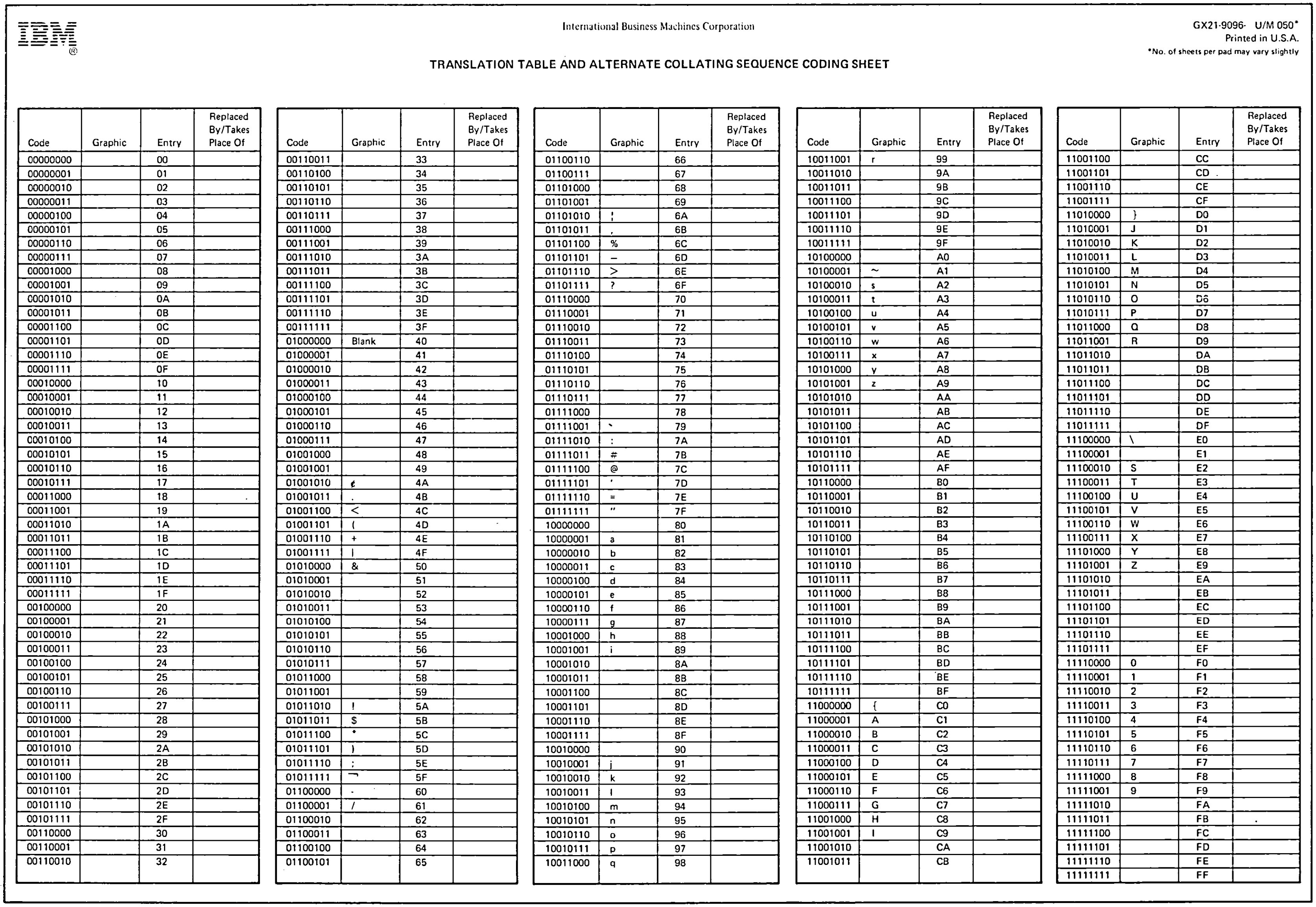
If a character is to be inserted between 2 consecutive characters, you must specify every character that is altered by this insertion. For example, if the dollar sign ($) is to be inserted between A and B, specify the changes for characters B through I on the coding sheet (see Figure 2-4).

The translation table and alternate collating sequence coding sheet show many hexadecimal values between the characters I and }, R and S, and Z and 0 that have no printable graphic associated with them. Because of this arrangement of nongraphics in the collating sequence, a character such as $, when inserted between A and B, changes only the position of characters B through I. Because the I replaces the nongraphic represented by hexadecimal CA, the remaining characters in the collating sequences are not affected.

Collating Sequence	Character	Hexadecimal Value
1	Blank	40
2	¢	4A
3	.	4B
4	<	4C
5	(	4D
6	+	4E
7	|	4F
8	&	50
9	!	5A
10	$	5B
11	*	5C
12	)	5D
13	;	5E
14	¬	5F
15	- (minus)	60
16	/	61
17	¦	6A
18	,	6B
19	%	6C
20	_(underscore)	6D
21	>	6E
22	?	6F
23	`	79
24	:	7A
25	#	7B
26	@	7C
27	'	7D
28	=	7E
29	"	7F
30	a	81
31	b	82
32	c	83
33	d	84
34	e	85
35	f	86
36	g	87
37	h	88
38	i	89
39	j	91
40	k	92
41	l	93
42	m	94
43	n	95
44	o	96
45	p	97
46	q	98
47	r	99
48	~	A1
49	s	A2
50	t	A3
51	u	A4
52	v	A5
53	w	A6
54	x	A7
55	y	A8
56	z	A9
57	{	C0
58	A	C1
59	B	C2
60	C	C3
61	D	C4
62	E	C5
63	F	C6
64	G	C7
65	H	C8
66	I	C9
67	}	D0
68	J	D1
69	K	D2
70	L	D3
71	M	D4
72	N	D5
73	O	D6
74	P	D7
75	Q	D8
76	R	D9
77	\	E0
78	S	E2
79	T	E3
80	U	E4
81	V	E5
82	W	E6
83	X	E7
84	Y	E8
85	Z	E9
86	0	F0
87	1	F1
88	2	F2
89	3	F3
90	4	F4
91	5	F5
92	6	F6
93	7	F7
94	8	F8
95	9	F9

Note: When zones are specified for record identification codes, the & is considered to have a hex C zone, the - (minus sign) is considered to have a hex D zone, and the blank is considered to have a hex F zone, to be consistent with card punches.

Figure 2-2. Normal Collating Sequence and Hexadecimal Value of Characters



IBM ®

International Business Machines Corporation

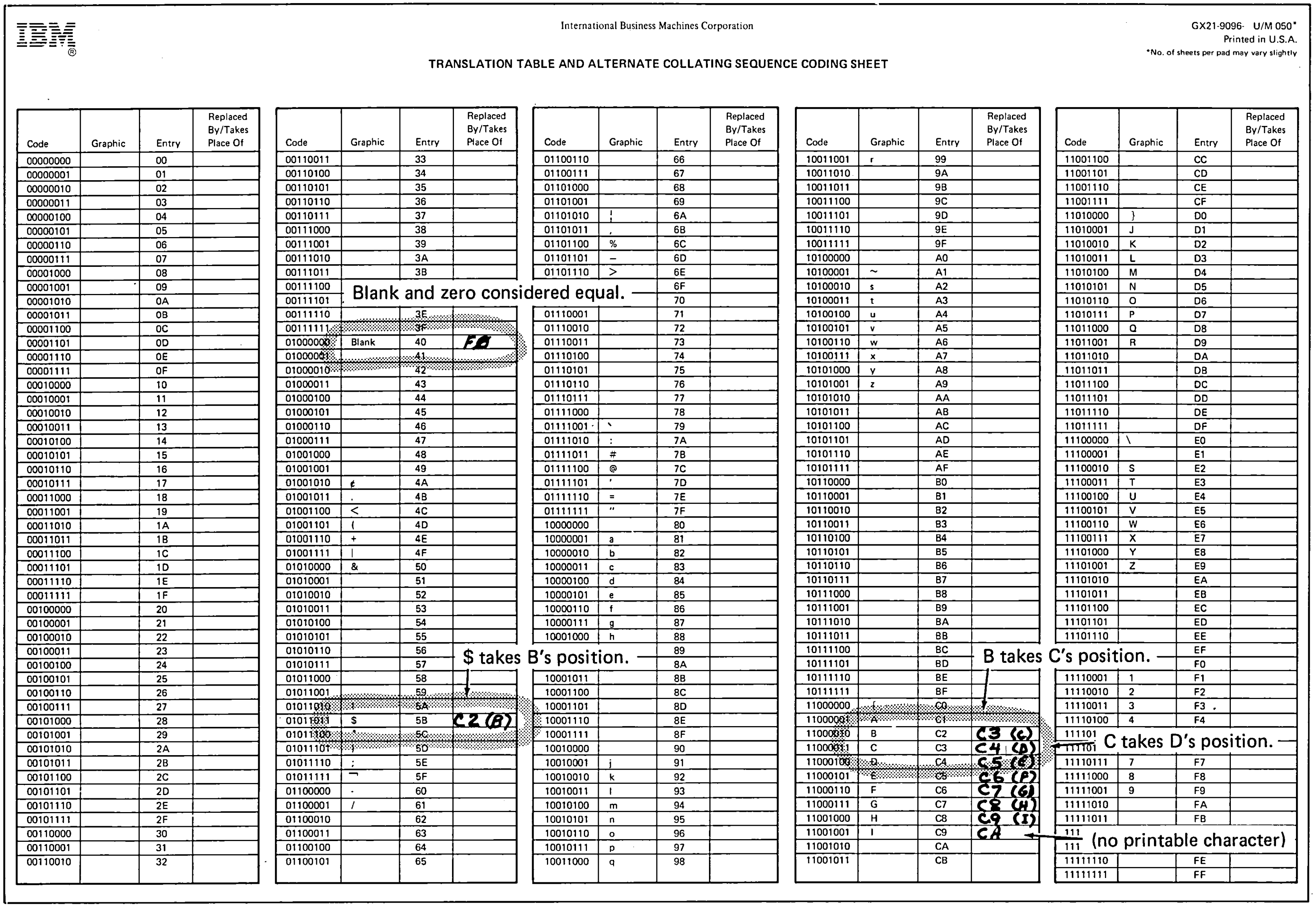
GX21-9096- U/M 050*
Printed in U.S.A.
*No. of sheets per pad may vary slightly

TRANSLATION TABLE AND ALTERNATE COLLATING SEQUENCE CODING SHEET

Code	Graphic	Entry	Replaced By/Takes Place Of
00000000		00	
00000001		01	
00000010		02	
00000011		03	
00000100		04	
00000101		05	
00000110		06	
00000111		07	
00001000		08	
00001001		09	
00001010		0A	
00001011		0B	
00001100		0C	
00001101		0D	
00001110		0E	
00001111		0F	
00010000		10	
00010001		11	
00010010		12	
00010011		13	
00010100		14	
00010101		15	
00010110		16	
00010111		17	
00011000		18	
00011001		19	
00011010		1A	
00011011		1B	
00011100		1C	
00011101		1D	
00011110		1E	
00011111		1F	
00100000		20	
00100001		21	
00100010		22	
00100011		23	
00100100		24	
00100101		25	
00100110		26	
00100111		27	
00101000		28	
00101001		29	
00101010		2A	
00101011		2B	
00101100		2C	
00101101		2D	
00101110		2E	
00101111		2F	
00110000		30	
00110001		31	
00110010		32	
00110011		33	
00110100		34	
00110101		35	
00110110		36	
00110111		37	
00111000		38	
00111001		39	
00111010		3A	
00111011		3B	
00111100		3C	
00111101		3D	
00111110		3E	
00111111		3F	
01000000	Blank	40	
01000001		41	
01000010		42	
01000011		43	
01000100		44	
01000101		45	
01000110		46	
01000111		47	
01001000		48	
01001001		49	
01001010	¢	4A	
01001011	.	4B	
01001100	<	4C	
01001101	(	4D	
01001110	+	4E	
01001111	|	4F	
01010000	&	50	
01010001		51	
01010010		52	
01010011		53	
01010100		54	
01010101		55	
01010110		56	
01010111		57	
01011000		58	
01011001		59	
01011010	!	5A	
01011011	$	5B	
01011100	*	5C	
01011101	)	5D	
01011110	;	5E	
01011111	¬	5F	
01100000	-	60	
01100001	/	61	
01100010		62	
01100011		63	
01100100		64	
01100101		65	
01100110		66	
01100111		67	
01101000		68	
01101001		69	
01101010	¦	6A	
01101011	,	6B	
01101100	%	6C	
01101101	–	6D	
01101110	>	6E	
01101111	?	6F	
01110000		70	
01110001		71	
01110010		72	
01110011		73	
01110100		74	
01110101		75	
01110110		76	
01110111		77	
01111000		78	
01111001	`	79	
01111010	:	7A	
01111011	#	7B	
01111100	@	7C	
01111101	'	7D	
01111110	=	7E	
01111111	"	7F	
10000000		80	
10000001	a	81	
10000010	b	82	
10000011	c	83	
10000100	d	84	
10000101	e	85	
10000110	f	86	
10000111	g	87	
10001000	h	88	
10001001	i	89	
10001010		8A	
10001011		8B	
10001100		8C	
10001101		8D	
10001110		8E	
10001111		8F	
10010000		90	
10010001	j	91	
10010010	k	92	
10010011	l	93	
10010100	m	94	
10010101	n	95	
10010110	o	96	
10010111	p	97	
10011000	q	98	
10011001	r	99	
10011010		9A	
10011011		9B	
10011100		9C	
10011101		9D	
10011110		9E	
10011111		9F	
10100000		A0	
10100001	~	A1	
10100010	s	A2	
10100011	t	A3	
10100100	u	A4	
10100101	v	A5	
10100110	w	A6	
10100111	x	A7	
10101000	y	A8	
10101001	z	A9	
10101010		AA	
10101011		AB	
10101100		AC	
10101101		AD	
10101110		AE	
10101111		AF	
10110000		B0	
10110001		B1	
10110010		B2	
10110011		B3	
10110100		B4	
10110101		B5	
10110110		B6	
10110111		B7	
10111000		B8	
10111001		B9	
10111010		BA	
10111011		BB	
10111100		BC	
10111101		BD	
10111110		BE	
10111111		BF	
11000000	{	C0	
11000001	A	C1	
11000010	B	C2	
11000011	C	C3	
11000100	D	C4	
11000101	E	C5	
11000110	F	C6	
11000111	G	C7	
11001000	H	C8	
11001001	I	C9	
11001010		CA	
11001011		CB	
11001100		CC	
11001101		CD	
11001110		CE	
11001111		CF	
11010000	}	D0	
11010001	J	D1	
11010010	K	D2	
11010011	L	D3	
11010100	M	D4	
11010101	N	D5	
11010110	O	D6	
11010111	P	D7	
11011000	Q	D8	
11011001	R	D9	
11011010		DA	
11011011		DB	
11011100		DC	
11011101		DD	
11011110		DE	
11011111		DF	
11100000	\	E0	
11100001		E1	
11100010	S	E2	
11100011	T	E3	
11100100	U	E4	
11100101	V	E5	
11100110	W	E6	
11100111	X	E7	
11101000	Y	E8	
11101001	Z	E9	
11101010		EA	
11101011		EB	
11101100		EC	
11101101		ED	
11101110		EE	
11101111		EF	
11110000	0	F0	
11110001	1	F1	
11110010	2	F2	
11110011	3	F3	
11110100	4	F4	
11110101	5	F5	
11110110	6	F6	
11110111	7	F7	
11111000	8	F8	
11111001	9	F9	
11111010		FA	
11111011		FB	
11111100		FC	
11111101		FD	
11111110		FE	
11111111		FF	

Note: Not all graphic symbols shown here are available on all systems.

Figure 2-3. Translation Table and Alternate Collating Sequence Coding Sheet



IBM®

International Business Machines Corporation

GX21-9096- U/M 050*
Printed in U.S.A.
*No. of sheets per pad may vary slightly

TRANSLATION TABLE AND ALTERNATE COLLATING SEQUENCE CODING SHEET

Code	Graphic	Entry	Replaced By/Takes Place Of
00000000		00	
00000001		01	
00000010		02	
00000011		03	
00000100		04	
00000101		05	
00000110		06	
00000111		07	
00001000		08	
00001001		09	
00001010		0A	
00001011		0B	
00001100		0C	
00001101		0D	
00001110		0E	
00001111		0F	
00010000		10	
00010001		11	
00010010		12	
00010011		13	
00010100		14	
00010101		15	
00010110		16	
00010111		17	
00011000		18	
00011001		19	
00011010		1A	
00011011		1B	
00011100		1C	
00011101		1D	
00011110		1E	
00011111		1F	
00100000		20	
00100001		21	
00100010		22	
00100011		23	
00100100		24	
00100101		25	
00100110		26	
00100111		27	
00101000		28	
00101001		29	
00101010		2A	
00101011		2B	
00101100		2C	
00101101		2D	
00101110		2E	
00101111		2F	
00110000		30	
00110001		31	
00110010		32	
00110011		33	
00110100		34	
00110101		35	
00110110		36	
00110111		37	
00111000		38	
00111001		39	
00111010		3A	
00111011		3B	
00111100			
00111101			
00111110		3E	
00111111		3F	
01000000	Blank	40	F0
01000001		41	
01000010		42	
01000011		43	
01000100		44	
01000101		45	
01000110		46	
01000111		47	
01001000		48	
01001001		49	
01001010	¢	4A	
01001011	.	4B	
01001100	<	4C	
01001101	(	4D	
01001110	+	4E	
01001111	|	4F	
01010000	&	50	
01010001		51	
01010010		52	
01010011		53	
01010100		54	
01010101		55	
01010110		56	
01010111		57	
01011000		58	
01011001		59	
01011010	!	5A	
01011011	$	5B	C2 (B)
01011100	*	5C	
01011101	)	5D	
01011110	;	5E	
01011111	¬	5F	
01100000	-	60	
01100001	/	61	
01100010		62	
01100011		63	
01100100		64	
01100101		65	
01100110		66	
01100111		67	
01101000		68	
01101001		69	
01101010	¦	6A	
01101011	,	6B	
01101100	%	6C	
01101101	–	6D	
01101110	>	6E	
		6F	
		70	
01110001		71	
01110010		72	
01110011		73	
01110100		74	
01110101		75	
01110110		76	
01110111		77	
01111000		78	
01111001	`	79	
01111010	:	7A	
01111011	#	7B	
01111100	@	7C	
01111101	'	7D	
01111110	=	7E	
01111111	"	7F	
10000000		80	
10000001	a	81	
10000010	b	82	
10000011	c	83	
10000100	d	84	
10000101	e	85	
10000110	f	86	
10000111	g	87	
10001000	h	88	
		89	
		8A	
10001011		8B	
10001100		8C	
10001101		8D	
10001110		8E	
10001111		8F	
10010000		90	
10010001	j	91	
10010010	k	92	
10010011	l	93	
10010100	m	94	
10010101	n	95	
10010110	o	96	
10010111	p	97	
10011000	q	98	
10011001	r	99	
10011010		9A	
10011011		9B	
10011100		9C	
10011101		9D	
10011110		9E	
10011111		9F	
10100000		A0	
10100001	~	A1	
10100010	s	A2	
10100011	t	A3	
10100100	u	A4	
10100101	v	A5	
10100110	w	A6	
10100111	x	A7	
10101000	y	A8	
10101001	z	A9	
10101010		AA	
10101011		AB	
10101100		AC	
10101101		AD	
10101110		AE	
10101111		AF	
10110000		B0	
10110001		B1	
10110010		B2	
10110011		B3	
10110100		B4	
10110101		B5	
10110110		B6	
10110111		B7	
10111000		B8	
10111001		B9	
10111010		BA	
10111011		BB	
10111100		BC	
10111101		BD	
10111110		BE	
10111111		BF	
11000000	{	C0	
11000001	A	C1	
11000010	B	C2	C3 (C)
11000011	C	C3	C4 (D)
11000100	D	C4	C5 (E)
11000101	E	C5	C6 (F)
11000110	F	C6	C7 (G)
11000111	G	C7	C8 (H)
11001000	H	C8	C9 (I)
11001001	I	C9	CA
11001010		CA	
11001011		CB	
11001100		CC	
11001101		CD	
11001110		CE	
11001111		CF	
11010000	}	D0	
11010001	J	D1	
11010010	K	D2	
11010011	L	D3	
11010100	M	D4	
11010101	N	D5	
11010110	O	D6	
11010111	P	D7	
11011000	Q	D8	
11011001	R	D9	
11011010		DA	
11011011		DB	
11011100		DC	
11011101		DD	
11011110		DE	
11011111		DF	
11100000	\	E0	
11100001		E1	
11100010	S	E2	
11100011	T	E3	
11100100	U	E4	
11100101	V	E5	
11100110	W	E6	
11100111	X	E7	
11101000	Y	E8	
11101001	Z	E9	
11101010		EA	
11101011		EB	
11101100		EC	
11101101		ED	
11101110		EE	
		EF	
		F0	
11110001	1	F1	
11110010	2	F2	
11110011	3	F3	
11110100	4	F4	
111101			
111101			
11110111	7	F7	
11111000	8	F8	
11111001	9	F9	
11111010		FA	
11111011		FB	
111			
111			
11111110		FE	
11111111		FF	

Blank and zero considered equal.

$ takes B's position.

B takes C's position.

C takes D's position.

(no printable character)

Figure 2-4. Altering the Collating Sequence

Note: Not all graphic symbols shown here are available on all systems.

Formatting the Alternate Collating Sequence Records

The changes to the collating sequence that are described on the coding sheet must be transcribed into the correct record format so that the operator can enter them into the system. The alternate collating sequence must be formatted as follows:

Record Position	Entry
1-6	ALTSEQ (This indicates to the system that the normal sequence is being altered.)
7-8	Leave these positions blank.
9-10	Enter the hexadecimal value for the character whose normal sequence is being changed. Figure 2-2 shows each printable character and its hexadecimal value.
11-12	Enter the hexadecimal value of the character replacing the character whose normal sequence is being changed.
13-16, 17-20, 21,-24 ...	Use these positions in the same way as positions 9 through 12. The first two positions specify the character to be replaced by the character specified in the next two positions. There can be as many four-position entries as can be contained in the record. The first blank position terminates the record.

The records that describe the alternate collating sequence must be preceded by a record with **ƀ (ƀ = blank) in positions 1 through 3. The remaining positions in this record can be used for comments.

Example of Alternate Collating Sequence Record

The record for the preceding example of inserting the dollar sign ($) between A and B is formatted as follows. The changes specified on the alternate collating sequence coding sheet are shown in Figure 2-4.

Record Position	Entry
1-6	ALTSEQ
7-8	Blanks
9-12	5BC2 ($ takes B's position)
13-16	C2C3 (B takes C's position)
17-20	C3C4 (C takes D's position)
21-24	C4C5 (D takes E's position)
25-28	C5C6 (E takes F's position)
29-32	C6C7 (F takes G's position)
33-36	C7C8 (G takes H's position)
37-40	C8C9 (H takes I's position)
41-44	C9CA (I takes the position of an unprintable character)

COLUMNS 27-36

Columns 27 through 36 are not used. Leave them blank.

COLUMN 37 (INQUIRY)

Entry	Explanation
Blank or I	Program when interrupted will not allow the operator to enter new procedures or commands (does not recognize an inquiry request).
B	Program when interrupted will allow the operator to enter new procedures or commands (does recognize an inquiry request).

Use column 37 to specify whether an executing program can be interrupted to allow another program to execute.

The operator requests an interruption (called an inquiry request) by pressing the ATTN key on the display station. The procedure or command statements for the interrupting program must be entered from the display station.

The program to be loaded following an inquiry request (the interrupting program) can have an I, B, or blank in column 37 of its control specifications. However, even if it has a B in column 37, the interrupting program cannot be interrupted by another inquiry request.

If column 37 contains a B, the inquiry function of System/34 allows the operator to interrupt a program that is currently using the display station and to enter new procedures or commands. If column 37 contains any of the valid entries, the operator can set the inquiry latch for SUBR95, cancel a single requestor terminal (SRT) program that the operator initiated, or release the display station from a multiple requestor terminal (MRT) program.

For more information on inquiry, including restrictions on the use of system utilities in inquiry mode, see *Interrupting a Program (Inquiry)* in Chapter 2 of the *System/34 Concepts and Design Guide.*

File Sharing

Single Program Mode

When System/34 is operating in single program mode, an inquiry program can access the files being used by the interrupted program (that is, the active files) for input and update processing only. Update processing of an active file is allowed only if the interrupted program is not updating or adding to the file. Active output files cannot be accessed by the inquiry program.

Multiple Program Mode

When System/34 is operating in multiple program mode, an inquiry program can access active input, update, and add files. However, an inquiry program cannot access indexed sequential add file types or output files. Each file to be shared in both the interrupted and the inquiry programs must be designated by the DISP-SHR parameter on the FILE OCL statement.

For a description of the valid file sharing combinations, see the *System Support Reference Manual.*

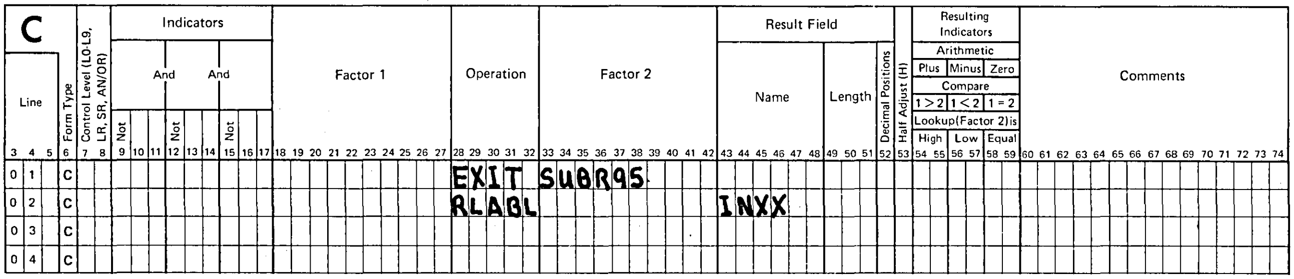
Inline Inquiry Subroutine (SUBR95)

The IBM-written subroutine SUBR95 can be used to perform an inquiry type function if the RPG II program does not have an MRT attribute (that is, the MRTMAX parameter on the COMPILE OCL statement has a value of 0 or is not specified). Column 37 can be blank or contain an I or B. (See *Interrupting a Program (Inquiry)* in Chapter 2 of the *System/34 Concepts and Design Guide* for restrictions on the inquiry function.)

The linkage to SUBR95 must be specified on the calculation specifications at every point in the program where a check is to be made for an inquiry request (see Figure 2-5). The indicator specified in columns 45 and 46 must be an RPG II indicator. For a detailed discussion of this linkage, see *Linking to External Subroutines* in Chapter 10.

When SUBR95 is called, it checks to determine whether an inquiry request was made. If one was made (that is, the operator selected option 4 in response to the inquiry display), the indicator specified in the RLABL operation is turned on and the inquiry request is reset. This indicator can then be used to condition subsequent calculation and output operations.

Calculation Specifications



Line	Form Type	Control Level (L0-L9, LR, SR, AN/OR)	Indicators	Factor 1	Operation	Factor 2	Result Field Name	Length	Decimal Positions	Half Adjust (H)	Resulting Indicators	Comments
01	C				EXIT	SUBR95						
02	C				RLABL		INXX					
03	C											
04	C											

Figure 2-5. Linkage for SUBR95

COLUMNS 38-40

Columns 38 through 40 are not used. Leave them blank.

COLUMN 41 (1P FORMS POSITION)

Entry	Explanation
Blank	First line is printed only once.
1	First line can be printed repeatedly.

Use column 41 only when the first output line is written to a printer file. If the program contains more than one printer file, the 1P entry in column 41 applies to each printer file that has 1P (first page) output.

When forms are first inserted in the printer, they may not be in perfect alignment. Sometimes several lines must be printed to determine the correct positioning of the form. If 1P forms position is specified, the system prints the first line of output and issues a message. The operator can then align the forms and select the appropriate option to retry printing the line or to continue printing. The 1P forms specification is also valid if the output is spooled. The page counter is not incremented until the forms have been positioned correctly.

The 1P forms position specification can be overridden on the PRINTER OCL statement, or forms alignment can be specified on the PRINTER OCL statement.

COLUMN 42

Column 42 is not used. Leave it blank.

COLUMN 43 (FILE TRANSLATION)

Entry	Explanation
Blank	No file translation is needed.
F	Input, output, update, or combined files are to be translated.

Use column 43 only when information contained in an input, output, update, or combined file is in a character code different from the character code used by System/34.

The file translation function of RPG II translates 1 character into another. The characters can be translated when they are read into the system from an input file or before they are written to an output file. An F in column 43 indicates one or both of the following: (1) a character code used in the input data must be translated into System/34 code; (2) the output data must be translated from System/34 code into a different code. If file translation is specified for update or combined files, the input data is translated into System/34 code and then translated back into the different code for output.

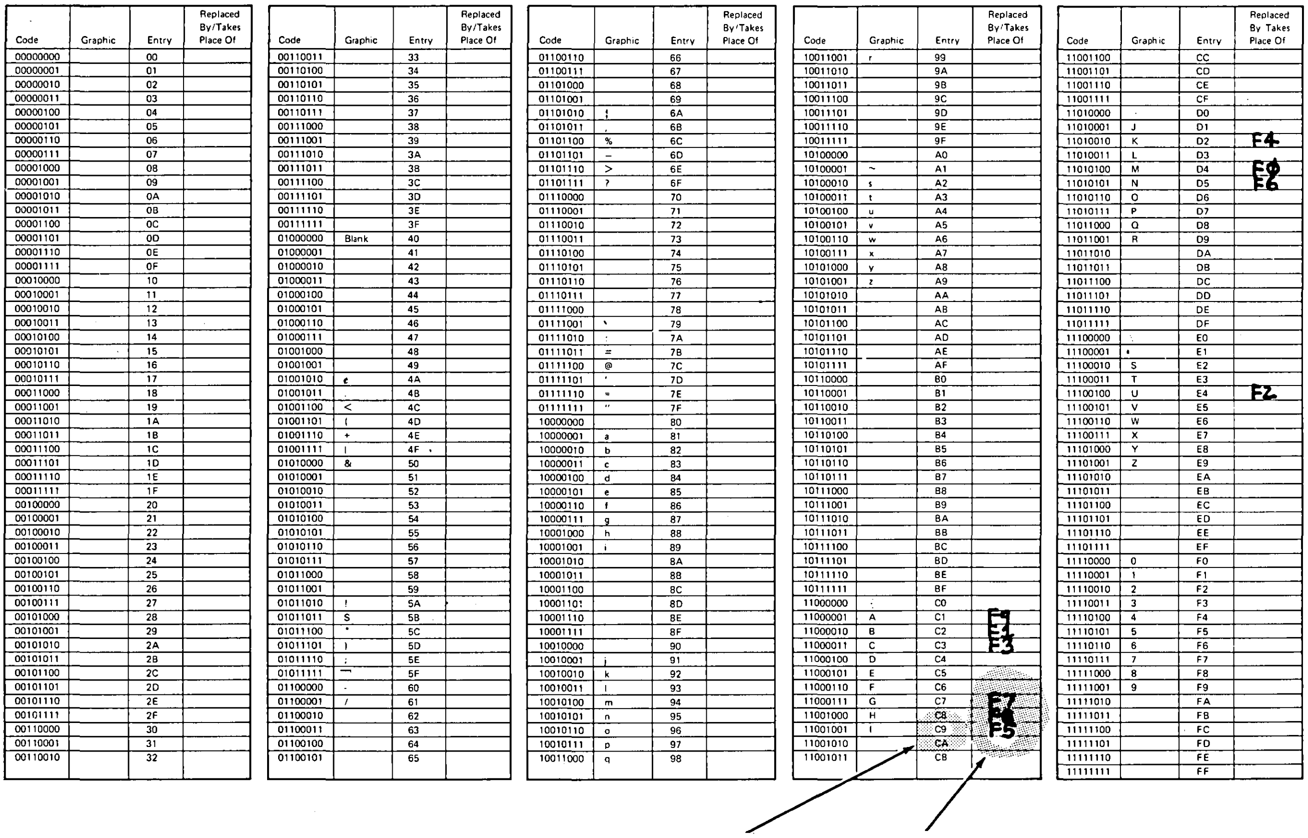
File translation is generally used as a security measure when input or output data is coded to prevent access to classified information. The system cannot process this classified information until it is translated into System/34 code. Figure 2-2 shows the hexadecimal code used by the System/34 to internally represent alphabetic, numeric, and special characters.

To specify file translation, enter an F in column 43. Then describe the character that is to be translated on the Translation Table and Alternate Collating Sequence Coding Sheet (see Figure 2-3). The translations described must also be transcribed into the correct record format for entry into the system. These records, called the file translation table records, must be entered following the RPG II specifications. This special table requires no file description or extension specifications, and is printed with the compiled program.

Filling Out the Translation Table Coding Sheet

Identify each character that is to be translated on the translation table coding sheet. In the *Replaced By* column, enter the hexadecimal value of the character that is to replace the character presently associated with the hexadecimal value shown in the *Entry* column. For example, if the input data is in a code in which the character B is used to represent the number 1, enter the hexadecimal value of 1 (which is F1) in the *Replaced By* column next to the character B (see Figure 2-6).

TRANSLATION TABLE AND ALTERNATE COLLATING SEQUENCE CODING SHEET



Code	Graphic	Entry	Replaced By/Takes Place Of
00000000		00	
00000001		01	
00000010		02	
00000011		03	
00000100		04	
00000101		05	
00000110		06	
00000111		07	
00001000		08	
00001001		09	
00001010		0A	
00001011		0B	
00001100		0C	
00001101		0D	
00001110		0E	
00001111		0F	
00010000		10	
00010001		11	
00010010		12	
00010011		13	
00010100		14	
00010101		15	
00010110		16	
00010111		17	
00011000		18	
00011001		19	
00011010		1A	
00011011		1B	
00011100		1C	
00011101		1D	
00011110		1E	
00011111		1F	
00100000		20	
00100001		21	
00100010		22	
00100011		23	
00100100		24	
00100101		25	
00100110		26	
00100111		27	
00101000		28	
00101001		29	
00101010		2A	
00101011		2B	
00101100		2C	
00101101		2D	
00101110		2E	
00101111		2F	
00110000		30	
00110001		31	
00110010		32	
00110011		33	
00110100		34	
00110101		35	
00110110		36	
00110111		37	
00111000		38	
00111001		39	
00111010		3A	
00111011		3B	
00111100		3C	
00111101		3D	
00111110		3E	
00111111		3F	
01000000	Blank	40	
01000001		41	
01000010		42	
01000011		43	
01000100		44	
01000101		45	
01000110		46	
01000111		47	
01001000		48	
01001001		49	
01001010	¢	4A	
01001011	.	4B	
01001100	<	4C	
01001101	(	4D	
01001110	+	4E	
01001111	|	4F	
01010000	&	50	
01010001		51	
01010010		52	
01010011		53	
01010100		54	
01010101		55	
01010110		56	
01010111		57	
01011000		58	
01011001		59	
01011010	!	5A	
01011011	$	5B	
01011100	*	5C	
01011101	)	5D	
01011110	;	5E	
01011111	¬	5F	
01100000	-	60	
01100001	/	61	
01100010		62	
01100011		63	
01100100		64	
01100101		65	
01100110		66	
01100111		67	
01101000		68	
01101001		69	
01101010	¦	6A	
01101011	,	6B	
01101100	%	6C	
01101101	_	6D	
01101110	>	6E	
01101111	?	6F	
01110000		70	
01110001		71	
01110010		72	
01110011		73	
01110100		74	
01110101		75	
01110110		76	
01110111		77	
01111000		78	
01111001	`	79	
01111010	:	7A	
01111011	#	7B	
01111100	@	7C	
01111101	'	7D	
01111110	=	7E	
01111111	"	7F	
10000000		80	
10000001	a	81	
10000010	b	82	
10000011	c	83	
10000100	d	84	
10000101	e	85	
10000110	f	86	
10000111	g	87	
10001000	h	88	
10001001	i	89	
10001010		8A	
10001011		88	
10001100		8C	
10001101		8D	
10001110		8E	
10001111		8F	
10010000		90	
10010001	j	91	
10010010	k	92	
10010011	l	93	
10010100	m	94	
10010101	n	95	
10010110	o	96	
10010111	p	97	
10011000	q	98	
10011001	r	99	
10011010		9A	
10011011		9B	
10011100		9C	
10011101		9D	
10011110		9E	
10011111		9F	
10100000		A0	
10100001	~	A1	
10100010	s	A2	
10100011	t	A3	
10100100	u	A4	
10100101	v	A5	
10100110	w	A6	
10100111	x	A7	
10101000	y	A8	
10101001	z	A9	
10101010		AA	
10101011		AB	
10101100		AC	
10101101		AD	
10101110		AE	
10101111		AF	
10110000		B0	
10110001		B1	
10110010		B2	
10110011		B3	
10110100		B4	
10110101		B5	
10110110		B6	
10110111		B7	
10111000		B8	
10111001		B9	
10111010		BA	
10111011		BB	
10111100		BC	
10111101		BD	
10111110		BE	
10111111		BF	
11000000	{	C0	
11000001	A	C1	F9
11000010	B	C2	F1
11000011	C	C3	F3
11000100	D	C4	
11000101	E	C5	
11000110	F	C6	
11000111	G	C7	F7
11001000	H	C8	F8
11001001	I	C9	F5
11001010		CA	
11001011		CB	
11001100		CC	
11001101		CD	
11001110		CE	
11001111		CF	
11010000	}	D0	
11010001	J	D1	
11010010	K	D2	F4
11010011	L	D3	
11010100	M	D4	FØ
11010101	N	D5	F6
11010110	O	D6	
11010111	P	D7	
11011000	Q	D8	
11011001	R	D9	
11011010		DA	
11011011		DB	
11011100		DC	
11011101		DD	
11011110		DE	
11011111		DF	
11100000	\	E0	
11100001		E1	
11100010	S	E2	
11100011	T	E3	
11100100	U	E4	F2
11100101	V	E5	
11100110	W	E6	
11100111	X	E7	
11101000	Y	E8	
11101001	Z	E9	
11101010		EA	
11101011		EB	
11101100		EC	
11101101		ED	
11101110		EE	
11101111		EF	
11110000	0	F0	
11110001	1	F1	
11110010	2	F2	
11110011	3	F3	
11110100	4	F4	
11110101	5	F5	
11110110	6	F6	
11110111	7	F7	
11111000	8	F8	
11111001	9	F9	
11111010		FA	
11111011		FB	
11111100		FC	
11111101		FD	
11111110		FE	
11111111		FF	

This is the hexadecimal value of the character to be translated.

This is the hexadecimal value of the character that will be substituted for the character that is to be translated.

Figure 2-6. Specifications for File Translation Input Records

Formatting File Translation Table Records

File translation table records must be formatted as follows:

Record Position	Entry
1-6 (To translate) all files)	Enter *FILES to indicate that all input, output, update, and combined files are to be translated (both the input and output data of the update and combined files are translated). Complete the file translation table record beginning with positions 9 and 10.
1-8 (To translate a specific file)	Enter the filename of the input, output, update, or combined file to be translated (both the input and output data of update and combined files are translated). Complete the file translation table record beginning with positions 9 and 10. The *FILES entry is not made in positions 1 through 6 when only a specific file is to be translated.
9-10	Enter the hexadecimal equivalent of the external character to be translated-from on input, or to be translated-to on output.
11-12	Enter the hexadecimal equivalent of the internal character that RPG works with. It will replace the character in positions 9 and 10 on input and be replaced by the character in positions 9 and 10 on output.
13-16, 17-20, 21-24, ... , 93-96	Use these positions in the same way as positions 9 through 12. The first two positions contain the hexadecimal value of the character to be replaced by the character whose hexadecimal value is specified in the next two positions.

If the number of translations exceeds 96 positions, duplicate positions 1 through 8 on the next record and continue with the translation pairs as before in positions 9 through 96.

All table records for one file must be kept together. The file translation table records must be preceded by one record with **ƀ (ƀ = blank) in positions 1 through 3. The remaining positions of this record can be used for comments.

Example of File Translation

A department store processes sales slips that contain the wholesale price and the retail price of each item. Because the wholesale price must remain confidential, the store substitutes the individual letters of a code name for the numbers comprising the wholesale costs. The store uses the code name BUCKINGHAM to represent the numbers 1 through 9 and 0, respectively.

In order for the system to perform any calculations with the wholesale prices, file translation must be specified for this file by an F entry in column 43 of the control specifications for this program. Figure 2-6 shows the entries made on the translation table coding sheet that translate the code name into System/34 characters.

The file translation table record for translating this code word is as follows:

Record Position	Entry
1-6	*FILES (All files in the program are to be translated.)
7-8	Blank
9-12	C2F1 (Each B read from the file is translated into a 1.)
13-16	E4F2 (Each U read from the file is translated into a 2.)
17-20	C3F3 (Each C read from the file is translated into a 3.)
21-24	D2F4 (Each K read from the file is translated into a 4.)
25-28	C9F5 (Each I read from the file is translated into a 5.)
29-32	D5F6 (Each N read from the file is translated into a 6.)
33-36	C7F7 (Each G read from the file is translated into a 7.)
37-40	C8F8 (Each H from the file is translated into an 8.)
41-44	C1F9 (Each A read from the file is translated into a 9.)
45-48	D4F0 (Each M read from the file is translated into a 0.)

Note: On input, the alphabetic characters are translated to their corresponding numbers. On output, the numbers are translated back to their corresponding alphabetic characters.

When the program with this file translation is executed, each character given in the file translation table record is translated for every field in each file in the program (unless a specific file was given in positions 1 through 8 of the file translation table record). All characters that are not in the file translation table record are handled in the normal manner.

COLUMN 44

Column 44 is not used. Leave it blank.

COLUMN 45 (NONPRINT CHARACTERS)

Entry	Explanation
Blank	Program halts if the last line printed contained an unprintable character.
1	Program does not halt for unprintable characters.

Use column 45 to bypass machine halts for unprintable characters. This column applies only to printer files.

All characters are represented in the system by a hexadecimal value, which is a numeric code. If a hexadecimal value is formed during a calculation that is not in the System/34 character set and that character is to be printed, the system halts after printing the line. In the printed line, the unprintable characters are replaced with blanks.

To bypass this halt, enter a 1 in column 45. An unprintable character is then replaced with a blank, and no halt occurs. Note, however, that your output is not correct, and, by bypassing the halt, the incorrect output may not become known (for example, when a packed key field is printed or when a nonprintable field is built by calculation specifications).

COLUMNS 46-47

Columns 46 and 47 are not used. Leave them blank.

COLUMN 48 (SHARED I/O)

Entry	Explanation
Blank	All disk files use a separate input/output area.
1	All disk files share a single input/output area.

Use column 48 to allow all disk files to use one input/output area.

Normally an RPG II program uses one input/output area for each file. Specifying a shared input/output area can reduce the amount of main storage needed to process a program. This is particularly important if a program is so large that it cannot run in the main storage available. However, the use of a shared input/output area can increase the time required to process the program. Therefore, before indicating that all disk files are to share one input/output area, be sure that the program would otherwise exceed the capacity of the system.

Additional input/output areas, which can be specified in column 32 of the file description specifications, cannot be specified for disk files using a shared input/output area. Also when an update file using a shared input/output area is processed, you must ensure that retrieval of another record did not occur between retrieval and update of the specified record; otherwise an invalid record update operation message is issued.

COLUMNS 49-51

Columns 49 through 51 are not used. Leave them blank.

COLUMNS 52-53 (NUMBER OF FORMATS)

Entry	Explanation
Blank	Program assumes an entry of 32.
0-32	Enter the number of formats in the display screen format member for the WORKSTN file.

Use columns 52 and 53 to indicate the number of individual formats in the display screen format member. This number must include all the formats in the display screen format load member, not just the number of formats used by the program.

COLUMNS 54-56

Columns 54 through 56 are not used. Leave them blank.

COLUMN 57 (TRANSPARENT LITERAL)

Entry	Explanation
Blank	No transparent literals or constants are present in the program.
1	Transparent literals or constants can be present in the program.

The transparent literal option must be specified if there are transparent literals or constants present in your program. Transparent literals or constants must begin with an apostrophe followed immediately by the shift-out (S/O) control character (hex 0E), and must end with the shift-in (S/I) control character (hex 0F) followed immediately by the terminating apostrophe. Transparent literals and constants are not checked for embedded apostrophes.

If the transparent literal option is specified and a literal or constant is found that begins with an apostrophe immediately followed by the S/O control character, the RPG II compiler checks for a valid transparent literal or constant. The following conditions cause a literal or constant to be diagnosed as an invalid transparent literal or constant:

- A second S/O control character is found before the S/I control character.
- An odd number of 1-byte characters are found between the S/O and S/I control characters.
- The S/I control character is not immediately followed by the terminating apostrophe.

If a literal or constant is found to be an invalid transparent literal or constant, it is rechecked as an alphameric literal or constant.

COLUMNS 58-74

Columns 58 through 74 are not used. Leave them blank.

COLUMNS 75-80 (PROGRAM IDENTIFICATION)

Entry	Explanation
Blank	RPGOBJ is the program identification assumed by the compiler.
Any valid program name	The first character of the program identification must be alphabetic, but cannot be #, $, or @. The remaining characters must be alphameric; however, no special character can be used and blanks must not appear between characters.

Use columns 75 through 80 to assign a unique name to your object program. The compiler uses the program name in a program directory that contains the location of your program on disk.

If the program contains a CONSOLE or WORKSTN device, the compiler also uses this program identification to name the display screen format load member for the program. The display screen format load member is generated by RPG II only for CONSOLE files; however, the name is created for both CONSOLE and WORKSTN files. This name is used by RPG II; therefore the user *must* generate his own load member with this name for WORKSTN files. For the display screen format load member name, the compiler uses the name specified as the value of the FMTS continuation line option. If the FMTS continuation line option is not specified, the compiler uses the characters specified in columns 75 through 80 of the control specifications (the program name) and adds the characters FM to the end of the program name. FM is added to the end of the program name regardless of its length, and the resulting name contains no embedded blanks.

If a cross-reference listing is to be generated for the program, this program identification is also used to identify the listing.

Chapter 3. File Description Specifications

File description specifications describe each file used by a program. One file description specifications statement is required for each file, and a maximum of 20 files can be described per program.

Write the file description specifications on the Control and File Description Specifications sheet (see Figure 3-1).

Charts at the end of this chapter show all possible files that can be defined on the file description specifications sheet (see Figure 3-20 through Figure 3-30). The charts are arranged by device and show the basic entries for all possible DISK, PRINTER, CRT, KEYBORD, CONSOLE, WORKSTN, BSCA, and SPECIAL files.

COLUMNS 1-2 (PAGE)

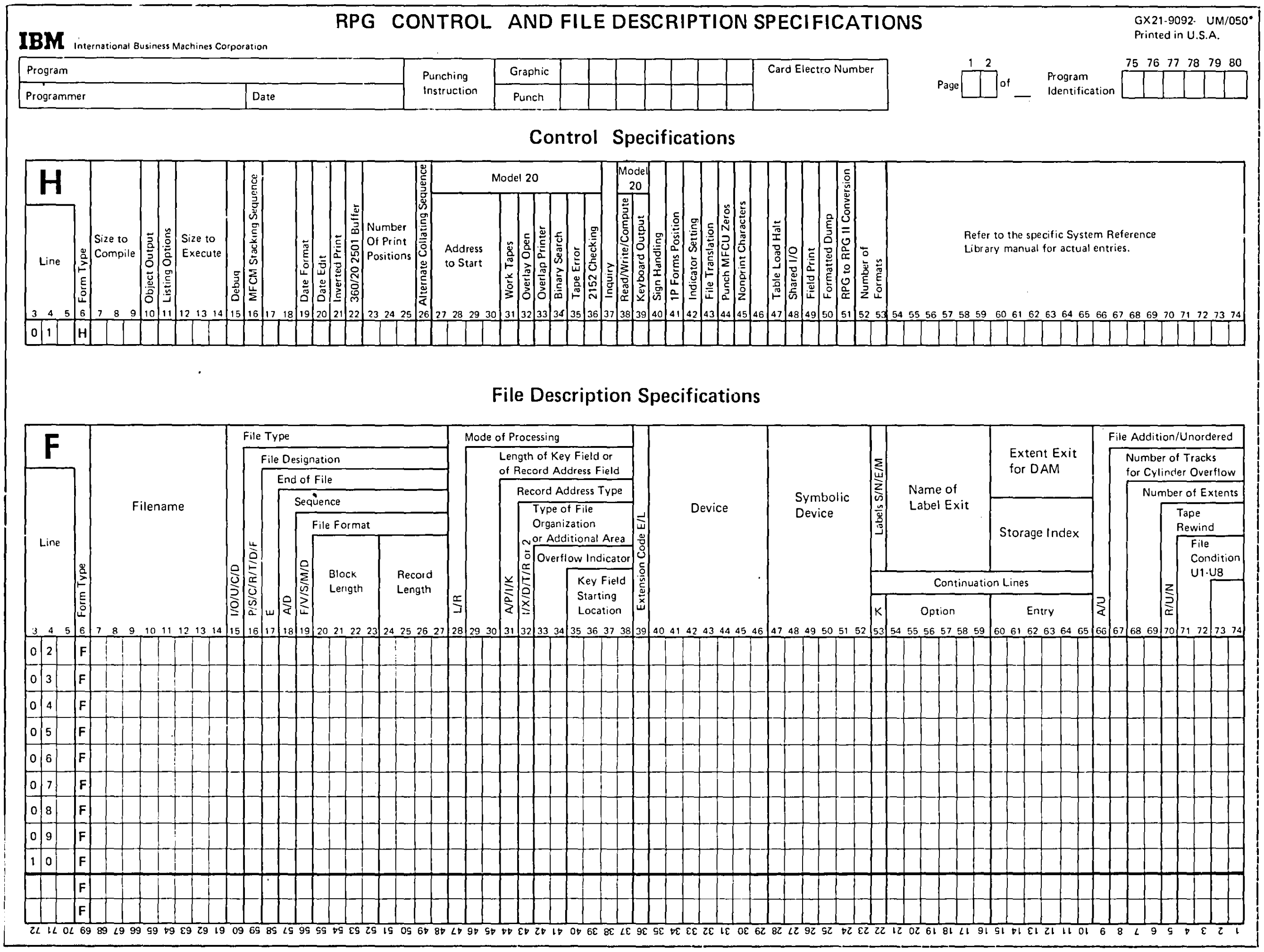
See *Common Entries* in Chapter 1.

COLUMNS 3-5 (LINE)

See *Common Entries* in Chapter 1.

COLUMN 6 (FORM TYPE)

An F must appear in column 6 to identify this line as a file description specifications statement.



IBM International Business Machines Corporation

RPG CONTROL AND FILE DESCRIPTION SPECIFICATIONS

GX21-9092- UM/050*
Printed in U.S.A.

Program | Punching Instruction | Graphic | Card Electro Number
Programmer | Date | Punch

Page (1 2) of ___ Program Identification (75 76 77 78 79 80)

Control Specifications

H

Line (3 4 5) | Form Type (6) | Size to Compile (7 8 9) | Object Output (10) | Listing Options (11) | Size to Execute (12 13 14) | Debug (15) | MFCM Stacking Sequence (16) | (17 18) | Date Format (19) | Date Edit (20) | Inverted Print (21) | 360/20 2501 Buffer (22) | Number Of Print Positions (23 24 25) | Alternate Collating Sequence (26) | Model 20: Address to Start (27 28 29 30), Work Tapes (31), Overlay Open (32), Overlap Printer (33), Binary Search (34), Tape Error (35), 2152 Checking (36) | Inquiry (37) | Model 20: Read/Write/Compute (38), Keyboard Output (39) | Sign Handling (40) | 1P Forms Position (41) | Indicator Setting (42) | File Translation (43) | Punch MFCU Zeros (44) | Nonprint Characters (45) | (46) | Table Load Halt (47) | Shared I/O (48) | Field Print (49) | Formatted Dump (50) | RPG to RPG II Conversion (51) | Number of Formats (52 53) | Refer to the specific System Reference Library manual for actual entries. (54–74)

0 1 H

File Description Specifications

F

Line (3 4 5) | Form Type (6) | Filename (7–14) | File Type: I/O/U/C/D (15) | File Designation: P/S/C/R/T/D/F (16) | End of File: E (17) | Sequence: A/D (18) | File Format: F/V/S/M/D (19) | Block Length (20 21 22 23) | Record Length (24 25 26 27) | Mode of Processing: L/R (28) | Length of Key Field or of Record Address Field (29 30) | Record Address Type: A/P/I/K (31) | Type of File Organization or Additional Area: I/X/D/T/R or 2 (32) | Overflow Indicator (33 34) | Key Field Starting Location (35 36 37 38) | Extension Code E/L (39) | Device (40–46) | Symbolic Device (47–52) | Labels S/N/E/M (53) | Name of Label Exit (54–59) | Extent Exit for DAM / Storage Index (60–65) | Continuation Lines: K (53), Option (54–59), Entry (60–65) | A/U (66) | File Addition/Unordered (66) | Number of Tracks for Cylinder Overflow (67 68 69) | Number of Extents (68 69) | Tape Rewind: R/U/N (70) | File Condition U1-U8 (71 72) | (73 74)

0 2 F
0 3 F
0 4 F
0 5 F
0 6 F
0 7 F
0 8 F
0 9 F
1 0 F
F
F

Figure 3-1. RPG Control and File Description Specifications

COLUMNS 7-14 (FILENAME)

Entry	Explanation
A valid filename	Every file used in a program must have a unique name. The first character must be alphabetic. The remaining characters can be any combination of alphabetic and numeric characters; however, special characters are not allowed. Blanks cannot appear between characters in the filename. The filename can be from 1 to 8 characters long, and must begin in column 7.

Use columns 7 through 14 to assign a unique name to every file used in your program, with the following exceptions:

- Compile-time tables and arrays do not require a filename.

- If multiple tables or arrays are read in at preexecution time from the same device, multiple filenames are required.

For naming tables and arrays, see *Columns 27-32* in Chapter 4, *Extension Specifications.*

COLUMN 15 (FILE TYPE)

Entry	Explanation
I	Input file
O	Output file
U	Update file
C	Combined file

Input Files

Input files contain records that a program uses as a source of data. All input files must be further described on input specifications, with the following exceptions:

- Preexecution-time tables and arrays and record address files are described on the extension specifications. There is, however, a method of loading arrays using extension and input specifications. See Chapter 14, *Tables and Arrays*, for complete information.

- Input files using the device name KEYBORD are further described on the calculation specifications when the KEY operation code is used.

All input files must be described within the first 24 noncommented file description specifications, including continuation statements.

Output Files

Output files contain records written or printed by a program. All output files, except table output files, must be further described on the output specifications. Table output files are further described on extension specifications.

Update Files

Update files are disk files from which a program reads a record, updates fields in the record, and writes the record back in the location from which it was read. The fields to be updated in this file must be described on the input and output specifications.

Records can be deleted from update files. The file must be defined as delete-capable when it is built (for further information on defining delete-capable files, see *FILE Statement* in the *System Support Reference Manual*).

Records are deleted from files by specifying DEL in columns 16 through 18 of the output specifications (for further information on deleting records, see *Columns 16-18 (ADD/DEL)* in Chapter 9). Deleted records are filled with hex FFs. The record is not physically removed from the file. When a direct file load of a delete-capable file is executed, the entire file is initialized to deleted records (hex FFs). (For further information on direct file loading of delete-capable files, see *Direct Files* in this chapter.)

A chained file or a demand file can be updated at detail time, total time, or at exception output time. However, all other disk files can be updated only at detail time or exception output time during the same program cycle in which the record is read.

An invalid record update operation message is issued if the record to be updated has not previously been read by the program, or if another record in the same file is retrieved between the retrieval and update of the specified record. After a record is retrieved, only one update is allowed if the files are shared (DISP-SHR). A second update attempt results in an error message, without retrieving the record again.

You should use care when updating disk files in any program that supports multiple display stations (including MRT programs). *If a file is shared by two or more display stations in a program and if the present record is not updated before the next read from the file,* the following error conditions can occur:

- An update can be lost. For example, if a record is read from file X and displayed at display station 1, then the same record is read from file X and displayed at display station 2. The update performed from one display station might then be destroyed by an update performed by the other display station.

  If this condition occurs and DISP-SHR was specified for file X, a diagnostic message is issued and the second update is not performed.

- The wrong record can be updated. For example, if a record is read from file X and displayed at display station 1, then a different record is read from file X and displayed at display station 2. If display station 1 then tries to update the first record, but the program does not reread that record, the program attempts to update the last record read from file X. If this condition occurs during an attempt to update an indexed file, a diagnostic message is issued and the requested update is not performed. Otherwise, the wrong record is updated.

- An update performed by another program sharing the file can be lost. For example, if a record is read from file X and is displayed at display station 1, then a record in a different sector is read from file X and displayed at display station 2. The second read from file X causes the system support program product to free the sector containing the first record. Another program sharing file X can then update the first record. If display station 1 also tries to update that record using the original field values, the updates made by the other program may be lost.

You can avoid the preceding error conditions by using one of the following techniques:

- Before performing an update, reread the record and check that none of the fields being updated have been changed since the record was displayed for updating. (If any of the fields were changed, you might want to redisplay the field for updating again or, if possible, perform the update using the field values currently in the record.)

- Within the program, define an array for each disk file. The array should contain one element for each display station. When a display station operator enters a relative record number, or index key of a record to be updated, the program should check the array to ensure that no other display station is updating that record. If no other display station is updating the requested record, the program should place the specified relative record number, or index key, into the array element corresponding to the display station. The program can then read the record and display it at the display station. After the operator enters the updates, the program must reread the record being updated and update it using the information entered from the display station. The program should then blank out the array element corresponding to the display station.

  If the possibility exists that another program may update the same file concurrently, the file should be defined as a different logical file for each display station using the program. Shared input/output must be used. If you use logical files and specify DISP-SHR for each logical file, the system support program product protects the sector containing the record from updates by other programs. Using shared input/output reduces the amount of space required for input/output buffers. In addition, shared input/output automatically performs the reread operation before updating the record.

  Note: Additional input/output areas (numeric entry in column 32 of the file description specifications) cannot be specified for disk files using a shared input/output area.

  CAUTION
  If shared input/output is not used and the reread of the record is performed by the program, the record is momentarily not protected for update by the system support program product. Another program can then update that same record before the reread is completed.

Combined Files

A combined file is both an input file and an output file. A combined file can be specified only if the device name in columns 40 through 46 of the file description specifications is SPECIAL or WORKSTN. A program reads records from a combined file and includes output data on the records in the file. The result is one file that contains both input and output data. Combined files must be further described on both the input and output specifications.

COLUMN 16 (FILE DESIGNATION)

Entry	Explanation
P	Primary file
S	Secondary file
C	Chained file
R	Record address file
T	Table file (preexecution-time tables or arrays)
D	Demand file
Blank	Output file (except chained output files)

Use column 16 to further identify the use of input, combined, and update files. Leave the column blank for all output files except chained output files.

Primary Files

An input, combined, or update file can be specified as a primary file. A primary file is not required in a program. However, if specified, the primary file is the main file from which a program reads records. In multifile processing, the primary file is used to control the order in which records are selected for processing. (See Chapter 11, *Multifile Processing,* for more information on record selection in primary files.)

If a primary file is not specified and one or more secondary files are specified, the first secondary file is assigned as the primary file. If no primary or secondary files are specified, you *must* provide an exit for your program by setting on the LR indicator.

If KEYBORD is specified as the device for a primary input file, no other files in the program can be specified as primary or secondary files. In this case, you must provide an exit for your program by setting on the LR indicator.

If WORKSTN is specified as the device for a primary file, no other files in the program can be specified as secondary files.

Secondary Files

Secondary files are used only in programs that do multifile processing. All files used in multifile processing, except the primary file, are secondary files. A secondary file can be an input, update, or combined file. Secondary files are processed in the order in which they are written in the file description specifications.

See Chapter 11, *Multifile Processing,* for more information on record selection for primary and secondary files.

Chained Files

A chained file is a disk file for which the CHAIN operation code is used to do one of the following:

- Read records randomly
- Load a nondelete-capable direct file

A chained file can be an input, output, or update file. See *Column 28 (Mode of Processing), Random by Relative Record Number or Key* in this chapter for a discussion of random processing. See Chapter 10, *Operation Codes, CHAIN,* for information about the CHAIN operation code.

No more than 15 chained and demand files can be specified for one program.

Record Address Files

A record address file is an input file that indicates which records are to be read from a disk file, and the order in which these records are to be read. You can use only one record address file in a program. All record address files must be further defined on the extension specifications. Record address files contain either record-key limits or relative record numbers.

Record address files that contain record-key limits can be disk files or CONSOLE files. These record address files are used with indexed files only. See *Column 28 (Mode of Processing), Sequential Within Limits* in this chapter for a complete discussion of this topic.

Record address files that contain relative record numbers in binary format can only be disk files. These files are called addrout (address output) files, and they are produced by the sort program. An addrout file can be used with a sequential, indexed, or direct file. See *Column 28 (Mode of Processing), Random by Addrout File* in this chapter for a complete discussion of this topic.

Table or Array Files

A table or array file is an input file that contains table or array entries. Only preexecution-time table or array files are described on the file description specifications. However, all tables and arrays (compile time, preexecution time, and execution time) must be described by extension specifications. (For a complete description of tables and arrays, see Chapter 14, *Tables and Arrays.*)

Table files are not used in record selection and processing. Table files are only a means of supplying entries for tables used by the program. When preexecution-time table or array files are read during the execution of the program, the program reads all the entries from the table or array file before it begins record processing.

Table or array files must be sequential files.

Demand Files

Demand files can be input, update, or combined files. The READ operation code must be used in the calculation specifications to read any demand files except those entered from files assigned to the KEYBORD. (The KEY operation code must be used in calculation specifications to read from KEYBORD demand files.) See Chapter 10, *Operation Codes, READ,* for a complete discussion of processing demand files.

No more than 15 demand and chained files can be specified for one program.

COLUMN 17 (END OF FILE)

Entry	Explanation
Blank	The program can end whether or not all records from the file are processed. However, if column 17 is blank for all files, all records from every file must be processed before the program can end. This column must be blank for WORKSTN or KEYBORD files.
E	All records from the file must be processed before the program can end. This entry is not valid for files processed by record address files.

Use column 17 to indicate whether the program can end before all records from the file are processed. Column 17 applies only to files used in a program that does multifile processing.

Column 17 can be used only for input, update, or combined files used as primary, secondary, or record address files. The devices associated with column 17 are DISK and CONSOLE. End of file for CONSOLE files is identified when the operator presses command key 12, that is, the Cmd key and the = (equal) key.

A program that performs multifile processing could reach the end of one file before reaching the end of the others. It needs, therefore, some indication of whether it is to continue reading records from the other files or end the program. An entry in column 17 provides that indication.

If the records from all files must be processed, column 17 must be blank for all files or contain Es for all files.

Note: An entry cannot be made in column 17 for files assigned to the KEYBORD and WORKSTN devices. To terminate the program with a primary file assigned to the KEYBORD, the LR indicator must be set on by calculation specifications.

COLUMN 18 (SEQUENCE)

Entry	Explanation
Blank	No sequence checking is to be done. This column must be blank for a WORKSTN file.
A	Sequence checking is to be done. Records in the file are in ascending order.
D	Sequence checking is to be done. Records in the file are in descending order.

Use column 18 to indicate whether the program is to check the sequence of records. Column 18 applies to input, update, or combined files used as primary or secondary files. Sequence checking can be done for disk files (except those processed randomly) and CONSOLE files. Use columns 61 and 62 of the input specifications to identify the record fields containing the sequence information.

Sequence checking is required when match fields are used in the records from the file. When a record from a matching input file is found to be out of sequence, the program halts and the operator has three options:

- Bypass the record out of sequence and read the next record from the same file.
- Bypass the record out of sequence, turn on the LR indicator, and perform all end-of-job and final total procedures.
- Cancel the entire program.

If column 18 contains an entry and matching records are specified, the entry in column 18 must be the same for all files. If column 18 is left blank and matching records are specified, then ascending order is assumed for a primary file and the sequence of the primary file is assumed for all secondary files.

COLUMN 19 (FILE FORMAT)

Entry	Explanation
F or blank	Fixed-length records

An F in column 19 indicates that all records in the file are of the same length. If this column is blank, F is assumed.

COLUMNS 20-23 (BLOCK LENGTH)

Entry	Explanation
Blank	The block length for this file equals the record length. These columns must be blank for a WORKSTN file and can be blank for any other file.
1-9999	Block length for disk equals the record length or is a multiple of the record length.
1-9999	Block length for a SPECIAL file equals the record length or is greater than the record length.
1-4075	Block length for a BSCA file equals the record length or is a multiple of the record length.
2-1518	Block length for a CONSOLE file, if entered, must equal the record length.
1-79	Length of largest field keyed for a KEYBORD file.
1-79	Length of largest output record for a CRT file.
1-198	Length of largest output record for a printer file. (Entries from 133 through 198 should only be used for printers with 198 print positions.)

Use columns 20 through 23 to specify the block length for the file. The entry made in columns 20 through 23 depends on the device named for the file. The block length entry must end in column 23, and leading zeros can be omitted (see Figure 3-2).

The function of the block length entry is to specify the amount of main storage to use for the input/output area. The maximum block length is 9999. The block length entered for disk files must equal the record length or be a multiple of the record length. If the record length is entered but the block length is not specified, RPG assumes the block length equals the record length.

COLUMNS 24-27 (RECORD LENGTH)

Entry	Explanation
1-4096	Record length for disk or SPECIAL files.
1-4075	Record length for BSCA files.
2-1518	Record length for CONSOLE files.
1-79	Length of largest field keyed for KEYBORD files.
1-79	Length of largest output record for CRT files.
1-198	Length of largest output record for printer files. (Entries from 133 through 198 should only be used for printers with 198 print positions.)
2-58	Twice the record address field length for a record address file assigned to the CONSOLE device.
1-9999	Length of largest input or output record for a WORKSTN file.

Use columns 24 through 27 to indicate the length of the records in a file. An entry must be made for each file, and the entry depends on the device named for the file. Entries in these columns must end in column 27, and leading zeros can be omitted (Figure 3-2).

All records in one file must be the same length. (For update files, the length of the record after the record is updated must be the same as it was before the record was updated.) The maximum length allowed depends upon the device assigned to the file (see Figure 3-2). The record length specified can be shorter than the maximum length allowed for the device but not longer.

Cols 40–46 Device	Cols 20–23 Block Length[1]	Cols 24–27 Record Length	Maximum Record Length
DISK	Record length or a multiple of record length	Record length	4096
CONSOLE	Record length	Record length	1518
	Record address file record length	Record length	58
KEYBORD	Length of largest field to be keyed	Length of largest field to be keyed	79 – alphameric 15 – numeric
PRINTER	Record length	Record length	198
CRT	Length of longest output record	Length of longest output record	79
SPECIAL	Record length or greater than the record length	Record length	4096
BSCA	Record length or a multiple of record length	Record length	4075
WORKSTN	Must be blank	Length of longest input or output record	9999

[1] Block length must be blank for a WORKSTN file and can be blank for any other file.

Figure 3-2. Block Length and Record Length Entries

The record length for KEYBORD files should be the length of the largest field to be keyed (that is, the record length equals the largest field length specified in columns 49 through 51 of the calculation specifications when the KEY operation code is used). If the KEY operation is used to display a message, you must also consider the length of the message when you specify the record length for the KEYBORD file. The maximum alphameric field length is 79 characters, and the maximum numeric field length is 15 characters. If the record length specified for a KEYBORD file is 40 or less, a display of six lines with 40 characters per line is centered both vertically and horizontally on the display screen. If the record length is greater than 40, the display consists of 24 lines with 79 characters per line.

COLUMN 28 (MODE OF PROCESSING)

Entry	Explanation
Blank	Consecutive[1] Sequential by key[1]
L	Sequential within limits[1]
R	Random by relative record number Random by key Random by addrout file Direct file load (random load)

[1] See *Shared File Considerations* in this chapter.

Use column 28 to indicate the method by which records are to be read from the file, or to indicate that a direct file load (random load) is to take place.

For disk files specified as primary, secondary, demand, or chained, the possible processing methods depend upon the organizations of the files (see Figure 3-3). For the other types of files, consecutive processing is the only possible method.

Column 31 further identifies the access method for the program. See *Column 31 (Record Address Type)* in this chapter.

	Primary, Secondary, or Demand Files
File Organization	**Possible Processing Methods**
Sequential	– Consecutively – Randomly by addrout file
Direct	– Consecutively – Randomly by addrout file (except demand files)
Indexed	– Sequentially by key – Sequentially within limits – Randomly by addrout file – Consecutively (not using the index)

	Chained Files
File Organization	**Possible Processing Methods**
Sequential	Randomly by relative record number
Direct	Randomly by relative record number
Indexed	Randomly by key or relative record number

Figure 3-3. Possible Processing Methods for Disk Files

Consecutive

The consecutive processing method applies to sequential, indexed, and direct input disk files (blank in column 31). During consecutive processing, records are read in the order they appear in the file. The contents of spaces left for missing records in direct files are read as though the records were there, unless the file is defined as delete-capable. If the file is delete-capable, deleted or missing records are bypassed. (When a non-delete-capable direct file is loaded, such spaces are filled with blanks. If the file is delete-capable, the spaces are filled with hex FFs to designate deleted records.) If an indexed file is processed consecutively, the index is not used. Records in the file can only be read; they cannot be updated or added to the file.

The program reads records from the file until either the end of that file is reached or the program ends because of an end-of-file condition of another file. See *Column 17 (End of File)* in this chapter for more information about the second condition.

Sequential by Key

The sequential-by-key method of processing applies only to indexed disk files that are used as primary files, secondary files, or demand files.

Records are read in ascending key sequence. However, records added to the file since the last sort of the index cannot be accessed unless the IFILE attribute is specified. If the file is to be processed sequentially as an input file or if the file is to be updated or added to, the keys must be sorted to allow access to all the added records in the file. To ensure that the keys are sorted, make sure that the FILE OCL statement does not have DISP-SHR specified, or use the KEYSORT procedure (see *Shared File Considerations* in this chapter). The program reads records until all records in the file are processed or the program ends because of the end-of-file condition of another file. See *Column 17 (End of File)* in this chapter for more information about the second condition.

Sequential Within Limits

The sequential-within-limits method of processing applies only to indexed disk files used as primary files, secondary files, or demand files. A limits record consists of the lowest record key and the highest record key of the records in the indexed disk file that are to be read. Limits records are contained in a record address file. The record address file can be a disk file or a CONSOLE file.

The sequential-within-limits method can be executed when you use either (1) a record address file containing limits records or (2) the SETLL operation code in calculation specifications. However, records added to the file since the last sort of the index cannot be accessed unless the IFILE attribute is specified (see *Shared File Considerations* in this chapter).

To process sequentially within limits, the program reads:

- A limits record from the record address file

- Records with keys greater than or equal to the low record key and less than or equal to the high record key

The program repeats this procedure until either the end of the record address file is reached or the program ends because of the end-of-file condition of another file. See *Column 17 (End of File)* in this chapter for more information about the second condition.

The format of records in a record address file containing limits must conform to these rules:

- Only one set of limits is allowed per record in the record address file. The length of a record in a record address file, therefore, must be twice the length of the record key.

- The low record key must begin in position 1 of the record. The high record key must immediately follow the low record key. A record key can be from 1 to 29 characters in length.

- The low record key and the high record key must have the same length, and each key must have the same length as the key field length specified in columns 29 and 30. Therefore, leading zeros may be necessary when numeric record keys are specified.

- An alphameric record key can contain blanks.

The record address file containing the limits and the files being processed by limits can have record keys in different formats. For example, one file can have packed keys and the other zoned decimal keys. During execution time, the format of the key from the record address file is changed to the format of the record key in the file being processed by limits. If the formats differ, the format of the keys for each file must be indicated by an A or a P in column 31. Also, the zoned decimal key length must be twice the packed length, minus one or two. See *Packed Decimal Format (P)* in Chapter 7 for more information concerning this calculation.

Note: A key cannot contain any hex FF characters.

The same set of limits can appear in more than one record in the record address file. Data records, therefore, can be processed as many times as you want.

If the two record keys in a limits record are equal, only one data record is read.

Note: Double buffering (column 32) should not be specified for the record address file.

The SETLL operation code method of limits processing applies to any indexed disk file used as a demand file (D in column 16 and L in column 28 of the file description specifications). You cannot, however, process an indexed demand file with SETLL if you are using a record address file to set the limits of the file.

The maximum number of files that can be processed with the SETLL operation is limited by the number of demand files permitted in an RPG II program. A maximum of 15 demand and/or chained files is allowed per program. See Figure 3-4 for an example of SETLL. For more information on how to use the SETLL operation code to set limits, see Chapter 10, *Operation Codes*.

When the end-of-file indicator is turned on, another SETLL can be specified and processing of the file can continue. However, it is not necessary to wait for end of file before you specify another SETLL operation.

Random by Relative Record Number or Key

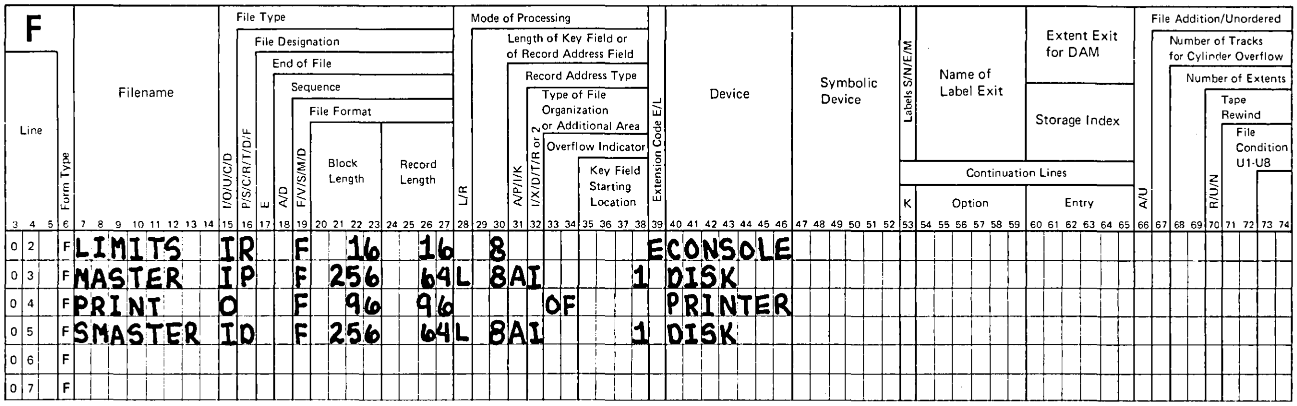
Random processing by relative record number or by key applies to chained files only. Either method requires use of the CHAIN operation code. The records of a file to be read or written must be processed by the CHAIN operation code. The records are read only when the CHAIN statements that identify them are executed.

For sequential and direct files, relative record numbers must be used to identify the records (see Figure 3-5). Relative record numbers identify the positions of the records relative to the beginning of the file. For example, the relative record numbers of the first, fifth, and seventh records in a file are 1, 5, and 7 respectively.

For indexed files, record keys must be used to identify the records (see Figure 3-6). A record key is the information from the key field of a record. The information is used in the index portion of the file to identify the record.

When random processing is used, records are read from the chained update file during the calculation phase of the program. Records can be read during total calculations and updated during total output, or be read during detail calculations and updated during detail output (see Figure 3-7).

File Description Specifications

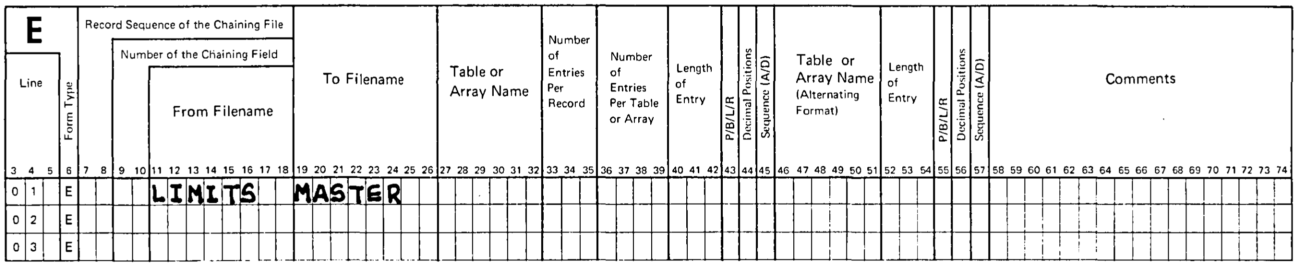


Line	Form Type	Filename	File Type (15)	File Designation (16)	File Format (19)	Block Length	Record Length	L/R (28)	Length of Key Field or of Record Address Field	Record Address Type	Type of File Organization or Additional Area	Overflow Indicator	Key Field Starting Location	Extension Code E/L	Device
02	F	LIMITS	I	R	F	16	16		8					E	CONSOLE
03	F	MASTER	I	P	F	256	64	L	8	A	I		1		DISK
04	F	PRINT	O		F	96	96					OF			PRINTER
05	F	SMASTER	I	D	F	256	64	L	8	A	I		1		DISK
06	F														
07	F														

The input disk file, MASTER, described as an indexed file to be processed by record keys is to be processed within the limits contained on the record address file, LIMITS. The LIMITS file, which is further described on the extension specifications sheet, is entered from the CONSOLE device.

Each set of limits read from LIMITS consists of the low and high account numbers to be processed. Because the account number key field (ACCT) is eight positions long, each set of limits includes two 8-position keys.

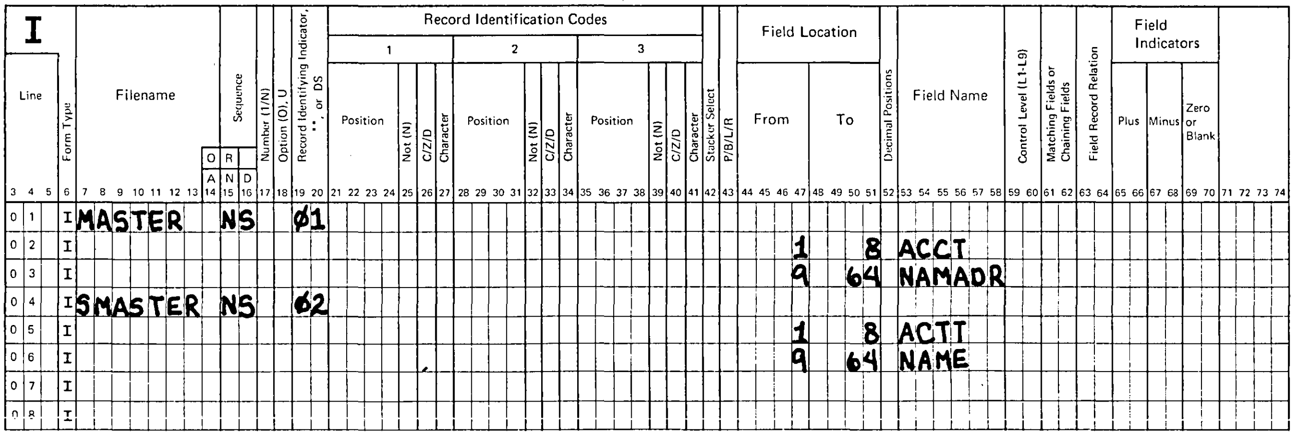
Extension Specifications



Line	Form Type	From Filename	To Filename
01	E	LIMITS	MASTER
02	E		
03	E		

As MASTER is processed within each set of limits, the corresponding records are written out on the printer output file, PRINT. Processing is complete when all sets of limits have been processed.

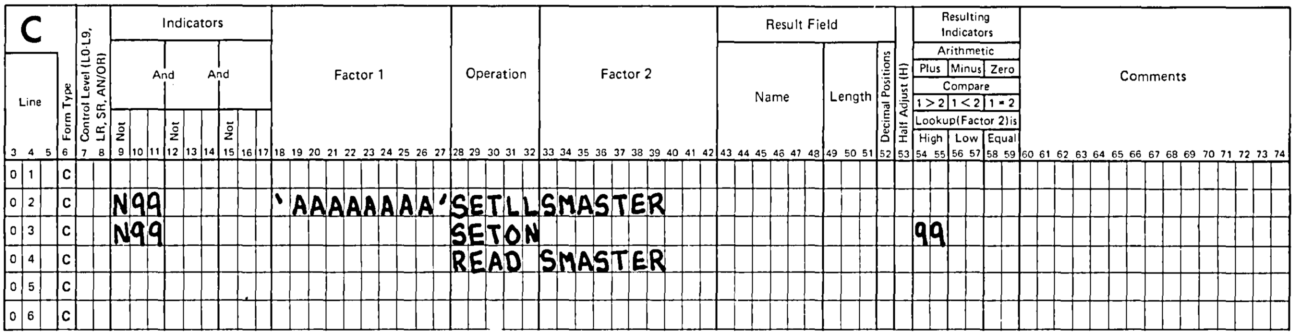
Input Specifications



Line	Form Type	Filename	Sequence	Record Identifying Indicator	Field Location From	Field Location To	Field Name
01	I	MASTER	NS	Ø1			
02	I				1	8	ACCT
03	I				9	64	NAMADR
04	I	SMASTER	NS	Ø2			
05	I				1	8	ACTT
06	I				9	64	NAME
07	I						
08	I						

Figure 3-4 (Part 1 of 2). Processing Indexed Files Sequentially Within Limits

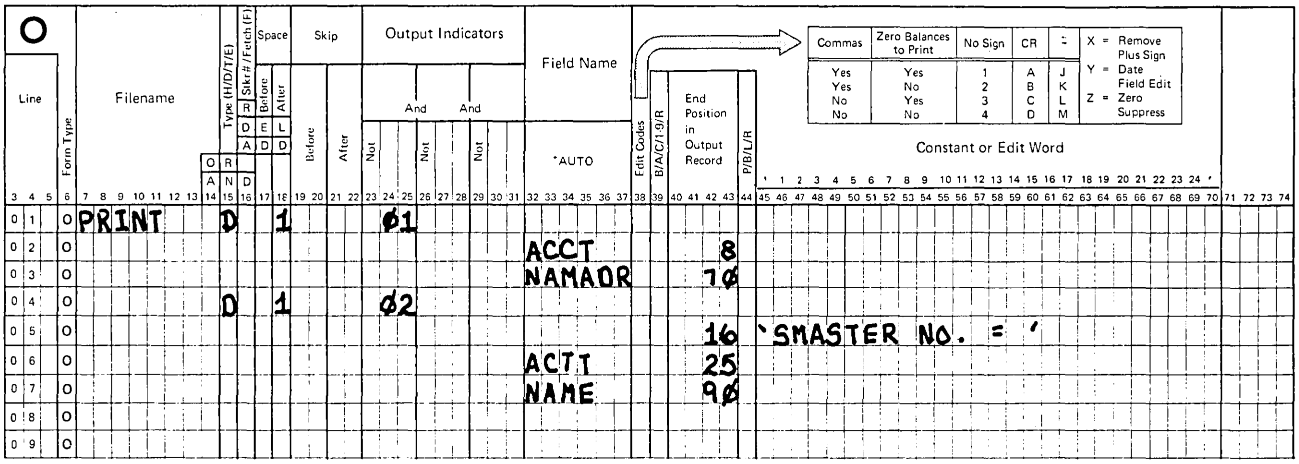
Calculation Specifications



```
01 C
02 C   N99      'AAAAAAAA'SETLLSMASTER
03 C   N99                SETON                        99
04 C                      READ SMASTER
05 C
06 C
```

File SMASTER is processed by the SETLL operation code. It is characterized by having no extension specifications, and its filename appears in factor 2 of the SETLL operation code. In this example the first record read from file SMASTER would be the one whose key is equal to or the next higher than the literal 'AAAAAAAA'. Records are read sequentially to end of file unless the cycle is interrupted by additional SETLL operations.

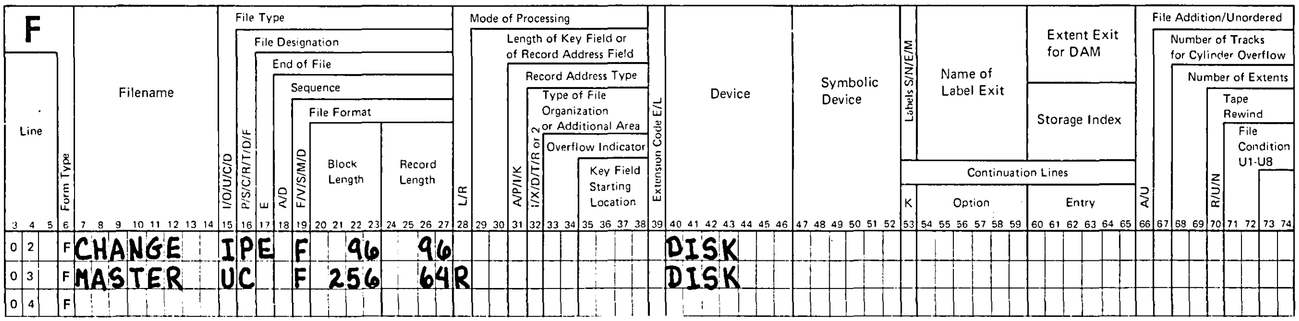
Output Specifications



```
01 OPRINT   D  1     Ø1
02 O                           ACCT      8
03 O                           NAMADR   7Ø
04 O        D  1     Ø2
05 O                                    16 'SMASTER NO. = '
06 O                           ACTT     25
07 O                           NAME     9Ø
08 O
09 O
```

Figure 3-4 (Part 2 of 2). Processing Indexed Files Sequentially Within Limits

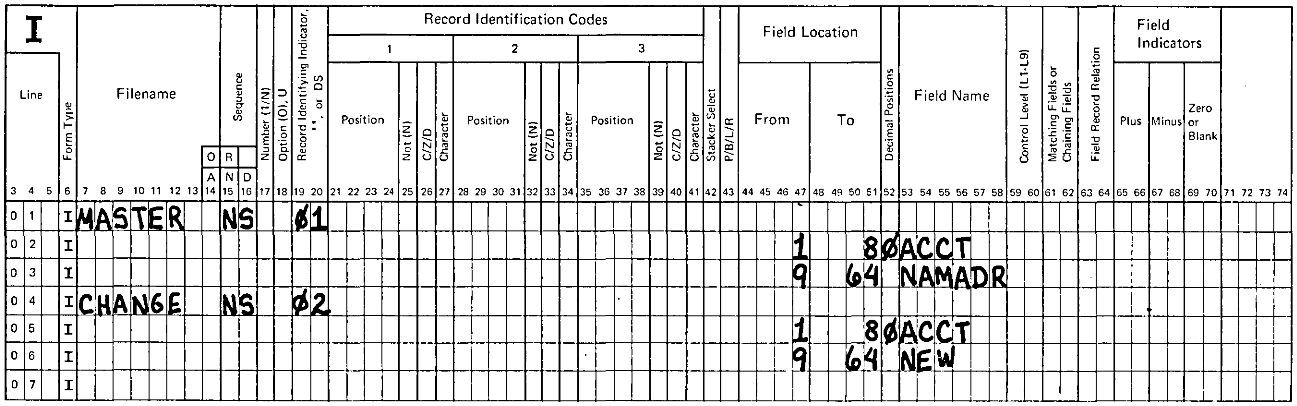
File Description Specifications


F
Line
Form Type
Filename
File Type
File Designation
End of File
Sequence
File Format
I/O/U/C/D
P/S/C/R/T/D/F
E
A/D
F/V/S/M/D
Block Length
Record Length
Mode of Processing
Length of Key Field or of Record Address Field
Record Address Type
Type of File Organization or Additional Area
Overflow Indicator
Key Field Starting Location
L/R
A/P/I/K
I/X/D/T/R or 2
Extension Code E/L
Device
Symbolic Device
Labels S/N/E/M
Name of Label Exit
Extent Exit for DAM
Storage Index
Continuation Lines
K
Option
Entry
A/U
File Addition/Unordered
Number of Tracks for Cylinder Overflow
Number of Extents
Tape Rewind
File Condition U1-U8
R/U/N

```
  02 FCHANGE  IPE F  96  96            DISK
  03 FMASTER  UC  F 256  64R           DISK
  04 F
```

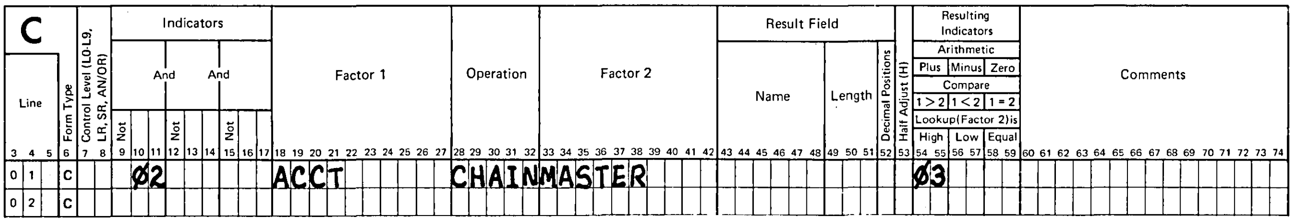
The direct update file, MASTER, is to be processed randomly by relative record numbers. The account number (ACCT) from the primary DISK file, CHANGE, is used as the relative record number. As each record is read from CHANGE, the MASTER record corresponding to the account number is read during calculation time by the CHAIN operation code. At detail output time, the data in the NEW field replaces the original data in the NAMADR field and the updated MASTER record is output to its original relative record location on the disk file.

Input Specifications


I
Line
Form Type
Filename
O R
A N D
Sequence
Number (1/N)
Option (O), U
Record Identifying Indicator, **, or DS
Record Identification Codes
1
2
3
Position
Not (N)
C/Z/D
Character
Stacker Select
P/B/L/R
Field Location
From
To
Decimal Positions
Field Name
Control Level (L1-L9)
Matching Fields or Chaining Fields
Field Record Relation
Field Indicators
Plus
Minus
Zero or Blank

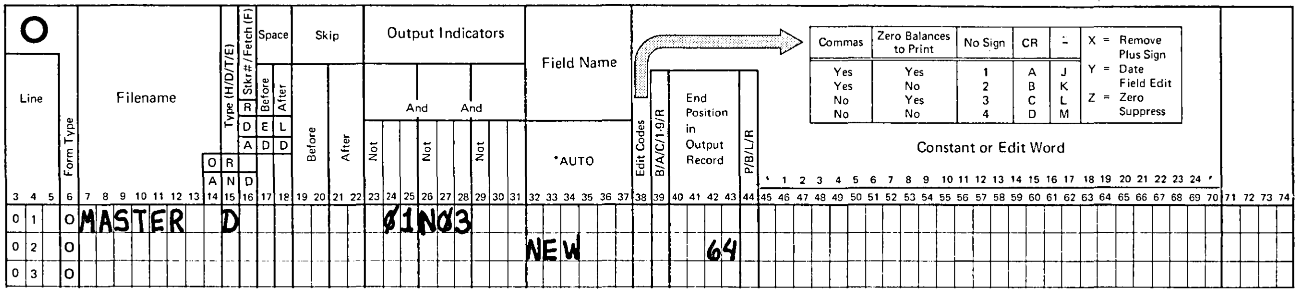
```
  01 IMASTER  NS  Ø1
  02 I                                        1   8ØACCT
  03 I                                        9  64 NAMADR
  04 ICHANGE  NS  Ø2
  05 I                                        1   8ØACCT
  06 I                                        9  64 NEW
  07 I
```

Calculation Specifications


C
Line
Form Type
Control Level (L0-L9, LR, SR, AN/OR)
Indicators
And
And
Not
Factor 1
Operation
Factor 2
Result Field
Name
Length
Decimal Positions
Half Adjust (H)
Resulting Indicators
Arithmetic
Plus
Minus
Zero
Compare
1 > 2
1 < 2
1 = 2
Lookup(Factor 2)is
High
Low
Equal
Comments

```
  01 C   Ø2      ACCT      CHAINMASTER                Ø3
  02 C
```

Output Specifications


O
Line
Form Type
Filename
O R
A N D
Type (H/D/T/E)
Stkr #/Fetch (F)
Space
Before
After
Skip
Output Indicators
And
And
Not
Field Name
*AUTO
Edit Codes
B/A/C/1-9/R
End Position in Output Record
P/B/L/R
Commas
Zero Balances to Print
No Sign
CR
-
X = Remove Plus Sign
Y = Date Field Edit
Z = Zero Suppress
Constant or Edit Word

Commas	Zero Balances to Print	No Sign	CR	-
Yes	Yes	1	A	J
Yes	No	2	B	K
No	Yes	3	C	L
No	No	4	D	M

```
  01 OMASTER  D       Ø1NØ3
  02 O                           NEW       64
  03 O
```

Figure 3-5. Random Processing of a Direct File by Relative Record Number

File Description Specifications

Line	Form Type	Filename	File Type	File Designation	End of File	Sequence	File Format	Block Length	Record Length	L/R	Length of Key Field or of Record Address Field	Record Address Type	Type of File Organization or Additional Area	Overflow Indicator	Key Field Starting Location	Extension Code E/L	Device	Symbolic Device	Labels S/N/E/M	Name of Label Exit	Extent Exit for DAM / Storage Index	A/U	File Addition/Unordered / Number of Tracks for Cylinder Overflow / Number of Extents / Tape Rewind / File Condition U1-U8
0 2	F	CHANGE	I	P	E		F	96	96								DISK						
0 3	F	MASTER	U	C			F	256	64	R	8	A	I		1		DISK						
0 4	F																						

MASTER, a chained update file, is described on the file description specifications sheet as an indexed file to be processed by keys. As each record is read from the input disk file, CHANGE, the account number (ACCT) is used as the key to chain to the corresponding record in MASTER at calculation time. At detail output time, the data in the NEW field of CHANGE replaces the original data in the NAMADR field. The updated MASTER record is then written on its original disk location. See *Column 32 (File Organization or Additional I/O Area)* in this chapter for a description of indexed file organization.

Input Specifications

Line	Form Type	Filename	Sequence	Number (1/N)	Option (O), U	Record Identifying Indicator, **, or DS	Record Identification Codes 1	Record Identification Codes 2	Record Identification Codes 3	Stacker Select	P/B/L/R	Field Location From	Field Location To	Decimal Positions	Field Name	Control Level (L1-L9)	Matching Fields or Chaining Fields	Field Record Relation	Field Indicators Plus	Field Indicators Minus	Field Indicators Zero or Blank
0 1	I	MASTER	NS			Ø1															
0 2	I											1	8		ACCT						
0 3	I											9	64		NAMADR						
0 4	I	CHANGE	NS			Ø2															
0 5	I											1	8		ACCT						
0 6	I											9	64		NEW						
0 7	I																				

Calculation Specifications

Line	Form Type	Control Level (L0-L9, LR, SR, AN/OR)	Indicators	And	And	Factor 1	Operation	Factor 2	Result Field Name	Length	Decimal Positions	Half Adjust (H)	Resulting Indicators Plus / 1>2 / High	Minus / 1<2 / Low	Zero / 1=2 / Equal	Comments
0 1	C		Ø2			ACCT	CHAIN	MASTER					Ø3			
0 2	C															

Output Specifications

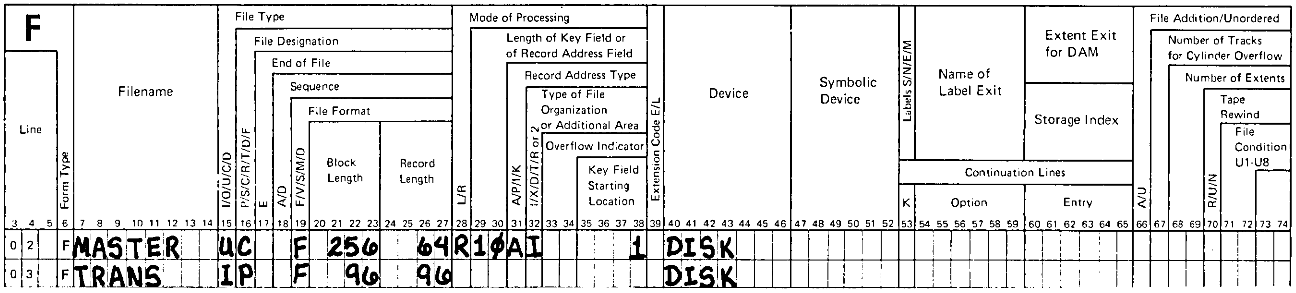
Line	Form Type	Filename	Type (H/D/T/E)	Stkr #/Fetch (F)	Space Before	Space After	Skip Before	Skip After	Output Indicators	And	And	Field Name / *AUTO	Edit Codes	B/A/C/1-9/R	End Position in Output Record	P/B/L/R	Constant or Edit Word
0 1	O	MASTER	D						Ø1	NØ3							
0 2	O											NEW			64		
0 3	O																

Commas	Zero Balances to Print	No Sign	CR	–
Yes	Yes	1	A	J
Yes	No	2	B	K
No	Yes	3	C	L
No	No	4	D	M

X = Remove Plus Sign
Y = Date Field Edit
Z = Zero Suppress

Figure 3-6. Random Processing of an Indexed File by Key

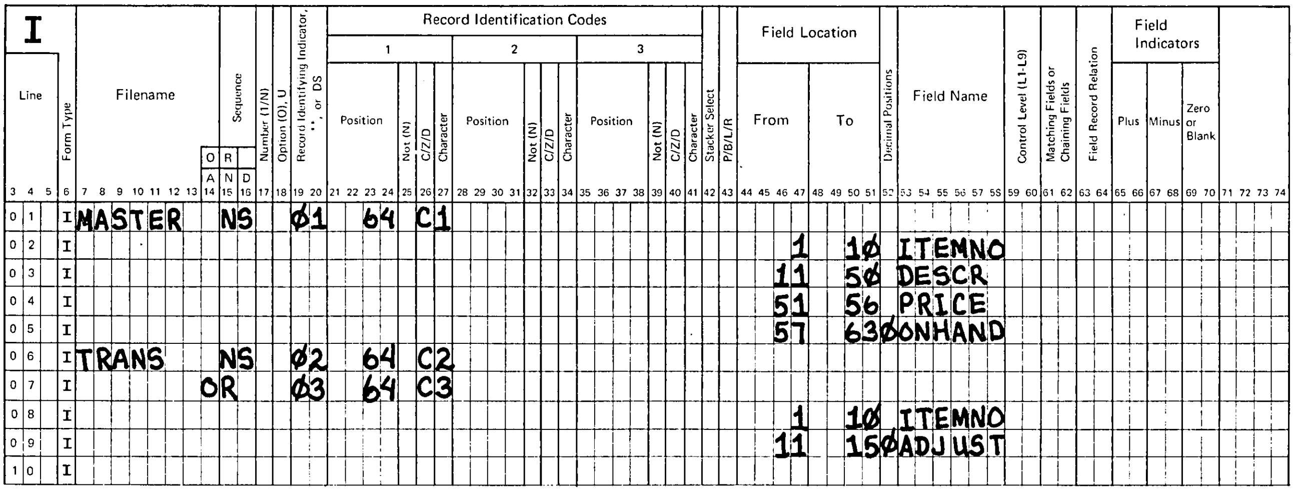
File Description Specifications



```
  02 FMASTER  UC  F 256  64R1ØAI     1 DISK
  03 FTRANS   IP  F  96  96            DISK
```

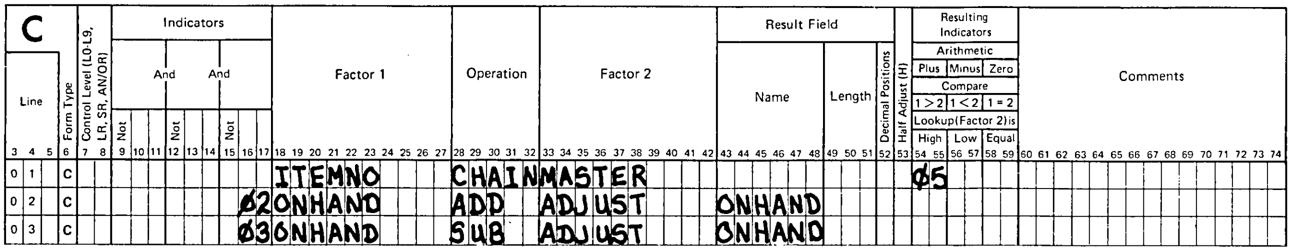
Indexed file, MASTER, is described as a chained update file to be processed by keys. The key field in MASTER is ITEMNO, in positions 1 through 10. As each record is read from TRANS, the input transaction file, the ITEMNO field is used as the key to chain to MASTER during calculations. If the character 2 is in position 64 of the transaction record, the quantity in ADJUST is added to the ONHAND field of MASTER. If the character 3 is in position 64, ADJUST is subtracted from ONHAND. If the character 1 appears in location 64 of the MASTER record, the updated ONHAND field is written out on its original location in the MASTER record at detail output time.

Input Specifications



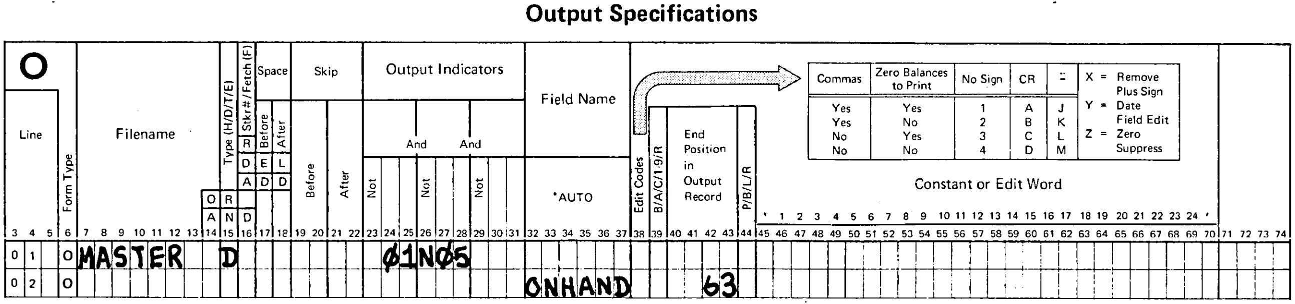
```
  01 IMASTER  NS  Ø1  64 C1
  02 I                                        1  1Ø ITEMNO
  03 I                                       11  5Ø DESCR
  04 I                                       51  56 PRICE
  05 I                                       57  63ØONHAND
  06 ITRANS    NS  Ø2  64 C2
  07 I       OR   Ø3  64 C3
  08 I                                        1  1Ø ITEMNO
  09 I                                       11  15ØADJUST
  10 I
```

Calculation Specifications



```
  01 C           ITEMNO    CHAINMASTER               Ø5
  02 C         Ø2ONHAND    ADD  ADJUST    ONHAND
  03 C         Ø3ONHAND    SUB  ADJUST    ONHAND
```

Output Specifications



```
  01 OMASTER  D        Ø1NØ5
  02 O                         ONHAND    63
```

Figure 3-7. Updating an Indexed File

Random by Addrout File

An addrout (address output) file is a record address file produced by the sort program. It contains the relative record numbers of the records in a disk file. (Each relative record number is a 3-byte binary field.) You can use addrout files to process input or update files that are designated as primary or secondary files.

When an RPG II program uses an addrout file, it reads a binary relative record number from that file. The binary relative record number is then converted to a disk address, and the record at that address in the original file is located and read. Records are read in this manner until either the end of the addrout file is reached or the program ends because of the end-of-file condition of another file (see Figure 3-8). See *Column 17 (End of File)* in this chapter for more information about the second condition.

Because addrout files are record address files, they must be further described on the extension specifications. Both the addrout file and the file to be processed by the addrout file must be described on the file description specifications.

Shared File Considerations

After an indexed file has records added to it, the keys might be out of sequence. If the file is to be processed sequentially as an input file (blank or L in column 28) or if the file is to be updated or added to, the keys must be sorted to allow access to all the records in the file, unless the IFILE attribute is specified. To ensure that the keys are sorted, make sure the FILE OCL statement does not have DISP-SHR specified, or use the KEYSORT command. An indexed file that is created (output file) has its keys sorted at job termination if an unordered load was specified (U in column 66 of the file description specifications). For further information on key sorting, see *Key Sorting for Indexed Files* in Chapter 2 of the *Concepts and Design Guide.*

IFILE support allows shared, indexed sequential processing of all added records in an indexed file. Without IFILE support, indexed sequential processing is limited to only those records having index entries in the primary portion of the index. Indexed files can be given the IFILE attribute by specifying it on the FILE OCL statement or in the SETFILE or BLDFILE procedure. For more information about IFILE support, see *Indexed Files with the IFILE Attribute* in Chapter 2 of the *System/34 Concepts and Design Guide.*

COLUMNS 29-30 (LENGTH OF KEY FIELD OR RECORD ADDRESS FIELD)

Entry	Explanation
1-29	Length of record key or relative record number

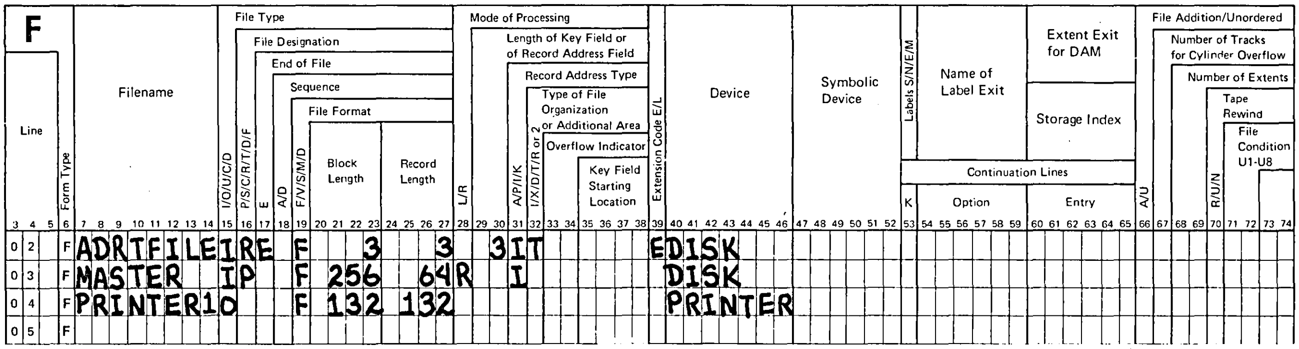
Use columns 29 and 30 to indicate:

- The length in bytes of the record keys in indexed files and record address files
- The length of the relative record numbers in addrout files, which is always 3

Columns 29 and 30 apply only to indexed files and record address files.

All of the key fields in the records in an indexed file must be the same length. The maximum length is 29 bytes (8 bytes for record keys in packed format). All of the relative record numbers contained in an addrout file are 3 characters long.

File Description Specifications

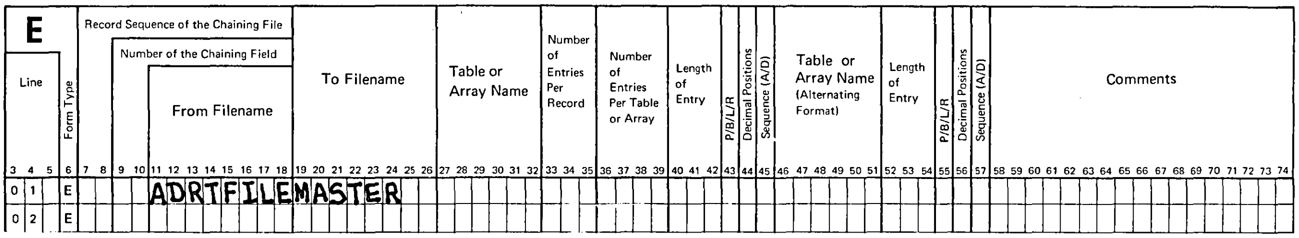


```
02 FADRTFILEIRE F   3   3  3IT      EDISK
03 FMASTER  IP  F 256  64R  I        DISK
04 FPRINTER O   F 132 132            PRINTER
05 F
```

The record address file, ADRTFILE, defined as an addrout disk file, consists of 3-byte binary relative record numbers that correspond to locations of records on the input disk file, MASTER. As each record is read from ADRTFILE, the indicated record from MASTER is located and read. For each record read from MASTER (indicator 01 is on), a detail line is printed on the printer output file, PRINTER.

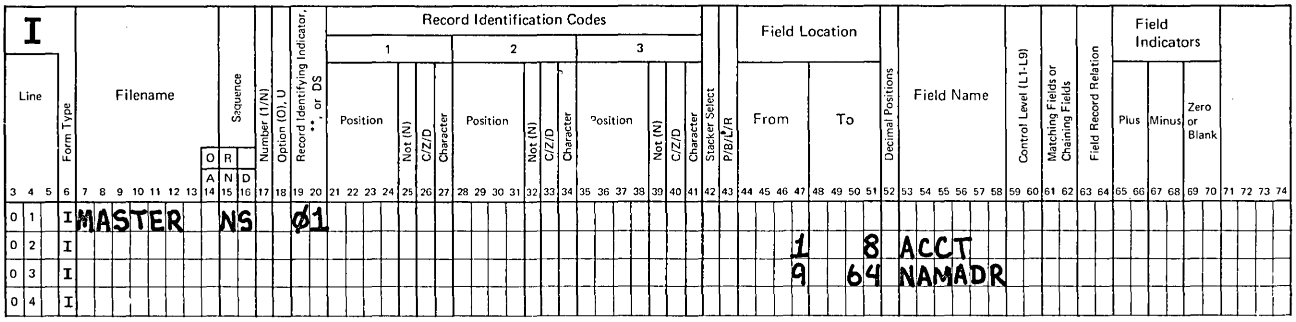
Because end of file (E in column 17 of the file description specifications sheet) is specified for the addrout file, processing continues until all records in ADRTFILE have been read.

Extension Specifications



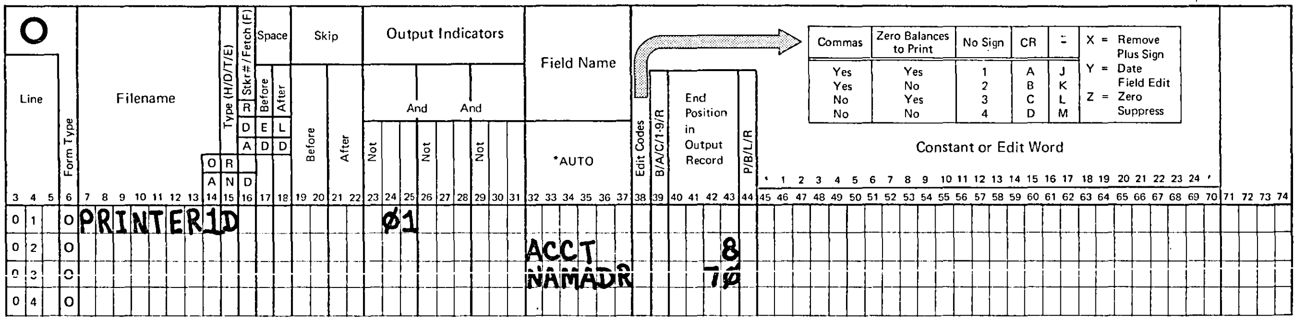
```
01 E    ADRTFILEMASTER
02 E
```

Input Specifications



```
01 IMASTER  NS  01
02 I                                       1   8 ACCT
03 I                                       9  64 NAMADR
04 I
```

Output Specifications



```
01 OPRINTER D        01
02 O                         ACCT       8
03 O                         NAMADR    70
04 O
```

Figure 3-8. Processing a Sequential Disk File With an Addrout File

COLUMN 31 (RECORD ADDRESS TYPE)

Entry	Explanation
Blank	– Relative record numbers are used in processing sequential and direct files. – A sequential or direct file is being loaded. – Records are read consecutively. – Keys in the record address file are in the same format as keys in the indexed files.
A	Record keys in zoned decimal format are used in processing or loading indexed files and processing record address files.
I	Relative record numbers from the addrout file are used to process the file, or the file is an addrout file consisting of relative record numbers.
P	Record keys in packed format are used in processing or loading indexed files and processing record address files.

Use column 31 to indicate how records in a disk file are identified. Column 31 applies to disk files specified as input, update, or chained output files. Together, columns 28 and 31 indicate:

- The method by which records are read from the file
- A direct file load

Following is the specification identifying methods for retrieving records:

Primary, Secondary, or Demand Files

Processing Method	Entry in Column 28	Entry in Column 31
Consecutive	Blank	Blank
By addrout (except demand files)	R	I
Sequential by key	Blank	A or P
Sequential within limits	L	A or P

Chained Files

Processing Method	Entry in Column 28	Entry in Column 31
Random by relative record number	R	Blank
Random by key	R	A or P
Direct file load (random load)	R	Blank[1]

[1] For nondelete-capable file, a direct file load requires an O in column 15 and a C in column 16. For delete-capable files, a direct file load requires an O in column 15 and a blank in column 16.

For addrout files, column 31 must contain an I, indicating that binary relative record numbers are used in processing.

Note: When building a file with packed keys (P in column 31), you must specify the key field as packed in your output specifications.

COLUMN 32 (FILE ORGANIZATION OR ADDITIONAL INPUT/OUTPUT AREA)

Entry	Explanation
Blank	Sequential file or direct file. The program uses one input/output area for the file.
I	Indexed file.
T	Addrout file.
1-9	Sequential file. The program uses two input/output areas for the file.

Use column 32 to (1) identify the organization of all disk files except addrout files, (2) identify addrout files, and (3) indicate whether one or two input/output areas are to be used for sequential files.

File Organization

File organization is the arrangement of records in a file. The three organizations are indexed, direct, and sequential. Files other than disk files are always sequential files. Disk files can be sequential, direct, or indexed files.

Indexed Files

An indexed file is a disk file in which the location of the records is recorded in a separate portion of the file called an index. The index and its associated records occupy adjacent positions on the disk. The index contains the record key and disk address of every record in the file (see Figure 3-9).

See *Key Sorting for Indexed Files* in Chapter 2 of the *Concepts and Design Guide* for an explanation of when keys for an indexed file are sorted.

A record key is the information from the key field of a record. The record key identifies the records in an indexed file. Record keys are always required in an indexed file. Indexed files can be loaded with the keys in ascending sequence or in unordered sequence. See *Column 66 (File Addition)* in this chapter for a definition of the unordered load function.

If an indexed file is processed consecutively, the index is not used. Records are processed in the order in which they are physically stored in the file; that is, in relative record number order. They are not retrieved in key sequence.

Direct Files

Direct files are disk files in which records are assigned specific record positions. Regardless of the order in which the records are put in the file, they always occupy a specific position in the file relative to the beginning of the file. Relative record numbers identify the position of a record within the file.

Direct File Load (Nondelete-Capable Files): To define a nondelete-capable direct file load, you must specify the disk file to be loaded as a chained output file in the file description specifications (columns 15 and 16). On the calculation specifications, factor 1 must contain either the name of a field containing the relative record number or the relative record number itself; columns 28 through 32 must contain the operation code CHAIN; and factor 2 must contain the name of the disk file to be loaded (see Figure 3-10). The field or the relative record number entered in factor 1 defines the record position for each record in the direct disk file. The relative record number can be a field or part of a field in the input records. Such fields are used for record identification of the input record, as well as for the disk records after the disk file is loaded.

Before a nondelete-capable direct file is loaded, the disk space required for the file is automatically filled with blanks. When a record is read in, the relative record number is used to chain to the corresponding relative record position in the disk file, and the information contained in the input record is then written on disk, replacing the blanks with data. If a record is missing from the input file when a nondelete-capable direct file is loaded, the space reserved for that record in the disk file remains blank until the proper record is read in later (see Figure 3-11).

Once the direct file is loaded, records can be inserted or changed in the file if you define the direct file as an update file on the file description specifications. This file can then be processed consecutively or by the CHAIN operation. (Remember that any file defined as a chained output file is cleared entirely to blanks before any records are processed.)

You might have to allow for *synonyms* when you load a nondelete-capable direct file. Synonyms are two or more records with the same relative record number. If you have synonyms, you can load the file in one of two ways:

- Clear the file to blanks in the first job by defining it as a chained output file. Once the file is cleared, you can run one or more subsequent jobs using the update function to read record locations and check for synonyms while loading the file.
- Load the direct file with records without synonyms, then run another job to identify synonyms and load them into the file.

Note: Adding records to direct disk files is different from adding records to indexed files or extending sequential files. For sequential disk files, the new record is added at the first available position at the end of the file. The same process occurs for an indexed file, except that the record key and disk address are added to the file index. Any new records added to a direct disk file already have a space reserved for them. Hence, the record is inserted in its proper place, not merely added to the physical end of the file.

Direct File Load (Delete-Capable Files): To specify a delete-capable direct file load, you must define the disk file as an output file that is processed randomly (O in column 15 and R in column 28 of the file description specifications). Also required is a file description specifications continuation line with the keyword RECNO and its associated field. The relative record number of the record to be loaded in the file must be placed in the RECNO field before the record is loaded (for an example of this method of direct file load, see Figure 3-12).

Unlike the nondelete-capable direct file load, this method of direct file load does not use a CHAIN operation to indicate where the record is to be located. You must place the relative record number in the RECNO field. The records are written to the output file in the same way they are written to any device. The records can be written as heading, detail, total, or exception output. If there is a record present with the same relative record number as the record you are loading, an error message is displayed and the operator can continue, bypassing the duplicate record. With this method of direct file load, the system checks for synonyms.

Before a delete-capable direct file is loaded, the disk space required for the file is initialized to deleted records (hex FFs). The relative record number that the programmer places in the RECNO field indicates where the record is to be loaded in the file. The information in the record is written over the deleted record, replacing the hex FFs with data. If a deleted record is not replaced with data, it remains in the file. A record can later be added at this relative record number (see *Adding Records to Delete-Capable Direct and Sequential Files* in this chapter). A deleted record cannot be accessed by the program. If a deleted record is accessed by a CHAIN operation, the no-record-found indicator (specified in columns 54 and 55) is set on.

This method of direct file load can be used only for delete-capable files. If you attempt to load a nondelete-capable file this way, an error occurs and a message is displayed. Nondelete-capable files must be loaded using the CHAIN operation.

Sequential Files

Sequential files are files in which the order of the records is determined by the order in which the records are put in the file. For example, the tenth record put in the file occupies the tenth record position (see Figure 3-13).

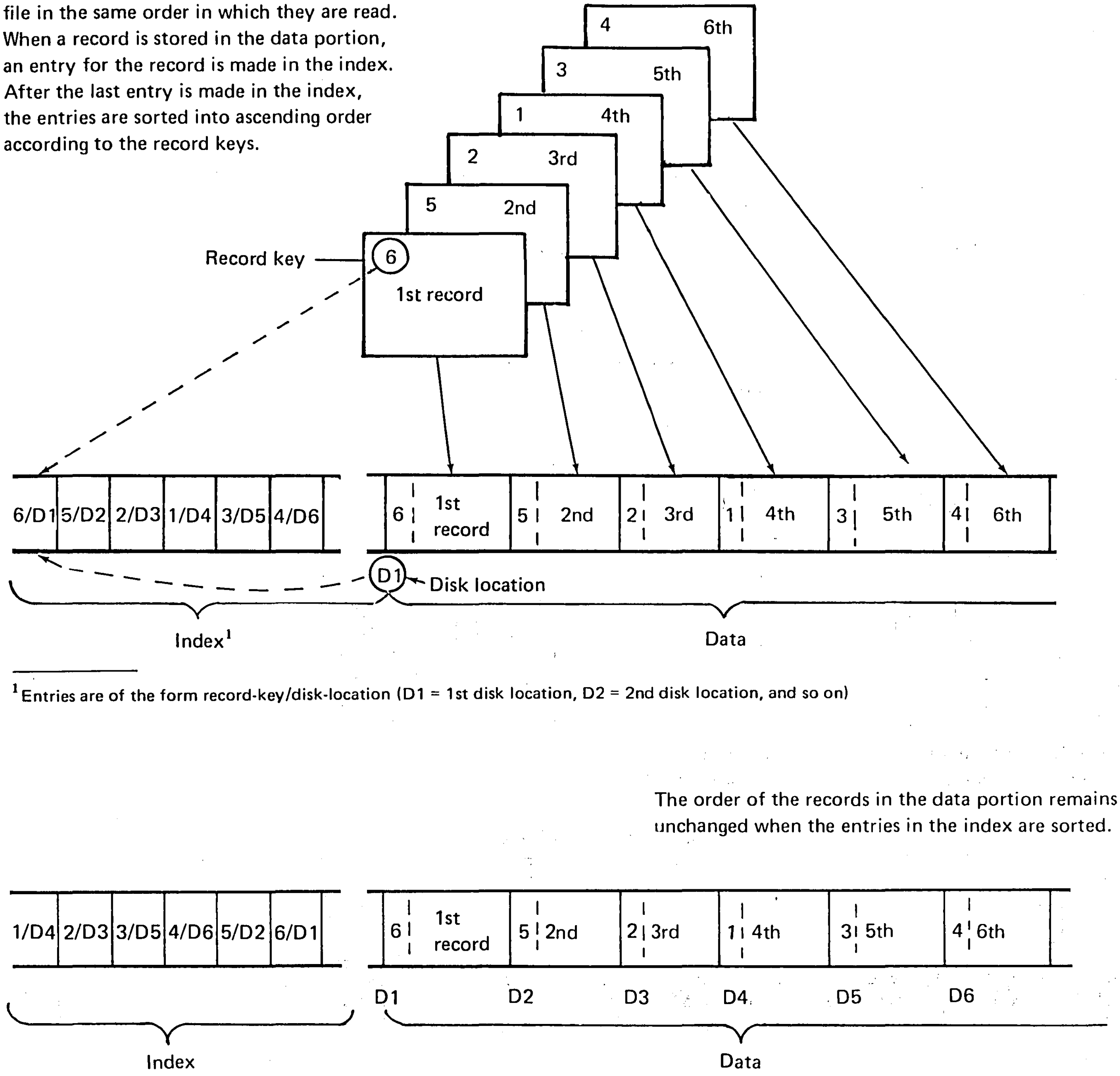
Additional Input/Output Area

Normally the program uses one input/output area for each file. A second area, however, can be used for sequential files specified as input or output files in column 15.

The use of two input/output areas increases the efficiency of the program. However, it also increases the size of the program. Therefore, before indicating that two areas are to be used for a file, be sure that the increase in size does not make your program exceed the capacity of your system.

Additional input/output areas cannot be specified for table files, for demand files (except for BSCA demand files), for indexed files, for chained files, for direct files, or for disk files with a shared input/output area (a 1 in column 48 of the control specifications). If both additional and shared input/output areas are specified, the additional input/output area specification is ignored and a warning message is issued.

Records are stored in the data portion of the file in the same order in which they are read. When a record is stored in the data portion, an entry for the record is made in the index. After the last entry is made in the index, the entries are sorted into ascending order according to the record keys.

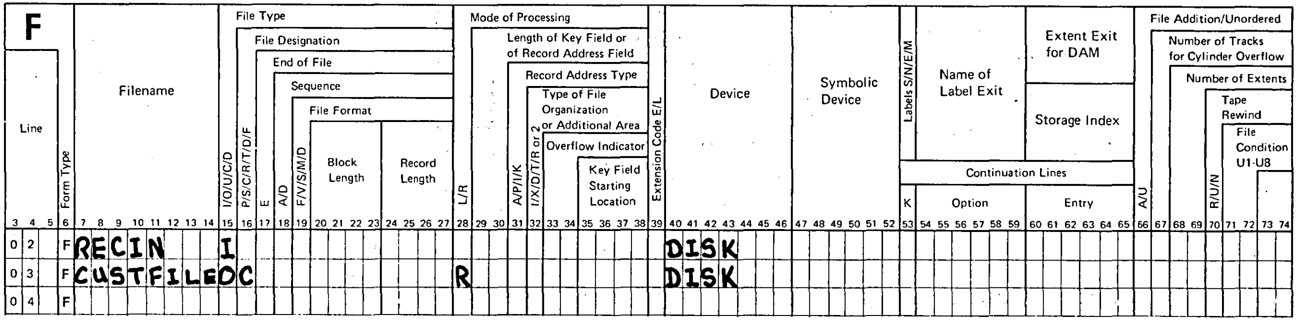


[1] Entries are of the form record-key/disk-location (D1 = 1st disk location, D2 = 2nd disk location, and so on)

The order of the records in the data portion remains unchanged when the entries in the index are sorted.

Figure 3-9. Indexed File Organization

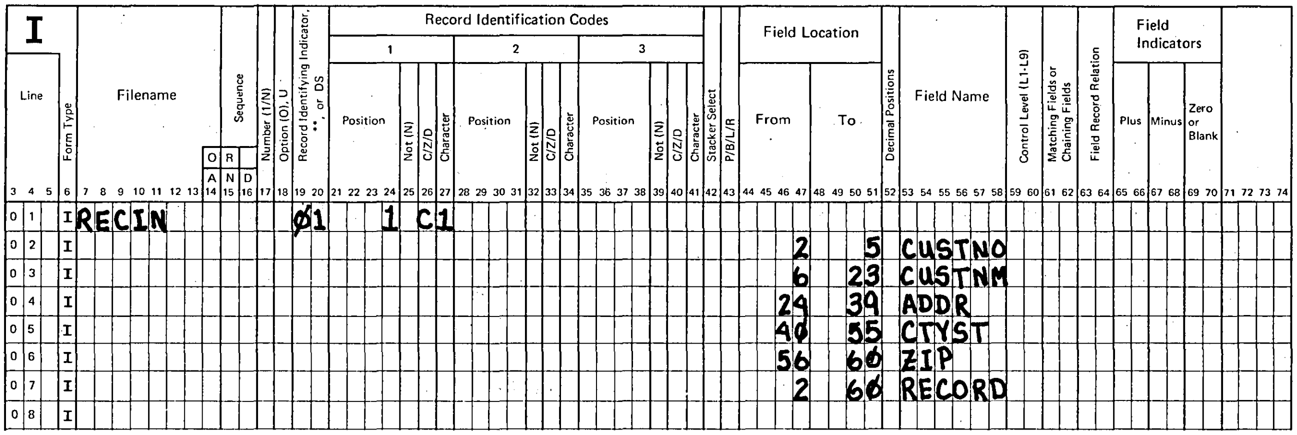
File Description Specifications



Line	Form Type	Filename	File Type	File Designation	L/R	Device
02	F	RECIN	I			DISK
03	F	CUSTFILE	O	C	R	DISK
04	F					

The direct file being created, CUSTFILE, is defined as chained output file.

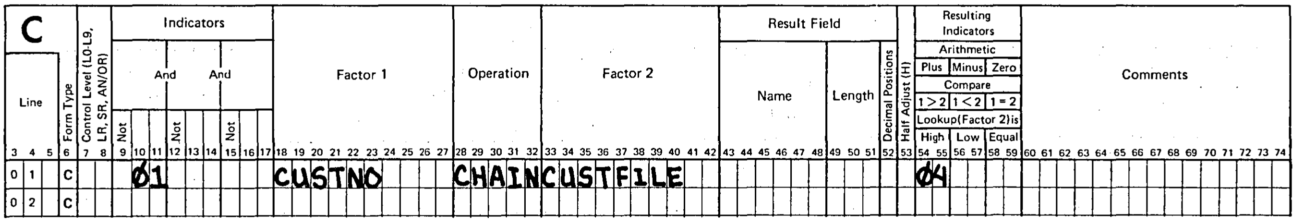
Input Specifications



Line	Form Type	Filename	Record Identifying Indicator	Position	C/Z/D	Character	From	To	Field Name
01	I	RECIN	Ø1	1	C	1			
02	I						2	5	CUSTNO
03	I						6	23	CUSTNM
04	I						24	39	ADDR
05	I						40	55	CTYST
06	I						56	60	ZIP
07	I						2	60	RECORD
08	I								

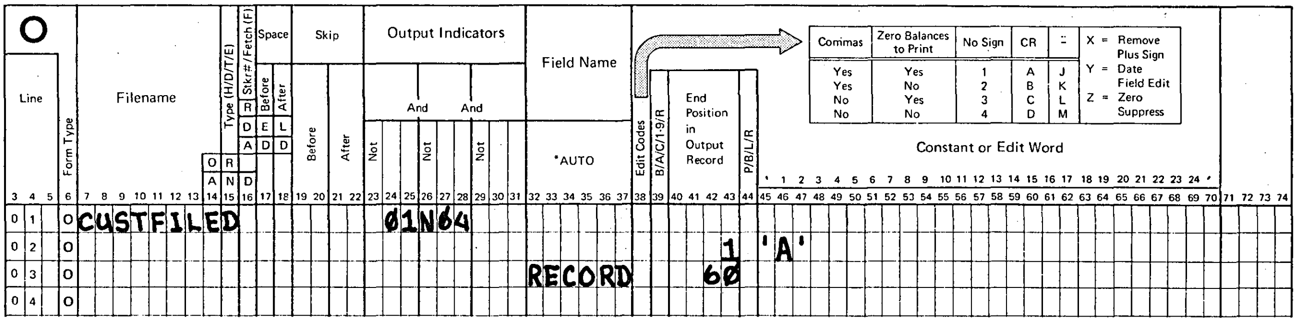
CUSTNO field from input records is used to chain to the direct file. Indicator 04 turns on if the record is not found.

Calculation Specifications



Line	Form Type	Indicators	Factor 1	Operation	Factor 2	Resulting Indicators (Plus / 1>2 / High)
01	C	Ø1	CUSTNO	CHAIN	CUSTFILE	Ø4
02	C					

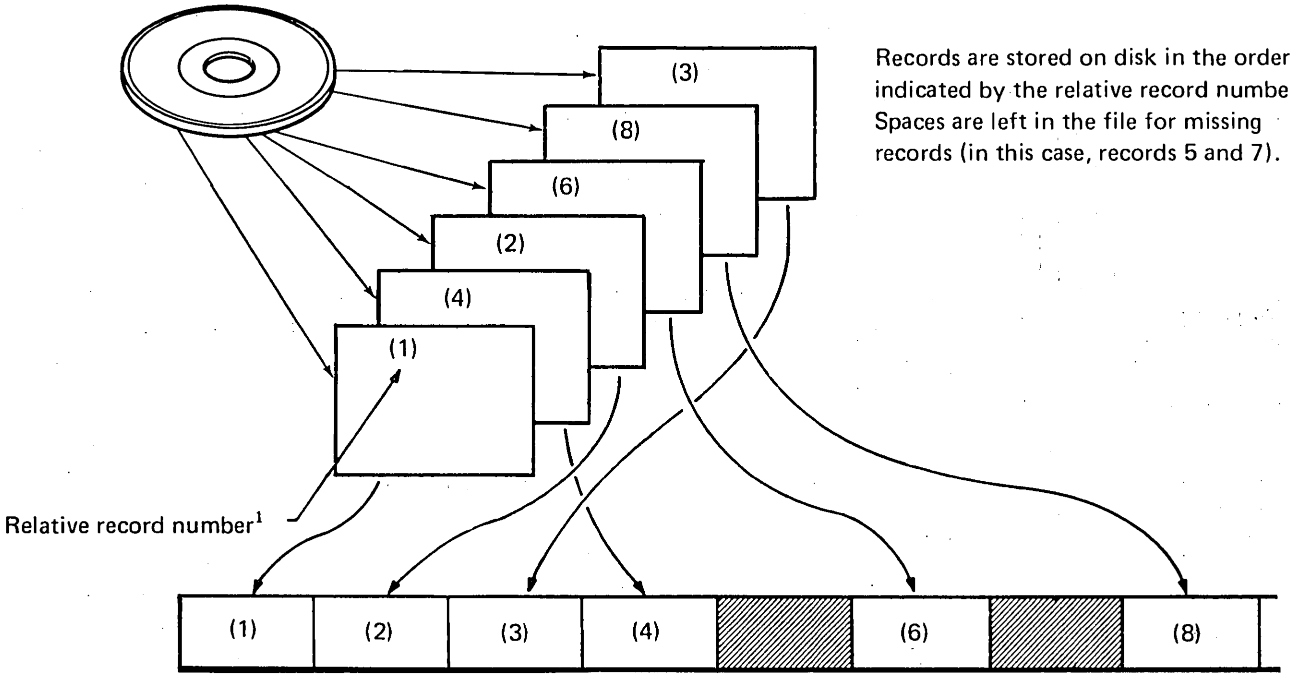
Output Specifications



Line	Form Type	Filename	Type (H/D/T/E)	Output Indicators	Field Name	End Position in Output Record	Constant or Edit Word
01	O	CUSTFILE	D	Ø1N64			
02	O					1	'A'
03	O				RECORD	6Ø	
04	O						

A record is written for each successful CHAIN operation.

Figure 3-10. Using CHAIN to Load a Direct File

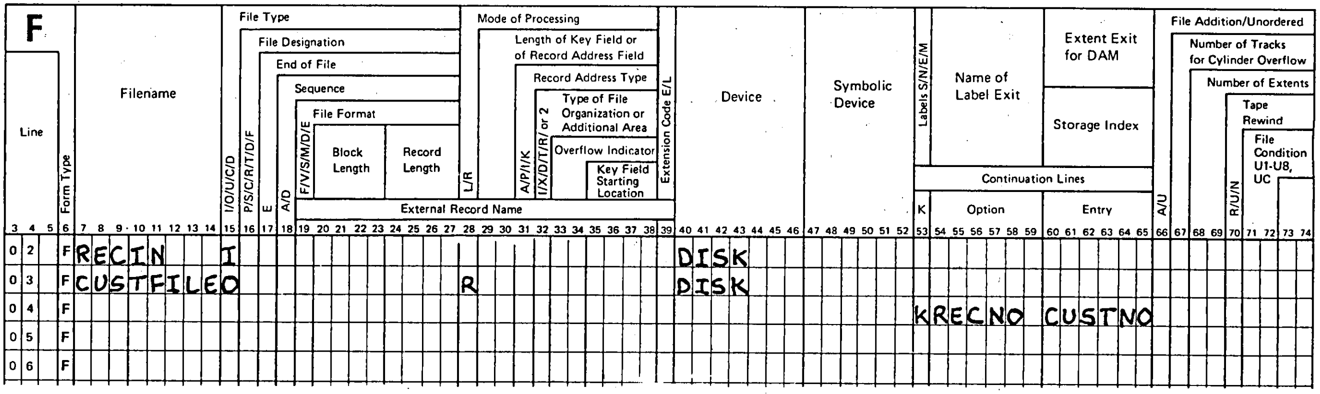


Records are stored on disk in the order indicated by the relative record numbers. Spaces are left in the file for missing records (in this case, records 5 and 7).

[1] The programmer usually derives relative record numbers from information in the records.

Figure 3-11. Direct File Organization

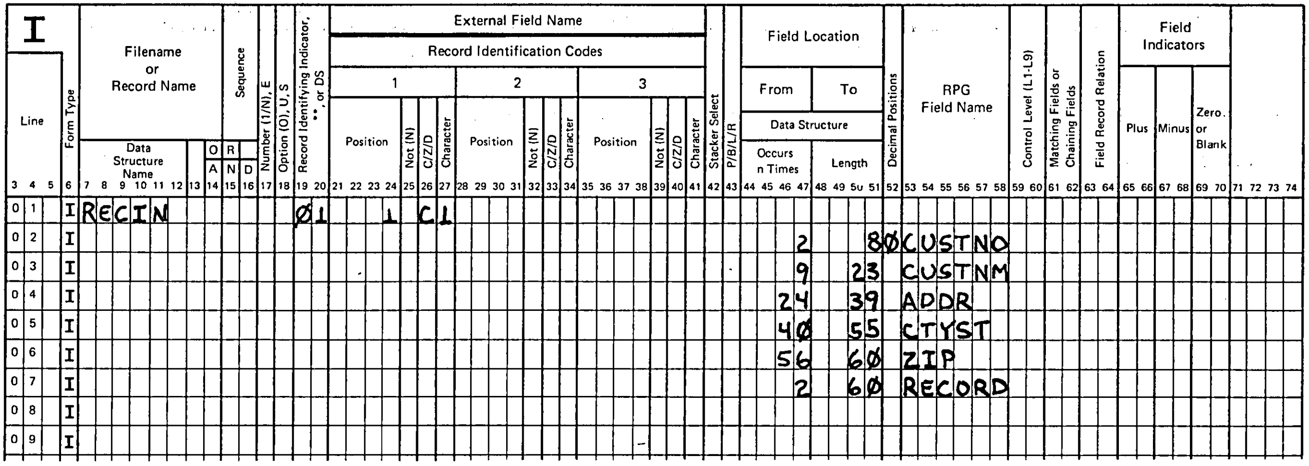
File Description Specifications



```
02 FRECIN   I                        DISK
03 FCUSTFILEO            R           DISK
04 F                                              KRECNO CUSTNO
05 F
06 F
```

The direct file being created, CUSTFILE, is defined as an output file that is processed randomly. A file description specifications continuation line containing the keyword RECNO specifies that RPG uses the contents of the field CUSTNO as the relative record number of the record to be loaded.

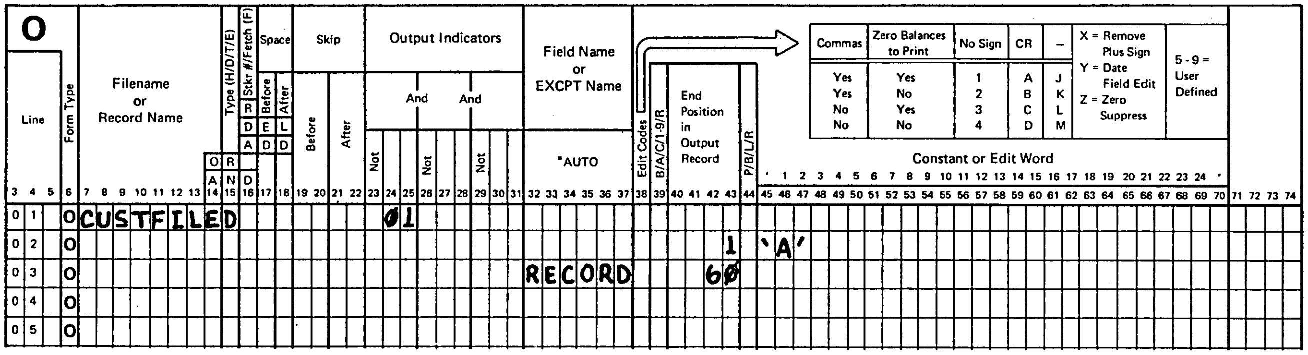
Input Specifications



```
01 IRECIN       Ø1   1 C1
02 I                                        2   8ØCUSTNO
03 I                                        9  23 CUSTNM
04 I                                       24  39 ADDR
05 I                                       4Ø  55 CTYST
06 I                                       56  6Ø ZIP
07 I                                        2  6Ø RECORD
08 I
09 I
```

These input specifications define the fields that make up the record to be loaded. The CUSTNO field from the input record is used as the relative record number to indicate the record to be loaded.

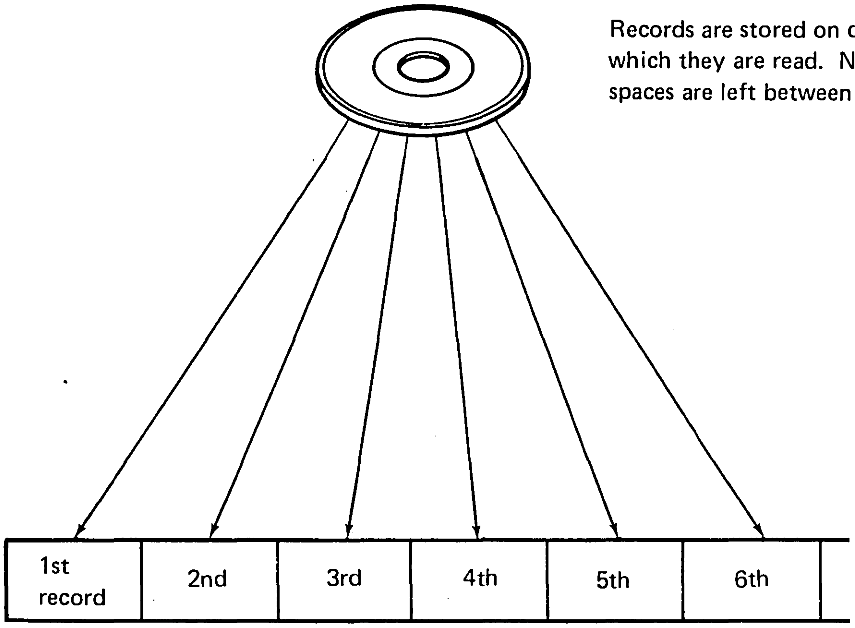
Output Specifications



```
01 OCUSTFILED        Ø1
02 O                                    1 'A'
03 O                         RECORD    6Ø
04 O
05 O
```

A record is loaded in the CUSTFILE for each record read from the RECIN file.

Figure 3-12. Using RECNO to Load a Direct File



Records are stored on disk in the same order in which they are read. No index is kept, and no spaces are left between disk records.

Figure 3-13. Sequential File Organization

Addrout Files

When describing an addrout file, you must place a T in column 32. The addrout file must be a disk file. See *Column 28 (Mode of Processing)* in this chapter for a description and example of addrout processing.

COLUMNS 33-34 (OVERFLOW INDICATOR)

Entry	Explanation
Blank	No overflow indicator is used.
OA-OG, OV	Specified overflow indicator conditions what records will be printed when overflow occurs.

Use columns 33 and 34 to specify an overflow indicator to condition which lines in each printer file will be printed when overflow occurs.

Only one overflow indicator can be assigned to a file. If more than one printer file in a program is assigned an overflow indicator, the indicator must be unique for each file.

Overflow occurs when a record is printed on the overflow line or when a space or skip instruction causes forms movement past the overflow line. When overflow occurs, the specified overflow indicator turns on and remains on for the rest of the program cycle. The indicator turns off after all lines conditioned by that indicator are printed. If no overflow indicator is specified and the fetch overflow routine is not used, the compiler automatically handles overflow. See *Column 16 (Fetch Overflow)* in Chapter 9, *Output Specifications*, for a description of the fetch overflow routine.

Automatic Overflow

If an overflow indicator is not specified, the compiler automatically advances the forms to the next page and continues printing once overflow is sensed. Printing begins on line 06 after the operator has positioned the first page, and overflow occurs at six lines less than the system page size unless the overflow line has been changed by line counter specifications. Detail lines, therefore, begin on line 06 for all pages except the first.

When overflow is handled automatically and the overflow line is sensed, the following steps occur:

1. All remaining detail lines in that program cycle are printed if a printer operation spaced or skipped to the overflow area.
2. All remaining total lines in that program cycle are printed.
3. A skip to line 06 occurs.

Overflow Indicator Specified

RPG II logic allows the overflow indicator to turn on at three different times in the program cycle: (1) at total time, (2) at detail time, and (3) at calculation time if exception output is used. There is only one time in the program cycle, however, when the program checks to determine whether the overflow indicator is on: right after all total records are printed.

When the overflow line is sensed, the overflow indicator turns on and the following steps occur:

1. Detail lines are printed if that part of the program cycle is not already completed.
2. Total lines are printed.
3. Total overflow lines are printed if conditioned by the overflow indicator.
4. Forms advance to the next page if indicated by the skip specifications on a heading line or total line.
5. Heading and detail lines are printed if conditioned by overflow indicators.

For more information on how to condition output operations with the overflow indicator, see *Columns 23-31 (Output Indicators)* in Chapter 9, *Output Specifications*.

COLUMNS 35-38 (KEY FIELD STARTING LOCATION)

Entry	Explanation
1-4096	Record position in which the key field begins

Use columns 35 through 38 to identify the record position in which the key field for an indexed file begins. Columns 35 through 38 apply only to indexed disk files, and an entry must be made in these columns for an indexed disk file. The key field of a record contains the information that identifies the record. This information is used in the index portion of the file. The key field must be in the same location in all of the records in the file. The entry in these columns must end in column 38. Leading zeros can be omitted.

Key fields cannot contain any hex FF characters. Therefore, if the key field is a binary field, you must be sure that no hex FF characters appear in the key field.

COLUMN 39 (EXTENSION CODE)

Entry	Explanation
Blank	No extension or line counter specifications are used.
E	Extension specifications further describe the file.
L	Line counter specifications further describe the file.

Use column 39 to indicate whether the file is further described on the extension specifications or on the line counter specifications. Column 39 applies only to (1) preexecution-time table and array files, (2) record address files, and (3) output files assigned to the printer. Describe printer output files on the line counter specifications, and describe table, array, and record address files on the extension specifications.

COLUMNS 40-46 (DEVICE)

Entry	Explanation
BSCA	Binary synchronous communications adapter
CONSOLE	Console data file or console record address file
CRT	Display screen
DISK	Disk
KEYBORD	Keyboard
PRINTER	132-position printer
SPECIAL	Used for a device not supported directly by RPG II
WORKSTN	Display station

Use columns 40 through 46 to identify the input/output device used for the file. All entries must begin in column 40. The devices and the associated file types that can be used with each device are shown in Figure 3-14. Figure 3-15 shows the columns that can be used for the devices named.

WORKSTN, CONSOLE, CRT, and KEYBORD all refer to the same physical device–a display station that consists of a display screen and a keyboard. CONSOLE, CRT, and KEYBORD devices can be used only with one display station. A CONSOLE file can be used only as an input file and cannot be used to display existing records in a file. A KEYBORD file can be used as an input/output file with the SET and KEY operation codes, which allow the output of prompts and the input of one field at a time. A CRT file can be used only as an output file to display information on the screen; this information cannot be changed by the operator. The WORKSTN file is an input/output (combined) file, which allows the programmer to specify multiple fields that can be output fields, input fields, or output/input fields. The WORKSTN file can be used with multiple display stations or SSP-ICF sessions.

BSCA

A BSCA device allows an RPG II program to transmit and receive binary synchronous data via a data communications network. For more information on the BSCA device, see Chapter 6, *Telecommunications Specifications*.

CONSOLE

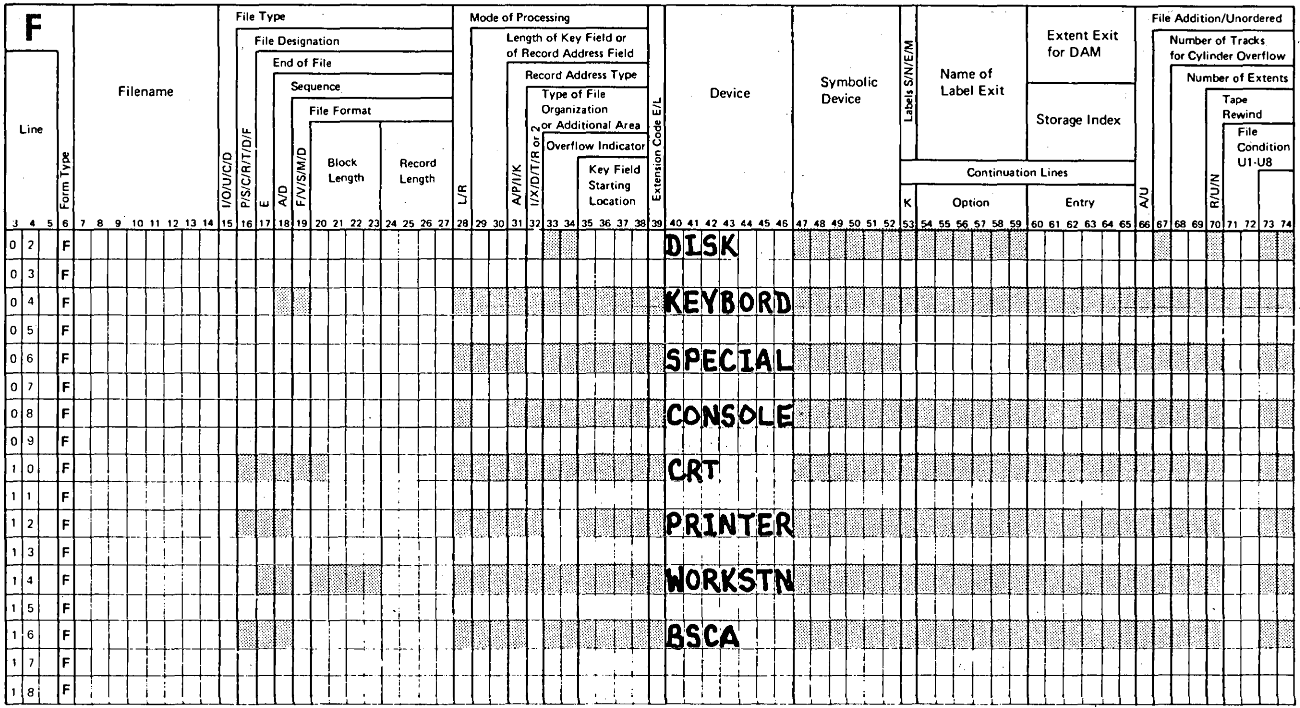
Use CONSOLE as the device name in one of two ways: (1) for a record address file, or (2) for an input data file. If CONSOLE is used for a record address file, the file must be further defined by extension specifications. Use CONSOLE when you want data records entered directly from a display station to an executing program. CONSOLE is an input file only and cannot be used to display existing records.

Device	Form of Data	File Type	Column 15	Column 16
DISK	Disk	Primary input	I	P
	Disk	Secondary input	I	S
	Disk	Record address file containing record key limits	I	R
	Disk	Record address file containing relative record numbers (addrout file)	I	R
	Disk	Chain input	I	C
	Disk	Demand	I	D
	Disk	Table or array (preexecution time only)	I	T
	Disk	Update (primary, secondary, chained, or demand)	U	P, S, C, or D
	Disk	Output	O	
	Disk	Direct file load (nondelete-capable file)[1]	O	C
WORKSTN	Keyed in by operator	Demand	C	D
	Keyed in by operator	Combined primary	C	P
CONSOLE	Keyed in by operator	Primary input	I	P
	Keyed in by operator	Secondary input	I	S
	Keyed in by operator	Demand	I	D
	Keyed in by operator	Record address files containing record key limits	I	R
KEYBORD	Keyed in by operator	Primary input	I	P
	Keyed in by operator	Demand	I	D
CRT	Displayed lines	Output	O	
PRINTER	Printed lines	Output	O	
BSCA	Data communications line	Primary input	I	P
	Data communications line	Secondary input	I	S
	Data communications line	Demand	I	D
	Data communications line	Output	O	
SPECIAL	Special device	Primary input	I	P
	Special device	Secondary input	I	S
	Special device	Demand	I	D
	Special device	Update (primary, secondary, or demand)	U	P, S, or D
	Special device	Combined (primary, secondary, or demand)	C	P, S, or D
	Special device	Output	O	

[1] For information on the specifications for the direct file load of a delete-capable file, see *Direct File Load (Delete-Capable File)* in this chapter.

Figure 3-14. Devices and Associated File Types

File Description Specifications



Line	Form Type	Device
0 2	F	DISK
0 3	F	
0 4	F	KEYBORD
0 5	F	
0 6	F	SPECIAL
0 7	F	
0 8	F	CONSOLE
0 9	F	
1 0	F	CRT
1 1	F	
1 2	F	PRINTER
1 3	F	
1 4	F	WORKSTN
1 5	F	
1 6	F	BSCA
1 7	F	
1 8	F	

F
Line
Form Type
Filename
File Type
File Designation
End of File
Sequence
File Format
Block Length
Record Length
I/O/U/C/D
P/S/C/R/T/D/F
E
A/D
F/V/S/M/D
L/R
Mode of Processing
Length of Key Field or of Record Address Field
Record Address Type
Type of File Organization or Additional Area
Overflow Indicator
Key Field Starting Location
A/P/I/K
I/X/D/T/R or 2
Extension Code E/L
Device
Symbolic Device
Labels S/N/E/M
Name of Label Exit
Extent Exit for DAM
Storage Index
Continuation Lines
K
Option
Entry
A/U
File Addition/Unordered
Number of Tracks for Cylinder Overflow
Number of Extents
Tape Rewind
R/U/N
File Condition U1-U8
3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74

Shaded columns must be blank.

Figure 3-15. Columns That Apply to Device Named

CRT

The CRT (display screen) can be used as an output device for normal and exception output. (See *Column 15* in Chapter 9, *Output Specifications*, for more information on exception output.) Any alphameric character can be written on the display screen. If the record length is less than or equal to 40, up to 40 characters can be written across the width of the screen, and a maximum of six such lines can appear on the screen at one time. The display is centered both vertically and horizontally on the screen. If the record length is greater than 40, up to 79 characters can be written across the width of the screen, and a maximum of 24 lines can appear on the screen at one time.

Data moves onto the screen from bottom to top: after the bottom line is written on the screen, if space-1-after is coded for the bottom line, the top line of a full screen moves off. Data is written on the display screen at the normal output times (total and detail) or at calculation time for exception output.

The display screen is designed to display messages and instructions to the operator and to display operator responses. It should not be used interchangeably with the printer as a major output device because of the speed with which data moves on and off the screen.

Output operations such as spacing and skipping can be specified with some restrictions. Spacing before and after (a 0 to 3 entry in columns 17 and 18 of the output specifications) and a skip-before to line 01 only (01 entry in columns 19 and 20 of the output specifications) can be specified. Specify a skip-before to 01 to erase data from the display screen. If a skip-before to any line other than 01 is specified, the system assumes the entry to be 01 and the screen is erased. A skip-after (columns 21 and 22 of the output specifications) cannot be specified for CRT files. When a line is written on the display screen over a previous line, the previous line is erased. Edit codes, edit words, and output indicators can be specified for CRT files.

DISK

DISK is an input/output device that allows the RPG II program to process data stored on disk. Disk files can be sequential, direct, or indexed files. See *Column 28 (Mode of Processing)* in this chapter for a description of the processing methods that can be used for disk files.

KEYBORD

The entries CONSOLE and KEYBORD refer to the same physical unit that includes both the keyboard and the display screen. Use KEYBORD when you use the KEY or SET operation codes. If KEYBORD is specified for the primary input file, you must provide a means of exit from the program by setting on the LR indicator.

Input specifications are not used for KEYBORD files. The input data is defined in the KEY or SET/KEY operation itself.

PRINTER

The print unit allows a separate output file to be printed on a 132-position printer. A maximum of eight printer files is allowed per program. PRINTER must be assigned as the device for each file, and each file must have a unique filename. Use the PRINTER OCL statement to assign a filename to a particular printer.

WORKSTN

The WORKSTN device allows an RPG II program to communicate with one or more display stations. A device can be specified as WORKSTN if you use the System/34 Display Screen Format Specifications to define its output and input and if it is allocated to the program. Only one WORKSTN file can be specified in a program. A program containing a WORKSTN file cannot contain KEYBORD, CONSOLE, or CRT files.

For a complete description of the file description specifications for a WORKSTN file, see Chapter 13, *WORKSTN File Considerations and Sample Programs.*

SPECIAL Device Support

Files using devices not directly supported by RPG II can be processed on System/34. To do this, enter SPECIAL in columns 40 through 46 of the file description specifications to indicate that the file is handled by a SPECIAL device.

You must also supply a subroutine to perform the input/output operations required to transfer data between the SPECIAL device and main storage. Enter the subroutine name in columns 54 through 59 of the file description specifications. Control cannot be transferred from one user assembler subroutine to another user assembler subroutine.

The SPECIAL device is also used with the IBM-written subroutine SUBR22 to read a transaction file created by the work station utility of the Utilities Program Product (see *Reading a Work Station Utility Transaction File* in this chapter).

Linkage for User-Written Input/Output Subroutines

The RPG II compiler generates the following DTF (define the file) for linking to an assembler input/output subroutine:

Bytes (hex)	**Description**
0	Device code (X'00')
1-2	Address of data management
3	Mask for external indicators
4-5	Backward chain pointer
6-7	Forward chain pointer
8-9	Logical record address
A	Completion code: X'42' = End of file X'41' = Controlled cancel X'40' = Normal completion
B	Operation code: X'80' = Get X'40' = Put X'20' = Update X'10' = Close
C-F	Attributes Byte 1: X'20' = Update file X'40' = Output file X'80' = Input file X'C0' = Combined file Byte 2: X'08' = Dual input/output X'01' = DTF open
10-11	Record length
12-19	Filename
1A-1B	Physical input address
1C-1D	Physical output address
1E-1F	Block length
20-21	Address of array DTT if array linkage is used

The address of byte 0 of the DTF is passed to the input/output subroutine in index register 2. Bytes 0 through 7 and C through 21 are filled in by RPG II at compile time. The contents of these fields depend on the entries specified in the file description specifications for the SPECIAL device. The input address (bytes 1A and 1B) and the output address (bytes 1C and 1D), when present, point to the physical buffer that has been allocated within the load module by RPG II for use by the SPECIAL device. The completion code (byte A) is inserted by the input/output subroutine when control is returned to RPG II. The operation code (byte B) and the logical record address (bytes 8 and 9) are inserted at object time.

Figure 3-16 shows an example of the DTF generated by RPG II for an input/output subroutine.

If array linkage is used, the RPG II compiler generates the following DTT (define the table):

Bytes	**Description**
0-1	Address of rightmost byte of the first element of the array
2-3	Address of rightmost byte of the last element of the array
4-5	RPG II last LOKUP element
6-7	Length of array element
8-13	Array name

```
            *******************************************************************
            *                                                                 *
            *            RPG II SPECIAL DTF OFFSETS AND EQUATES               *
            *                                                                 *
            *******************************************************************

            *       **** SPECIAL DTF LAYOUT ****
0000        SPDEV    EQU   0              DEVICE CODE (X'00')
0002        SPDMA    EQU   SPDEV+2        ADDRESS OF D.M.
0003        SPUPS    EQU   SPDMA+1        UPSI INDICATORS
0005        SPCHA    EQU   SPUPS+2        BACKWARD CHAIN POINTER
0007        SPCHB    EQU   SPCHA+2        FORWARD CHAIN POINTER
0009        SPLRA    EQU   SPCHB+2        LOGICAL RECORD ADDRESS
000A        SPCMP    EQU   SPLRA+1        COMPLETION CODE
000B        SPOPC    EQU   SPCMP+1        OPERATION CODE
000C        SPAT1    EQU   SPOPC+1        ATTRIBUTE BYTE ONE
000D        SPAT2    EQU   SPAT1+1        ATTRIBUTE BYTE TWO
000E        SPAT3    EQU   SPAT2+1        ATTRIBUTE BYTE THREE
000F        SPAT4    EQU   SPAT3+1        ATTRIBUTE BYTE FOUR
0011        SPRCL    EQU   SPAT4+2        RECORD LENGTH
0019        SPNAM    EQU   SPRCL+8        FILE NAME
001B        SPPBI    EQU   SPNAM+2        PHYSICAL INPUT I/O ADDRESS
001D        SPPBO    EQU   SPPBI+2        PHYSICAL OUTPUT I/O ADDRESS
001F        SPBKL    EQU   SPPBO+2        BLOCK LENGTH
0021        SPDTT    EQU   SPBKL+2        ADDR OF ARRAY DTT IF SPECIFIED
0022        SPLEN    EQU   SPDTT+1        LENGTH OF SPECIAL DTF

            *       **** SPCMP EQUATES ****
0040        SPNORM   EQU   X'40'          NORMAL
0042        SPEOF    EQU   X'42'          END OF FILE
0041        SPCCNL   EQU   X'41'          CONTROLLED CANCEL

            *       **** SPAT1 EQUATES ****
0020        SPUPDT   EQU   X'20'          UPDATE FILE
0040        SPOUT    EQU   X'40'          OUTPUT FILE
0080        SPINP    EQU   X'80'          INPUT FILE
00C0        SPCMB    EQU   X'C0'          COMBINED

            *       **** SPAT2 EQUATES ****
0008        SPDIO    EQU   X'08'          DUAL I/O
0001        SPOPEN   EQU   X'01'          DTF OPEN

            *       **** SPOPC EQUATES ****
0040        SPPUT    EQU   X'40'          PUT
0080        SPGET    EQU   X'80'          GET
0020        SPUPD    EQU   X'20'          UPDATE
0010        SPCLS    EQU   X'10'          CLOSE
            *** END OF EXPANSION **
```

Figure 3-16. Example of RPG II Generated DTF

Considerations for the Assembler Programmer

The input/output subroutine must save and restore the registers altered in the subroutine. Control should be returned to the address in the address recall register (ARR).

When an input operation is done, the input/output subroutine must move the address of the physical buffer currently being used to the logical record address location in the DTF (bytes 8 and 9). This logical record address points to the record within the physical buffer that is to be processed by the RPG II program.

When an output operation is requested, the input/output subroutine must move the data from the logical buffer (address in bytes 8 and 9 of the DTF) to the physical buffer (address in bytes 1C and 1D of the DTF). The logical record address (bytes 8 and 9) points to the RPG II common output buffer that contains the record to be output by the SPECIAL device.

The assembler programmer must also consider the following:

- The input/output subroutine must do its own open when the first call to it is issued. It must also do its own close to the file on a close call.
- If a dual input/output area is requested, the second area will be immediately behind the first.
- Subroutines of the type SUBRxx cannot be overlaid; however, subroutines of the type SRyzzz can be overlaid.
- Only consecutive processing is supported for SPECIAL files.

File Description Specifications for SPECIAL Device

The following file description specifications apply to files assigned to the SPECIAL device:

Column	**Entry**
7-14	Valid RPG II filename.
15	I, O, U, or C.
16	P, S, D, or blank.
17	E or blank.
18	A, D, or blank.
19	F.
20-23	Block length.
24-27	Record length.
28-31	Must be blank.
32	1 through 9 or blank.
33-39	Must be blank.
40-46	SPECIAL.
47-53	Must be blank.
54-59	Name of the user-written or IBM-written subroutine that performs the input/output operations. The subroutine name must be in the form SUBRxx, where x is any alphabetic character or in the form SRyzzz, where y is any of the following 15 characters: B, C, D, F, G, H, I, L, M, O, P, R, S, T, or U; and z is any of the following 16 characters: A, B, C, D, F, G, H, I, L, M, O, P, R, S, T, or U.
60-70	Must be blank.
71-72	U1 through U8 or blank.
73-74	Must be blank.

The following can be used with SPECIAL files:

- FORCE operation code
- READ operation code
- File translation

The following cannot be used with SPECIAL files:

- CHAIN operation code
- Spacing and skipping
- *PRINT

SPECIAL files can only be processed consecutively. (See Figure 3-31 for possible file description entries for SPECIAL files.)

Reading and Updating a Work Station Utility Transaction File (SUBR22)

The IBM-written subroutine SUBR22 allows an RPG II program to read and update records from a transaction file created by the work station utility of the System/34 Utilities Program Product. To link to this subroutine, use the SPECIAL device and an array with one 13-character element.

For an example of an RPG II program that reads records from a work station utility transaction file, see Figure 3-17.

Notes:

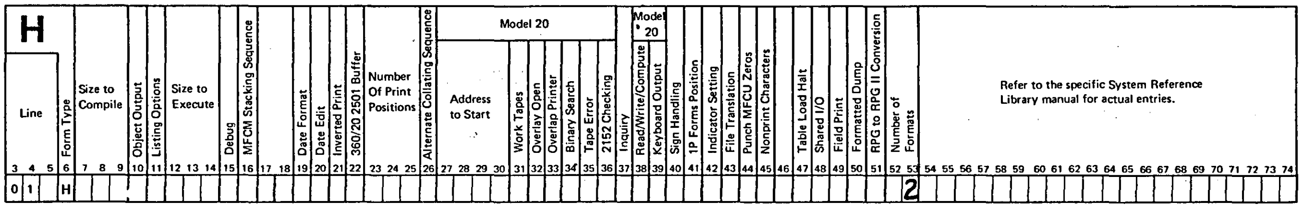
1. The user program must initialize positions 1 through 11 of the array before the first data record is read by SUBR22 or before SUBR22 begins to read a new logical chain.
2. The last 13 bytes of the work station utility transaction file record (the trailer information) are not returned to the RPG program and cannot be updated.
3. The user program should check position 13 of the array (the error indicator) after each data record is read to determine if any errors were encountered.

File Description Entries

To use SUBR22, the following entries must be made on the file description specifications:

Column	Entry
6	F.
7-14	Name of the transaction file created by the work station utility.
15	I, U.
16	P, S, or D.
17	Blank or E.
18	Blank (assumed blank if an entry is present).
19	F.
20-23	Enter the block length, which is calculated by the following formula: • Block length = 256 if the record length is a submultiple of 256 • Block length = the record length if the record length is a multiple of 256 • Otherwise, the block length must equal the record length plus 255 rounded up to the next multiple of 256
24-27	Valid entries for record length are 14 through 4096. *Note:* The record length specified must include 13 bytes for the work station utility file trailer information, of which bytes 11 and 12 contain the work station ID.
28-39	Must be blank.
40-46	SPECIAL.
47-53	Must be blank.
54-59	SUBR22.
60-70	Must be blank.
71-72	Blank or U1 through U8.
73-74	Must be blank.
75-80	Program identification.

Control Specifications

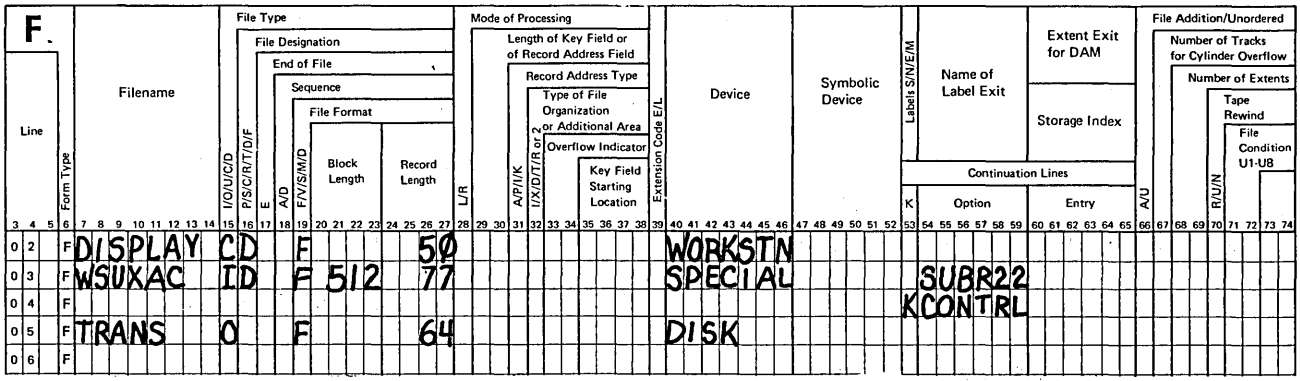


H

Refer to the specific System Reference Library manual for actual entries.

Line	Form Type	Size to Compile	Object Output	Listing Options	Size to Execute	Debug	MFCM Stacking Sequence	Date Format	Date Edit	Inverted Print	360/20 2501 Buffer	Number Of Print Positions	Alternate Collating Sequence	Model 20: Address to Start	Work Tapes	Overlay Open	Overlay Printer	Binary Search	Tape Error	2152 Checking	Inquiry	Model 20: Read/Write/Compute	Keyboard Output	Sign Handling	1P Forms Position	Indicator Setting	File Translation	Punch MFCU Zeros	Nonprint Characters	Table Load Halt	Shared I/O	Field Print	Formatted Dump	RPG to RPG II Conversion	Number of Formats
01	H																																		2

File Description Specifications

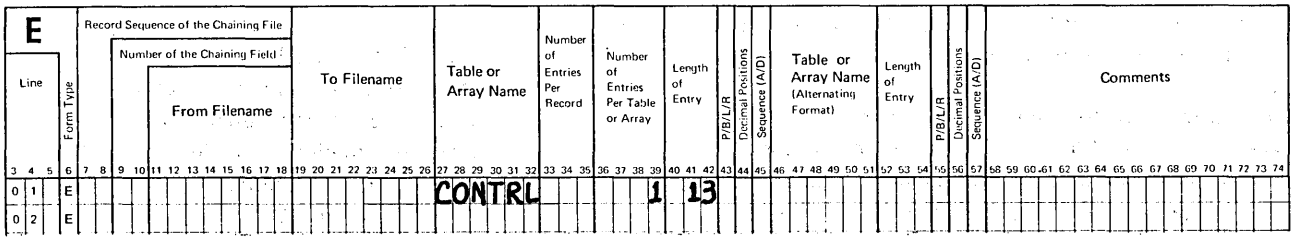


F.

Line	Form Type	Filename	I/O/U/C/D	P/S/C/R/T/D/F	E	A/D	F/V/S/M/D	Block Length	Record Length	L/R	Length of Key Field or of Record Address Field	Record Address Type	Type of File Organization or Additional Area	Overflow Indicator	Key Field Starting Location	Extension Code E/L	Device	Symbolic Device	Labels S/N/E/M	Name of Label Exit	Extent Exit for DAM	Continuation Lines: K	Option	Entry	A/U	Number of Tracks for Cylinder Overflow	Number of Extents	Tape Rewind R/U/N	File Condition U1-U8
02	F	DISPLAY	C	D			F		5Ø								WORKSTN												
03	F	WSUXAC	I	D			F	512	77								SPECIAL			SUBR22									
04	F																					K	CONTRL						
05	F	TRANS	O				F		64								DISK												
06	F																												

This program uses SUBR22 to selectively read chains of records from the work station transaction file (WSUXAC) and to write them to a permanent disk file (TRANS). The operator enters the ID of the work station whose transactions he wishes to copy. The program displays error messages when a file or chain containing an error is encountered, or when the work station ID entered cannot be found. The program also displays a message after all the records in a chain have been copied. The program ends when a file containing an error is found, or when Cmd key 7 is pressed.

Extension Specifications



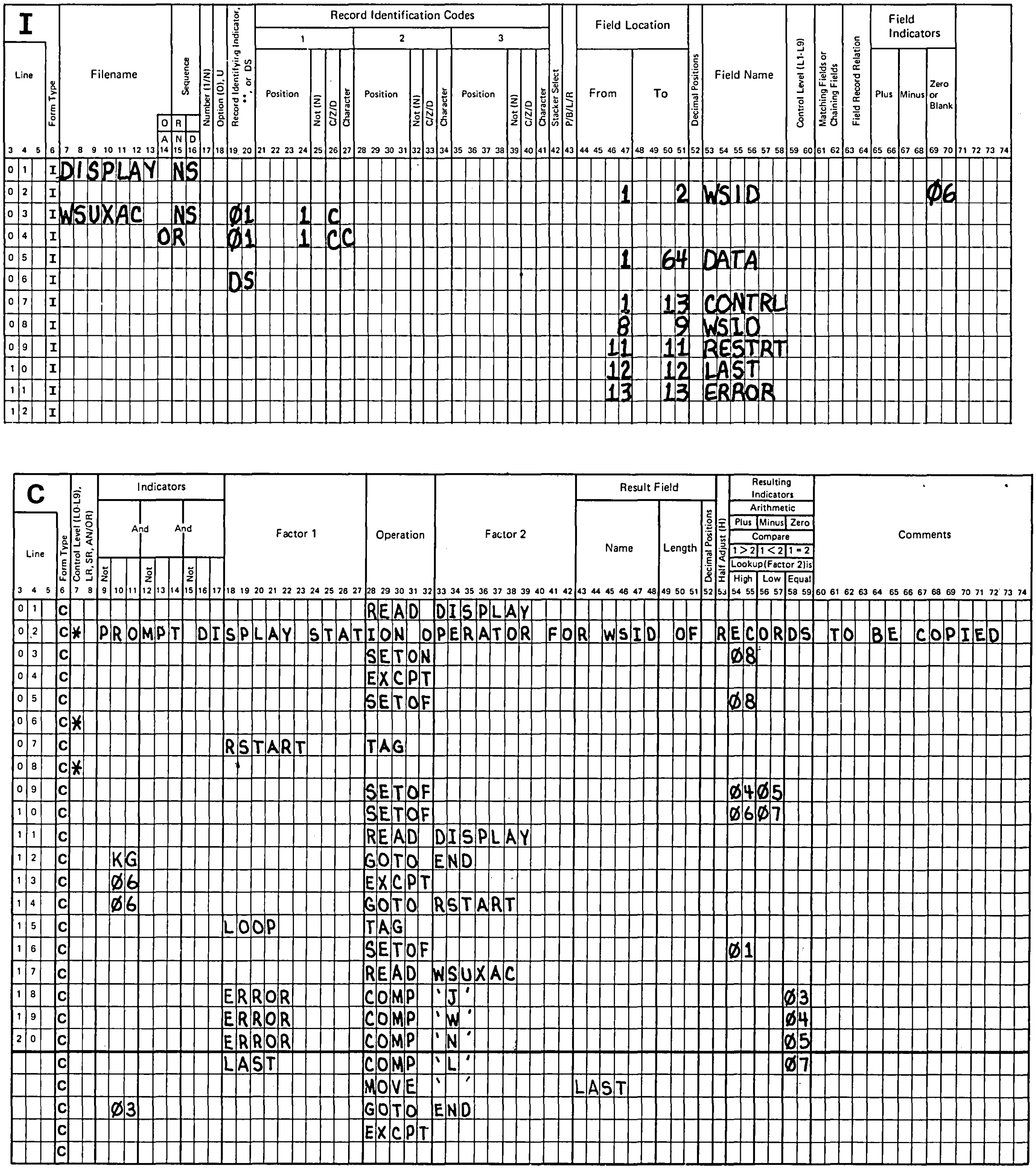
E

Line	Form Type	Record Sequence of the Chaining File	Number of the Chaining Field	From Filename	To Filename	Table or Array Name	Number of Entries Per Record	Number of Entries Per Table or Array	Length of Entry	P/B/L/R	Decimal Positions	Sequence (A/D)	Table or Array Name (Alternating Format)	Length of Entry	P/B/L/R	Decimal Positions	Sequence (A/D)	Comments
01	E					CONTRL		1	13									
02	E																	

SUBR22 requires an array (in this program, an execution-time array is used) to contain control information to be passed to and from SUBR22 and the program with each data record. This array (CONTRL) contains the work station ID, the error indications, and the last record flag.

Figure 3-17 (Part 1 of 4). Reading a Work Station Utility Transaction File

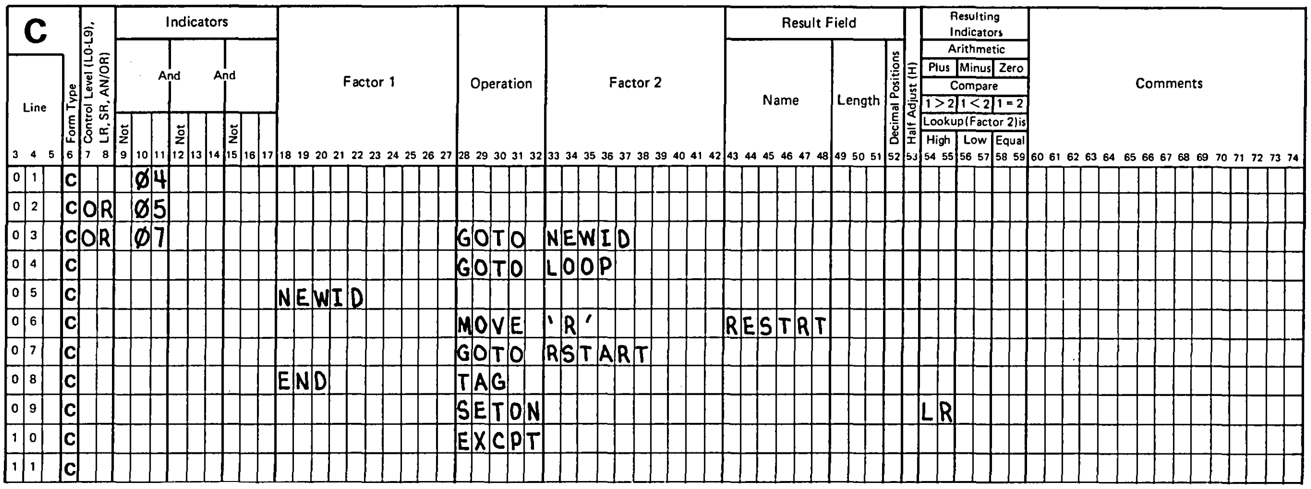
RPG Input Specifications



Line	Form Type	Filename	OR/AND	Sequence	Record Identifying Indicator or DS	Position	C/Z/D	Character	From	To	Field Name	Zero or Blank
01	I	DISPLAY		NS								
02	I								1	2	WSID	Ø6
03	I	WSUXAC		NS	Ø1	1	C					
04	I		OR		Ø1	1	C	C				
05	I								1	64	DATA	
06	I				DS							
07	I								1	13	CONTRL	
08	I								8	9	WSID	
09	I								11	11	RESTRT	
10	I								12	12	LAST	
11	I								13	13	ERROR	
12	I											

Line	Form Type	Control Level	Indicators	Factor 1	Operation	Factor 2	Result Field Name	High	Low	Equal
01	C				READ	DISPLAY				
02	C	*	PROMPT DISPLAY STATION OPERATOR FOR WSID OF RECORDS TO BE COPIED							
03	C				SETON			Ø8		
04	C				EXCPT					
05	C				SETOF			Ø8		
06	C	*								
07	C			RSTART	TAG					
08	C	*								
09	C				SETOF			Ø4	Ø5	
10	C				SETOF			Ø6	Ø7	
11	C				READ	DISPLAY				
12	C		KG		GOTO	END				
13	C		Ø6		EXCPT					
14	C		Ø6		GOTO	RSTART				
15	C			LOOP	TAG					
16	C				SETOF			Ø1		
17	C				READ	WSUXAC				
18	C			ERROR	COMP	'J'				Ø3
19	C			ERROR	COMP	'W'				Ø4
20	C			ERROR	COMP	'N'				Ø5
	C			LAST	COMP	'L'				Ø7
	C				MOVE	' '	LAST			
	C		Ø3		GOTO	END				
	C				EXCPT					
	C									

Figure 3-17 (Part 2 of 4). Reading a Work Station Utility Transaction File



Line	Form Type	Control Level (L0-L9, LR, SR, AN/OR)	Indicators	And	And	Factor 1	Operation	Factor 2	Result Field Name	Length	Decimal Positions	Half Adjust (H)	Resulting Indicators Plus / High / 1>2	Minus / Low / 1<2	Zero / Equal / 1=2	Comments
01	C		Ø4													
02	C	OR	Ø5													
03	C	OR	Ø7				GOTO	NEWID								
04	C						GOTO	LOOP								
05	C					NEWID										
06	C						MOVE	'R'	RESTRT							
07	C						GOTO	RSTART								
08	C					END	TAG									
09	C						SETON						LR			
10	C						EXCPT									
11	C															

The first READ operation reads the first record, which is blank, from the display. Indicator 08 is set on to display the initial ENTER WS ID prompt (displayed by the EXCPT operation) and is then set off. The program also sets off indicators 04, 05, 06, and 07.

The second READ operation is executed when:

- The Enter/Rec Adv key is pressed (WSID is blank). Indicator 06 is turned on because the input field is blank. The EXCPT operation is executed, displaying the ENTER WS ID prompt and the error message WSID BLANK. The program then goes to RSTART, sets off indicators 04, 05, 06, and 07, and waits for the READ operation to be executed.

- Two characters are entered in the WSID field.

  The program attempts to read a record from the transaction file (WSUXAC). The SUBR22 error flag (ERROR) is checked for the following:
  - If the WSU transaction file contains an error, indicator 03 is turned on.
  - If the record returned to the program is from a display station session that ended abnormally, indicator 04 is turned on.
  - If no records exist for the work station ID that was entered, indicator 05 is turned on.

  After these comparisons are executed, the last record flag (LAST) is checked to determine whether this record is the last record in the logical chain. If it is the last record, indicator 07 is turned on.

Depending on which indicators have been turned on by the previous comparisions, the EXCPT operation causes one of the following output combinations to be performed:
- The ENTER WS ID prompt and the error message WS ID xx BAD CHAIN are displayed if indicator 04 is on.
- The ENTER WS ID prompt and the error message WS ID xx NOT FOUND are displayed if indicator 05 is on.
- The record is written to the permanent disk file (TRANS), and the ENTER WS ID prompt and the message WS ID XX CHAIN COPIED are displayed if indicator 07 is on.
- The record is written to the permanent disk file (TRANS) if indicator 01 is on.

If indicator 04, 05 or 07 is on, the program goes to NEWID after performing the output. Otherwise, the program goes to LOOP and continues reading records from the WSU transaction file until an error is encountered or the last record in the chain is reached.

The program ends in one of two ways. Either Cmd key 7 is pressed, turning on indicator KG, or a bad file is found, turning on indicator 03. Either condition causes the program to branch to END, turn on the LR indicator, and display the END OF JOB screen. If indicator 03 is on, the error message BAD FILE is also displayed.

Figure 3-17 (Part 3 of 4). Reading a Work Station Utility Transaction File

RPG Output Specifications

Commas	Zero Balances to Print	No Sign	CR	–
Yes	Yes	1	A	J
Yes	No	2	B	K
No	Yes	3	C	L
No	No	4	D	M

X = Remove Plus Sign
Y = Date Field Edit
Z = Zero Suppress

Line	Form Type	Filename	Type (H/D/T/E) / AND-OR	Space Before	Space After	Skip Before	Skip After	Output Indicators	And	And	Field Name	Edit Codes	B/A/C/1-9/R	End Position in Output Record	P/B/L/R	Constant or Edit Word
01	O	DISPLAY	E					Ø8								
02	O		OR					Ø4	NLR							
03	O		OR					Ø5	NLR							
04	O		OR					Ø6	NLR							
05	O		OR					Ø7	NLR							
06	O													K6		'PROMPT'
07	O							NØ6						6		'WS ID '
08	O							NØ6			WSID			8		
09	O							Ø7	NØ4					21		'CHAIN COPIED'
10	O							Ø5						18		'NOT FOUND'
11	O							Ø4						18		'BAD CHAIN'
12	O							Ø6	NØ8					1Ø		'WSID BLANK'
13	O		E					LR								
14	O													K3		'END'
15	O							Ø3						8		'BAD FILE'
16	O	TRANS	E					Ø1	NØ3	NØ4						
17	O		AND					NØ5	NØ6	NLR						
18	O										DATA			64		
19	O															
20	O															

Figure 3-17 (Part 4 of 4). Reading a Work Station Utility Transaction File

The following entries must also be made on the file description continuation line:

Column	**Entry**
7	F.
5-52	Must be blank.
53	K.
54-59	Name of an array with one 13-character element. This array must also be described on the extension specifications.
60-74	Must be blank.
75-80	Program identification.

Contents of the Array

The array named in columns 54 through 59 of the continuation line is used to pass parameters from the RPG II program to SUBR22 and from SUBR22 to the RPG II program. The entries that can be made in the array are described in the following text.

Positions 1-7 (Starting Record Number)

Entry: Any valid zoned decimal relative record number or blank

To read part of a logical chain in the transaction file, specify the relative record number of the first record to be read. Records are read from the file until the end of the logical chain is reached or until a restart parameter is specified. The starting record number is blanked by SUBR22 after it is used to process records.

Positions 8-9 (Work Station ID)

Entry: 2-character work station ID or blank

To read one logical chain from the transaction file, specify the work station ID of the display station whose logical chain is to be read. Records are read from the file until the end of the logical chain is reached or until a restart parameter is specified. The work station ID is blanked by SUBR22 after it is used to process records.

Note: When the file is read, this same value is in positions 11 and 12 of the trailer portion of the transaction file record.

Position 10 (Type)

Entry: A or blank

To read all the logical chains in the transaction file, enter an A in position 10. All logical chains in the file are read in the sequence in which they are chained (that is, all data records are read for the first display station in the chain, then for the second display station in the chain, and so on). The type entry is blanked by SUBR22 after it is used to process records.

Position 11 (Restart/Active)

Entry: A, R, or blank

The restart parameter allows the RPG II program to read/update more than one logical chain and to read/update active logical chains or chains from sessions that ended abnormally. When an R is specified in position 11, the program starts processing records from the file as specified by the parameter list. However, an R should not be moved into position 11 before the first read is executed for the program. When an A is specified in position 11, the program also starts processing records from the file as specified by the parameter list. An entry of A allows the program to process records from active work sessions and work sessions that terminated abnormally. If position 11 is blank, the next record in the logical chain is read.

Position 12 (Last Record Flag)

Entry: E, L, or blank

An E is returned to the RPG II program in position 12 of the array when an A was specified in position 11 of the array and one of the following conditions occurs:

- The end of the logical chain is reached when a starting record number was specified in positions 1 through 7.
- The end of the logical chain is reached when a work station ID was specified in positions 8 and 9.
- The last record of the transaction is reached when an A was specified in position 10.

Note: A blank record is returned to the program whenever an E is returned to the array.

An L is returned to the RPG II program in position 12 of the array when one of the following conditions occurs:

- The current record is the last record entered for a work session that ended normally when a starting record number is specified in positions 1 through 7.
- The current record is the last record entered for a work session that ended normally when a work station ID is specified in positions 8 and 9.
- The current record is the last record entered for a work session that ended normally, the work session is the last work session in the transaction file, and an A is specified in position 10.

When an E or L is returned to the RPG program, the programmer can specify the restart option (R in position 11). If the E or L parameter is ignored by the program, the next read request to the subroutine causes a normal end-of-file indication to be returned to the RPG II program. After the end-of-file indication is returned to the RPG II program, additional attempts to access the file also result in end-of-file.

SUBR22 blanks out the last record flag if a restart is specified.

Position 13 (Error Flag)

Entry:

Blank

No errors found. The subroutine has returned a good data record to the program.

W

This display station session ended abnormally. The last sequence set may be incomplete, or some inserted records may have been lost. A data record is returned to the program with this error flag. SUBR22 does not read any records that have been added during the work session that ended abnormally.

J

This file has not closed normally and contains display station sessions that may be incomplete. However, the data being processed by the program comes from a completed display station session. A data record is returned to the program with this error flag.

N

No-record-found error occurred for one of the following reasons:

- The work station ID specified in the parameter list is invalid.
- The parameter list does not contain a starting record number, type, or work station ID parameter for a first-time or restart option.
- The relative record number specified is invalid.

A blank record is returned with this error flag. If a no-record-found error occurs, the restart option must be specified the next time the subroutine is called, and the programmer must reset the no-record-found indicator.

P

An attempt was made to update a record, but no record was retrieved for the update.

D

An attempt was made to process a delete-capable file. The work station utility does not allow delete-capable files. Processing of this file is canceled and the file is closed.

Note: When the subroutine is accessed the first time or with a restart parameter specified, one of the three first-time options (starting record number, work station ID, or type) must be specified in the parameter list. If a valid option is not specified, the subroutine returns to the program without reading a record. If the subroutine is accessed twice consecutively with a valid option specified, a normal end-of-file indication is returned to the RPG II program. If more than one valid option is specified, the subroutine only processes the first valid option found. The subroutine checks for the first valid option in the following order: starting record number, type, and work station ID.

System Input Subroutine for SPECIAL Device (SUBR01)

The IBM-written subroutine SUBR01 (system input subroutine) can be used to input a record for use by system functions, such as system utility programs or program products. To use this subroutine, specify SUBR01 in columns 54 through 59 for a SPECIAL device. This subroutine makes 120-character records available to the SPECIAL file. RPG treats the records read by SUBR01 as data records.

The record length and block length entries for the SPECIAL file should be 120 to avoid execution errors. If the block length is not specified, it defaults to the record length. If the operation control language (OCL) statements to execute the program are entered from the keyboard, the records to be made available to the file must also be entered from the keyboard. If a procedure was called to execute the program, the records to be available to the file must follow the RUN OCL statement in the procedure. The last input record in the procedure should be followed by a END control statement. If the program is an SRT program and if there is no END statement in the procedure, successive OCL statements (entered from the keyboard) are read as input to the SPECIAL file in the program. If the program is an MRT program, a procedure must be used for the data records and the END control statement must follow the last data record.

If a program is to be run from the input job queue, a procedure must be used to execute the program. If CONSOLE is specified in the same program as SUBR01, the OCL and the data records for SUBR01 must be contained in a procedure or undesirable results can occur.

See the *System Support Reference Manual* for information on how to create a procedure.

1255 MICR Subroutine (SUBR08 and SUBR25)

The 1255 MICR subroutines (SUBR08 and SUBR25) allow you to use the 1255 as a SPECIAL device in an RPG II program. To use SUBR08 or SUBR25, you must place certain entries on the file description and extension specifications in order to define the SPECIAL device file and to name the subroutine. SUBR08 uses system and stacker specifications to describe each job to be performed by the 1255. SUBR25 uses a SUBR25 parameter list and a Device Control Language program to describe each job to be performed by the 1255. The system and stacker entries or the SUBR25 parameter list is placed in an array. See the *IBM System/34 1255 Magnetic Character Reader Reference Manual*, SC21-7740, for more information and examples of how to code the file description and extension specifications.

COLUMNS 47-52

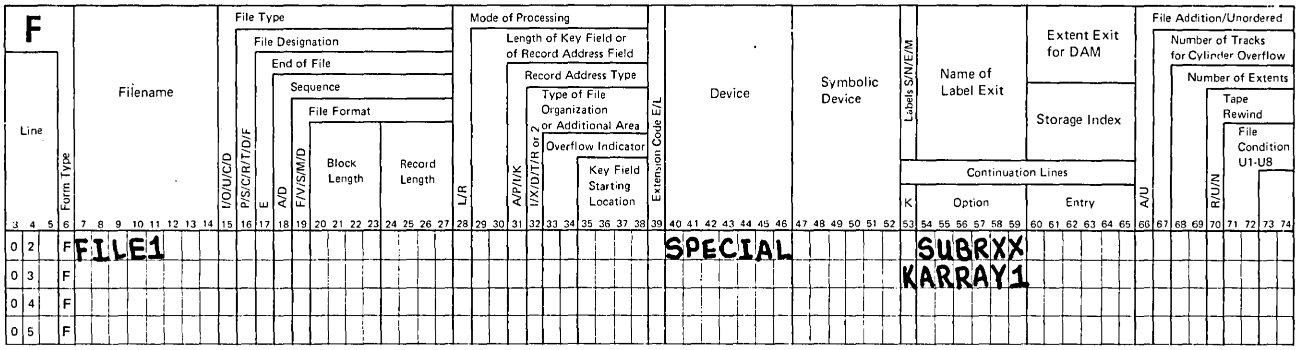
Columns 47 through 52 are not used. Leave them blank.

COLUMN 53 (CONTINUATION LINES–K)

Entry	Explanation
K	Continuation record

Use column 53 to indicate that a continuation record provides additional information about the SPECIAL file, disk file, or WORKSTN file being defined. Only one continuation record can be specified for each SPECIAL file or each disk file; however, several continuation records can be specified for a WORKSTN file. When you specify a continuation record for a SPECIAL device, columns 54 through 59 (continuation line option) must be coded. When you specify a continuation record for a disk or WORKSTN device, columns 54 through 65 must be coded. Figure 3-18 shows an example of the coding necessary on the file description specifications sheet for a continuation line for a SPECIAL file.

File Description Specifications



Line	Form Type	Filename	Device	K	Name of Label Exit / Option
02	F	FILE1	SPECIAL		SUBRXX
03	F			K	ARRAY1
04	F				
05	F				

F
File Type
File Designation
End of File
Sequence
File Format
Filename
Line
Form Type
I/O/U/C/D
P/S/C/R/T/D/F
E
A/D
F/V/S/M/D
Block Length
Record Length
L/R
Mode of Processing
Length of Key Field or of Record Address Field
Record Address Type
Type of File Organization or Additional Area
Overflow Indicator
A/P/I/K
I/X/D/T/R or 2
Key Field Starting Location
Extension Code E/L
Device
Symbolic Device
Labels S/N/E/M
Name of Label Exit
Continuation Lines
K
Option
Entry
Extent Exit for DAM
Storage Index
A/U
File Addition/Unordered
Number of Tracks for Cylinder Overflow
Number of Extents
Tape Rewind
R/U/N
File Condition U1-U8

Figure 3-18 Specifications for a SPECIAL Device

COLUMNS 54-59

Name of Label Exit

Entry	Explanation
Blank	No SPECIAL device is used.
SUBRxx	Name of the user-written or IBM-written subroutine that cannot be overlaid and that performs the input/output operation for a SPECIAL device. (For a user-written subroutine, x = any alphabetic character. Numeric characters are reserved for IBM-supplied subroutines.)
SRyzzz	Name of the IBM-written subroutine (5-character name in library is @yzzz) that performs the input/output operation for a SPECIAL device (y = any of the following 15 characters: B, C, D, F, G, H, I, L, M, O, P, R, S, T, or U; z = any of the following 16 characters: A, B, C, D, F, G, H, I, L, M, O, P, R, S, T, or U).

Note: Subroutines of the type SRyzzz can be overlaid. Modifications within the subroutine code may or may not be present the next time the subroutine is used.

Use columns 54 through 59 to specify the subroutine that performs the input/output operations for a file assigned to a SPECIAL device. Columns 54 through 59 must contain an entry for each file assigned to a SPECIAL device. The subroutine name entered in columns 54 through 59 can be from 4 to 6 characters long. For a user-written subroutine the first 4 characters must be SUBR; the remaining characters can be any alphabetic character.

Note: User-written subroutines should be placed in #LIBRARY, not in the same library as the RPG II source program.

Continuation Line Option for SPECIAL Device

Entry	Explanation
Table/array name	Name of table/array used by the user-written subroutine. The array name cannot be ASCII or BUFOFF.

Continuation Line Options for WORKSTN File

The following options can be specified for a WORKSTN file if more than one device is attached to a program or if you want to specify the WORKSTN file information data structure (INFDS) or the WORKSTN exception/error processing subroutine (INFSR). The NUM keyword is required if the program attaches more than one device to a file at the same time. Enter the keyword in columns 54 through 59 and the value in columns 60 through 65 (columns 60 through 67 can be used if the FMTS option is specified).

Note: For WORKSTN files, a device can be either a display station or an SSP-ICF session.

Keyword	Value
NUM	Maximum number of devices that can be attached to this file at one time. The number specified must be right-justified in columns 60 through 65. If not specified, 1 is assumed. If specified, NUM must be greater than or equal to the number of requestors specified by the MRTMAX parameter on the COMPILE statement plus the number of acquired devices (those specified on the WORKSTN OCL statement or in the ACQ operation). The number specified on the MRTMAX parameter is reserved for requestors. The difference between the MRTMAX value and the NUM value is the maximum number of devices (nonrequestors) that can be attached to the program at one time via OCL statements or the ACQ operation code. For example, if the MRTMAX value is 5 and the NUM value is 6, only one nonrequestor device can be attached to the program, even if only one requestor is presently signed on.
SAVDS	Name of a data structure that is to be saved and restored for each device attached to this file. This data structure cannot be a display station local data area, and it cannot contain a compile-time array or a preexecution-time array. If SAVDS is not specified or NUM equals 1, no data area swapping is done.
IND	Number of indicators, beginning with 01, that are to be saved and restored by display station. Indicators 01 through the number specified here are saved and restored for each display station. If IND is not specified, or if NUM equals 1, no indicator swapping is done. The entry must be right-justified in columns 60 through 65. *Note:* For SAVDS and IND, only one copy of the data structure and indicators is available at a time. The indicators and data structure that are available are those associated with the device from which the last input was read. The data structure and indicators that are available change each time an input operation (either a primary file input or a demand file read) is executed. On an input operation, the present copy of the data structure and indicators in the program is written to a save area for the device from which the previous input was read. The data structure and indicators for the device now being read from are then written from the save area associated with the device to the program SAVDS and IND areas (for further information, see Figure 13-2 in Chapter 13).
SLN	Name of a two-digit numeric field whose value determines the first line on the display screen where the display format is to begin if a variable starting line number (V in column 17 of the display screen format S specification) was specified in the format. If SLN is not specified, all formats having a variable starting line number begin on line 1.

Keyword Value

FMTS *NONE. Indicates that there are only SSP-ICF formats present in this program.

Display screen format load member name. The compiler uses the name specified here as the display screen format load member name. The format load member name entered here can be from 1 to 8 characters in length, left-justified in columns 60 through 67. If a name is not entered, the compiler assumes the display screen format load member name is the program name (from columns 75 through 80 of the control specifications) with FM added to the end of the name.

ID Name of a 2-character self-defining alphameric field that contains the ID of the device that supplied the record being processed in this file. The ID field is updated whenever a record is read from the WORKSTN file. Therefore, it always contains the identification of the device from which the last record was read (unless your program moves a different identification into the ID field). This field is considered self-defining because it need not be specified as an input or result field. For a multiple device file, you can direct an operation to a device other than the one currently being processed by changing the value in the ID field to the symbolic ID of another device in the file before performing the output operation.

The device IDs are assigned at system configuration time. Display station IDs are in the form AX, where A is an alphabetic character (A-Z, #, @, or $) and X is any character. If a WORKSTN OCL statement exists for the display station, the ID is the same as the value of the SYMID parameter.

SSP-ICF session IDs can be in two formats. They are either NN where N is numeric (0-9), or NA where N is numeric and A is alphabetic (A-Z, #, @, or $). If the format is NA, a SESSION OCL statement must be specified with a SYMID parameter whose value is also in an NA format.

Keyword Value

INFSR Name of the user-written calculation subroutine designated as the WORKSTN exception/error processing subroutine to which control may be passed if an exception/error occurs during the following operations: ACQ, REL, NEXT, POST, input (READ or primary input), or output (EXCPT operation or normal cycle output). If INFSR is not specified, the program halts if an exception/error occurs. See *WORKSTN Exception/Error Handling* in Chapter 13 for more information on INFSR.

INFDS Name of the data structure that contains the identification of the type of exception/error condition and an indication of the WORKSTN operation that was executing when the exception/error condition occurred. It also specifies the area that is posted with the display station screen size (960 or 1920 characters) and information about ideographic support. If INFDS is not specified, this information is not available to the RPG II program. See *WORKSTN Exception/Error Handling* in Chapter 13 for more information on INFDS.

For more information on the keywords and their associated values, see Chapter 13, *WORKSTN File Considerations and Sample Programs.*

Continuation Line Keyword Option

The RECNO keyword is used to randomly add records to, or to load, a delete-capable disk file.

Keyword	Value
RECNO	Name of a numeric field that is seven digits long with zero decimal positions. This field name must be specified if records are to be added randomly to a delete-capable direct or sequential file. This field name is also required for a direct file load of a delete-capable file. The programmer places the relative record number of the record to be added to the file in the RECNO field. It must be the relative record number of a deleted record, a record that has been initialized to hex FFs. RPG uses the relative record number in the RECNO field to determine where a record is to be loaded (direct file load) or added (ADD on output specifications). *Note:* If a CHAIN operation successfully reads a record from a sequential or direct file, RPG places the relative record number of this record in the RECNO field.

COLUMNS 60-65 (STORAGE INDEX)

Entry	Explanation
Blank	No storage index is kept in storage.
6-9999	Number of bytes reserved for the storage index.

Use columns 60 through 65 to specify the number of bytes to be reserved for the storage index. Columns 60 through 65 apply to indexed files processed randomly via the CHAIN operation code and indexed files processed sequentially within limits. Specifying a storage index provides faster retrieval of records. Storage index cannot be specified with a shared input/output area. Entries must end in column 65. Leading zeros are not required.

The storage index is a table containing entries for tracks in the index portion of a file. Each entry contains a sector address and the lowest key field from the next track in the file index. The index portion of a file contains the position of the records in the file. Each index entry consists of a key and a 3-byte disk address for each record in the file. The last entry of the file index contains all FFs to indicate the end of the index. Figure 3-19 shows the layout of the index for the indexed file, INDEXT, and the most efficient storage index for the file, INDEXT. Notice that the storage index contains one entry for each track in the file index if the space reserved is large enough.

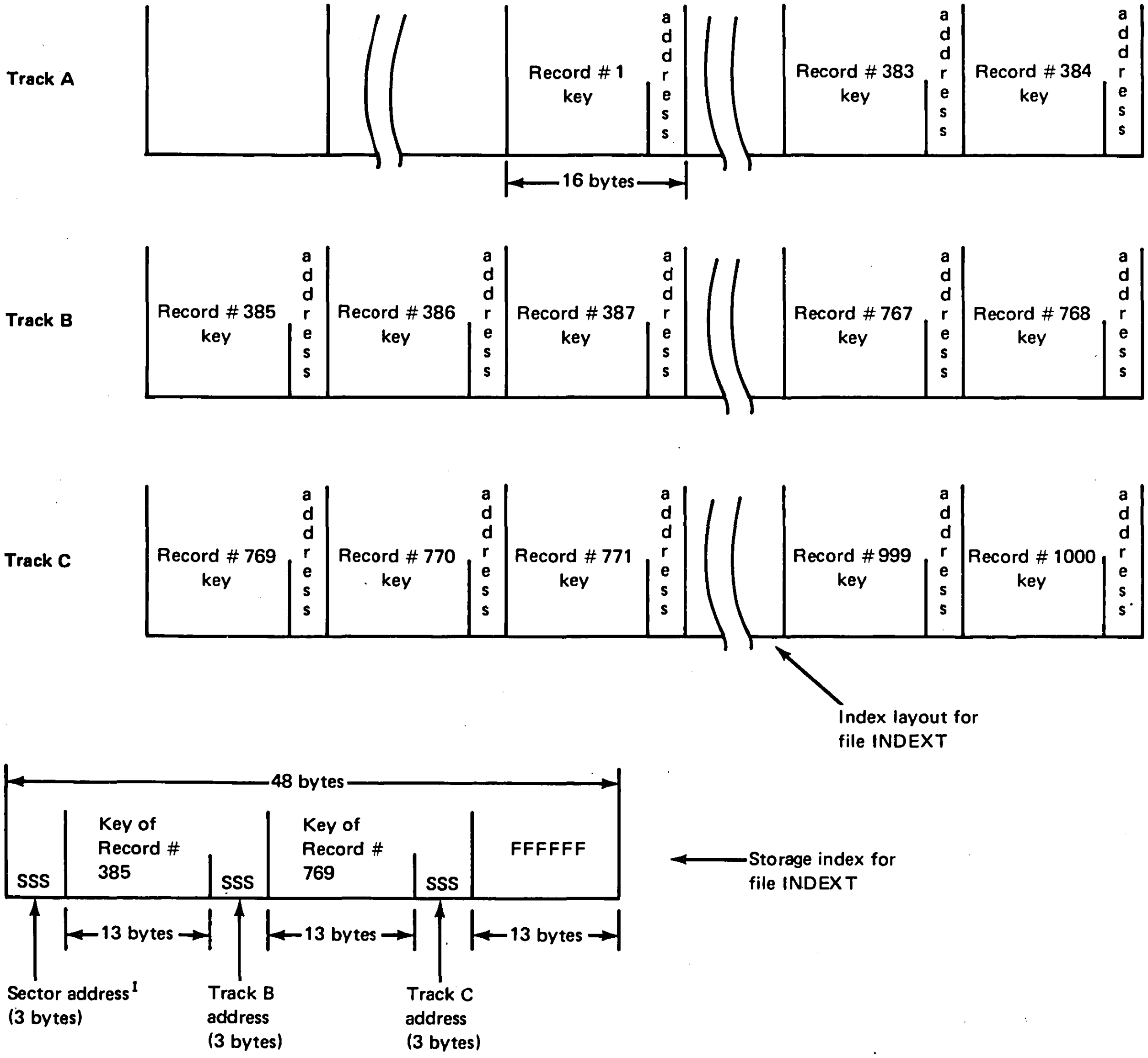
Use of the storage index significantly reduces the amount of time needed to process an indexed file. It enables the system to go more directly to the specific record you want by searching only a small portion of the file index. Without the storage index, all file index entries that precede the record you want must be searched. When the storage index is used, as shown in Figure 3-19, the record 767 can be found in the following manner:

1. Search the storage index until the first key field higher than key 767 is located. In this instance the key is 769 on track C; therefore, key 767 must reside on track B.

2. Search track B in the file index until key 767 is located.

3. Chain directly to the associated data record.

In columns 60 through 65, specify the number of storage positions (bytes) you want reserved for the storage index. Using the amount of main storage you specify, the system builds the most efficient storage index it can. The storage index is built immediately before your RPG II program is executed.



[1] If the index begins on a track boundary, this address will also be a track address.

Figure 3-19. Example of Storage Index

For efficient processing, the storage index you specify should be large enough to contain one entry for each track in the index, plus one entry that is the delimiter for the storage index. The minimum number of bytes required for a storage index entry for one track equals (the key length + 3) multiplied by 2. For example, if file INDEXT has one track that contains index entries and a key length of 4, the most efficient storage index is 14 bytes, that is, (4 + 3) times 2. If the file index occupies more than one track, the number of bytes required for the storage index equals (the key length + 3) multiplied by (the number of tracks containing index entries + 1). For example, if the key length equals 4 and the number of tracks containing index entries equals 5, the number of bytes required for the storage index is 42; that is, (4 + 3) multiplied by (5 + 1) equals 42.

To determine the number of bytes in the storage index, use the following calculation:

1. Use the CATALOG procedure to find the total number of records that the file can contain.

2. Determine the length of an entry in the file index by adding 3 to the key length.

3. Determine the number of keys in each sector by dividing 256 by the entry length. Drop the remainder.

4. Determine the number of sectors in the index by dividing the number of records in the file (the result of step 1) by the number of keys in each sector (the result of step 3). Round up the result.

5. Determine the number of tracks in the index by dividing the number of sectors (the result of step 4) by 60 if your disk has 27.1 megabytes or less. Divide by 64 if your disk has more than 27.1 megabytes. If the quotient is not a whole number, round it up to the next whole number. (To determine the number of megabytes on your disk, use the STATUS command to display the status of your work session. The AVAILABLE DISK SIZE field on the third display of work station status shows the number of megabytes on the disk.)

6. Determine the optimal size for the storage index by multiplying the length of an entry in the file index (the result of step 2) times the number of tracks in the file index (the result of step 5).

If the storage space specified in columns 60 through 65 is not large enough to contain an entry for every track of file index, the system divides the given area into as many entries as there is room for with each entry pointing to a larger track of the file index. As the number of entries in the storage index decreases, the amount of processing time increases.

COLUMN 66 (FILE ADDITION)

Entry	Explanation
A	New records are added to the file.
U	Records are to be loaded for an indexed file in unordered sequence (random sequence).

Use column 66 to indicate:

- The program is to add new records to the file (Figure 3-20). Records can be added at detail, total, or exception time during the program cycle.
- The program is to load records in an unordered sequence (Figure 3-21).

Column 66 applies to direct, sequential, and indexed disk files.

Note: Adding records to a file also requires a corresponding ADD entry in columns 16 through 18 of the output specifications.

Adding Records to an Indexed File (A)

Records added to an indexed file are added at the end of the file and entries for the new records are made in the index. (See *Key Sorting for Indexed Files* in the *Concepts and Design Guide* for an explanation of when the system sorts the keys into ascending sequence.) File addition (column 66) cannot be specified for indexed files from which records are read by the sequential-within-limits method. Records added to an indexed file should be in ascending sequence to achieve better performance.

When indexed chained files are specified with add, the records to be added may contain either of the following:

- Keys that are above the highest presently in the file. In this case, the records constitute an extension of the file.

- Keys that are either lower than the lowest presently in the file, or fall between those already in the file.

If records are to be added to an indexed file processed sequentially:

- The key of the record to be added must be lower than the key currently in process and higher than the preceding key, or

- The file must be at end of file.

If these requirements are not met, a halt occurs; otherwise, the record is added.

To add a record to any indexed file processed randomly, the program searches the index to the file to determine whether the record is on the file; if it is, a halt occurs. Otherwise, the record is added.

Adding Records to Delete-Capable Direct and Sequential Files

Records can be added to direct and sequential files that are processed randomly by relative record number. The file must be defined as delete-capable when it is built. When a delete-capable direct file is built, it is initialized to all deleted records (hex FFs). For further information on defining delete-capable files, see *FILE Statement* in the *System Support Reference Manual.*

If records are added to direct or sequential files, the file must be defined as an input or update chained file on the file description specifications. The file description specifications must also contain an A in column 66 to specify file addition. The keyword RECNO and its associated field must be specified on a file description specification continuation line for each file that records are added to (for further information on the RECNO keyword, see *Continuation Line Keyword Option* in this chapter).

You must place in the RECNO field the relative record number of the record to be added to the file. It must be the relative record number of a deleted record (one that is filled with hex FFs). You code output specifications that contain ADD in columns 16 through 18 to add records to a file. RPG uses the relative record number from the RECNO field to locate where the record is to be added to the file. If the relative record number is not the number of a deleted record, a halt occurs and a message is issued stating that a duplicate record exists in the file. An option to continue can be taken, but the record is not added to the file.

If a CHAIN operation accesses a nondeleted record in the file, that record is returned to the program and RPG places the relative record number in the RECNO field.

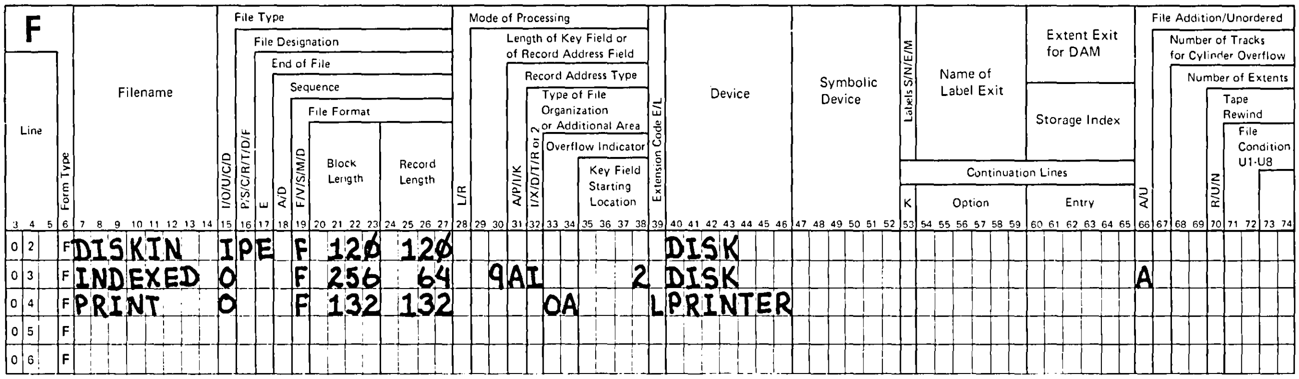
Records cannot be added to sequential files past end of file by using the RECNO keyword. For example, if a sequential file has relative record numbers 1 through 5 and 7 through 10, a record can be added at relative record number 6 but not at relative record number 11. Attempting to add a record past end of file results in error RPG-9038, THIS FILE IS FULL.

Extending a Sequential File

Records can be added to a sequential file without specifying the RECNO keyword. The file must be defined as an output file and column 66 of the file description specifications must contain an A to specify file addition.

Records added to a sequential file are added to the end of the file. They cannot be added between other records.

File Description Specifications

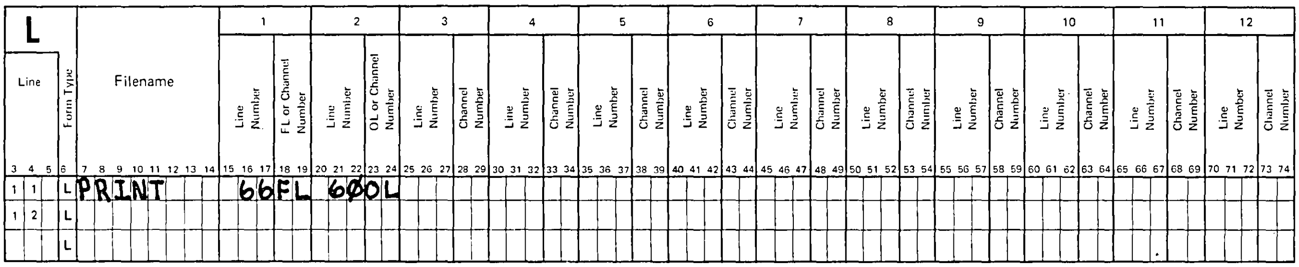


```
  02 FDISKIN  IPE F 120 120            DISK
  03 FINDEXED O   F 256  64  9AI     2 DISK                      A
  04 FPRINT   O   F 132 132     OA    LPRINTER
  05 F
  06 F
```

The new records are contained in disk file DISKIN. The file INDEXED is the existing disk file to which new records are added. A printer file, PRINT, provides a report showing all the records in DISKIN with an indication of which records are not added.

On the file description specifications sheet, an A must appear in column 66 for the file INDEXED. On the output specifications sheet, ADD must appear in columns 16 through 18 for the new record to be added.

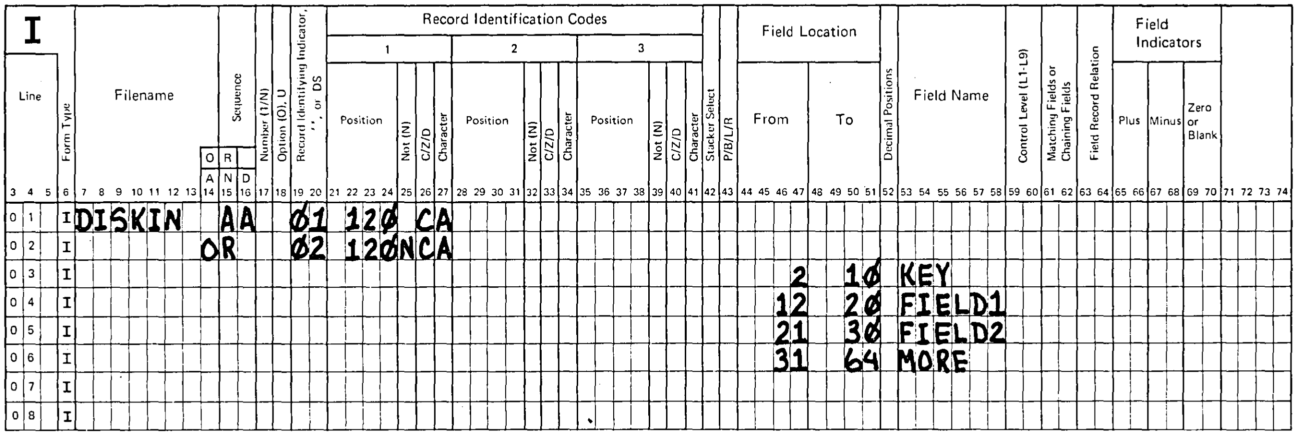
Line Counter Specifications



```
  11 LPRINT    66FL 60OL
  12 L
     L
```

As defined on the input specifications sheet, all the records in DISKIN should have an A in position 120. The code identifies a record to be added to the disk file, and this record type is assigned indicator 01. On the output specifications sheet, notice that when 01 is on, the data from DISKIN is written on the disk file INDEXED and is also printed on the file PRINT to keep a visual report of new records.

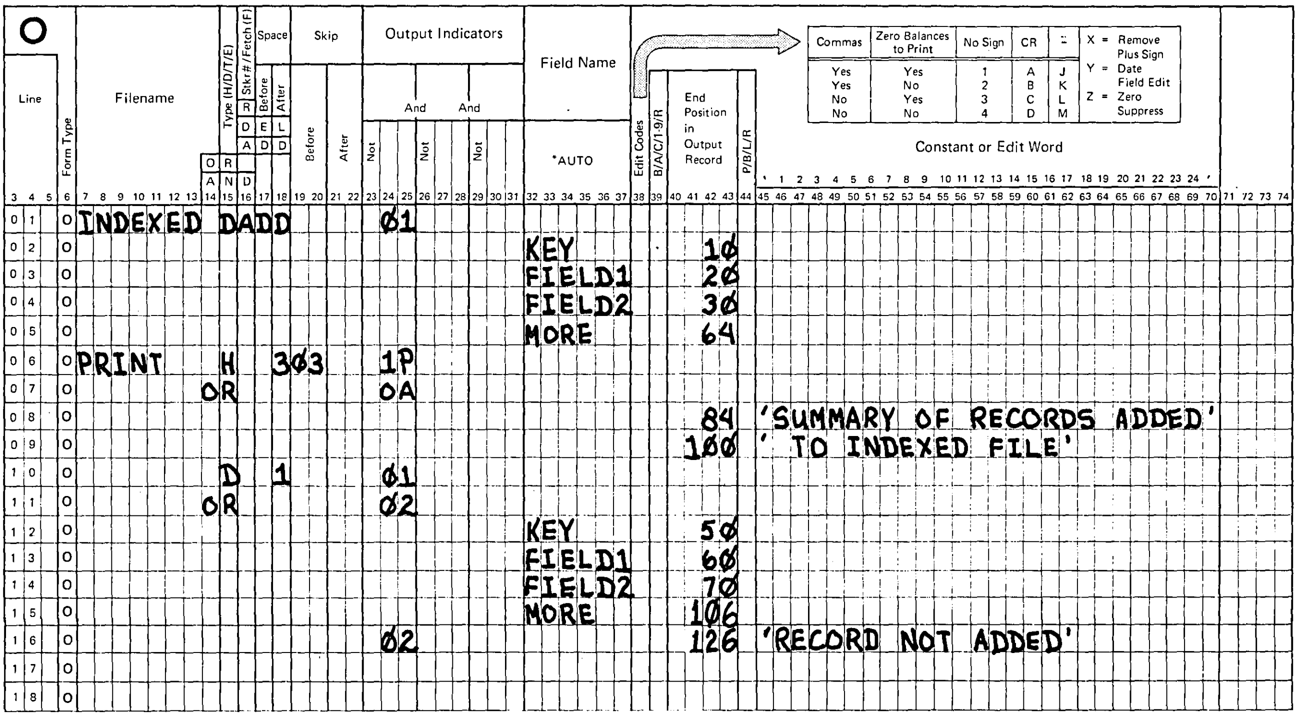
Input Specifications



```
  01 IDISKIN  AA  01 120 CA
  02 I       OR   02 120NCA
  03 I                                        2  10 KEY
  04 I                                       12  20 FIELD1
  05 I                                       21  30 FIELD2
  06 I                                       31  64 MORE
  07 I
  08 I
```

Figure 3-20 (Part 1 of 2). Adding New Records to a File

Output Specifications



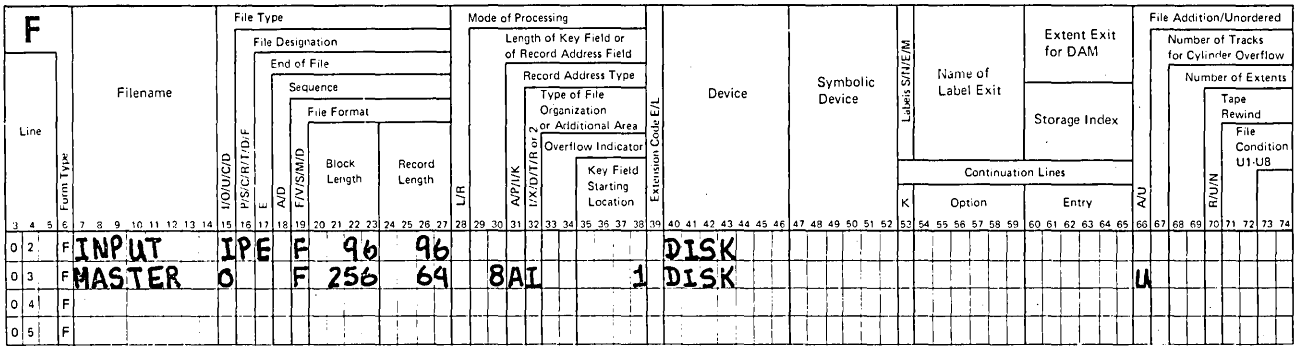
Commas	Zero Balances to Print	No Sign	CR	–	X = Remove Plus Sign Y = Date Field Edit Z = Zero Suppress
Yes	Yes	1	A	J	
Yes	No	2	B	K	
No	Yes	3	C	L	
No	No	4	D	M	

```
01 OINDEXED DADD     Ø1
02 O                         KEY       1Ø
03 O                         FIELD1    2Ø
04 O                         FIELD2    3Ø
05 O                         MORE      64
06 OPRINT   H  3Ø3   1P
07 O       OR        OA
08 O                                   84 'SUMMARY OF RECORDS ADDED'
09 O                                  1ØØ ' TO INDEXED FILE'
10 O        D  1     Ø1
11 O       OR        Ø2
12 O                         KEY       5Ø
13 O                         FIELD1    6Ø
14 O                         FIELD2    7Ø
15 O                         MORE     1Ø6
16 O                 Ø2               126 'RECORD NOT ADDED'
17 O
18 O
```

There may be records in DISKIN that do not belong in that file, or some records may have an error. Those records are identified on the input specifications sheet as not having the character A in position 120. These records turn on indicator 02 and are not to be added to the disk file INDEXED. However, these records are printed on the file PRINT for a visual report, but they must be identified in the printed report as records that were not added to the disk file INDEXED. On the output specifications sheet, the constant RECORDS NOT ADDED is printed only on indicator 02, indicating a record that was not added to the disk file. In this manner, a report of all records in DISKIN is printed, and the records not added to INDEXED are identified by the constant RECORD NOT ADDED.

Figure 3-20 (Part 2 of 2). Adding New Records to a File

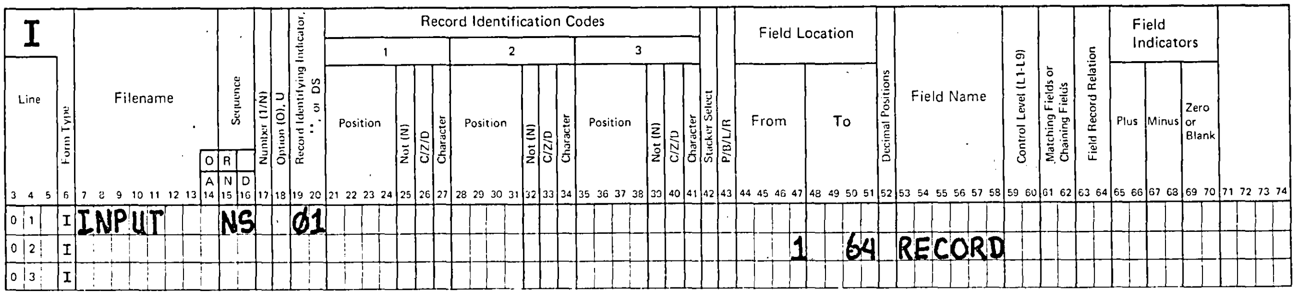
File Description Specifications



Line	Form Type	Filename	File Type	File Designation	End of File	File Format	Block Length	Record Length	Length of Key Field	Record Address Type	Type of File Organization	Key Field Starting Location	Device	File Addition/Unordered
02	F	INPUT	I	P	E	F	96	96					DISK	
03	F	MASTER	O			F	256	64	8	A	I	1	DISK	U
04	F													
05	F													

The output file, MASTER, is described as an indexed file to be loaded and processed by record keys. The U in column 66 of the file description specifications sheet indicates that an unordered load is to be done. The input file, INPUT, is described by the input specifications as being unsequenced.

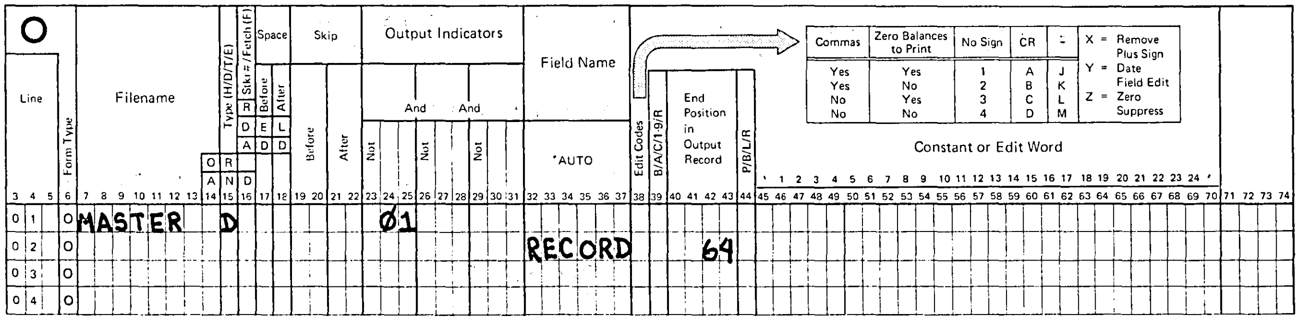
Input Specifications



Line	Form Type	Filename	Sequence	Record Identifying Indicator	Field Location From	Field Location To	Field Name
01	I	INPUT	NS	01			
02	I				1	64	RECORD
03	I						

The keys from which the index is to be built appear as the first eight positions of the output record. As the disk file is loaded, the key is extracted from the record and an index entry containing the location of the record on the disk is built. After the entire file is loaded and an index entry is constructed for each record, the index entries are sorted into ascending sequence.

Output Specifications



Line	Form Type	Filename	Type (H/D/T/E)	Output Indicators	Field Name	End Position in Output Record
01	O	MASTER	D	01		
02	O				RECORD	64
03	O					
04	O					

Figure 3-21. Unordered Load of An Indexed File

Loading Records in an Unordered Sequence (U)

Unordered load (U in column 66) is specified when an indexed file is to be built from records in an unordered sequence. After records are loaded and an index is built in the unordered sequence, the system sorts the index into ascending sequence.

File Addition Functions for Indexed Files

In the following chart, combinations of entries in column 15 and column 66 show the functions that can be performed for indexed files (I in column 32).

Col 15	Col 66	Function
O	Blank	Load records in ascending key sequence to an indexed file.
O	U	Load records in unordered key sequence to an indexed file.
O	A	Add records to an existing indexed file.
I	Blank	Read records of an indexed file without adding new records or updating records.
I	A	Read records of an indexed file and add new records to the file that are not presently there. No updating is performed.
U	Blank	Update records of an indexed file without adding new records.
U	A	Update records of an indexed file and add new records to the file.

File Addition Functions for Sequential Files (File Extension)

Col 15	Col 66	Function
O	A	Extend an existing sequential file by adding records to the end of the file.
O	Blank	Load records in a sequential file.

Random File Addition Functions for Delete-Capable Direct and Sequential Files

Col 15	Col 16	Col 66	Function
I	C	A[1]	Add records to an existing direct or sequential file.
U	C	A[1]	Add records to an existing direct or sequential file.
O		Blank[1]	Load a direct file.

[1] Adding records to delete-capable files requires that the keyword RECNO and its associated field be specified on a file description specifications continuation line.

COLUMN 67

Column 67 is not used. Leave it blank.

COLUMNS 68-69

Columns 68 and 69 are not used. Leave them blank.

COLUMN 70

Column 70 is not used. Leave it blank.

COLUMNS 71-72 (FILE CONDITION)

Entry	Explanation
Blank	The file is not conditioned by an external indicator.
U1-U8	The file is conditioned by the specified external indicator.

Use columns 71 and 72 to indicate whether the file is conditioned by an external indicator. Columns 71 and 72 apply to input (excluding table input files, and KEYBORD files), update, and output files. A file conditioned by an external indicator is used only when the indicator is on. When the indicator is off, the file is treated as though the end of the file is reached; that is, no records can be read from or written to the file.

The external indicators are normally set prior to processing by the SWITCH OCL statement or by a previous RPG II program. Their setting can be changed during processing, allowing the program to alter the status of these indicators. However, if an external indicator conditions a file, that indicator must be set on when the program is loaded in order to use the file in the program. For information on how to save and restore the external indicators for each display station attached to a WORKSTN file, see Chapter 13, *WORKSTN File Considerations and Sample Programs.*

If a file is conditioned by an external indicator, any calculations that are not done when the file is not used should also be conditioned by the same indicator.

COLUMNS 73-74

Columns 73 and 74 are not used. Leave them blank.

COLUMNS 75-80 (PROGRAM IDENTIFICATION)

See *Common Entries* in Chapter 1.

FILE DESCRIPTION CHARTS

Figures 3-22 through 3-33 show the file description specification entries for disk files (which are presented by disk file organization and processing method), KEYBORD files, CONSOLE files, PRINTER files, CRT files, WORKSTN files, SPECIAL files, and BSCA files. When you use the charts, keep the following in mind:

- The entries in the chart must be made for the processing method and type of file described on that line.
- The shaded columns must be blank for the file described on that line.
- The other columns may be required or optional, but cannot be indicated on the chart because the entries represent information that changes from program to program.

How to Use the Charts

If you are updating an indexed disk file using the CHAIN operation code, see Figure 3-22 and refer to indexed disk files, random processing by CHAIN operation code. Then choose the chained update file with or without record addition.

In this example, the following columns are required but may change from one program to another: filename, record length, length of key field, and key field starting location. Optional entries are line, block length, storage index, and file condition.

Figure 3-22. Processing Methods for Indexed Disk Files (Using the Index)

File Description Specifications

Type of Processing		Line (3-5)	Form Type (6)	I/O/U/C/D (15)	P/S/C/R/T/D/F (16)	F/V/S/M/D (19)	L/R (28)	A/P/I/K (31)	I/X/D/T/R or 2 (32)	Device (40-46)	A/U (66)
Sequential[1]	by Key, no ADD	02	F	I	P	F		A	I	DISK	
	by Key, no ADD	03	F	I	S	F		A	I	DISK	
	by Key, no ADD	04	F	I	D	F		A	I	DISK	
	by Key, with ADD	05	F	I	P	F		A	I	DISK	A
	by Key, with ADD	06	F	I	S	F		A	I	DISK	A
	by Key, with ADD	07	F	I	D	F		A	I	DISK	A
	by Key, no ADD	08	F	U	P	F		A	I	DISK	
	by Key, no ADD	09	F	U	S	F		A	I	DISK	
	by Key, no ADD	10	F	U	D	F		A	I	DISK	
	by Key, with ADD	11	F	U	P	F		A	I	DISK	A
	by Key, with ADD	12	F	U	S	F		A	I	DISK	A
	by Key, with ADD	13	F	U	D	F		A	I	DISK	A
	by Limits	14	F	I	P	F	L	A	I	DISK	
	by Limits	15	F	I	S	F	L	A	I	DISK	
	by Limits	16	F	I	D	F	L	A	I	DISK	
	by Limits	17	F	U	P	F	L	A	I	DISK	
	by Limits	18	F	U	S	F	L	A	I	DISK	
	by Limits	19	F	U	D	F	L	A	I	DISK	
		20	F								
Random	by CHAIN, no ADD[2]	21	F	I	C	F	R	A	I	DISK	
	by CHAIN, with ADD[2]	22	F	I	C	F	R	A	I	DISK	A
	by CHAIN, no ADD[2]	23	F	U	C	F	R	A	I	DISK	
	by CHAIN, with ADD[2]	24	F	U	C	F	R	A	I	DISK	A
		25	F								
	by Addrout	27	F	I	P	F	R	I	I	DISK	
	by Addrout	28	F	I	S	F	R	I	I	DISK	
	by Addrout	29	F	U	P	F	R	I	I	DISK	
	by Addrout	30	F	U	S	F	R	I	I	DISK	
		32	F								
Load	Unordered	33	F	O		F		A	I	DISK	U
	Ordered	34	F	O		F		A	I	DISK	
		35	F								
Add records only	ADD only	36	F	O		F		A	I	DISK	A
		37	F								

Shaded columns must be blank.

[1] Sequential processing by key or limits must use the file index, which is always arranged in ascending sequence. When an indexed file is processed record by record from beginning to end, the sequential-by-key method is used to process the file through the index.

[2] If chained files are processed by key, column 31 should contain an A; however, if chained files are processed by relative record number, columns 31 and 32 must be blank.

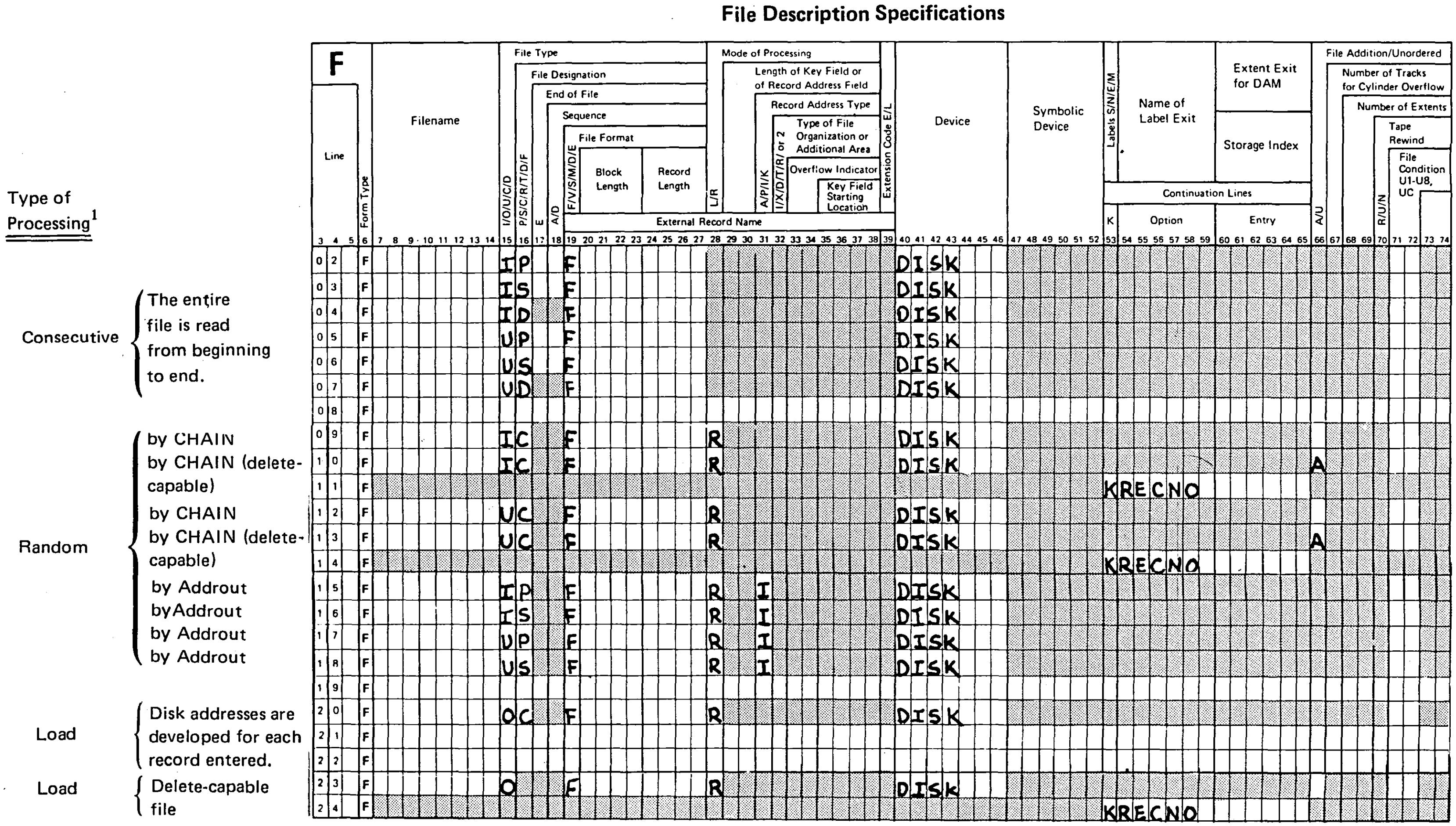
Figure 3-23. Processing Methods for Sequential Disk Files

File Description Specifications

Type of Processing		Line	Form Type	File Type (I/O/U/C/D)	File Designation (P/S/C/R/T/D/F)	File Format (F/V/S/M/D/E)	Mode of Processing (L/R)	Record Address Type (A/P/I/K)	Extension Code E/L	Device	K	Option	A/U
Consecutive	The entire file is read from beginning to end	02	F	I	P	F				DISK			
		03	F	I	S	F				DISK			
		04	F	I	T	F			E	DISK			
		05	F	I	D	F				DISK			
		06	F	U	P	F				DISK			
		07	F	U	S	F				DISK			
		08	F	U	D	F				DISK			
		09	F										
Random	by CHAIN	10	F	I	C	F	R			DISK			
	by CHAIN (delete-capable)	11	F	I	C	F	R			DISK			A
		12	F								K	RECNO	
	by CHAIN	13	F	U	C	F	R			DISK			
	by CHAIN (delete-capable)	14	F	U	C	F	R			DISK			A
		15	F								K	RECNO	
		16	F										
	by Addrout	17	F	I	P	F	R	I		DISK			
	by Addrout	18	F	I	S	F	R	I		DISK			
	by Addrout	19	F	U	P	F	R	I		DISK			
	by Addrout	20	F	U	S	F	R	I		DISK			
		21	F										
Load	The file is written on disk as entered	22	F	O		F				DISK			
		23	F										
Add records only	ADD only	24	F	O		F				DISK			A
		25	F										
		26	F										

Shaded columns must be blank.

Figure 3-24. Processing Methods for Direct Disk Files



File Description Specifications

Type of Processing[1]		Line (3-5)	Form Type (6)	Filename (7-14)	File Type I/O/U/C/D (15)	File Designation P/S/C/R/T/D/F (16)	File Format F/V/S/M/D/E (19)	Mode of Processing L/R (28)	Record Address Type A/P/I/K (31)	Device (40-46)	K (53)	Option (54-59)	A/U (66)
Consecutive	The entire file is read from beginning to end.	02	F		I	P	F			DISK			
		03	F		I	S	F			DISK			
		04	F		I	D	F			DISK			
		05	F		U	P	F			DISK			
		06	F		U	S	F			DISK			
		07	F		U	D	F			DISK			
		08	F										
Random	by CHAIN	09	F		I	C	F	R		DISK			
	by CHAIN (delete-capable)	10	F		I	C	F	R		DISK			A
		11	F								K	RECNO	
	by CHAIN	12	F		U	C	F	R		DISK			
	by CHAIN (delete-capable)	13	F		U	C	F	R		DISK			A
		14	F								K	RECNO	
	by Addrout	15	F		I	P	F	R	I	DISK			
	byAddrout	16	F		I	S	F	R	I	DISK			
	by Addrout	17	F		U	P	F	R	I	DISK			
	by Addrout	18	F		U	S	F	R	I	DISK			
		19	F										
Load	Disk addresses are developed for each record entered.	20	F		O	C	F	R		DISK			
		21	F										
		22	F										
Load	Delete-capable file	23	F		O		F	R		DISK			
		24	F								K	RECNO	

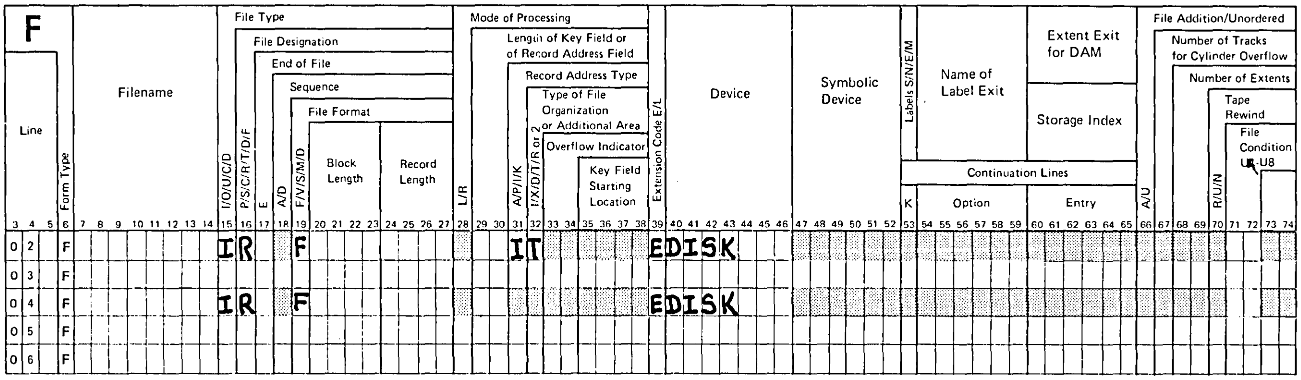
[1] To insert or change records in a direct file, define the file as an update file processed consecutively, or an update file processed randomly by the CHAIN operation code.

Shaded columns must be blank.

Record Address Files[1]

Containing: 1. Relative record numbers (addrout file)
2. Record key limits

File Description Specifications

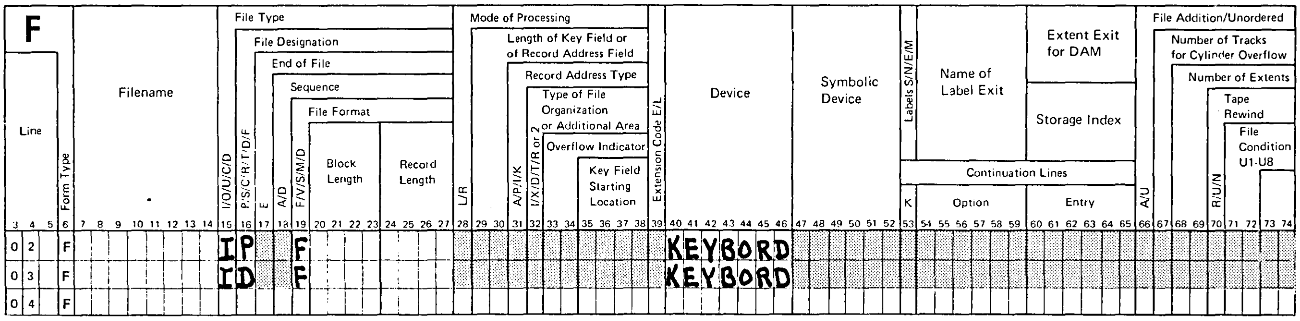

F
Line
Form Type
Filename
File Type
File Designation
End of File
Sequence
File Format
I/O/U/C/D
P/S/C/R/T/D/F
E
A/D
F/V/S/M/D
Block Length
Record Length
L/R
Mode of Processing
Length of Key Field or of Record Address Field
Record Address Type
Type of File Organization or Additional Area
Overflow Indicator
Key Field Starting Location
Extension Code E/L
Device
Symbolic Device
Labels S/N/E/M
Name of Label Exit
Extent Exit for DAM
Storage Index
Continuation Lines
K
Option
Entry
A/U
File Addition/Unordered
Number of Tracks for Cylinder Overflow
Number of Extents
Tape Rewind
File Condition U1-U8
R/U/N
3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74

```
02 F        IR  F           IT      EDISK
03 F
04 F        IR  F                   EDISK
05 F
06 F
```

Shaded columns must be blank

Figure 3-25. Record Address Files Located on Disk

File Description Specifications


F
Line
Form Type
Filename
File Type
File Designation
End of File
Sequence
File Format
I/O/U/C/D
P/S/C/R/T/D/F
E
A/D
F/V/S/M/D
Block Length
Record Length
L/R
Mode of Processing
Length of Key Field or of Record Address Field
Record Address Type
Type of File Organization or Additional Area
Overflow Indicator
Key Field Starting Location
Extension Code E/L
Device
Symbolic Device
Labels S/N/E/M
Name of Label Exit
Extent Exit for DAM
Storage Index
Continuation Lines
K
Option
Entry
A/U
File Addition/Unordered
Number of Tracks for Cylinder Overflow
Number of Extents
Tape Rewind
File Condition U1-U8
R/U/N
3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74

```
02 F        IP  F                    KEYBORD
03 F        ID  F                    KEYBORD
04 F
```

Shaded columns must be blank.

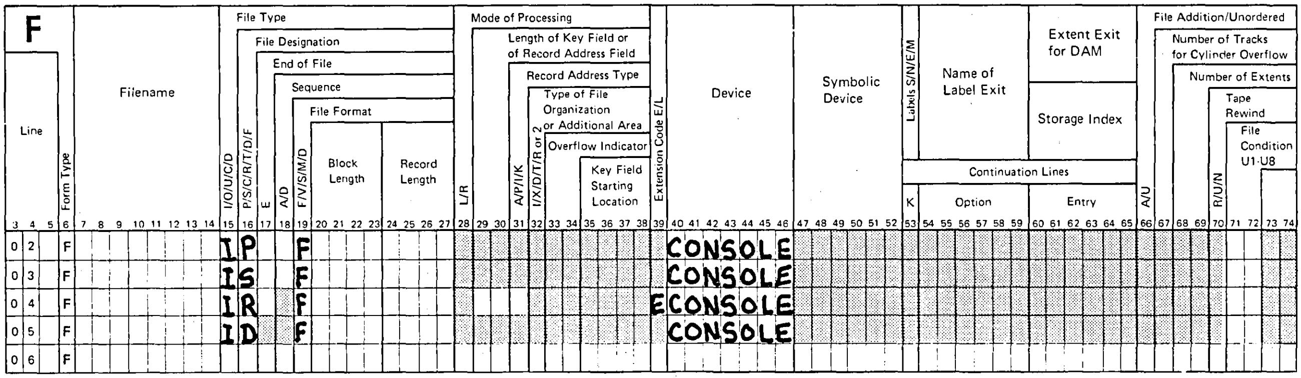
- If KEYBORD is specified as a *primary input* file, no other *input* files in the program can be specified as *primary* or *secondary* files.
- Input data entered from the KEYBORD device must be defined in calculation specifications for a KEY operation.
- No input specifications can be used for KEYBORD files.

Figure 3-26. KEYBOARD Files

[1] Record address files containing relative record numbers can be associated with indexed, sequential, or direct disk files.

Record address files containing record key limits can only be associated with indexed disk files, but can be a disk or CONSOLE file. (See chart for CONSOLE files.)

File Description Specifications



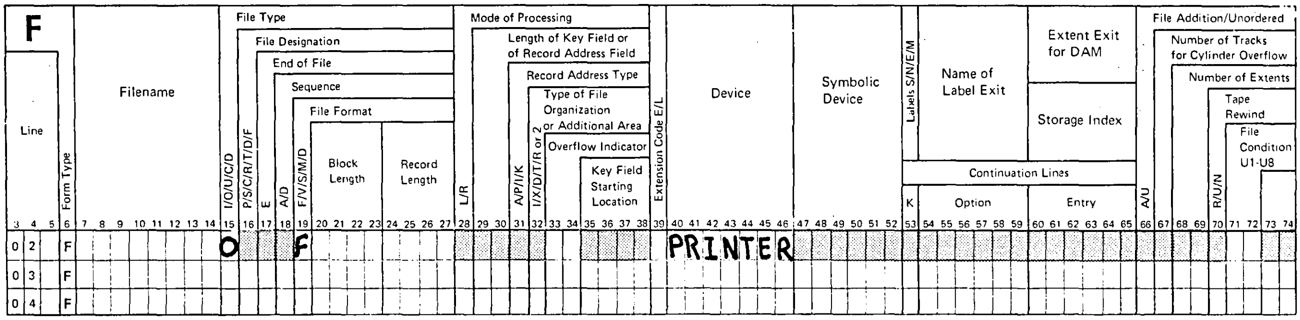
Line (3-5)	Form Type (6)	Filename (7-14)	File Type I/O/U/C/D (15)	File Designation P/S/C/R/T/D/F (16)	End of File E (17)	Sequence A/D (18)	File Format F/V/S/M/D (19)	Block Length (20-23)	Record Length (24-27)	L/R (28)	Length of Key Field or of Record Address Field (29-30)	Record Address Type A/P/I/K (31)	Type of File Organization or Additional Area I/X/D/T/R or 2 (32)	Overflow Indicator (33-34)	Key Field Starting Location (35-38)	Extension Code E/L (39)	Device (40-46)	Symbolic Device (47-52)	Labels S/N/E/M (53)	Name of Label Exit (54-59)	Extent Exit for DAM (60-65)	A/U (66)	Number of Tracks for Cylinder Overflow (67)	Number of Extents (68-69)	Tape Rewind R/U/N (70)	File Condition U1-U8 (71-72)	(73-74)
02	F		I	P			F										CONSOLE										
03	F		I	S			F										CONSOLE										
04	F		I	R			F									E	CONSOLE										
05	F		I	D			F										CONSOLE										
06	F																										

Input file records are displayed on the display screen when keyed into the program.

Shaded columns must be blank.

Figure 3-27. CONSOLE Files

File Description Specifications

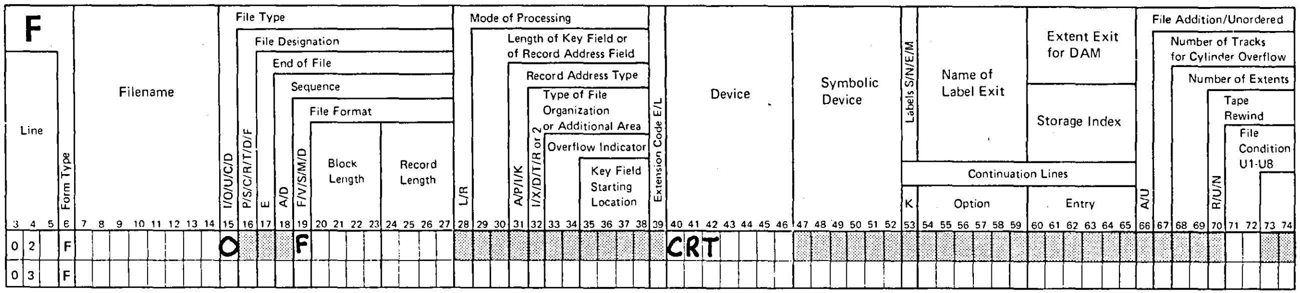


Line (3-5)	Form Type (6)	Filename (7-14)	File Type I/O/U/C/D (15)	File Designation P/S/C/R/T/D/F (16)	End of File E (17)	Sequence A/D (18)	File Format F/V/S/M/D (19)	Block Length (20-23)	Record Length (24-27)	L/R (28)	Length of Key Field or of Record Address Field (29-30)	Record Address Type A/P/I/K (31)	Type of File Organization or Additional Area I/X/D/T/R or 2 (32)	Overflow Indicator (33-34)	Key Field Starting Location (35-38)	Extension Code E/L (39)	Device (40-46)	Symbolic Device (47-52)	Labels S/N/E/M (53)	Name of Label Exit (54-59)	Extent Exit for DAM (60-65)	A/U (66)	Number of Tracks for Cylinder Overflow (67)	Number of Extents (68-69)	Tape Rewind R/U/N (70)	File Condition U1-U8 (71-72)	(73-74)
02	F		O				F										PRINTER										
03	F																										
04	F																										

Shaded columns must be blank.

Figure 3-28. PRINTER Files

File Description Specifications

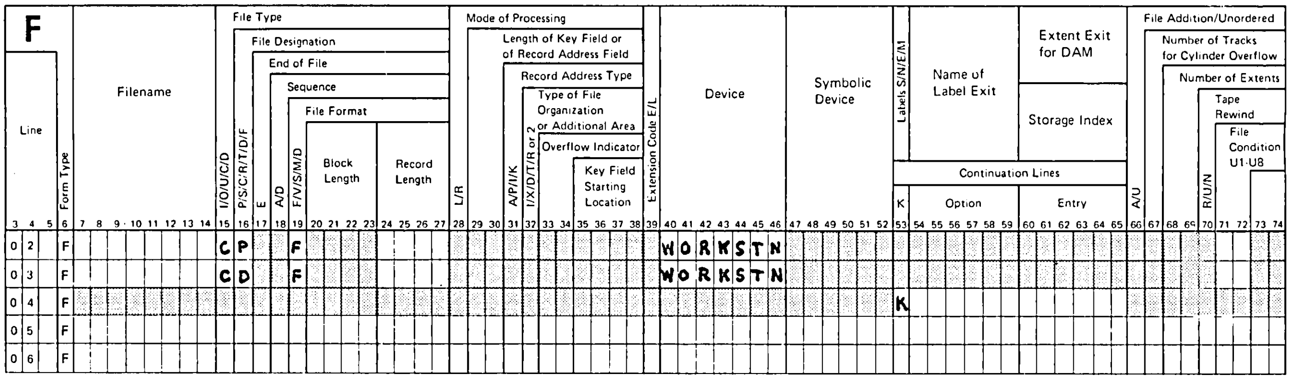


Line (3-5)	Form Type (6)	Filename (7-14)	File Type I/O/U/C/D (15)	File Designation P/S/C/R/T/D/F (16)	End of File E (17)	Sequence A/D (18)	File Format F/V/S/M/D (19)	Block Length (20-23)	Record Length (24-27)	L/R (28)	Length of Key Field or of Record Address Field (29-30)	Record Address Type A/P/I/K (31)	Type of File Organization or Additional Area I/X/D/T/R or 2 (32)	Overflow Indicator (33-34)	Key Field Starting Location (35-38)	Extension Code E/L (39)	Device (40-46)	Symbolic Device (47-52)	Labels S/N/E/M (53)	Name of Label Exit (54-59)	Extent Exit for DAM (60-65)	A/U (66)	Number of Tracks for Cylinder Overflow (67)	Number of Extents (68-69)	Tape Rewind R/U/N (70)	File Condition U1-U8 (71-72)	(73-74)
02	F		O				F										CRT										
03	F																										

Shaded columns must be blank.

Figure 3-29. CRT Files

File Description Specifications

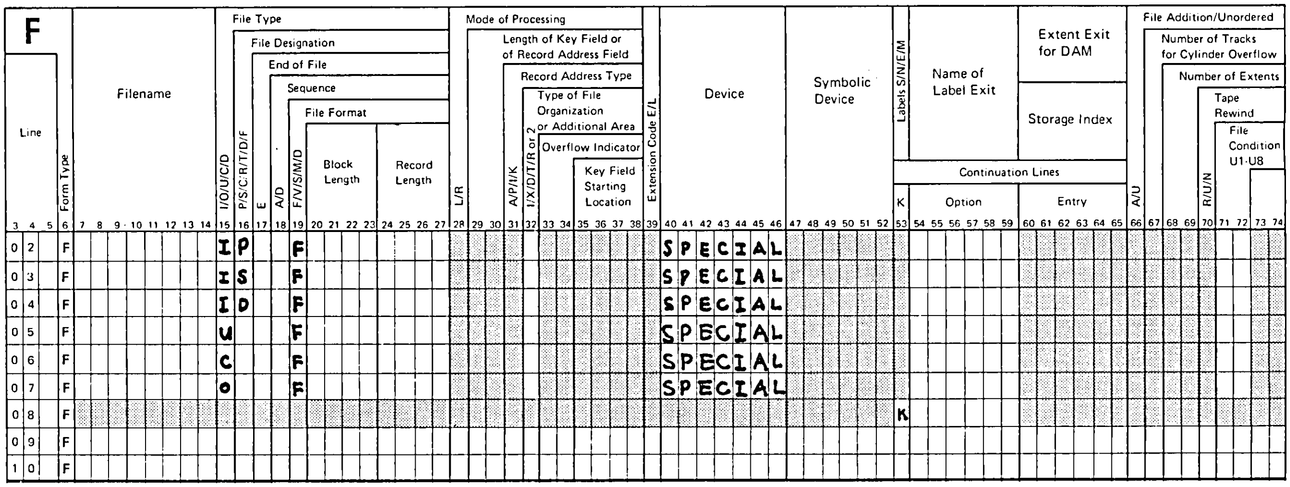


Line	Form Type	Filename	I/O/U/C/D (15)	P/S/C/R/T/D/F (16)	E (17)	A/D (18)	F/V/S/M/D (19)	Block Length	Record Length	Device (40-46)	Symbolic Device	K (53)	Option	Entry
02	F		C	P			F			WORKSTN				
03	F		C	D			F			WORKSTN				
04	F											K		
05	F													
06	F													

Shaded columns must be blank.

Figure 3-30. WORKSTN Files

File Description Specifications

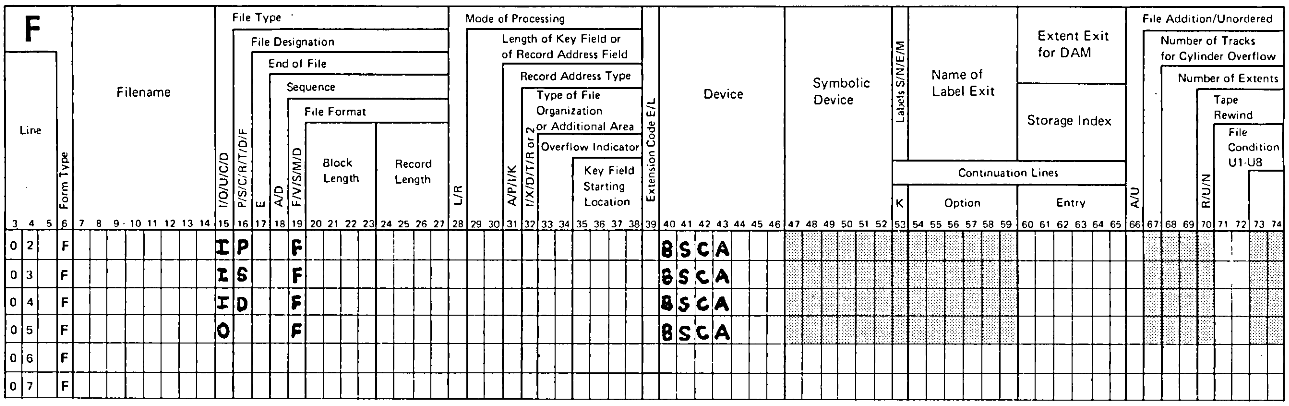


Line	Form Type	Filename	I/O/U/C/D (15)	P/S/C/R/T/D/F (16)	E (17)	A/D (18)	F/V/S/M/D (19)	Block Length	Record Length	Device (40-46)	Symbolic Device	K (53)	Option	Entry
02	F		I	P			F			SPECIAL				
03	F		I	S			F			SPECIAL				
04	F		I	D			F			SPECIAL				
05	F		U				F			SPECIAL				
06	F		C				F			SPECIAL				
07	F		O				F			SPECIAL				
08	F											K		
09	F													
10	F													

Shaded columns must be blank.

Figure 3-31. SPECIAL Files

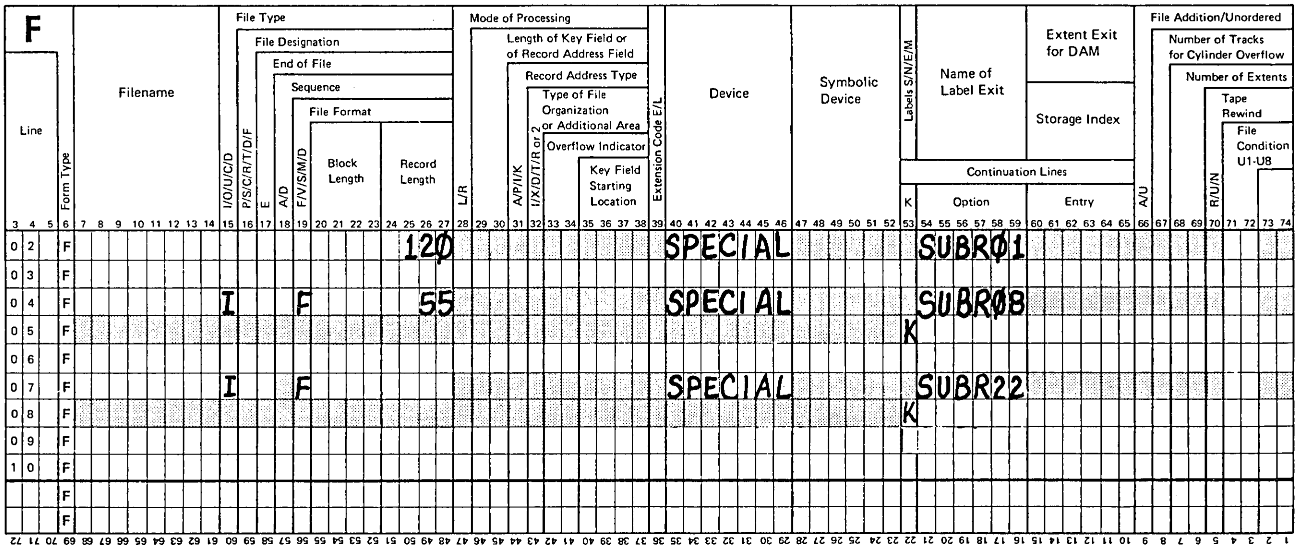
File Description Specifications



Line	Form Type	Filename	I/O/U/C/D (15)	P/S/C/R/T/D/F (16)	E (17)	A/D (18)	F/V/S/M/D (19)	Block Length	Record Length	Device (40-46)	Symbolic Device	K (53)	Option	Entry
02	F		I	P			F			BSCA				
03	F		I	S			F			BSCA				
04	F		I	D			F			BSCA				
05	F		O				F			BSCA				
06	F													
07	F													

Shaded columns must be blank.

Figure 3-32. BSCA Files



File Description Specifications

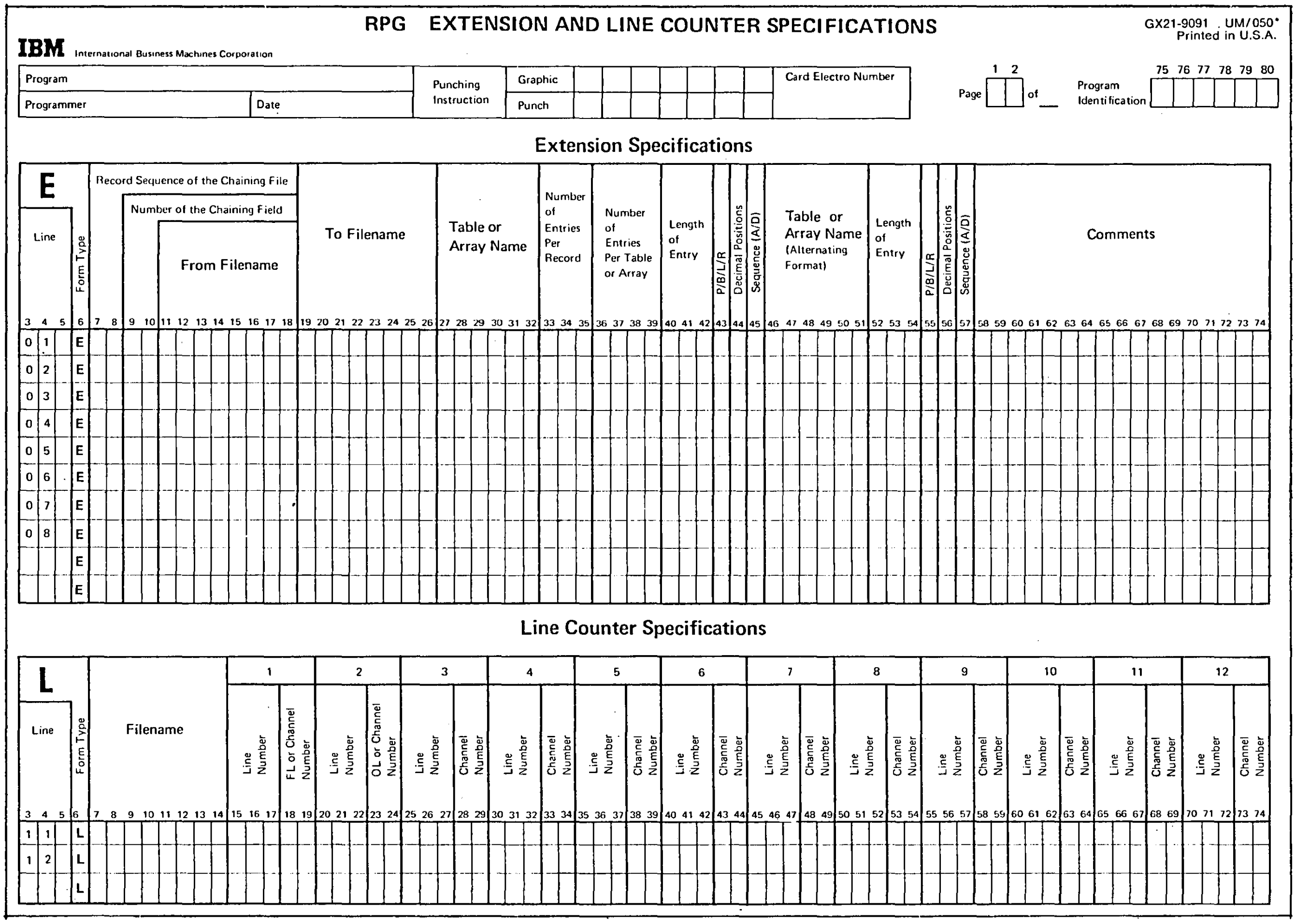
Line	Form Type	Filename	I/O/U/C/D	P/S/C/R/T/D/F	E	A/D	F/V/S/M/D	Block Length	Record Length	L/R	Mode of Processing	Length of Key Field or of Record Address Field	A/P/I/K	I/X/D/T/R or 2	Overflow Indicator	Key Field Starting Location	Extension Code E/L	Device	Symbolic Device	Labels S/N/E/M K	Name of Label Exit Option	Extent Exit for DAM Storage Index Entry	A/U	Number of Tracks for Cylinder Overflow Number of Extents	R/U/N	File Condition U1-U8
0 2	F								12Ø									SPECIAL			SUBRØ1					
0 3	F																									
0 4	F		I				F		55									SPECIAL			SUBRØ8					
0 5	F																			K						
0 6	F																									
0 7	F		I				F											SPECIAL			SUBR22					
0 8	F																			K						
0 9	F																									
1 0	F																									
	F																									
	F																									

Figure 3-33. File Description Specifications for IBM-Supplied Subroutines

Chapter 4. Extension Specifications

Extension specifications describe all record address files, compile-time or preexecution-time tables, and compile-time, preexecution-time, or execution-time arrays used in a program. Write these specifications on the RPG Extension and Line Counter Specifications sheet (Figure 4-1).

Record address files require entries in columns 11 through 26. Preexecution-time tables and arrays require entries in columns 11 through 45. Compile-time tables and arrays require entries in columns 19 through 45. If an alternating table or array is specified with the table or array described in columns 11 through 45 or 19 through 45, it must be described in columns 46 through 57 of the same line. A maximum of 75 tables or arrays or data structures can be used in a program; however, only 70 of these can be compile-time tables or arrays.



RPG EXTENSION AND LINE COUNTER SPECIFICATIONS

IBM International Business Machines Corporation

GX21-9091 . UM/050*
Printed in U.S.A.

Program
Programmer
Date
Punching Instruction
Graphic
Punch
Card Electro Number
Page 1 2 of
Program Identification 75 76 77 78 79 80

Extension Specifications

Line (3-5)	Form Type (6)	Record Sequence of the Chaining File (7-8)	Number of the Chaining Field (9-10)	From Filename (11-18)	To Filename (19-26)	Table or Array Name (27-32)	Number of Entries Per Record (33-35)	Number of Entries Per Table or Array (36-39)	Length of Entry (40-42)	P/B/L/R (43)	Decimal Positions (44)	Sequence (A/D) (45)	Table or Array Name (Alternating Format) (46-51)	Length of Entry (52-54)	P/B/L/R (55)	Decimal Positions (56)	Sequence (A/D) (57)	Comments (58-74)
01	E																	
02	E																	
03	E																	
04	E																	
05	E																	
06	E																	
07	E																	
08	E																	
	E																	
	E																	

Line Counter Specifications

Line (3-5)	Form Type (6)	Filename (7-14)	1 Line Number (15-17)	1 FL or Channel Number (18-19)	2 Line Number (20-22)	2 OL or Channel Number (23-24)	3 Line Number (25-27)	3 Channel Number (28-29)	4 Line Number (30-32)	4 Channel Number (33-34)	5 Line Number (35-37)	5 Channel Number (38-39)	6 Line Number (40-42)	6 Channel Number (43-44)	7 Line Number (45-47)	7 Channel Number (48-49)	8 Line Number (50-52)	8 Channel Number (53-54)	9 Line Number (55-57)	9 Channel Number (58-59)	10 Line Number (60-62)	10 Channel Number (63-64)	11 Line Number (65-67)	11 Channel Number (68-69)	12 Line Number (70-72)	12 Channel Number (73-74)
11	L																									
12	L																									
	L																									

Figure 4-1. RPG Extension and Line Counter Specifications

Figure 4-6 at the end of this chapter shows possible extension specifications. See Chapter 14, *Tables and Arrays* for a complete discussion of tables and arrays.

COLUMNS 1-2 (PAGE)

See *Common Entries* in Chapter 1.

COLUMNS 3-5 (LINE)

See *Common Entries* in Chapter 1.

COLUMN 6 (FORM TYPE)

An E must appear in column 6 to identify this line as an extension specifications statement.

COLUMNS 7-10

Columns 7 through 10 are not used. Leave them blank.

COLUMNS 11-18 (FROM FILENAME)

Entry	Explanation
Blank	– Table or array is loaded at compilation time if there is an entry in columns 33 through 35. – Array is loaded at execution time (via input and/or calculation specifications) if there is no entry in columns 33 through 35.
Record address filename	Name of the record address file.
Table or array filename	Name of the table or array file loaded at preexecution time.

Use columns 11 through 18 to name a table file, array file, or record address file. Filenames must begin in column 11. The record address filename must always be entered in these columns and also on the file description specifications. The filename of every preexecution-time table or array used in the program must be entered in these columns. Leave columns 11 through 18 blank for compile-time tables or arrays and for arrays loaded via input and/or calculation specifications (execution-time arrays).

When a table or array is loaded at compilation time, it is compiled along with the source program and included in the object program. Such a table file does not need to be loaded separately every time the program is run. Only those tables and arrays that contain constant data should be compiled with the program.

When tables or arrays are compiled with the program, table records must always follow the RPG II source program. A record with **ƀ (ƀ = blank) in positions 1 through 3 must separate the RPG II source program from the table or array records. Tables or arrays must be separated from each other by records with **ƀ in positions 1 through 3. Because **ƀ in positions 1 through 3 indicates the start of a table or array, **ƀ must not be specified in positions 1 through 3 of the table input records.

Short tables (tables that contain blank entries) can be compiled with the program, but a warning is issued. See *Columns 36-39* in this chapter for a discussion of short tables.

COLUMNS 19-26 (TO FILENAME)

Entry	Explanation
Name of an input or update file	File processed via the record address file named in columns 11 through 18.
Name of an output file	Output file to which a table or array is to be written at end of job.

Use columns 19 through 26 to define the relationship between a file named in these columns and a file named in columns 11 through 18. Filenames must begin in column 19.

If a record address file is named in columns 11 through 18, the name of the input or update file that contains the data records to be processed must be entered in columns 19 through 26. Do not enter the record address filename in these columns.

If a table or array is to be written at end of job (that is, after LR processing), enter the filename of the output file in columns 19 through 26. This output file must be named previously in the file description specifications. A table or array can be written to only one output device. Leave columns 19 through 26 blank if the table or array is not to be written.

If a table or array is assigned to an output file, it is automatically written at the end of the execution–after all other records are written. The table or array is written in the same format in which it was entered.

Because there is no program control over the output format when an entry is made in columns 19 through 26, those cases where formatting is required should be provided for in the program through the output specifications or by the EXCPT operation that writes one item at a time (see Chapter 10, *Operation Codes, EXCPT*). Tables or arrays should be written only after all records are processed (last record indicator is on).

COLUMNS 27-32 (TABLE OR ARRAY NAME)

Entry	Explanation
Table or array name	Name of table or array used in the program

Use columns 27 through 32 to name the table or array. No two tables or arrays can have the same name. The rules for forming table and array names are discussed in the following text.

Table Name

Every table used in your program must have a name that begins with the letters TAB. The entire table name can be from 3 to 6 characters long.

After the letters TAB, 1 to 3 alphabetic or numeric characters can be used (no special characters are allowed). Blanks cannot appear between characters in the table name. Any name in columns 27 through 32 that does not begin with TAB is considered an array name.

The table name entered in columns 27 through 32 is used throughout the program. However, different results can be obtained depending upon how the table name is used. When the table name is used in factor 2 or the result field of the calculation specifications with a LOKUP operation, the name refers to the entire table. When the table name is used with any other operation code, the name refers to the table entry last selected from the table by a LOKUP operation (see *Operation Codes, LOKUP* in Chapter 10 and *Tables and Arrays* in Chapter 14).

Table files are processed in the order they are specified on the extension specifications. Therefore, if you have more than one table file, the files are to be loaded in the same order as they appear on the extension specifications.

If two tables are in alternating format in one file, the table whose entry appears first must be named in columns 27 through 32. The second table is named in columns 46 through 51 (Figure 4-2).

Array Name

Each array used in a program must be given a unique name that does not begin with the letters TAB. The name can be from 1 to 6 characters long and must begin with an alphabetic character. This array name is used throughout the program. The array name should be used by itself only to reference the entire array. See *Array Name and Index* in Chapter 14 for more information on array names and on referencing array entries.

COLUMNS 33-35 (NUMBER OF ENTRIES PER RECORD)

Entry	Explanation
1-999	Number of table or array entries in each table or array input record

Use columns 33 through 35 to indicate the exact number of table or array entries in each table or array input record. The number must end in column 35. Every table or array input record except the last must contain the same number of entries as indicated in columns 33 through 35. The last record can contain fewer entries than indicated, but not more. Comments can be entered on table input records in the positions following the table entries.

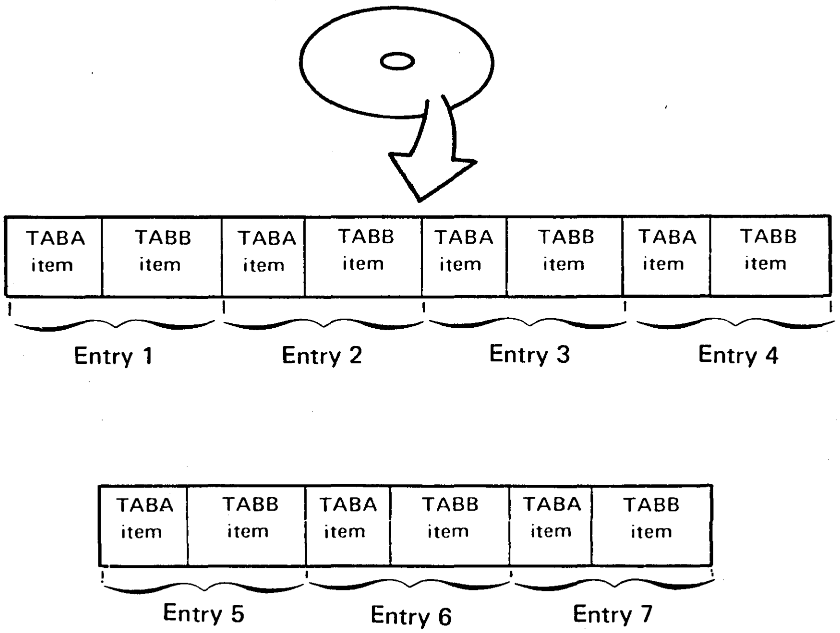
If two tables or arrays are in alternating format in one file, each table or array input record must contain the corresponding entries from each table or array. The corresponding entries from the two tables or arrays are considered one entry and must be on the same record.

Two tables (TABA and TABB) are described in alternating format. An item for TABA appears first. Thus, TABA is named in columns 27 through 32 of the extension specifications sheet (see Part 2 of this figure); TABB is named in columns 46 through 51.

Table A (account number)	Table B (amount due)
00126	56.75
03240	39.00
03648	156.72
15632	17.98
28887	2.97
29821	290.98
30001	579.95
5 positions	7 positions

Corresponding table items

Note: The decimal points shown in Table B are only for illustration purposes. Decimal points are not a part of table or array input data.

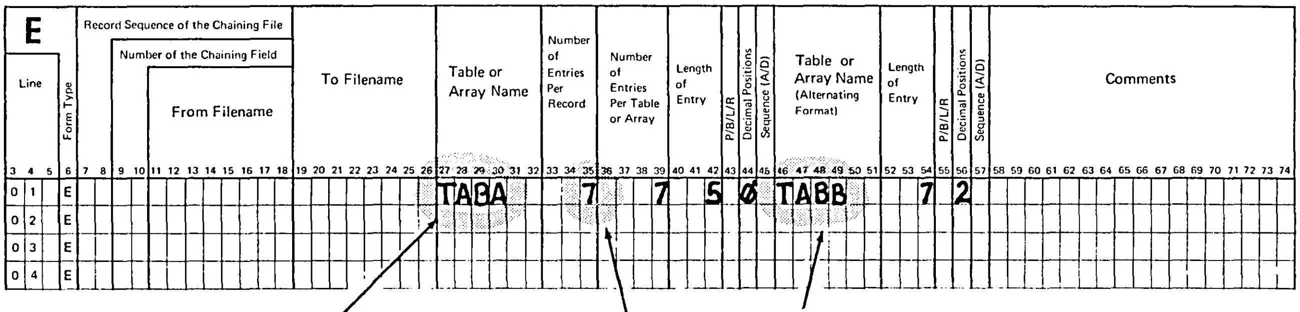


The corresponding items from the tables are entered in the system in alternating format. Corresponding items from the two tables are considered as one entry.

Figure 4-2 (Part 1 of 2). Related Tables

Table entries for the two tables, A and B, are entered in alternating format. A1 and B1, the corresponding items in tables A and B, are considered one entry. Even though 14 table items are listed, there are only seven table entries.

Extension Specifications



Line	Form Type	Record Sequence of the Chaining File (7-8)	Number of the Chaining Field (9-10)	From Filename (11-18)	To Filename (19-26)	Table or Array Name (27-32)	Number of Entries Per Record (33-35)	Number of Entries Per Table or Array (36-39)	Length of Entry (40-42)	P/B/L/R (43)	Decimal Positions (44)	Sequence (A/D) (45)	Table or Array Name (Alternating Format) (46-51)	Length of Entry (52-54)	P/B/L/R (55)	Decimal Positions (56)	Sequence (A/D) (57)	Comments (58-74)
01	E					TABA	7	7	5		Ø		TABB	7		2		
02	E																	
03	E																	
04	E																	

Table whose items are loaded first is named in columns 27 through 32.

Table whose items are loaded second is named in columns 46 through 51.

This entry indicates the number of table entries in each input record. Remember, corresponding items from the two tables are considered as one entry.

Figure 4-2 (Part 2 of 2). Related Tables

When columns 27 through 32 contain an array name, the following rules apply to the use of columns 11 through 18 and 33 through 35:

- For a preexecution-time array, columns 11 through 18 must contain a filename and columns 33 through 35 must have an entry.
- For a compile-time array, columns 11 through 18 must be blank and columns 33 through 35 must have an entry.
- For an execution-time array, columns 11 through 18 and columns 33 through 35 must be blank.

COLUMNS 36-39 (NUMBER OF ENTRIES PER TABLE OR ARRAY)

Entry	Explanation
1-9999	Maximum number of table or array entries

Use columns 36 through 39 to indicate the maximum number of entries that can be contained in the table or array named in columns 27 through 32. This number applies to one table or array or to two tables or arrays in alternating format. The number entered must end in column 39.

Because the number of entries for two tables or arrays written in alternating format must be the same, the number in these columns also gives the number of entries in the second table or array specified in columns 46 through 51.

If the table or array is full, these columns give the exact number of entries in it. However, if the table or array is not full, these columns give the number of entries that can be put into it (Figure 4-3). A table or array that is not full is one that contains unused entries and is known as a short table or array.

A compile-time table or array should be full. However, if it is not full (a short table or array), the table or array is compiled with the program and a warning is issued. In storage, the unused entries in a short table or array are filled with blanks or zeros (for alphameric or numeric tables or arrays, respectively). A preexecution-time table or array need not be full.

COLUMNS 40-42 (LENGTH OF ENTRY)

Entry	Explanation
1-15	Length of a numeric entry
1-256	Length of an alphameric entry

Use columns 40 through 42 to specify the length of each entry in the table or array named in columns 27 through 32. The number entered must end in column 42. For numeric tables or arrays in packed decimal format, enter the zoned decimal length in columns 40 through 42. For numeric tables or arrays in binary format, enter the number of digits required in storage for the binary field. For a two-position binary field, the entry in columns 40 through 42 is 4; for a four-position binary field, the entry is 9.

All table or array entries must have the same number of characters. It is almost impossible, however, for every item to be the same length. Therefore, add leading zeros for numeric entries and add blanks after alphameric entries to make them the same length (see Figure 4-4).

For compile-time arrays, the maximum length of an alphameric entry is 96 because the maximum length of a record in the source program is 96 characters.

If two tables or arrays are entered in alternating format, the specification in columns 40 through 42 applies to the table or array whose entry appears first in the record (see Figure 4-5).

See Chapter 14, *Tables and Arrays*, for more information.

COLUMN 43 (PACKED OR BINARY FIELD)

Entry	Explanation
Blank	Data for table or array is in zoned decimal format or is alphameric.
P	Data for table or array is in packed decimal format on disk.
B	Data for table or array is in binary format on disk.

Use column 43 to indicate that a numeric field in a preexecution-time table or array file is in packed or binary format. Leave column 43 blank if the field is in zoned decimal format. See *Column 43* under *Field Description Entries* in Chapter 7 for more information on packed or binary format.

COLUMN 44 (DECIMAL POSITIONS)

Entry	Explanation
Blank	Alphameric table or array
0-9	Number of positions to the right of the decimal in numeric table or array items

Use column 44 to indicate the number of decimal positions in a numeric table or array entry. Column 44 must always have an entry for a numeric table or array. If the entries in a table or array have no decimal positions, enter a 0.

Note: The decimal points shown in these tables are only for illustration purposes. Decimal points are not part of table input data.

TABPRT (part number)	TABAMT (price)
001	127.62
002	198.32
003	.27
004	.01
005	1.98
009	3.79
010	5.67
014	2.33
026	14.67
045	29.33
096	29.34
097	.05
098	.09
099	1.19
100	2.22
101	126.73
110	596.74
115	393.75
126	697.75
137	1.92

If this data is entered into the system, TABPRT and TABAMT will be full (20 entries fill the table).

TABPRT (part number)	TABAMT (price)
001	127.62
002	198.32
003	.27
004	.01
005	1.98

If this data is entered into the system, TABPRT and TABAMT will not be full.

Extension Specifications

E

Line	Form Type	Record Sequence of the Chaining File	Number of the Chaining Field	From Filename	To Filename	Table or Array Name	Number of Entries Per Record	Number of Entries Per Table or Array	Length of Entry	P/B/L/R	Decimal Positions	Sequence (A/D)	Table or Array Name (Alternating Format)	Length of Entry	P/B/L/R	Decimal Positions	Sequence (A/D)	Comments
3 4 5	6	7 8	9 10	11 12 13 14 15 16 17 18	19 20 21 22 23 24 25 26	27 28 29 30 31 32	33 34 35	36 37 38 39	40 41 42	43	44	45	46 47 48 49 50 51	52 53 54	55	56	57	58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74
0 1	E					TABPRT	12	2Ø	3		Ø		TABAMT	5		2		
0 2	E																	
0 3	E																	
0 4	E																	

This entry indicates that TABPRT and TABAMT can both have a maximum of 20 entries.

Figure 4-3. Table Entries (Number per Table)

JANUARY
FEBRUARY
MARCH
APRIL
MAY
JUNE
JULY
AUGUST
SEPTEMBER
OCTOBER
NOVEMBER
DECEMBER

```
JANUARYbb
FEBRUARYb
MARCHbbbb
APRILbbbb
MAYbbbbbb
JUNEbbbbb
JULYbbbbb
AUGUSTbbb
SEPTEMBER
OCTOBERbb
NOVEMBERb
DECEMBERb
```

All entries must have the same length. Those items that are not as long as the longest item must be padded with blanks (b).

List of Months   TABMO

Figure 4-4. Length of Table Entries

COLUMN 45 (SEQUENCE)

Entry	Explanation
Blank	No particular order
A	Ascending order
D	Descending order

Use column 45 to describe the sequence (either ascending or descending) of the data in a table or array file.

When an entry is made in column 45, the table or array is checked for the specified sequence. If a compile-time table or array is out of sequence, a severe error occurs and the program halts after compilation. If a preexecution-time table or array is out of sequence, an error occurs and the program halts immediately. The program can be restarted from the point where it halted if you do not want to correct the out-of-sequence condition; however, if you do correct the out-of-sequence condition, program execution must be restarted from the beginning.

Ascending order means that the table or array entries start with the lowest data entry (according to the collating sequence) and proceed to the highest. Descending order means that the table or array entries start with the highest data entry and proceed to the lowest.

If two tables or arrays are entered in alternating format, the entry in column 45 applies to the table or array containing the entry that appears first on the record. When the LOKUP operation is used to search a table or array for an entry to determine whether the entry is high or low compared with the search word, the table or array must be in either ascending or descending order. See *LOKUP* in Chapter 10 for more information.

An execution-time array (built by input and/or calculation specifications) is not sequence checked. However, an A or D entry must be specified if a high or low LOKUP operation is performed.

COLUMNS 46-57

Entry	Explanation
Table or array name	Name of the alternating table or array

Use columns 46 through 57 only to describe a second table or array that is entered in an alternating format with the table or array specified in columns 27 through 32. All fields in this section have the same significance and require the same entries as the fields with corresponding titles in columns 27 through 45. See the previous discussion on those columns for information about correct specifications. Leave these columns blank for a single table or array.

TABCOD (code)	TABAMT (amount)
021	217.43
022	93.06
023	8.14
040	2166.58
041	39.23
060	1741.78
117	83.33
118	5.12
143	72.03
352	253.96
3 positions	6 positions

Two tables are entered in alternating format, TABCOD and TABAMT. Each item in TABCOD is 3 characters long; each item in TABAMT is 6 characters long. Since TABCOD is entered in the system first, its length, 3, is specified in columns 40 through 42. The length of items in TABAMT is in columns 52 through 54.

Note: The decimal points shown in these tables are only for illustration purposes. Decimal points are not a part of table input data.

Extension Specifications

E Line (3-5)	Form Type (6)	Record Sequence of the Chaining File (7-8)	Number of the Chaining Field (9-10)	From Filename (11-18)	To Filename (19-26)	Table or Array Name (27-32)	Number of Entries Per Record (33-35)	Number of Entries Per Table or Array (36-39)	Length of Entry (40-42)	P/B/L/R (43)	Decimal Positions (44)	Sequence (A/D) (45)	Table or Array Name (Alternating Format) (46-51)	Length of Entry (52-54)	P/B/L/R (55)	Decimal Positions (56)	Sequence (A/D) (57)	Comments (58-74)
01	E					TABCOD	1Ø	1Ø	3				TABAMT	6		2		
02	E																	
03	E																	
04	E																	

The length of the table item that is first entered in the system must appear in columns 40 through 42.

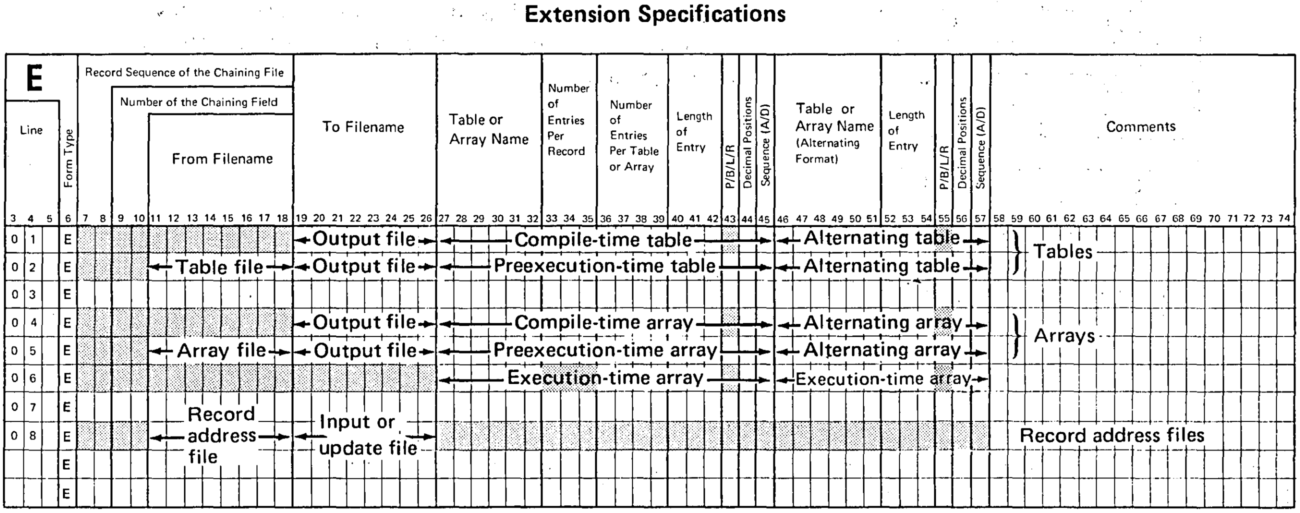
Figure 4-5. Length of Corresponding Table Items

COLUMNS 58-74 (COMMENTS)

Columns 58 through 74 can be used for comments to document the purpose of each specification line.

COLUMNS 75-80 (PROGRAM IDENTIFICATION)

See *Common Entries* in Chapter 1.



Extension Specifications

Line	Form Type	Record Sequence of the Chaining File	Number of the Chaining Field	From Filename	To Filename	Table or Array Name	Number of Entries Per Record	Number of Entries Per Table or Array	Length of Entry	P/B/L/R	Decimal Positions	Sequence (A/D)	Table or Array Name (Alternating Format)	Length of Entry	P/B/L/R	Decimal Positions	Sequence (A/D)	Comments
01	E				Output file	Compile-time table							Alternating table					Tables
02	E			Table file	Output file	Preexecution-time table							Alternating table					
03	E																	
04	E				Output file	Compile-time array							Alternating array					Arrays
05	E			Array file	Output file	Preexecution-time array							Alternating array					
06	E					Execution-time array							Execution-time array					
07	E			Record address file	Input or update file													
08	E																	Record address files
	E																	
	E																	

- The shaded columns must be blank for the file named.
- For tables and all arrays except execution-time arrays, columns 19 through 26 are optional. For all tables and arrays, columns 46 through 57 are optional.
- Execution-time arrays are loaded via input and/or calculation specifications.
- For record address files, columns 11 through 26 must have entries.

Figure 4-6. Possible File Entries for Extension Specifications

Chapter 5. Line Counter Specifications

Line counter specifications indicate at what line overflow occurs and the length of the form used in the printer. . Both of these entries must be specified on the RPG Extension and Line Counter Specifications sheet (Figure 5-1). Line counter specifications should be used for each printer file in your program. If no line counter specifications exist, the form length used is the form length specified on the PRINTER OCL statement. (See the LINES parameter of the PRINTER OCL statement in the *System Support Reference Manual* for a description of the defaults for the form length.)

In these instances, the overflow line is assumed to be six lines less than the specified form length.

COLUMNS 1-2 (PAGE)

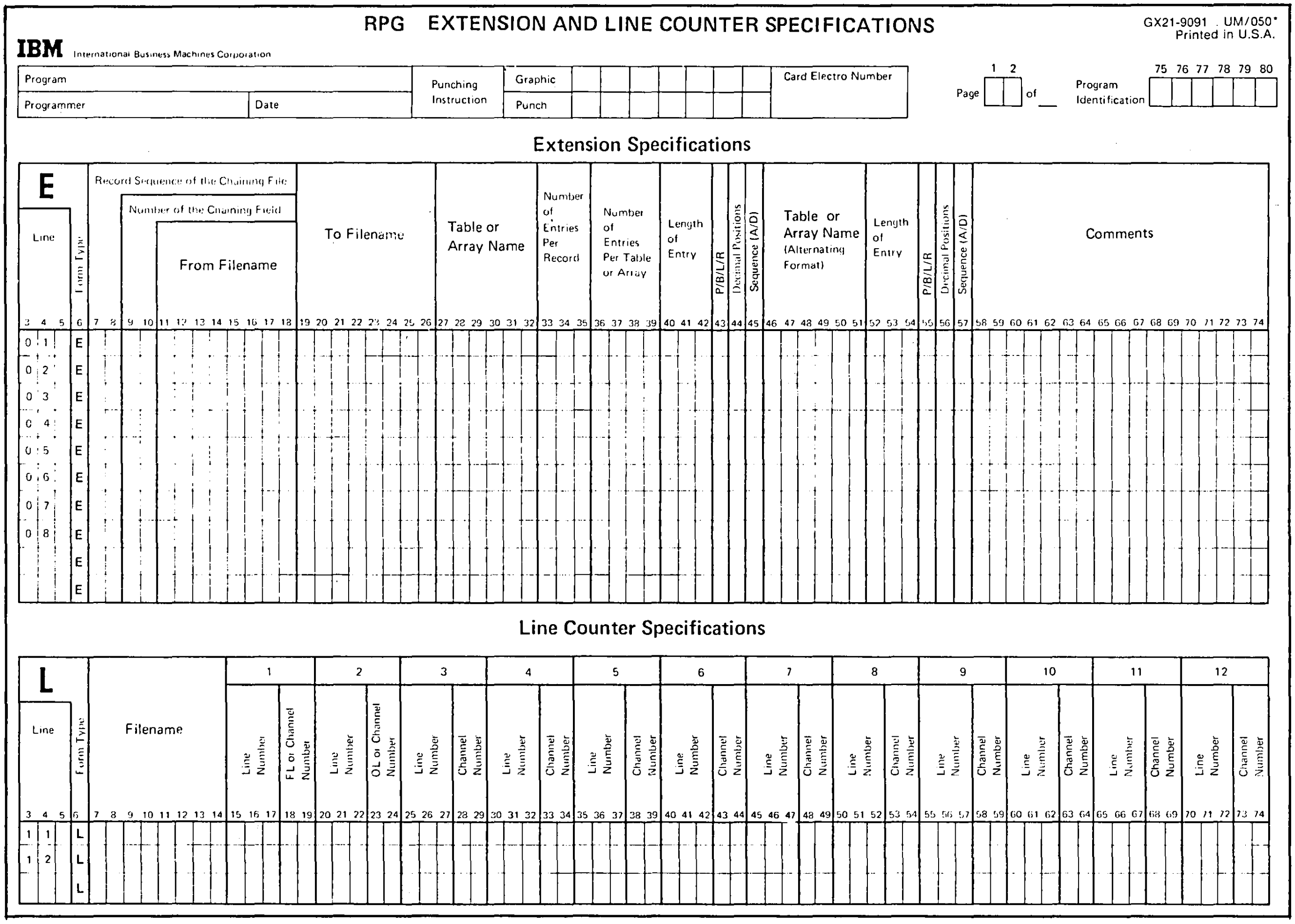
See *Common Entries* in Chapter 1.

COLUMNS 3-5 (LINE)

See *Common Entries* in Chapter 1.

COLUMN 6 (FORM TYPE)

An L must appear in column 6 to identify this line as a line counter specifications statement.



IBM International Business Machines Corporation

RPG EXTENSION AND LINE COUNTER SPECIFICATIONS

GX21-9091 . UM/050*
Printed in U.S.A.

Program		Punching Instruction	Graphic
Programmer	Date		Punch

Card Electro Number

1 2
Page □□ of __

75 76 77 78 79 80
Program Identification □□□□□□

Extension Specifications

E Line	Form Type	Record Sequence of the Chaining File	Number of the Chaining Field	From Filename	To Filename	Table or Array Name	Number of Entries Per Record	Number of Entries Per Table or Array	Length of Entry	P/B/L/R	Decimal Positions	Sequence (A/D)	Table or Array Name (Alternating Format)	Length of Entry	P/B/L/R	Decimal Positions	Sequence (A/D)	Comments
3 4 5	6	7 8	9 10	11 12 13 14 15 16 17 18	19 20 21 22 23 24 25 26	27 28 29 30 31 32	33 34 35	36 37 38 39	40 41 42	43	44	45	46 47 48 49 50 51	52 53 54	55	56	57	58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74
0 1	E																	
0 2	E																	
0 3	E																	
0 4	E																	
0 5	E																	
0 6	E																	
0 7	E																	
0 8	E																	
	E																	
	E																	

Line Counter Specifications

L			1		2		3		4		5		6		7		8		9		10		11		12	
Line	Form Type	Filename	Line Number	FL or Channel Number	Line Number	OL or Channel Number	Line Number	Channel Number	Line Number	Channel Number	Line Number	Channel Number	Line Number	Channel Number	Line Number	Channel Number	Line Number	Channel Number	Line Number	Channel Number	Line Number	Channel Number	Line Number	Channel Number	Line Number	Channel Number
3 4 5	6	7 8 9 10 11 12 13 14	15 16 17	18 19	20 21 22	23 24	25 26 27	28 29	30 31 32	33 34	35 36 37	38 39	40 41 42	43 44	45 46 47	48 49	50 51 52	53 54	55 56 57	58 59	60 61 62	63 64	65 66 67	68 69	70 71 72	73 74
1 1	L																									
1 2	L																									
	L																									

Figure 5-1. RPG Extension and Line Counter Specifications

COLUMNS 7-14 (FILENAME)

Entry	Explanation
A valid filename	Filename of the printer output file as previously defined on the file description specifications. The filename must begin in column 7.

Use columns 7 through 14 to identify the output file to be printed on the printer.

COLUMNS 15-17 (LINE NUMBER–NUMBER OF LINES PER PAGE)

Entry	Explanation
1-112	Number of printing lines available is 1 to 112.

Use columns 15 through 17 to specify the exact number of lines available on the form or page to be used. The entry must end in column 17. Leading zeros can be omitted.

COLUMNS 18-19 (FORM LENGTH)

Entry	Explanation
FL	Form length

Use columns 18 and 19 to indicate that the preceding entry (columns 15 through 17) is the form length. Columns 18 and 19 must contain the entry FL.

COLUMNS 20-22 (LINE NUMBER–OVERFLOW LINE)

Entry	Explanation
1-112	The line number specified is the overflow line.

Use columns 20 through 22 to specify the line number that is the overflow line. The entry must end in column 22. Leading zeros can be omitted. The entry must be less than or equal to the form length specified in columns 15 through 17. When the line that is specified as the overflow line is printed, the overflow indicator turns on. When the overflow indicator is on and fetch overflow is not specified, the following occurs before forms advance to the next page:

1. Detail lines are printed (if this part of the program cycle has not already been completed).

2. Total lines are printed (if conditions are met).

3. Total lines conditioned by the overflow indicator are printed.

Because all these lines are printed on the page after the overflow line, specify the overflow line high enough on the page to allow all these lines to print. See *Overflow Indicators* in Chapter 9 for more information.

Note: If the number of lines per page entry equals the overflow line entry, no overflow occurs.

COLUMNS 23-24 (OVERFLOW LINE)

Entry	Explanation
OL	Overflow line

Use columns 23 and 24 to indicate that the preceding entry (columns 20 through 22) is the overflow line. Columns 23 and 24 must contain OL.

COLUMNS 25-74

Columns 25 through 74 are not used. Leave them blank.

COLUMNS 75-80 (PROGRAM IDENTIFICATION)

See *Common Entries* in Chapter 1.

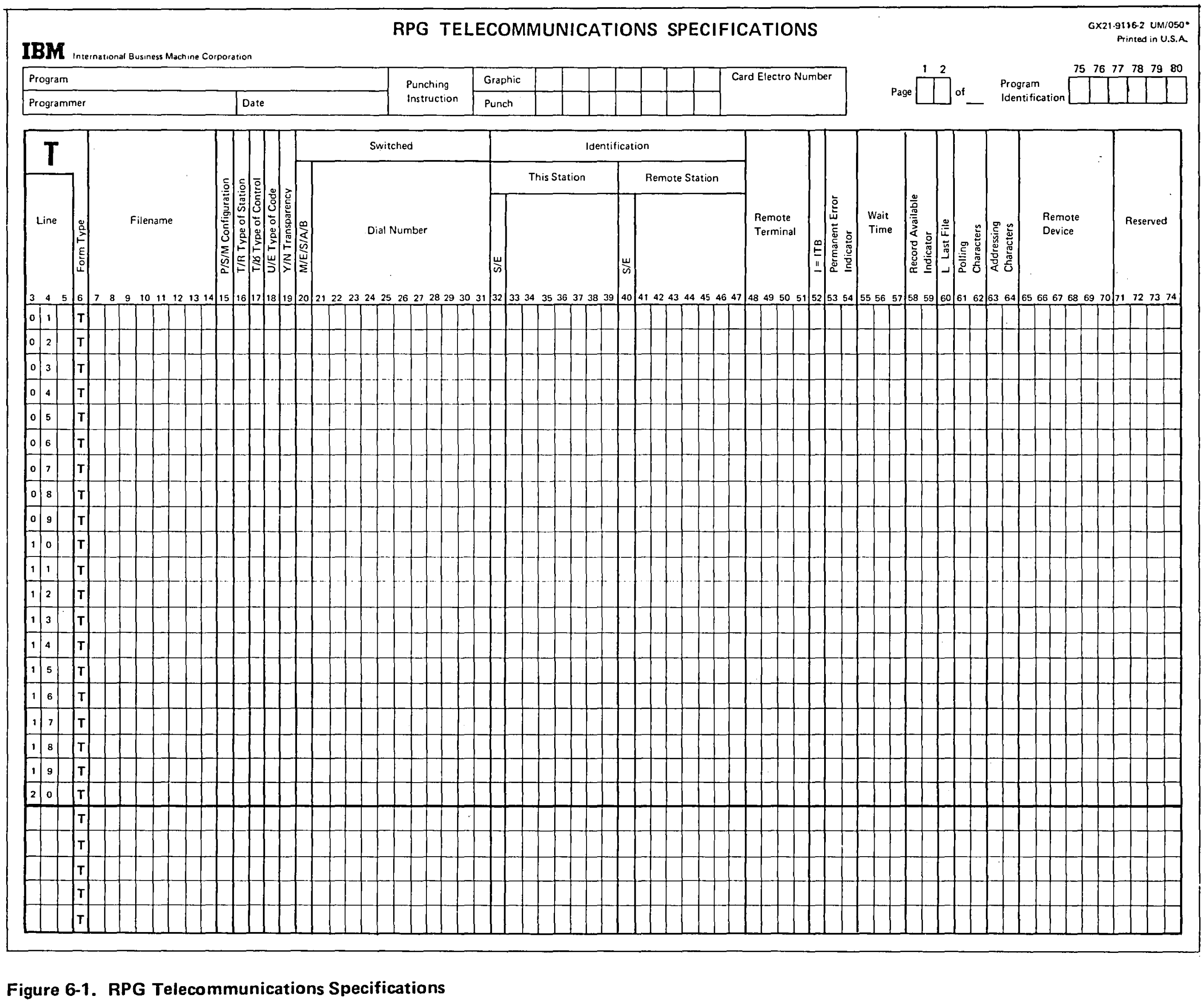
Chapter 6. Telecommunications Specifications

Telecommunications specifications describe the information necessary to establish and maintain the BSC (binary synchronous communications) link. Each BSCA file defined on the file description specifications sheet must have a corresponding entry on the RPG Telecommunications Specifications sheet (see Figure 6-1).

RPG II data communications programming enables you to transmit and receive binary synchronous data via a data communications network. RPG II data communications support performs all the functions necessary to establish the line connections, exchange identification sequences, send and receive data, and execute the correct termination or disconnect procedures.

RPG II permits System/34 to function as any of the following station types:

- Receive only (receive input data from a remote station).
- Transmit only (transmit data to a remote station).
- Transmit and receive, but no conversational reply. Three modes of operation are possible:
  - Transmitting a file, then receiving another file
  - Receiving a file, then transmitting another file
  - Transmitting records of one file interspersed with receiving records of another file



RPG TELECOMMUNICATIONS SPECIFICATIONS

IBM International Business Machine Corporation

GX21-9116-2 UM/050*
Printed in U.S.A.

Program

Programmer

Date

Punching Instruction

Graphic

Punch

Card Electro Number

Page (1 2) of ___

Program Identification (75 76 77 78 79 80)

T

								Switched		Identification													
										This Station		Remote Station											
Line	Form Type	Filename	P/S/M Configuration	T/R Type of Station	T/ƀ Type of Control	U/E Type of Code	Y/N Transparency	M/E/S/A/B	Dial Number	S/E		S/E		Remote Terminal	I = ITB	Permanent Error Indicator	Wait Time	Record Available Indicator	L Last File	Polling Characters	Addressing Characters	Remote Device	Reserved
3 4 5	6	7 8 9 10 11 12 13 14	15	16	17	18	19	20	21 22 23 24 25 26 27 28 29 30 31	32	33 34 35 36 37 38 39	40	41 42 43 44 45 46 47	48 49 50 51	52	53 54	55 56 57	58 59	60	61 62	63 64	65 66 67 68 69 70	71 72 73 74
0 1	T																						
0 2	T																						
0 3	T																						
0 4	T																						
0 5	T																						
0 6	T																						
0 7	T																						
0 8	T																						
0 9	T																						
1 0	T																						
1 1	T																						
1 2	T																						
1 3	T																						
1 4	T																						
1 5	T																						
1 6	T																						
1 7	T																						
1 8	T																						
1 9	T																						
2 0	T																						
	T																						
	T																						
	T																						
	T																						
	T																						

Figure 6-1. RPG Telecommunications Specifications

BSC is a flexible form of line control that provides a set of rules for communications between devices. For a description of the basic characteristics and operational concepts of BSC, a description of the RPG II interface to BSC, and a complete description of RPG II data communications programming, see the *Data Communications Reference Manual.*

Note: Telecommunications specifications are used only for RPG II data communications programming. Telecommunications specifications are not used for the Interactive Communications Feature (SSP-ICF).

COLUMNS 1-2 (PAGE)

See *Common Entries* in Chapter 1.

COLUMNS 3-5 (LINE)

See *Common Entries* in Chapter 1.

COLUMN 6 (FORM TYPE)

A T must appear in column 6 to identify this line as a telecommunications specifications statement.

COLUMNS 7-14 (FILENAME)

Entry	Explanation
A valid filename	Filename previously defined on the file description specifications for the BSCA device

COLUMN 15 (CONFIGURATION)

Entry	Explanation
P or blank	This is a point-to-point nonswitched network.
M	This is a multipoint network where the control station selects the tributary station through polling or addressing. System/34 cannot be the control station.
S	This is a point-to-point switched network.

If this column contains an M, column 17 must contain a T.

COLUMN 16 (TYPE OF STATION)

Entry	Explanation
T	This station transmits messages from the file named in columns 7 through 14. The file must be designated as an output file on the file description specifications and must be defined on the output specifications.
R	This station receives messages in the file named in columns 7 through 14. The file must be designated as an input file on the file description specifications and must be defined on the input specifications.

Note: This entry is independent of the entry in column 20.

COLUMN 17 (TYPE OF CONTROL)

Entry	Explanation
Blank	Polling is not used.
T	This is a tributary station on a multipoint network. Column 17 must contain a T if column 15 contains an M.

System/34 cannot be the control station.

COLUMN 18 (TYPE OF CODE)

Entry	Explanation
A or U	ASCII (formerly referred to as USASCII) transmission control characters are used. An A or U entry causes the necessary file translation to be done for System/34.
E or blank	EBCDIC transmission control characters are used.

ASCII and EBCDIC characters are listed in the *Data Communications Reference Manual.*

If your BSC program halts because of an invalid ASCII character in your data, check your data and the ASCII translation table.

COLUMN 19 (TRANSPARENCY)

Entry	Explanation
Y	EBCDIC transparency is used. The data being transferred can contain transmission control characters and/or packed numeric or alphameric characters. Column 18 must be E or blank.
N or blank	EBCDIC transparency is not used. Zoned decimal numeric or alphameric data is transmitted and received. The data being transferred cannot contain transmission characters.

COLUMN 20 (SWITCHED)

Entry	Explanation
Blank	This is not a switched network.
M	The operator using this program makes the connection by dialing the number (manual dial).
A	This program uses autoanswer.
B	This program uses manual answer.

Notes:

1. This entry is independent of the entry in column 16.
2. For a detailed explanation of how the initial line connection is made, see *Network Control* in the *Data Communications Reference Manual.*

COLUMNS 21-31

Columns 21 through 31 are not used. Leave them blank.

COLUMN 32 (LOCATION OF IDENTIFICATION–THIS STATION)

Entry	Explanation
Blank	This is a nonswitched or switched network, and no identification is used for this station.
S	This is a switched network. This station's identification is at the position specified by the symbolic name in columns 33 through 39.
E	This is a switched network. The entry in columns 33 through 39 is this station's identification.

COLUMNS 33-39 (IDENTIFICATION–THIS STATION)

Entry	Explanation
Alphameric characters	When column 32 contains an E, this entry is the actual identification sequence of this station (minimum of 2 characters). When column 32 contains an S, this entry is the symbolic name of the location of this station's identification.

If columns 33 through 39 contain a symbolic name, it must not be an array name. If the BSCA file is a primary or secondary file, the symbolic name must refer to the first entry of a table (the table might have only one entry) to ensure that the station identification is in storage before the communications line is open.

The station identification referred to by the symbolic name can be from 2 to 15 characters long, but it must not contain a transmission control character.

The station identification is translated if the BSCA files are translated.

COLUMN 40 (LOCATION OF IDENTIFICATION–REMOTE STATION)

Entry	Explanation
Blank	This is a nonswitched or switched network, and no identification is used for the remote station.
S	This is a switched network. The remote station's identification is at the position specified by the symbolic name in columns 41 through 47.
E	This is a switched network. The entry in columns 41 through 47 is the remote station's identification.

COLUMNS 41-47 (IDENTIFICATION–REMOTE STATION)

Entry	Explanation
Alphameric characters	When column 40 contains an E, this entry is the actual identification sequence of the remote station (minimum of 2 characters). When column 40 contains an S, this entry is the symbolic name of the location of the remote station's identification.

If columns 41 through 47 contain a symbolic name, it must not be an array name. If the BSCA file is a primary or secondary file, this symbolic name must refer to the first entry in a table (the table might have only one entry) to ensure that the station identification is in storage before the communications line is open.

The station identification referred to by the symbolic name can be from 2 to 15 characters long, but must not contain a transmission control sequence character. The station identification is translated if the BSCA files are translated.

The identification received from the remote station is compared with this entry. The session continues only if the identification matches.

COLUMNS 48-51

Columns 48 through 51 are not used. Leave them blank.

COLUMN 52 (ITB)

Entry	Explanation
Blank	Intermediate block checking is not used.
I	Intermediate block checking is used.

Intermediate block checking (ITB) can be used only if the records are blocked. ITB and EBCDIC transparency cannot both be specified for the same BSCA output file.

COLUMNS 53-54 (PERMANENT ERROR INDICATOR)

Entry	Explanation
Blank	No permanent error indicator is specified. If a permanent error occurs, a system halt occurs and the program cannot be restarted.
01-99, L1-L9, LR, H1-H9	A permanent error indicator can be specified for every BSCA file. If there is more than one BSCA file, each file can have a permanent error indicator. The indicator does not have to be unique for each file. Specifying a permanent error indicator is recommended when the system is running in an unattended mode.

Use columns 53 and 54 to specify a permanent error indicator for every BSCA file. When a permanent error occurs, the specified error indicator and the identification indicator of the record causing the error turn on (however, no hardware diagnostics are performed). The permanent error indicator can then be used to condition the appropriate programming response, such as printing a message or performing a controlled cancel.

Do not attempt further transmission while the permanent error indicator is on. This includes attempts to transmit more than one record during detail, total, or exception output. Further transmission can be prevented if each record to be transmitted is conditioned with the not-permanent-error indicator in columns 9 through 11 of the calculation specifications or columns 23 through 31 of the output specifications.

To retry an operation after a permanent error occurs, turn off the permanent error indicator. The RPG II program can then access the BSCA file on which the error occurred. If an error occurs on the retried operation, the permanent error indicator is turned on again; otherwise, processing continues.

Keep the following points in mind when retrying an operation:

- The permanent error indicator is the only indication to the RPG II program that an error occurred. A BSCA information message describing the type of error is displayed. If a halt (H1 through H9) is not issued as part of the permanent error routine, the BSCA information message may not be preserved on the display screen.
- Any data in the BSCA buffers at the time of an error is lost. The record in your buffers is not the same as the record in the BSCA buffers. Therefore, retrying the last operation will still result in lost data.
- Switched lines are not disabled when an error occurs unless a disconnect sequence is received or the hardware detects disconnect.
- Any data transmitted while the permanent error indicator is on is invalid. Unless your program is designed to recognize all data, the error condition can cause an unidentified record halt.
- A limit should be imposed by the RPG II program on the number of times an error can occur before the program is stopped.

Note: Avoid using H1 through H9 as permanent error indicators if you are going to condition operations on the permanent error indicator being off. Because H1 through H9 are reset at the end of the detail logic cycle, they can be set off before the program cycle in which the error occurred is completed. If H1 through H9 is used as a permanent error indicator, the H1 through H9 display can preempt the system halt display. If the H1 through H9 display appears before the system display, the operator should take the 0 option to prompt the system halt display.

COLUMNS 55-57 (WAIT TIME)

Entry	Explanation
Blank	The system convention for time-out, 180 seconds, is used.
Numeric	The length in time in seconds, 1 to 999, that BSC waits with no messages being sent or received before a permanent error occurs.

A permanent error indication is recognized by the system whenever the wait time on an idle line elapses. Therefore, when determining the wait time, consider the time the operator might require to respond to halts and other processing interruptions, and also time the program might require for special operations such as table searches and computing square roots.

The wait time limit specified applies only to delays caused by the System/34 program and does not apply to the remote device. In addition, the time limit applies only during the transmission or reception of a file, not between file transmissions. The occurrence of a permanent error indicates the end of processing of a file, but not the end of file.

COLUMNS 58-59 (RECORD AVAILABLE INDICATOR)

Entry	Explanation
01-99, L1-L9, LR, H1-H9	A record available indicator should be specified if a reverse interrupt (RVI is to be received. See *Device Dependent Considerations* in the *Data Communications Reference Manual* for examples of using a record available indicator. This indicator turns on whenever a reverse interrupt (RVI) is received.

COLUMN 60 (LAST FILE)

Entry	Explanation
Blank	This BSCA file may not be the last input file processed.
L	This BSCA file is processed only after all other input files are processed.

The entry in column 60 does not affect demand files.

COLUMNS 61-62 (POLLING CHARACTERS)

Entry	Explanation
Blank	This station is not transmitting on a multipoint network.
Alphameric characters	The polling identification of this station is required if this station is part of a multipoint network and the BSCA file is a transmit (output) file. Polling and addressing characters must be used in pairs as listed in the *Data Communications Reference Manual.*

COLUMNS 63-64 (ADDRESSING CHARACTERS)

Entry	Explanation
Blank	This station is not receiving on a multipoint network.
Alphameric character	The addressing identification of this station is required if this station is part of a multipoint network and the BSCA file is a receive (input) file. Polling and addressing characters must be used in pairs as listed in the *Data Communications Reference Manual.*

Enter polling and addressing characters in EBCDIC; the compiler converts the characters to the form required by the code specified in column 18. (If ASCII was specified, enter uppercase addressing characters; they are converted to lowercase ASCII characters.) See *Polling and Addressing Characters* in the *Data Communications Reference Manual.*

COLUMNS 65-74

Columns 65 through 74 are not used. Leave them blank.

COLUMNS 75-80 (PROGRAM IDENTIFICATION)

See *Common Entries* in Chapter 1.

FILE DESCRIPTION SPECIFICATIONS FOR BSCA FILES

The following entries are used on the file description specifications to define a BSCA file. Entries for the columns not listed are described in Chapter 3, *File Description Specifications.*

The only other specification sheet affected by a BSCA file is the control specifications. Because a BSC program must not be interrupted, a B must not be entered in column 37 of the control specifications.

Columns 7-14 (Filename)

Enter the name of the BSCA file. The same filename must be used on the telecommunications specifications.

Note: Look-ahead fields must not be specified for a BSCA file.

Column 15 (File Type)

Entry	Explanation
I	This is an input (receive) file.
O	This is an output (transmit) file.

Column 16 (File Designation)

The entries for this column are the same as those described in Chapter 3 except that:

- D (demand file) is the required entry for transmit interspersed with receive. BSCA files should also be designated as demand files for any receiving program that does not address the BSCA files immediately. For example, if the BSCA file is defined as a secondary file, the communications line opens as soon as the program begins. This means that your wait time might elapse before you are ready to process the BSCA file. If the BSCA file is defined as a demand file, however, the line opens once the program is ready to receive the first record from the BSCA file.
- R (record address file) is an invalid entry. A BSCA file cannot be a record address file.

Column 17 (End of File)

Enter an E if end of file on the input (receive) file is to determine end of job. The BSCA input file might be the only file with an E in column 17. However, if any other input file has an E in column 17, all BSCA input files should also have an E in column 17. This E is not necessary for the BSCA files; but when it is not specified and end of file is reached on another input file, the BSCA files close and the system on the other end of the communications line has no indication of what has happened. When an E is specified for the BSCA files, all systems can come to a successful end of job.

Column 19 (File Format)

Enter an F (fixed length) for BSCA files.

Columns 20-23 (Block Length)

Enter the size of the blocks of data processed by BSC. The block length must be a multiple of the record length. The maximum block length is 4,075 bytes.

Columns 24-27 (Record Length)

Enter the length of the BSCA records, right-justified. If the record length is not specified, it defaults to the maximum record length, which is 4,075 bytes.

If a record that has data of 0 length is received, it is ignored unless 3740 mode is used, in which case it is considered a file separator. If your program receives a record that has a length greater than 0 but shorter than the record size specified, the remainder of the record contains unpredictable characters.

Column 32 (File Organization or Addition I/O Area)

Assign dual input/output areas in this column. Any number, 1 through 9, assigns two input/output areas. If this column is blank, only one input/output area is assigned and throughput is decreased accordingly.

Columns 40-46 (Device)

Entry	Explanation
BSCA	This is the device entry for BSCA files.

For data communications programming considerations, see the *Data Communications Reference Manual.*

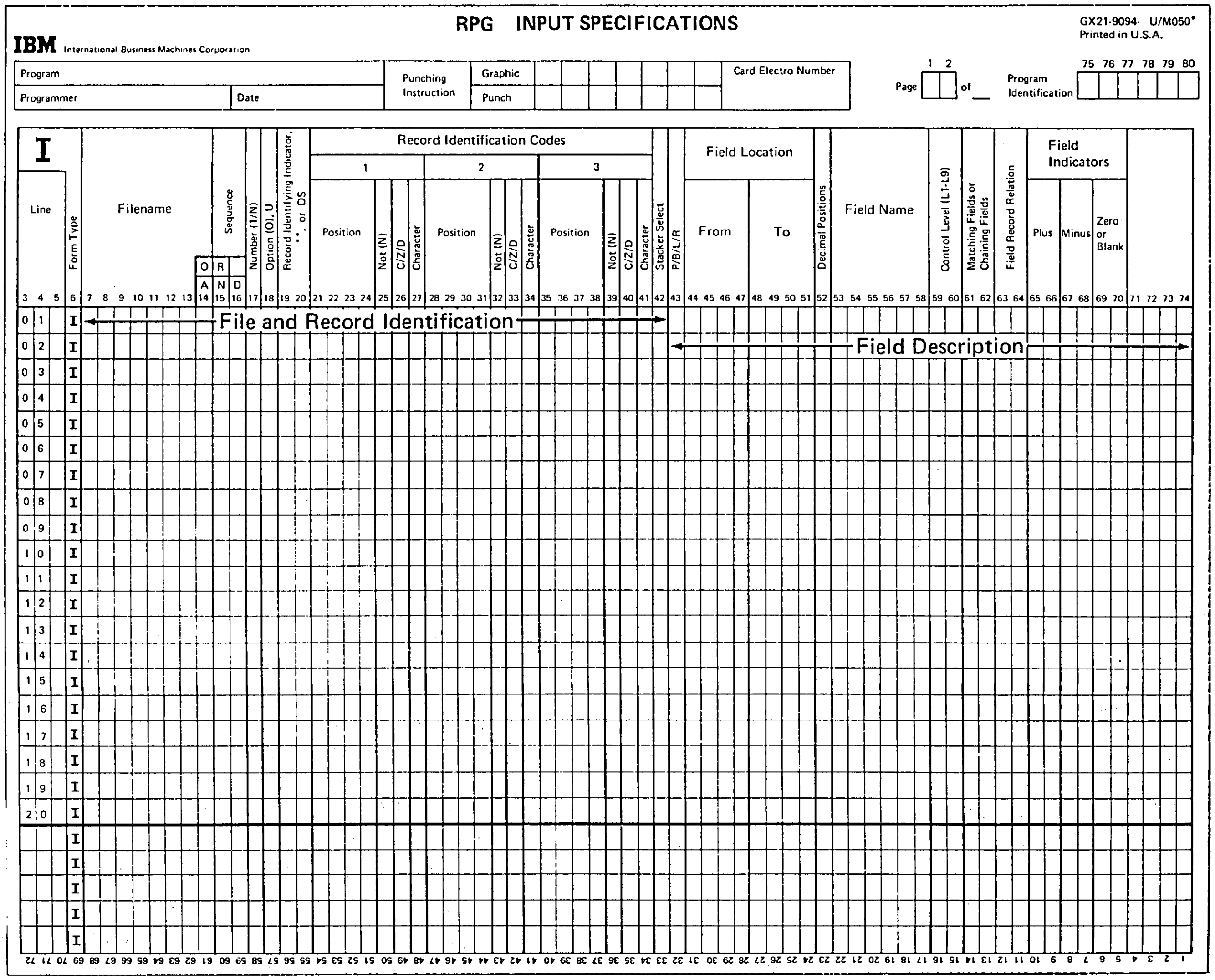
Chapter 7. Input Specifications

Input specifications describe the data files, records, and fields of the records used in the program. All input files are described on the input specifications except files assigned to the device KEYBORD, record address files, addrout files, and table files. KEYBORD files are described on the calculation specifications when the KEY operation code is used. Record address files, addrout files, and table files are described on the extension specifications.

The input specifications are divided into two categories:

- File and record-type identification entries (columns 7 through 42) describe the input record and its relationship to other records in the file.
- Field description entries (columns 43 through 74) describe the fields in the records. These specifications must start on the line below the file and record-type identification specifications.

Write these specifications on the RPG Input Specifications sheet (see Figure 7-1).



IBM International Business Machines Corporation

RPG INPUT SPECIFICATIONS

GX21-9094- U/M050*
Printed in U.S.A.

Program
Programmer
Date
Punching Instruction
Graphic
Punch
Card Electro Number
Page 1 2 of
Program Identification 75 76 77 78 79 80

I

Line 3 4 5 | Form Type 6 | Filename 7 8 9 10 11 12 13 | Sequence 14 15 (OR / AND) | Number (1/N) 16 | Option (O), U 17 | Record Identifying Indicator, **, or DS 18 19 20

Record Identification Codes: 1 — Position 21 22 23 24, Not (N) 25, C/Z/D 26, Character 27; 2 — Position 28 29 30 31, Not (N) 32, C/Z/D 33, Character 34; 3 — Position 35 36 37 38, Not (N) 39, C/Z/D 40, Character 41

Stacker Select 42 | P/B/L/R 43 | Field Location: From 44 45 46 47, To 48 49 50 51 | Decimal Positions 52 | Field Name 53 54 55 56 57 58 | Control Level (L1-L9) 59 60 | Matching Fields or Chaining Fields 61 62 | Field Record Relation 63 64 | Field Indicators: Plus 65 66, Minus 67 68, Zero or Blank 69 70 | 71 72 73 74

0 1 I ← File and Record Identification →
0 2 I ← Field Description →
0 3 I
0 4 I
0 5 I
0 6 I
0 7 I
0 8 I
0 9 I
1 0 I
1 1 I
1 2 I
1 3 I
1 4 I
1 5 I
1 6 I
1 7 I
1 8 I
1 9 I
2 0 I
I
I
I
I
I

Figure 7-1. RPG Input Specifications

File and Record-Type Identification Entries

COLUMNS 1-2 (PAGE)

See *Common Entries* in Chapter 1.

COLUMNS 3-5 (LINE)

See *Common Entries* in Chapter 1.

COLUMN 6 (FORM TYPE)

An I must appear in column 6 to identify this line as an input specifications statement.

COLUMNS 7-14 (FILENAME)

Entry	Explanation
A valid filename or data structure name	Same filename that appears on the file description specifications for the input file or the name of a data structure.

If a data structure is specified (DS in columns 19 and 20), columns 7 through 14 can contain:

- Blanks
- A name up to 6 characters long
- A name previously referenced in columns 53 through 58 of the input specifications

Data structure entries must be the last entries on the input specifications.

COLUMNS 14-16

Entry	Explanation
AND or OR	AND/OR indicates a relationship between record identifying indicators or record types. The entry must begin in column 14.

See *Columns 21-41 (Record Identification Codes)* and *Columns 53-58 (Field Name)* in this chapter for more information on AND/OR relationship.

COLUMNS 15-16 (SEQUENCE)

Entry	Explanation
Any two alphabetic characters	Program does not check for special sequence. Alphabetic characters must be used for chained files, demand files (except CONSOLE demand files), WORKSTN files, and look-ahead records.
01-99	Program checks for special sequence.

Use a numeric entry (01 through 99) in columns 15 and 16 to assign a special sequence to different record types in a file. The first sequence number must be 01. Gaps in sequence numbers are allowed, but the numbers must be in ascending order.

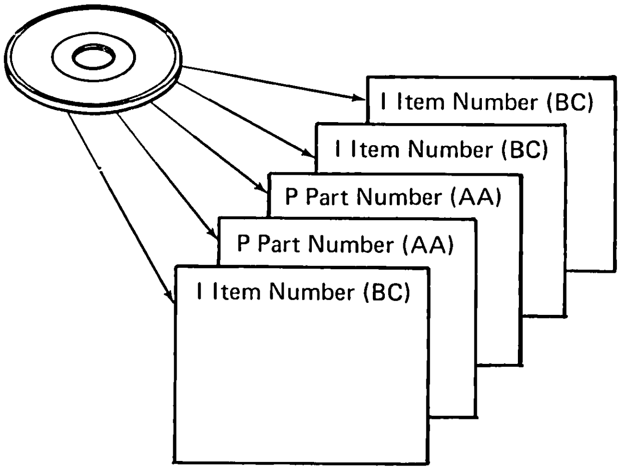
If the types of records do not need to be in any special order, use two alphabetic characters (see Figure 7-2). Within one file, all record types having alphabetic entries in columns 15 and 16 must be described before those types with numeric entries.

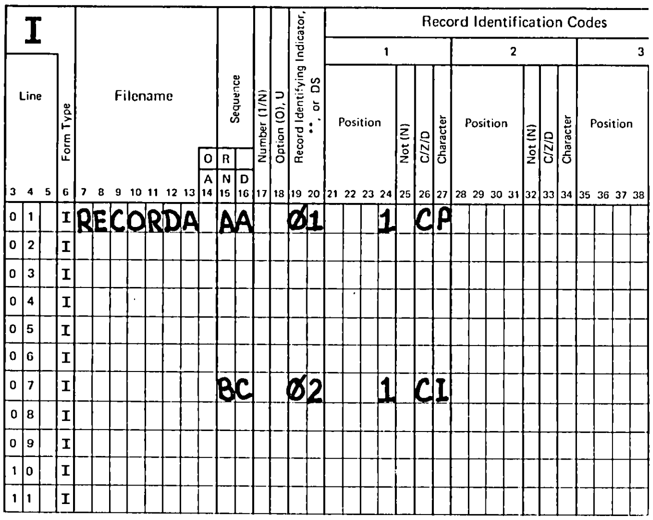
Assigning Sequence Numbers

Enter a numeric character in columns 15 and 16 if one record type (identified by a record identification code) must be read before another record type in a sequenced group. To specify sequence checking, each record type must have a record identification code, and the record types must be numbered in the order they should appear. The program checks this order as the records are read (see Figure 7-3). If a record type is out of sequence, the program stops. The operator can restart the program by selecting the appropriate option and pressing an entry function key. The program bypasses the record that caused the halt and reads the next record from the same file.

Sequence numbers ensure only that all records of the lowest record type precede the records of the next highest record type. The sequence numbers do not ensure that records within a record type are in any certain order. Sequence numbers are unrelated to control levels and do not provide for checking data in fields of a record for a special sequence (see Figure 7-4). Use columns 61 and 62 to indicate that data in fields of a record be checked for a special sequence.

Records in an OR or AND line cannot have a sequence entry in these columns. The entry in columns 15 and 16 on the previous line also applies to the OR or AND line. See *Columns 53-58 (Field Name)* in this chapter for information on OR relationships.

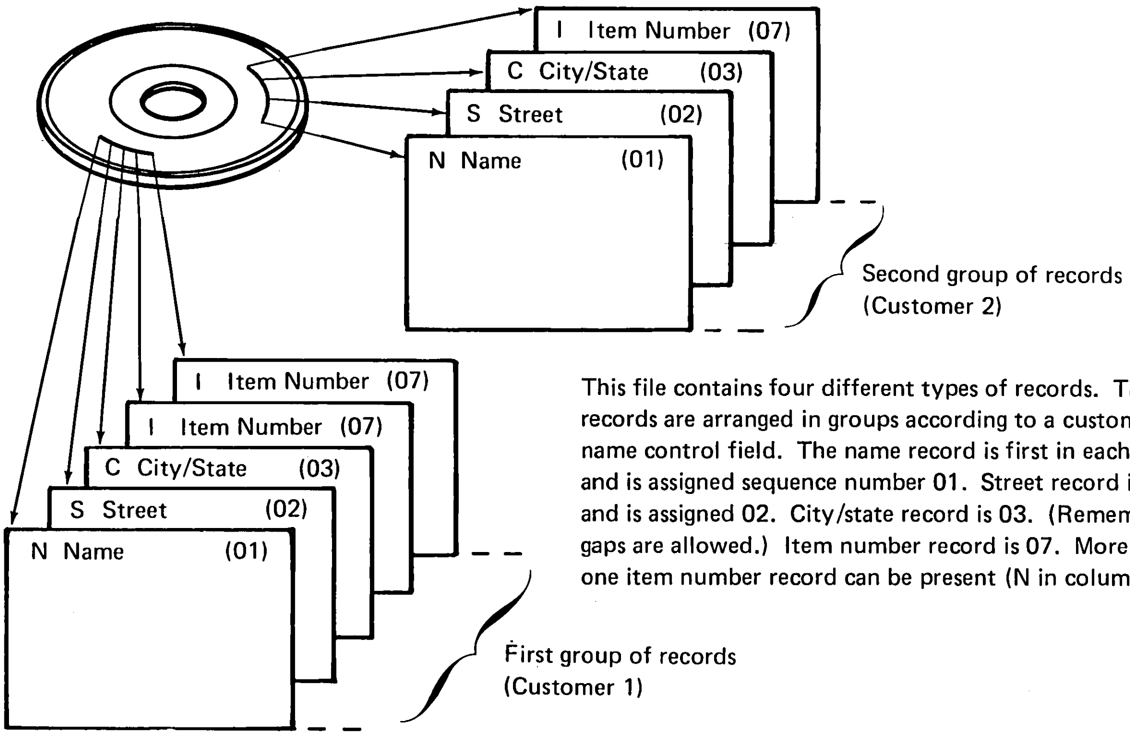




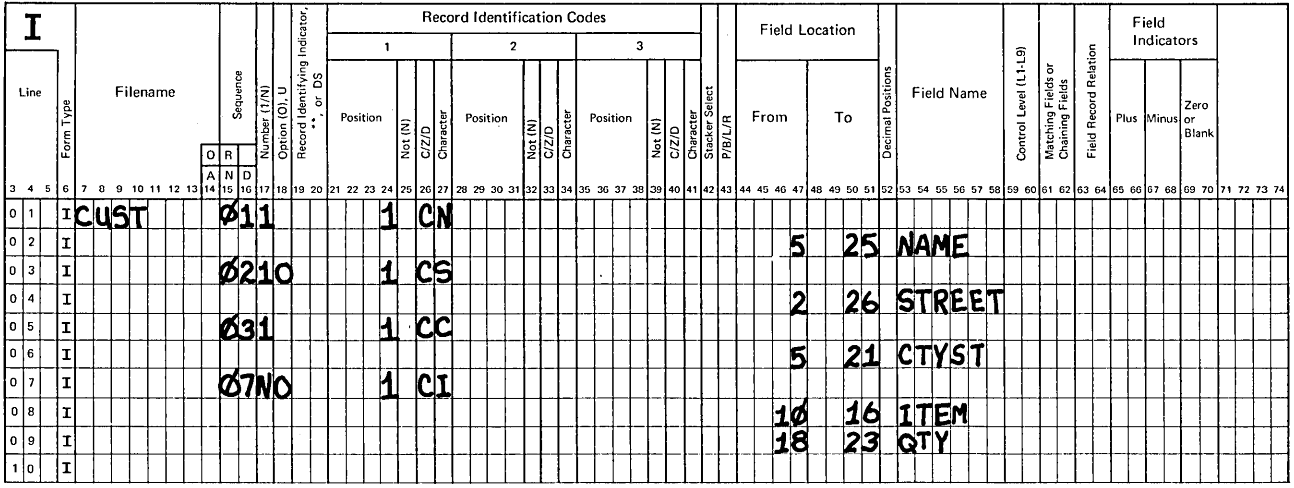
I							Record Identification Codes								
							1				2				3
Line	Form Type	Filename	Sequence	Number (1/N)	Option (O), U	Record Identifying Indicator, **, or DS	Position	Not (N)	C/Z/D	Character	Position	Not (N)	C/Z/D	Character	Position
		O R A N D													
3 4 5	6	7 8 9 10 11 12 13 14	15 16	17	18	19 20	21 22 23 24	25	26	27	28 29 30 31	32	33	34	35 36 37 38
0 1	I	RECORDA	AA			Ø1	1		C	P					
0 2	I														
0 3	I														
0 4	I														
0 5	I														
0 6	I														
0 7	I		BC			Ø2	1		C	I					
0 8	I														
0 9	I														
1 0	I														
1 1	I														

File RECORDA has two types of records (part number and item number) that can appear in any order. Because they are not to be checked for sequencing, they are assigned 2 alphabetic characters in columns 15 and 16 (AA and BC, respectively) instead of numbers.

Figure 7-2. Unsequenced Record Types in a File

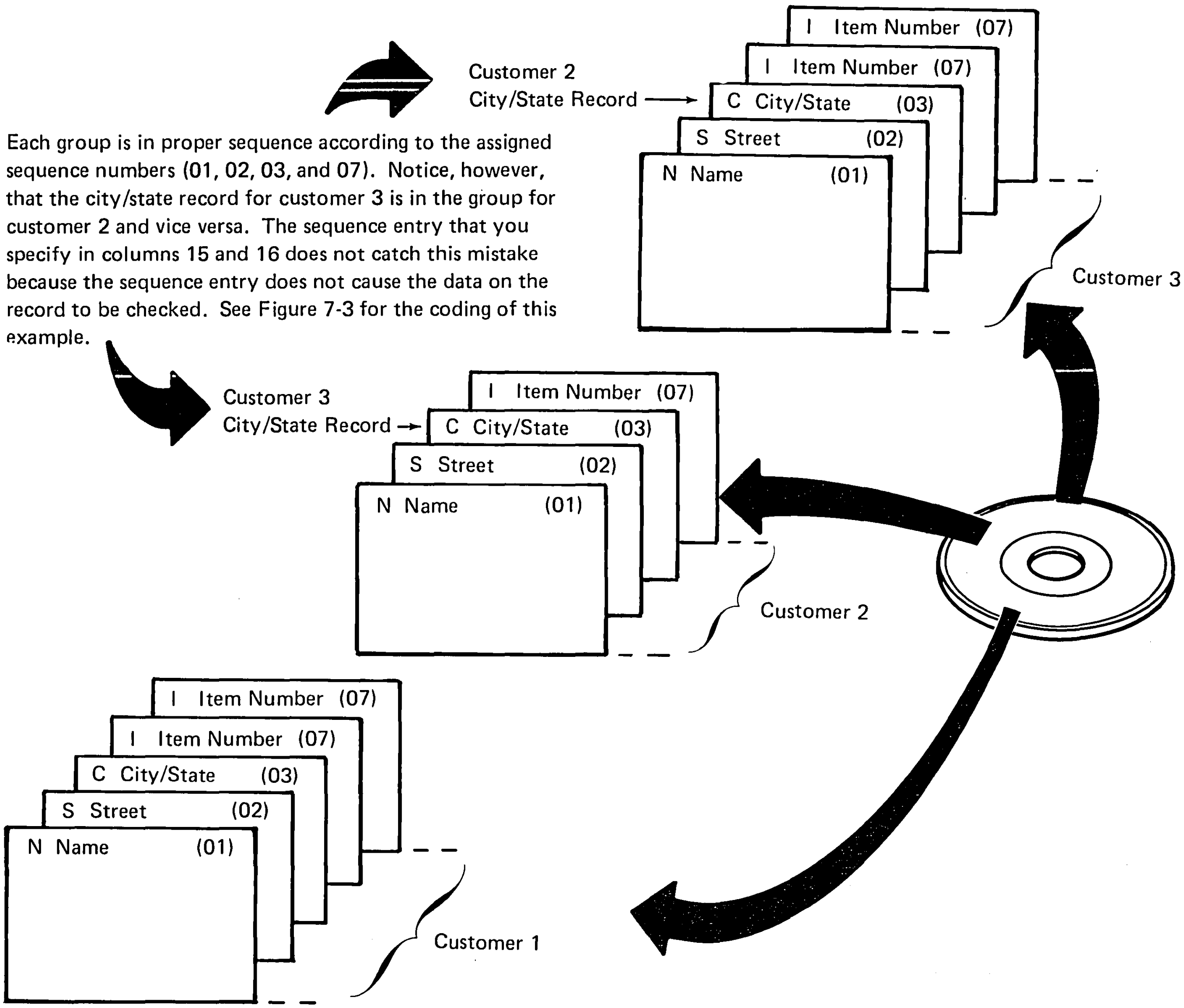


This file contains four different types of records. The records are arranged in groups according to a customer name control field. The name record is first in each group and is assigned sequence number 01. Street record is next and is assigned 02. City/state record is 03. (Remember gaps are allowed.) Item number record is 07. More than one item number record can be present (N in column 17).



Line (3-5)	Form Type (6)	Filename (7-14)	Sequence (15-16)	Number (1/N) (17)	Option (O), U (18)	Record Identifying Indicator, **, or DS (19-20)	1 Position (21-24)	Not (N) (25)	C/Z/D (26)	Character (27)	2 Position (28-31)	Not (N) (32)	C/Z/D (33)	Character (34)	3 Position (35-38)	Not (N) (39)	C/Z/D (40)	Character (41)	Stacker Select (42)	P/B/L/R (43)	Field Location From (44-47)	Field Location To (48-51)	Decimal Positions (52)	Field Name (53-58)	Control Level (L1-L9) (59-60)	Matching Fields or Chaining Fields (61-62)	Field Record Relation (63-64)	Field Indicators Plus (65-66)	Field Indicators Minus (67-68)	Field Indicators Zero or Blank (69-70)	(71-74)
01	I	CUST	Ø1	1			1		C	N																					
02	I																				5	25		NAME							
03	I		Ø2	1	O		1		C	S																					
04	I																				2	26		STREET							
05	I		Ø3	1			1		C	C																					
06	I																				5	21		CTYST							
07	I		Ø7	N	O		1		C	I																					
08	I																				1Ø	16		ITEM							
09	I																				18	23		QTY							
10	I																														

Figure 7-3. Sequence Checking of Record Type



Each group is in proper sequence according to the assigned sequence numbers (01, 02, 03, and 07). Notice, however, that the city/state record for customer 3 is in the group for customer 2 and vice versa. The sequence entry that you specify in columns 15 and 16 does not catch this mistake because the sequence entry does not cause the data on the record to be checked. See Figure 7-3 for the coding of this example.

Figure 7-4. Correct Record Sequence (Incorrect Data Within Groups)

COLUMN 17 (NUMBER)

Entry	Explanation
Blank	Program does not check record types for a special sequence (columns 15 and 16 have alphabetic entries).
1	Only one record of this type can be present in the sequenced group.
N	One or more records of this type can be present in the sequenced group.

Use column 17 only if columns 15 and 16 contain a numeric entry specifying sequence checking (see Figure 7-5).

OR lines (columns 14 and 15 contain OR) and AND lines (columns 14 through 16 contain AND) should not have an entry in this column. The entry in column 17 on the previous line also applies to the OR or AND line. See *Columns 53-58 (Field Name)* in this chapter for more information on OR lines.

COLUMN 18 (OPTION)

Entry	Explanation
Blank	Record type must be present if sequence checking is specified.
O	Record type is optional (that is, it may or may not be present) if sequence checking is specified.
U	The program uses the data structure defined on this specification line as a display station local data area.

Use column 18 only if columns 15 and 16 contain a numeric entry specifying sequence checking, or if the data structure defined on the following specification line is used as a display station local data area.

If sequence checking is specified and all record types are optional, no sequence error is found.

OR and AND lines should not have an entry in this column. The entry in column 18 on the previous line also applies to the OR or AND line. See *Columns 53-58 (Field Name)* in this chapter for more information on OR lines.

Display Station Local Data Area

To define a display station local data area for your program, enter a U in column 18 and define a data structure with a maximum of 256 characters (see Figure 7-6). See *Columns 19-20 (Record Identifying Indicator, **, DS)* for information on defining a data structure.

A local data area is provided for each command display station on System/34. The local data area for the requestor of a single requestor program is read into the RPG II display station local data area at the beginning of the program and written out automatically at the end of the program. This local data area allows data to be passed from program to program or from job to job for the requesting display station. Use the LOCAL OCL statement or another RPG II program to enter information into the display station local data area (see the *System Support Reference Manual*). The program can change this data and update the display station local data area at end of job.

For a multiple requestor program, the display station local data area (defined with U in column 18) contains a copy of the local data area for the first requestor only. Any modifications specified by LOCAL OCL statements are made to this copy of the local data area, not to the requestor's local data area. This copy of the local data area is read into the display station local data area at the beginning of the program, but it is not automatically written out at the end of job. To read and write the local data area for each requestor in a multiple requestor program, use SUBR21 (see *IBM-Written Subroutines SUBR20 and SUBR21* in Chapter 13, *WORKSTN File Considerations and Sample Programs*).

COLUMNS 19-20 (RECORD IDENTIFYING INDICATOR, **, DS)

Entry	Explanation
01-10	Record identifying indicator for CONSOLE files. Record identifying indicators 01 through 10 for CONSOLE files correspond to command keys 1 through 10.
01-99	Record identifying indicator.
L1-L9	Control level indicator used for a record identifying indicator when a record type rather than a control field signals the start of a new control group.
LR	Last record indicator.
H1-H9	Halt indicator used for a record identifying indicator when checking for a record type that causes an error condition.
**	Look-ahead field. Look-ahead can be used only with input or update files; however, these files cannot be chained or demand files. Look-ahead fields are not valid with CONSOLE files or WORKSTN files.
DS	Data structure. A data structure is considered to be alphameric data and can be from 1 to 9999 characters in length. Data structure entries must be the last entries on the input specifications.

Use columns 19 and 20 to:

- Assign a record identifying indicator to each record type in a file. These indicators do not have to be assigned in any order.
- Define the field that is described on the following line in columns 53 through 58 as a look-ahead field.
- Define the same internal area multiple times, subdivide fields, or group fields by declaring the area a data structure.

Record Identifying Indicators

To identify which record type is being processed during a program cycle, assign a unique record identifying indicator to each record type in a file. When a record type is selected for processing, its corresponding indicator turns on and remains on throughout the cycle. Therefore, this indicator can be used to condition calculation and output operations (see *Columns 9-17* in Chapter 8, *Calculation Specifications*, and *Columns 23-31* in Chapter 9, *Output Specifications*) and to associate a field with a particular record type (see *Columns 63-64 (Field Record Relation)* in this chapter).

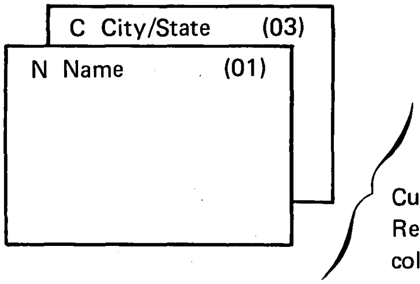
For a WORKSTN file, the first input record is blank unless a read under format is performed (see *Read Under Format* in Chapter 13) or unless program data-yes is specified in the procedure that called the program (see the $MAINT utility program description in the *System Support Reference Manual* or the description of SEU end of job in the *SEU Reference Manual*). This blank record must be identified on the input specifications, and a record identifying indicator must be assigned to that record.

The same indicator can be assigned to two or more different record types if the same operation is to be performed on all record types. To do this, use the OR relationship. See *Columns 21-41 (Record Identification Codes)* for more information. Record identifying indicators for OR lines can also be specified for every record type in the OR relationship that requires special processing. See *Columns 53-58 (Field Name)* for more information.

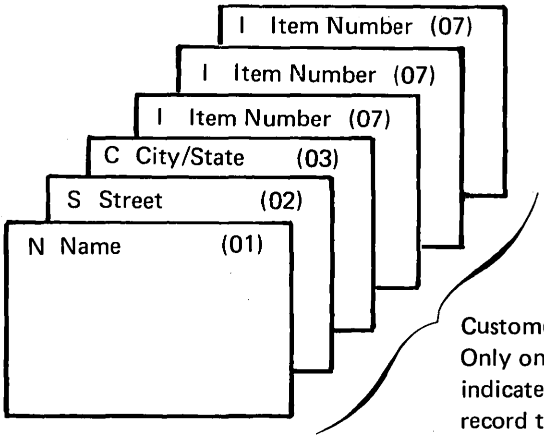
No record identifying indicator can be specified in the AND line of an AND relationship. See *Columns 21-41 (Record Identification Codes)* for more information.

When a control level indicator used as a record identifying indicator turns on to indicate the type of record read, only that one control level indicator turns on. All lower control level indicators remain unchanged.

If RPG telecommunications specifications are used, see *Columns 53-54 (Permanent Error Indicator)* in Chapter 6, *Telecommunications Specifications*, for a description of the permanent error indicator.

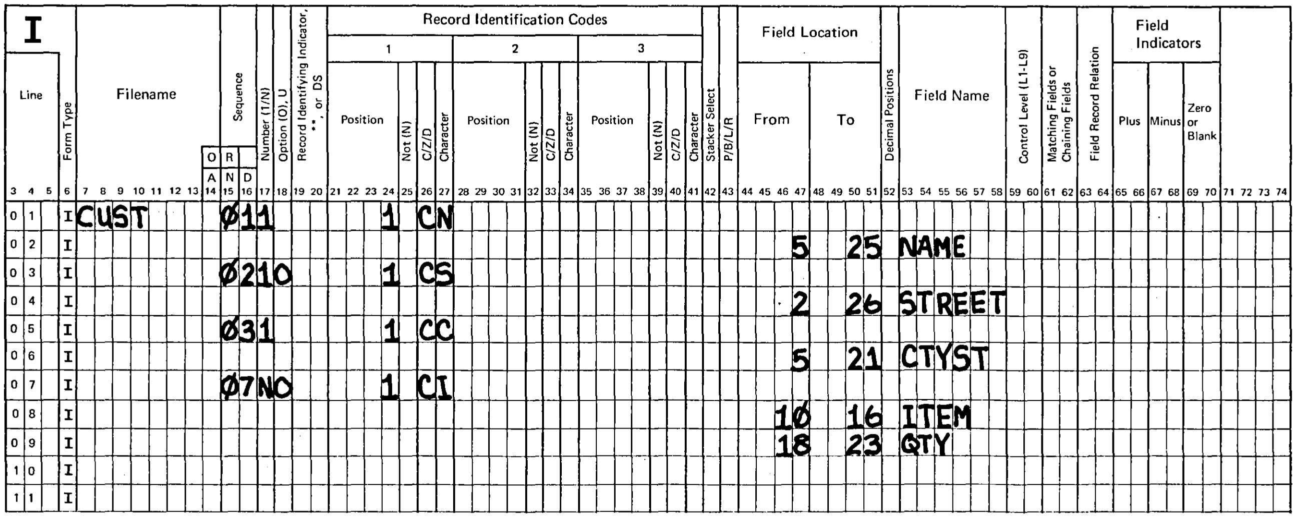


Customer 2
Record types 02 and 07 are optional as indicated by O in column 18.



Customer 1
Only one record of types 01, 02, and 03 can be present as indicated by 1 in column 17; however, any number of record types 07 can be present as indicated by N in column 17.

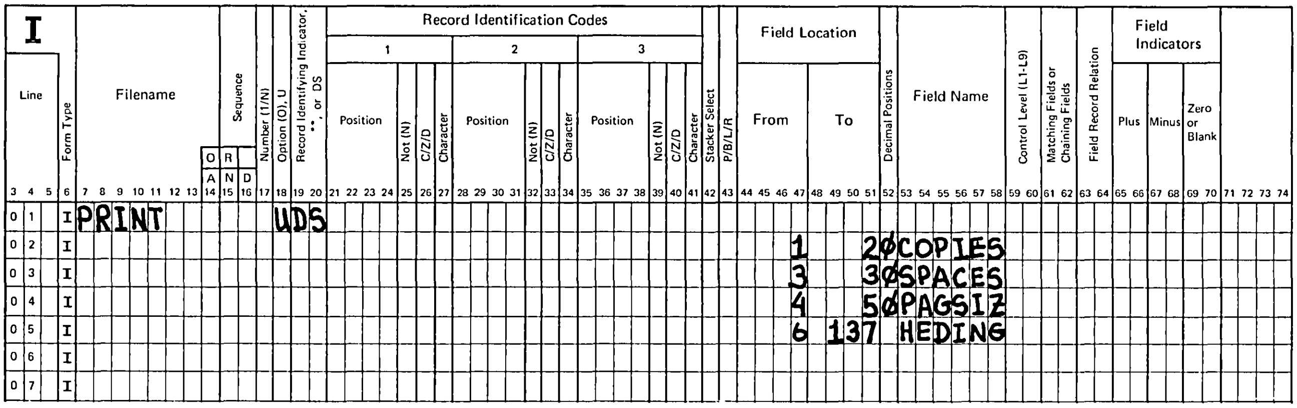
Input Specifications



Line	Form Type	Filename	Sequence	Number (1/N)	Option (O), U	Record Identifying Indicator, **, or DS	Record Identification Codes 1 Position	Not (N)	C/Z/D	Character	Field Location From	Field Location To	Field Name
01	I	CUST	Ø1	1			1		C	N			
02	I										5	25	NAME
03	I		Ø2	1	O		1		C	S			
04	I										2	26	STREET
05	I		Ø3	1			1		C	C			
06	I										5	21	CTYST
07	I		Ø7	N	O		1		C	I			
08	I										1Ø	16	ITEM
09	I										18	23	QTY
10	I												
11	I												

Figure 7-5. Sequenced Record File

Input Specifications



Line	Form Type	Filename	Sequence	Number (1/N)	Option (O), U	Record Identifying Indicator, ** or DS	Record Identification Codes 1 Position	Not (N)	C/Z/D	Character	2 Position	Not (N)	C/Z/D	Character	3 Position	Not (N)	C/Z/D	Character	Stacker Select	P/B/L/R	Field Location From	To	Decimal Positions	Field Name	Control Level (L1-L9)	Matching Fields or Chaining Fields	Field Record Relation	Field Indicators Plus	Minus	Zero or Blank	
01	I	PRINT				UDS																									
02	I																				1	2	Ø	COPIES							
03	I																				3	3	Ø	SPACES							
04	I																				4	5	Ø	PAGSIZ							
05	I																				6	137		HEDING							
06	I																														
07	I																														

The program determines (1) how many copies of the file are to be printed and places this number in COPIES; (2) how the report is to be spaced (single-, double-, or triple-spaced) and places a 1, 2, or 3 in SPACES; and (3) how many lines are to be printed per page and places this number in PAGSIZ. HEDING contains the heading lines to be printed on the top of each page.

Positions 138 through 256 are not used by the program and need not be defined.

Note: Although a data structure name is specified in the example, this entry is optional for data structures used as a display station local data area.

Figure 7-6. Defining a Display Station Local Data Area

Look-Ahead

A look-ahead field allows you to:

- Determine when the last record of a control group is being processed
- Extend the RPG II matching record capability

Because an RPG II program processes one record at a time, normally only the information from the record being processed is available for use. However, the look-ahead function enables information to be made available from records that follow the one currently being processed. This information can then be used to determine what operation should be done next.

Any or all of the fields in a file can be described as look-ahead fields. The description applies to all records in the file regardless of their type. Look-ahead fields can be described before or after the field descriptions for any of the records in the file. The line that signals that look-ahead fields are to be described must contain an alphabetic entry in columns 15 and 16 and must contain ** in column 19 and 20. All the other columns must be blank. Remember that specifications with an alphabetic sequence in columns 15 and 16 must precede specifications with a numeric sequence in columns 15 and 16.

Look-ahead fields are described on the lines immediately following the line that contains ** in column 19 and 20 (see Figure 7-7). Make the following entries for each look-ahead field description line:

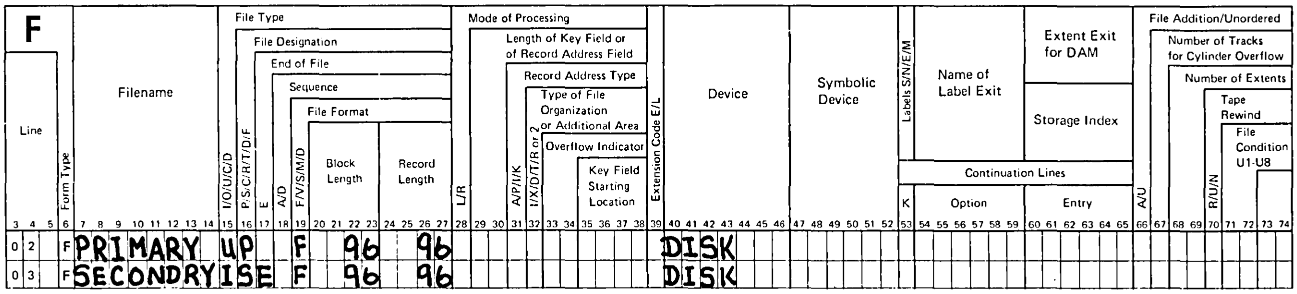
- *Columns 44-51:* Identify the record positions in which the field is located.
- *Column 52:* If the field is numeric, enter the number of digits to the right of the decimal point in column 52. If there are no decimal positions, enter a 0. If the field is alphameric, leave this column blank.
- *Columns 53-58:* Enter the name of the look-ahead field. If the field is also one of the normal fields in the record, use a different name for the look-ahead field.

For input files, look-ahead fields always apply to the next record in the file, provided the file is not an update file. Thus, if the information is used both before and after the record is selected for processing, describe the field twice, once as a look-ahead field and once as a normal field. See Figure 7-8 for an example of how records are selected for processing from two input files when look-ahead fields are used.

For update files, the look-ahead fields apply to the next record in the file only if the record currently selected for processing was read from another file. Therefore, when the program is reading from only one file and that file is an update file, look-ahead fields always apply to the current record and contain the same information as a normal field. See Figure 7-9 for an example of how records are selected for processing from an update file and an input file when look-ahead fields are used.

As the last record from a file is processed, every look-ahead field for the file is automatically filled with 9s. For example, a look-ahead field that is 3 characters long will contain 999. The 9s remain in the field until the job ends. The blank-after option (B in column 39 of the output specifications) cannot be used with look-ahead fields.

File Description Specifications

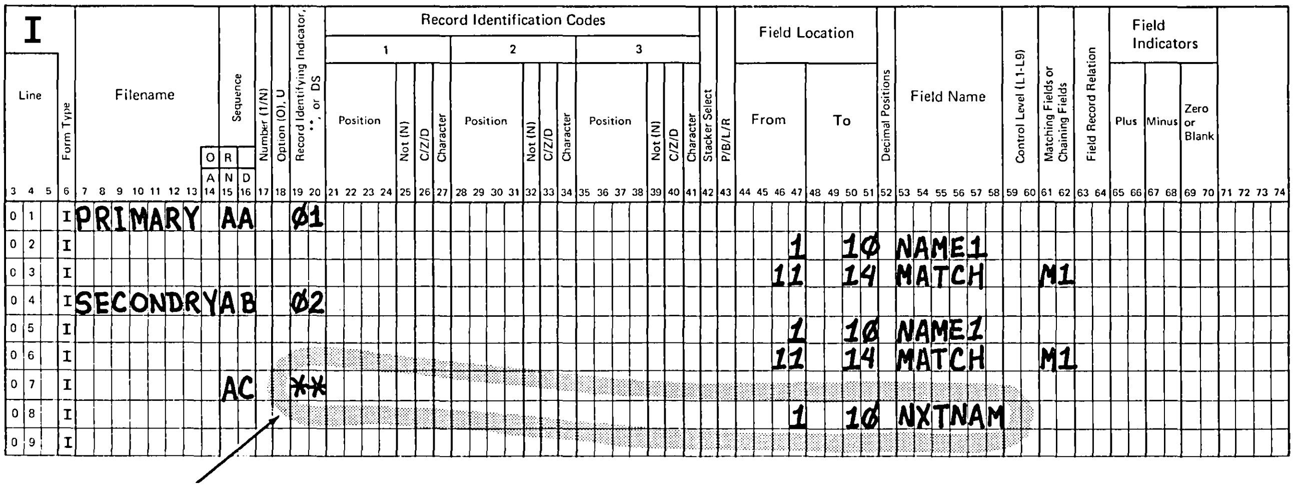


```
  02 FPRIMARY UP  F  96  96            DISK
  03 FSECONDRYISE F  96  96            DISK
```

This program reads records from two disk files. The primary file is named PRIMARY; the secondary file, SECONDRY. If a record from the primary file matches one from the secondary file, the information in positions 1 through 10 of the secondary file record is placed in positions 31 through 40 of the primary file record. When there is no match, a 6 is placed in position 1 of the primary file record. The 6 indicates an unmatched record in the primary file.

Because the primary file record is processed first when it matches a secondary file record, the information from the secondary file can be made available only by a look-ahead field.

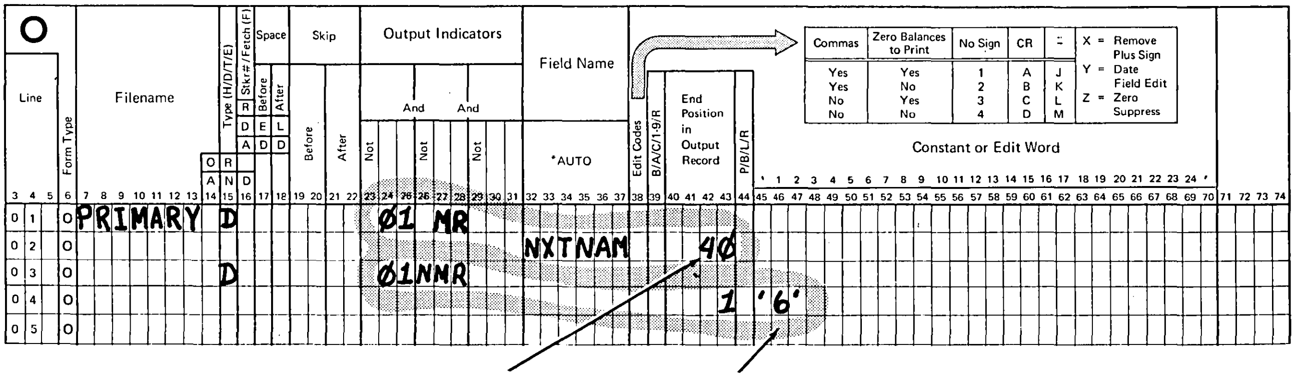
Input Specifications



```
  01 IPRIMARY AA  Ø1
  02 I                                        1  1Ø NAME1
  03 I                                       11  14 MATCH   M1
  04 ISECONDRYAB  Ø2
  05 I                                        1  1Ø NAME1
  06 I                                       11  14 MATCH   M1
  07 I        AC  **
  08 I                                        1  1Ø NXTNAM
  09 I
```

Look-ahead field (field from secondary file records needed in primary file records).

Output Specifications



```
  01 OPRIMARY D        Ø1 MR
  02 O                         NXTNAM    4Ø
  03 O        D        Ø1NMR
  04 O                                    1 '6'
  05 O
```

Place the look-ahead field from secondary records into positions 31 through 40 of the primary record if the two records match.

Place a 6 in position 1 of the primary record if the record matches no secondary record.

Figure 7-7. Look-ahead Fields

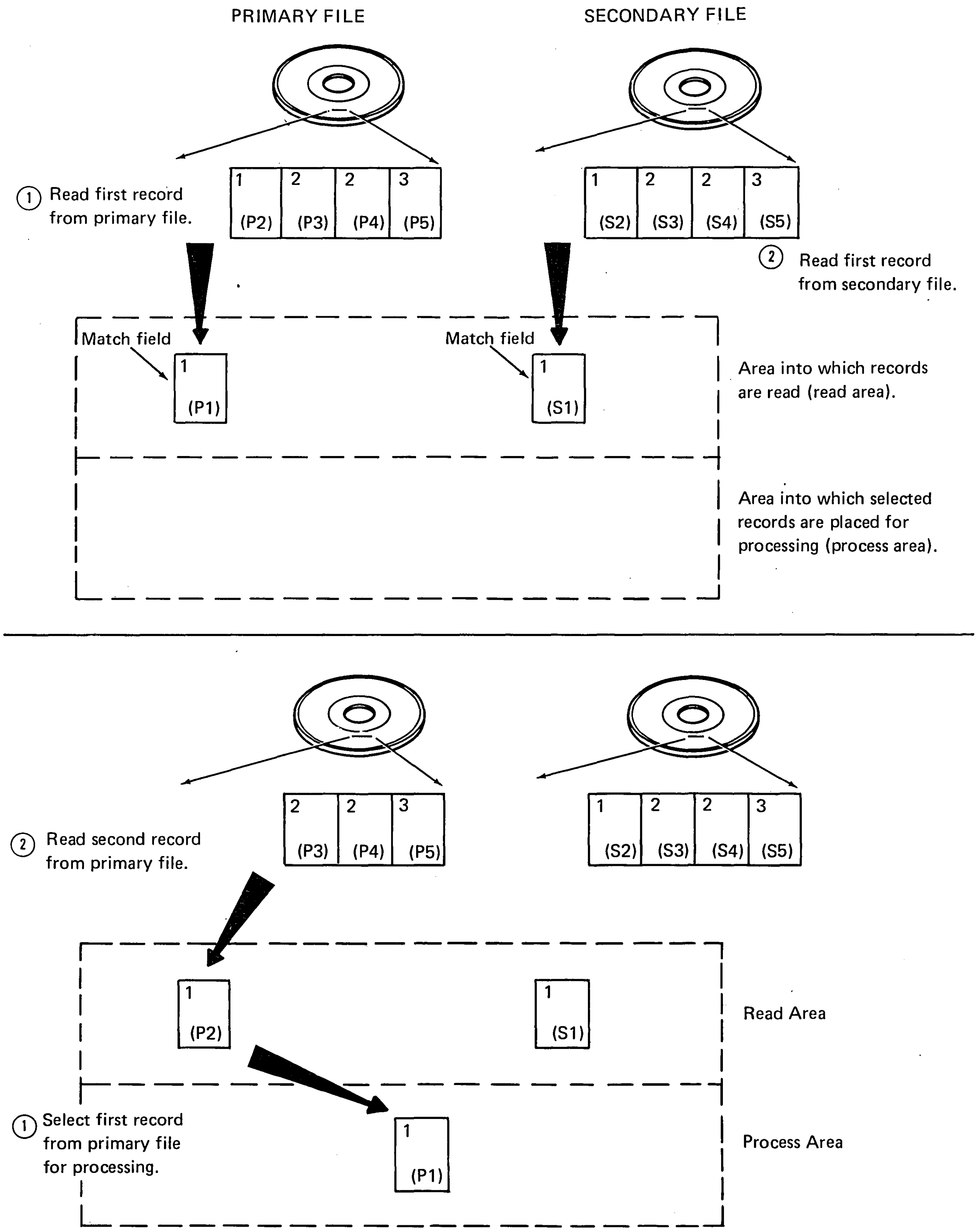
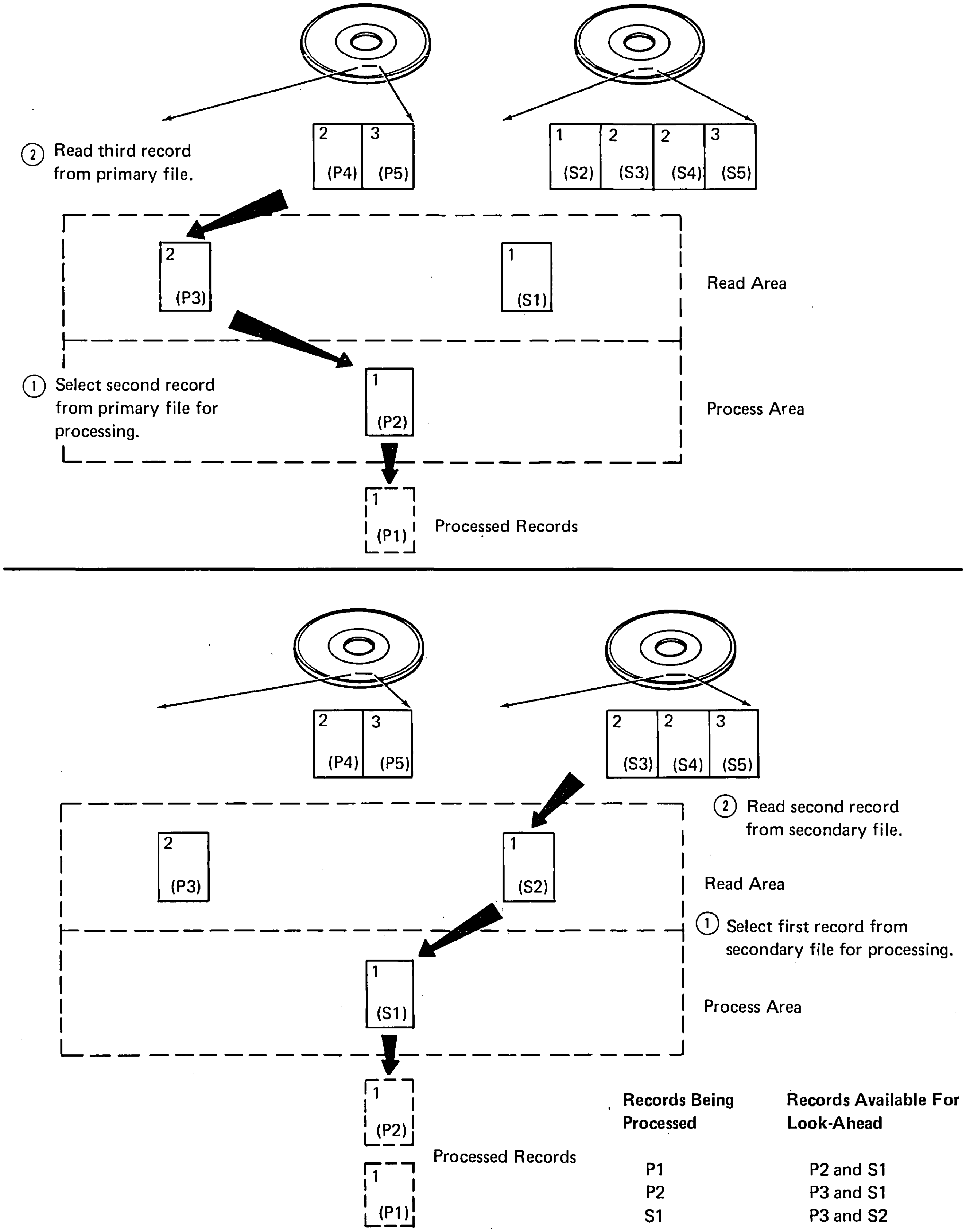
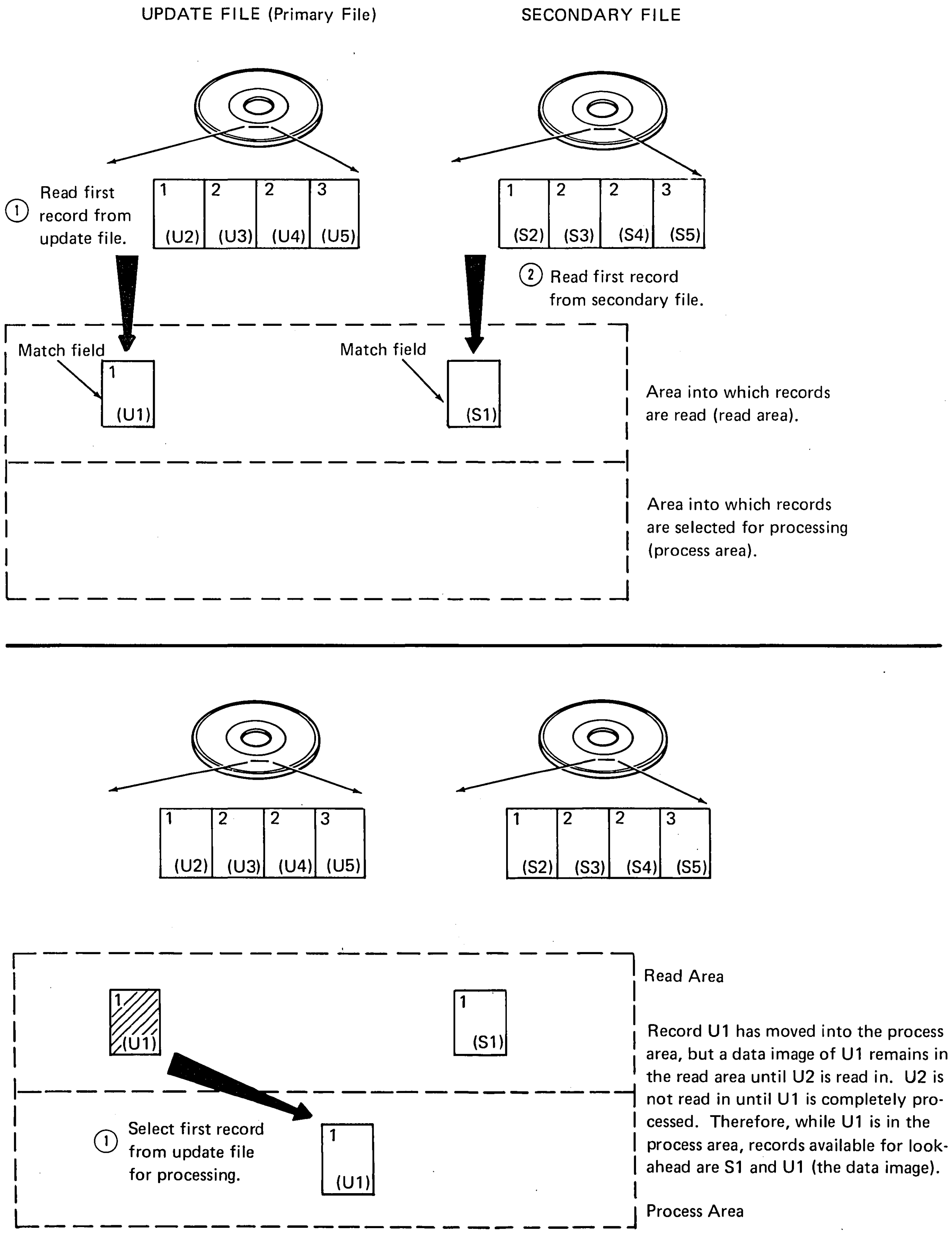

PRIMARY FILE
SECONDARY FILE
1 2 2 3
(P2) (P3) (P4) (P5)
1 2 2 3
(S2) (S3) (S4) (S5)
① Read first record from primary file.
② Read first record from secondary file.
Match field
Match field
1 (P1)
1 (S1)
Area into which records are read (read area).
Area into which selected records are placed for processing (process area).
2 2 3
(P3) (P4) (P5)
1 2 2 3
(S2) (S3) (S4) (S5)
② Read second record from primary file.
1 (P2)
1 (S1)
Read Area
① Select first record from primary file for processing.
1 (P1)
Process Area

Figure 7-8 (Part 1 of 2). Available Records: Two Input Files



Records Being Processed	Records Available For Look-Ahead
P1	P2 and S1
P2	P3 and S1
S1	P3 and S2

Figure 7-8 (Part 2 of 2). Available Records: Two Input Files



① Read first record from update file.

② Read first record from secondary file.

Area into which records are read (read area).

Area into which records are selected for processing (process area).

① Select first record from update file for processing.

Record U1 has moved into the process area, but a data image of U1 remains in the read area until U2 is read in. U2 is not read in until U1 is completely processed. Therefore, while U1 is in the process area, records available for look-ahead are S1 and U1 (the data image).

Figure 7-9 (Part 1 of 3). Available Records: One Input File, One Update File

① Read second record from update file.

① Select second record from update file for processing.

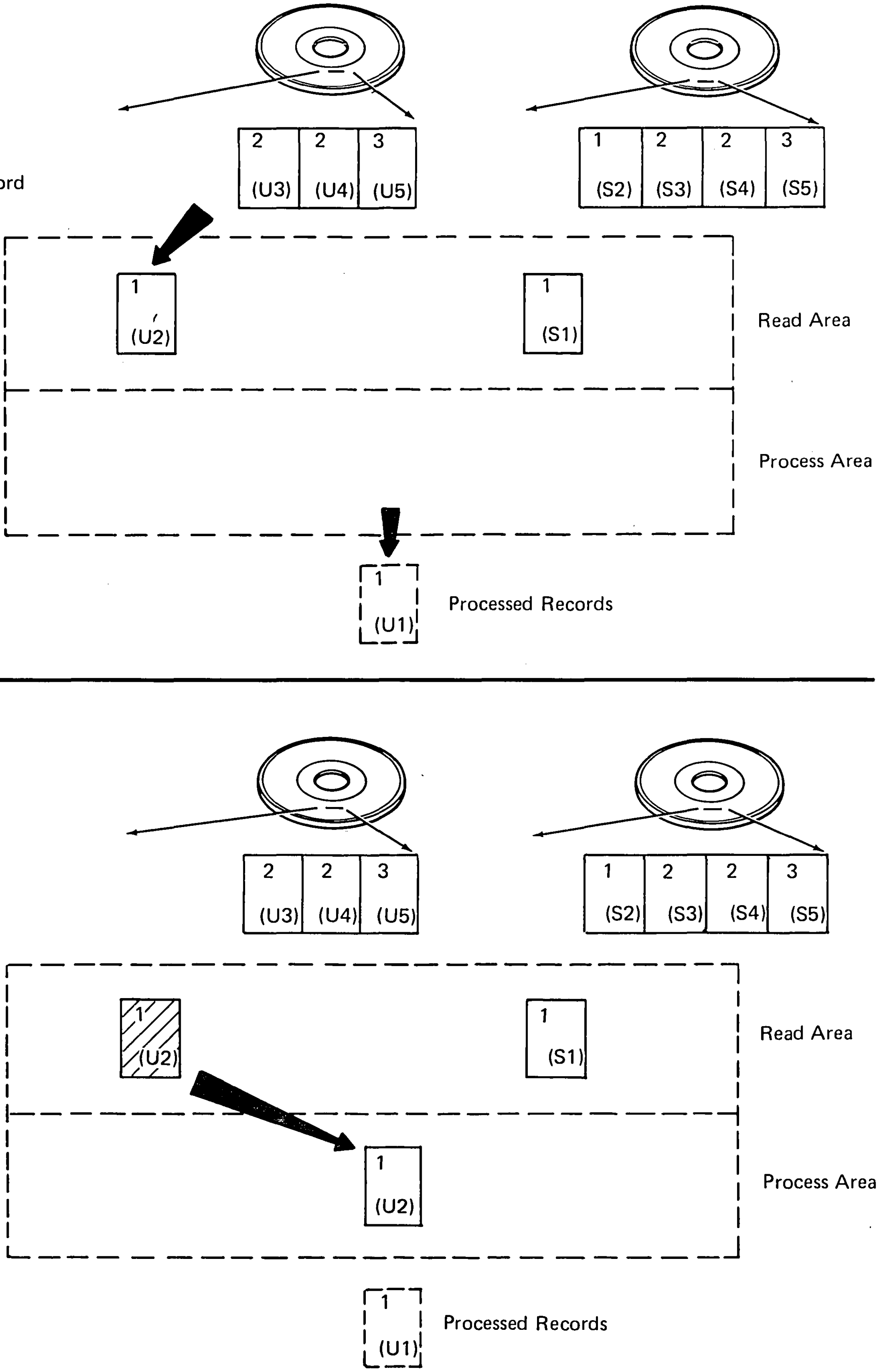
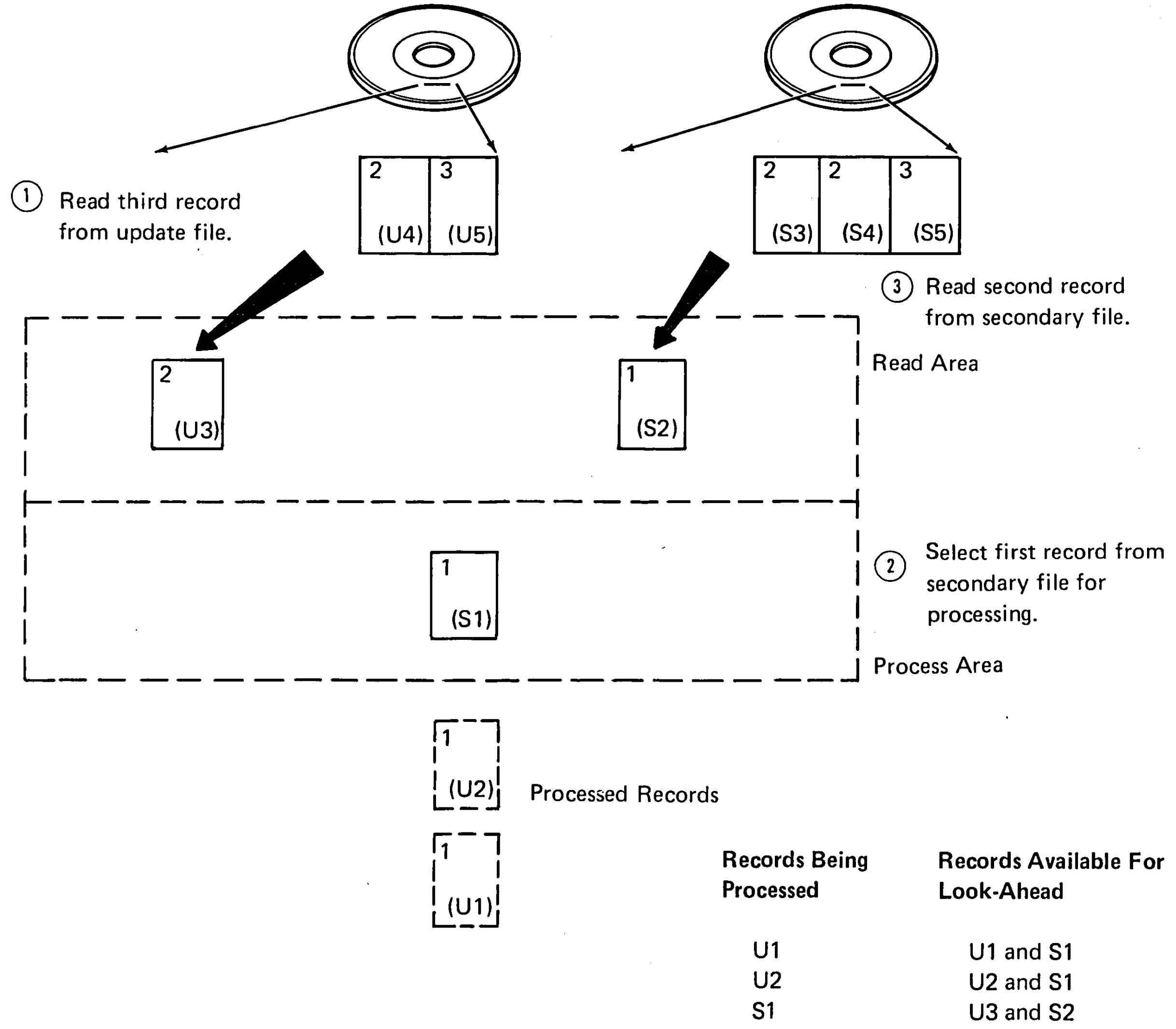


Figure 7-9 (Part 2 of 3). Available Records: One Input File, One Update File


2
3
(U4)
(U5)
2
2
3
(S3)
(S4)
(S5)
① Read third record from update file.
③ Read second record from secondary file.
2
(U3)
1
(S2)
Read Area
1
(S1)
② Select first record from secondary file for processing.
Process Area
1
(U2)
Processed Records
1
(U1)

Records Being Processed	**Records Available For Look-Ahead**
U1	U1 and S1
U2	U2 and S1
S1	U3 and S2

Figure 7-9 (Part 3 of 3). Available Records: One Input File, One Update File

Data Structures

A data structure can be used to:

- Define the same internal area multiple times using different data formats (see Figure 7-10).

- Subdivide a field so that either the entire field or its subfields can be referenced (see Figure 7-11).

- Group fields for easier reference (see Figure 7-12).

To specify a data structure, make the following entries (the columns not mentioned must be blank):

- *Column 6:* I.

- *Columns 7-14:* The name (maximum of 6 characters) of the data structure, which is optional. If specified, the name must meet the requirements of a field name. The data structure name can be referenced (1) as a field only on the input or output specifications, (2) as an RLABL, or (3) as a SAVDS or INFDS name in columns 60 through 65 of the file description specifications continuation lines for a WORKSTN file (see Chapter 13, *WORKSTN File Considerations and Sample Programs*, for more information on SAVDS or INFDS).

- *Column 18:* U, if this data structure is to be used as a display station local data area.

- *Columns 19-20:* DS, which identifies this as a data structure.

- *Columns 75-80:* Program identification.

Remember that data structure entries must be the last entries on the input specifications.

Figure 7-13 shows some common uses of a data structure.

Input Specifications

Line (3-5)	Form Type (6)	Filename (7-14)	Sequence (15-16)	Number (1/N) (17)	Option (O), U (18)	Record Identifying Indicator, or DS (19-20)	Record Identification Codes 1 Position (21-24)	1 Not (N) (25)	1 C/Z/D (26)	1 Character (27)	2 Position (28-31)	2 Not (N) (32)	2 C/Z/D (33)	2 Character (34)	3 Position (35-38)	3 Not (N) (39)	3 C/Z/D (40)	3 Character (41)	Stacker Select (42)	P/B/L/R (43)	Field Location From (44-47)	Field Location To (48-51)	Decimal Positions (52)	Field Name (53-58)	Control Level (L1-L9) (59-60)	Matching Fields or Chaining Fields (61-62)	Field Record Relation (63-64)	Field Indicators Plus (65-66)	Minus (67-68)	Zero or Blank (69-70)	(71-74)
01	I	SALESREC				Ø1	91		C	S	92		C	R																	
02	I																				1	32		SREC							
03	I	PURCHASE				Ø2	91		C	P	92		C	C																	
04	I																				1	35		PREC							
05	I	TRANSFER				Ø3	91		C	T	92		C	F																	
06	I																				1	4Ø		TREC							
07	I	*																													
08	I					DS																									
09	I	*SALES RECORD FIELDS																													
10	I																				1	32		SREC							
11	I																				1	9		ORDNO							
12	I																				1Ø	16		PART#							
13	I																				17	2Ø	Ø	QTY1							
14	I																				21	26	2	UNITCT							
15	I																				27	32	2	TOTCST							
16	I	*PURCASE RECORD FIELDS																													
17	I																				1	35		PREC							
18	I																				1	6		PARTNO							
19	I																				7	12	Ø	QTYOH							
20	I																				13	17	Ø	QTYORD							
21	I																				18	23	2	COST							
22	I																				24	29		ORDATE							
23	I																				3Ø	35	2	AMTDUE							
24	I	*TRANSFER RECORD FIELDS																													
25	I																				1	4Ø		TREC							
26	I																				1	6		PARTNO							
27	I																				7	1Ø	Ø	QTY2							
28	I																				11	2Ø		FRWHSE							
29	I																				21	3Ø		TOWHSE							
30	I																				31	4Ø		TIMDAT							
31	I																														

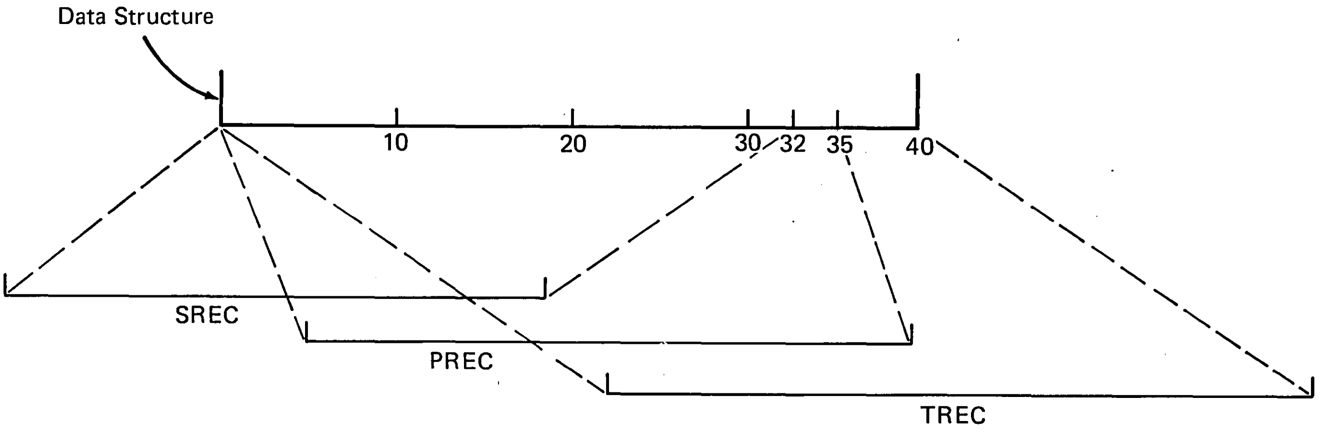
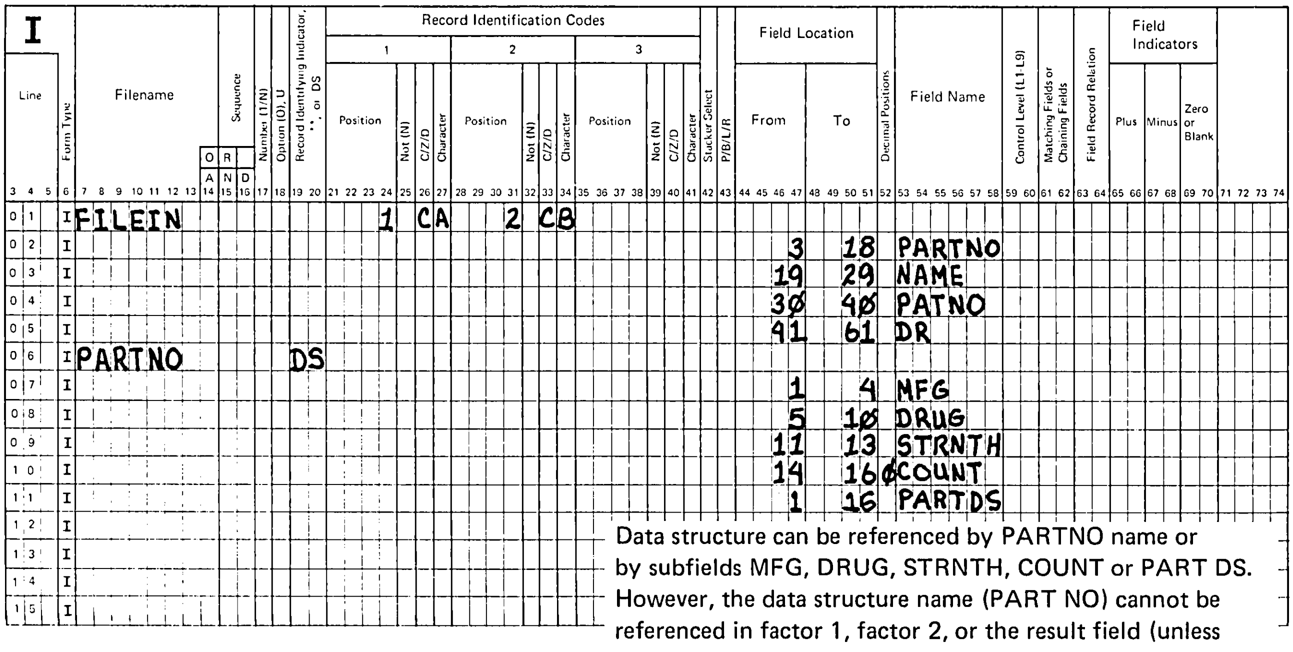


Figure 7-10. Using a Data Structure to Redefine the Same Internal Area Multiple Times

Input Specifications

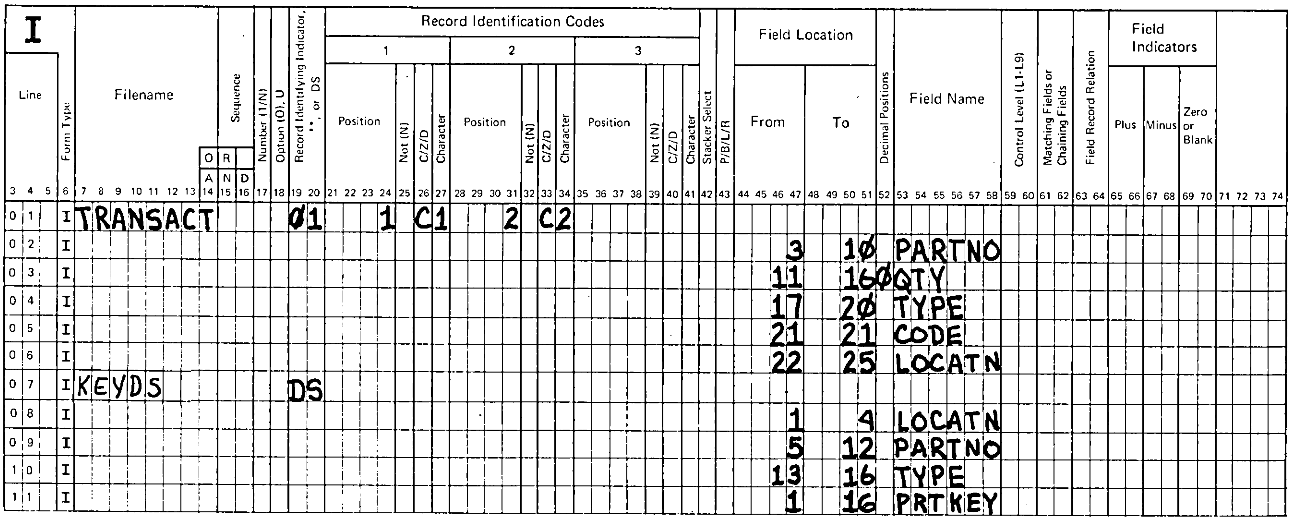


Line	Form Type	Filename	Record Identifying Indicator, **, or DS	Position (1)	C/Z/D (1)	Character (1)	Position (2)	C/Z/D (2)	Character (2)	From	To	Decimal Positions	Field Name
01	I	FILEIN		1	C	A	2	C	B				
02	I									3	18		PARTNO
03	I									19	29		NAME
04	I									3∅	4∅		PATNO
05	I									41	61		DR
06	I	PARTNO	DS										
07	I									1	4		MFG
08	I									5	1∅		DRUG
09	I									11	13		STRNTH
10	I									14	16	∅	COUNT
11	I									1	16		PARTDS
12	I												
13	I												
14	I												
15	I												

Data structure can be referenced by PARTNO name or by subfields MFG, DRUG, STRNTH, COUNT or PART DS. However, the data structure name (PART NO) cannot be referenced in factor 1, factor 2, or the result field (unless the operation is RLABL), but any of the subfield names can.

Figure 7-11. Using a Data Structure to Define Subfields within a Field

Input Specifications

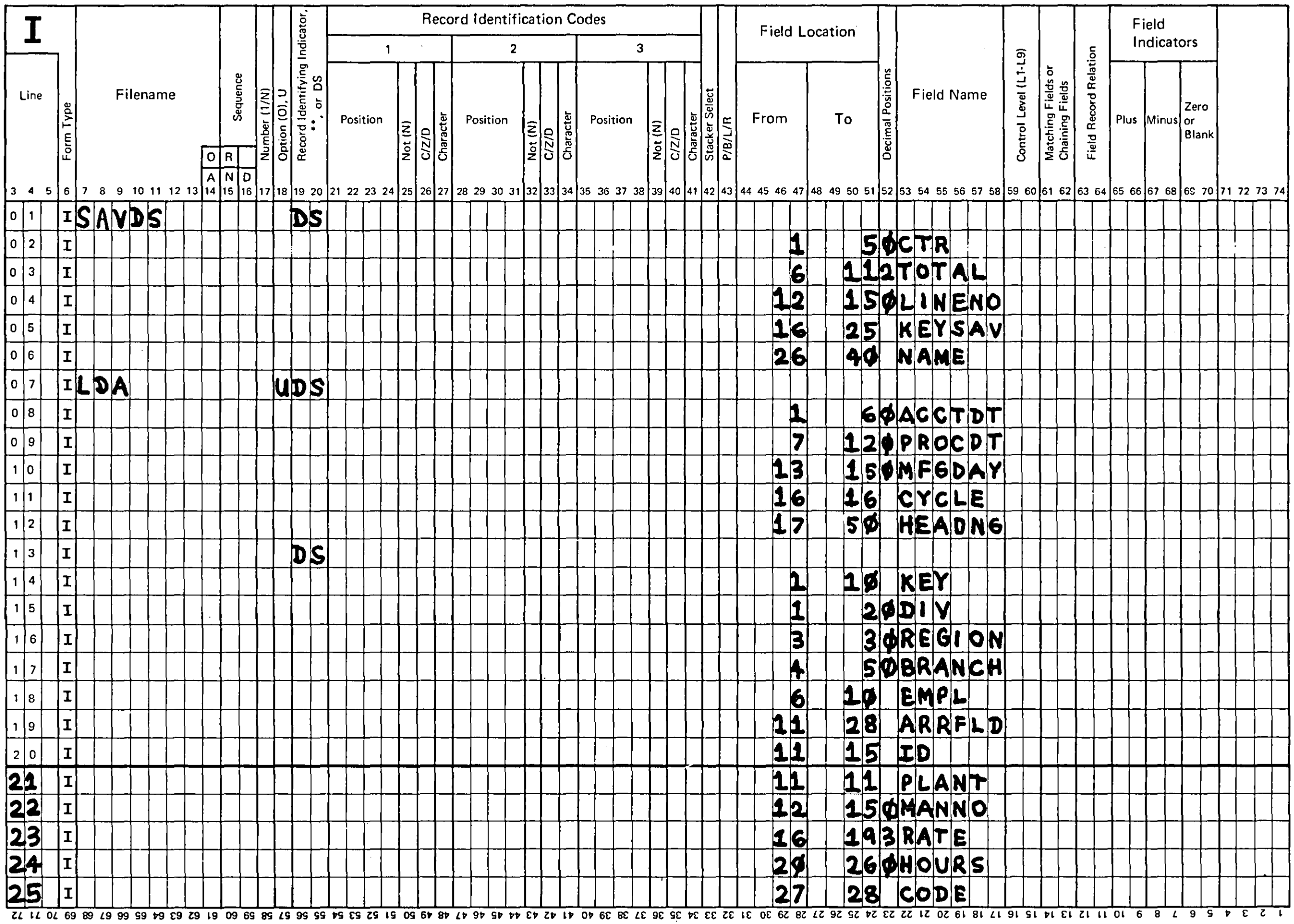


Line	Form Type	Filename	Record Identifying Indicator, **, or DS	Position (1)	C/Z/D (1)	Character (1)	Position (2)	C/Z/D (2)	Character (2)	From	To	Decimal Positions	Field Name
01	I	TRANSACT	∅1	1	C	1	2	C	2				
02	I									3	1∅		PARTNO
03	I									11	16	∅	QTY
04	I									17	2∅		TYPE
05	I									21	21		CODE
06	I									22	25		LOCATN
07	I	KEYDS	DS										
08	I									1	4		LOCATN
09	I									5	12		PARTNO
10	I									13	16		TYPE
11	I									1	16		PRTKEY

Figure 7-12. Using a Data Structure to Group Fields

When you use a data structure to group fields, fields from nonadjacent locations on the input record can be made to occupy adjacent storage locations internally. The storage area can then be referenced by the data structure name on output specifications, a data structure name entered as the result field of an RLABL operation, or by the subfield names. To reference the entire data structure in factor 1, factor 2, or the result field (except for RLABL), assign a subfield name to include the entire data structure (see line 11).

Input Specifications



Line	Form Type	Filename	Option (O), U	Record Identifying Indicator, or DS	From	To	Decimal Positions	Field Name
01	I	SAVDS		DS				
02	I				1	5	Ø	CTR
03	I				6	11	2	TOTAL
04	I				12	15	Ø	LINENO
05	I				16	25		KEYSAV
06	I				26	4Ø		NAME
07	I	LDA	U	DS				
08	I				1	6	Ø	ACCTDT
09	I				7	12	Ø	PROCDT
10	I				13	15	Ø	MFGDAY
11	I				16	16		CYCLE
12	I				17	5Ø		HEADNG
13	I			DS				
14	I				1	1Ø		KEY
15	I				1	2	Ø	DIV
16	I				3	3	Ø	REGION
17	I				4	5	Ø	BRANCH
18	I				6	1Ø		EMPL
19	I				11	28		ARRFLD
20	I				11	15		ID
21	I				11	11		PLANT
22	I				12	15	Ø	MANNO
23	I				16	19	3	RATE
24	I				2Ø	26	Ø	HOURS
25	I				27	28		CODE

The data structure defined on line 01 (SAVDS) is used as the SAVDS data structure (specified on a file description continuation line for the WORKSTN file). This data structure contains fields that are to be saved and restored for each display station attached to the WORKSTN file.

The data structure defined on line 07 (LDA) is used as the display station local data area (U in column 18). Information from the requesting command display station's local data area is read into this data structure at the beginning of the program and is written out at the end of the program. The program can change this data and update the display station local data area at end of job. This use of the data structure is usually related to a single requestor terminal (SRT) program.

The third data structure (defined on line 13) contains three subfields (KEY on line 14, ARRFLD on line 19, and ID on line 20) for which additional subfields are defined. The name ARRFLD (line 19) indicates a subfield into which an array is moved.

Figure 7-13. Typical Uses of Data Structures

To specify the subfields of a data structure, make the following entries. These entries must be made on the line below the DS specification. The columns not mentioned must be blank.

- *Column 6:* I.

- *Columns 44-47:* The number of the record position in which the subfield begins relative to the beginning of the data structure.

- *Columns 44-50:* To define the reserved, self-defining subfields for the file information data structure (INFDS), enter keywords (*OPCODE, *RECORD, *SIZE, *STATUS, *MODE, *INP, *OUT) on subfield specification lines for the data structure.

  In addition to these keyword-defined subfields, there is an alphameric subfield in the INFDS that contains WORKSTN data management or SSP-ICF return codes. This subfield is filled in for all WORKSTN files. The subfield is located in positions 23 through 26 and must be defined on the input specifications. (For information on return codes resulting from the use of the Interactive Communications Feature, see the *Interactive Communications Feature Reference Manual.*)

  Each subfield must be given a name in columns 53 through 58. This allows the program to reference the subfields. For more information concerning the INFDS data structure, see *WORKSTN Exception/Error Handling* in Chapter 13.

- *Columns 48-51:* The number of the record position in which the subfield ends relative to the beginning of the data structure.

- *Column 52:* The number of digits (0 through 9) to the right of the decimal position if the subfield is numeric. Leave the column blank if the subfield is alphameric. •*Columns 53-58:* The subfield name. The subfield name can be the same as an input field name or a result field name, and it can appear as an RLABL. Subfields can be used in factor 1, factor 2, or as an output field. However, the same subfield name cannot be specified in different data structures, and a data structure name cannot be specified as part of another data structure. RPG II field name rules apply to subfield names.

- *Columns 75-80:* Program identification (optional).

When using a data structure, consider the following:

- A data structure is considered an alphameric byte string. The data structure is initialized to blanks except for that part of the data structure that is initialized with an array or a display station local data area. You must ensure that numeric subfields are initialized with numeric data prior to their use in CHAIN, LOKUP, COMP, or editing operations. If the element is a compile-time array, the array data is placed in the data structure after the data structure has been initialized to blanks.

- You can redefine a subfield in the data structure by specifying the same or part of the same from/to positions for another subfield.

- The name of an input field or a result field that is being redefined in a data structure must be specified in the data structure or be the data structure name; however, it does not have to immediately precede the subfields redefining it.

- If a field appears as a data structure name or as a data structure subfield name, the physical space reserved for that field is in the data structure, regardless of where the field was defined.

- The from and to positions specified in a data structure for an input field that is being redefined are relative to the beginning of the data structure, not to the positions that the field occupies in the input record.

- A subfield can have the same length attributes as other fields or subfields.

- The maximum length of an alphameric subfield is 256 characters; the maximum length of a numeric subfield is 15 characters.

- If arrays are specified as subfields, the length specified must equal the amount of storage required to store the entire array.

The following restrictions apply to the use of a data structure:

- The maximum length of a data structure is 9,999 characters. However, if the data structure is defined as a display station local data area (U in column 18), the maximum length is 256.

- The length of a data structure is defined in one of two ways: (1)If the data structure name is specified as a field in an input record, the length of the data structure is the same as the length of the input field. If the to-position specified for a subfield exceeds the length defined for the input field, the input field specification is invalid. (2) If the data structure name is not specified as a field in an input record, the length is defined by the highest to-position specified for a subfield.

- Look-ahead fields cannot appear as a data structure or a subfield.

- Packed or binary numeric fields cannot be specified as a subfield within the data structure. The field can be defined as packed or binary in a file. RPG converts the field to zoned decimal format when it is placed in the data structure. The field is carried in zoned decimal format within the data structure.

- RPG II reserved words, array elements, and table names cannot be specified as a subfield.

- A maximum of 75 data structures or tables or arrays can be used in a program.

COLUMNS 21-41 (RECORD IDENTIFICATION CODES)

Use columns 21 through 41 to describe the information that identifies a record type. If all records are to be processed alike regardless of their type or if there is only one record type, leave columns 21 through 41 blank.

Note: Only columns 21 through 34 are valid for CONSOLE files (see Chapter 12, *CONSOLE File Considerations*, for more information).

When one file contains more than one record type, each record type is identified by a code consisting of a character or a combination of characters in certain positions in the record. If different operations are to be performed for each record type, this code must be described in columns 21 through 41 so that the program can determine the type of record selected for processing. Only one type of record is selected for processing during a program cycle, and the record identifying indicator for that record turns on at the time of selection.

Seven columns are used for the description of one character in the record identification code. Each specification line contains three sets of seven columns: columns 21 through 27, 28 through 34, and 35 through 41. Each set consists of four fields: Position, Not, C/Z/D, and Character. Coding is the same for all three sets.

Note: Any record that is read by the system and is not described by a record identification code in columns 21 through 41 causes the program to halt. The operator can continue, however, by selecting the appropriate option and pressing an entry function key. The record that causes the halt is not processed and the next record in that file is read.

Position (Columns 21-24, 28-31, and 35-38)

Entry	Explanation
Blank	No record identification code is needed.
1-4096	Record position of one character in the record identification code.

Use these columns to give the location in the record of every character in the identification code. These entries must end in columns 24, 31, and 38 respectively.

Not (N) (Columns 25, 32, and 39)

Entry	Explanation
Blank	Character is present in the specified record position.
N	Character should not be present in the specified record position (not valid for CONSOLE files; see Chapter 12).

Use these columns to indicate that a certain character should not be present in the specified position.

C/Z/D (Columns 26, 33, and 40)

Entry	Explanation
C	Entire character. C must be used for CONSOLE files (see Chapter 12).
Z	Zone portion of character.
D	Digit portion of character.

Use these columns to indicate what portion of a character is used as part of the record identifying code. Only the zone portion, only the digit portion, or both portions (the whole character) can be used (see Figure 7-14). When establishing record identifying codes, remember that many characters have either the same zone or the same digit portion. For a list of characters that have identical zone or digit portions, see Figure 7-15.

Character (Columns 27, 34, and 41)

In these columns, enter the alphabetic character, special character, or numeric character that is used in the record as the identification code or part of the code.

Character Grouping by Zone or Digit

When characters are used for record identification purposes on a digit or zone only basis, all characters having the same zone or digit are selected by the system as meeting record identification requirements. When a character is read into the system, it is converted into an 8-bit code. The program tests this 8-bit code to see whether the character meets the requirements of the record identifying character in the input specifications.

Figure 7-15 lists the characters that have identical zones or digits. For example, if column 26 contains D, which specifies digit only, and column 27 contains A, all records having a slash (/), A, J, or 1 in the specified column are selected as having the correct record identification code. If column 26 contains Z and column 27 contains A, all records containing & or A through I are selected as having the correct code.

The following three special cases are exceptions:

- The hex representation of an & (ampersand) is 50. However, when the ampersand is coded in the character entry, it is treated as though its hex representation were C0, that is, as if it had the same zone as the characters A through I. An ampersand in the input data satisfies two zone checks, for either a hex 5 zone or a hex C zone.
- The hex representation of a – (minus sign) is 60. However, when the minus sign is coded in the character entry, it is treated as though its hex representation were D0, that is, as if it had the same zone as the characters J through R. A minus sign in the input data satisfies two zone checks, for either a hex 6 zone or a hex D zone.
- The hex representation of a blank is 40. However, when the blank is coded in the character entry, it is treated as though its hex representation were F0, that is, as if it had the same zone as the numeric characters 0 through 9. A blank in the input data satisfies two zone checks, for either a hex 4 zone or a hex F zone.

AND Relationship

A maximum of three identifying characters can be described in one specification line. If the identification code consists of more than 3 characters, an AND line must be used to describe the additional characters. Write the word AND in columns 14 through 16 to indicate an AND line (see Figure 7-14).

A maximum of 20 AND lines can be used to describe the record identifying code for a record sequence if no OR lines are used. The record must contain all the characters indicated as its record identification code before the record identifying indicator turns on. AND lines are not allowed on CONSOLE files used for interactive data entry.

OR Relationship

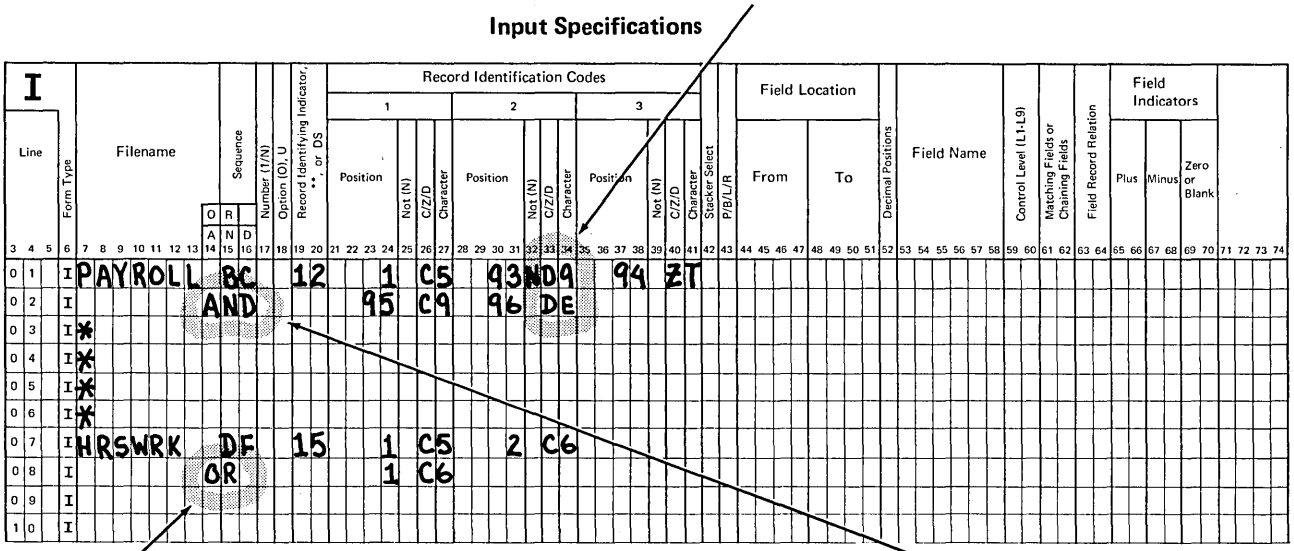
If a particular record type can be identified by two different codes, OR lines must be used to indicate that either of the codes can be present to identify the record. A maximum of 20 OR lines can appear for each record sequence if no AND lines are used. Write the word OR in columns 14 and 15 to indicate an OR line (see Figure 7-14).

Note: If AND lines and OR lines are combined, the total number of AND and OR lines for one record sequence cannot exceed 20.

COLUMN 42

Column 42 is not used. Leave it blank.

Character 5 must be present in position 1, zone of character T in position 94, character 9 in position 95, and digit E in position 96. However, digit 9 must not be present in column 93. Only the digit portions of 9 and E are checked, and only the zone portion of character T is checked.



Input Specifications

Line	Form Type	Filename	Sequence	AND / OR	Record Identifying Indicator	Position	Not (N)	C/Z/D	Character	Position	Not (N)	C/Z/D	Character	Position	Not (N)	C/Z/D	Character
01	I	PAYROLL	BC		12	1		C	5	93	N	D	9	94		Z	T
02	I			AND		95		C	9	96		D	E				
03	I	*															
04	I	*															
05	I	*															
06	I	*															
07	I	HRSWRK	DF		15	1		C	5	2		C	6				
08	I			OR		1		C	6								
09	I																
10	I																

Record type 15 can be identified by two different codes: a 5 in position 1 and a 6 in position 2 or a 6 in position 1.

AND must be used to describe last 2 characters of a 5-character code.

Figure 7-14. Record Identification Codes

Character Grouping by Zone (Z)		
Zone 4	Zone 9	Zone E
blank ¢ . < (+ |	j k l m n o p q r	\\ S T U V W X Y Z
Zone 5	Zone A	Zone F
! $ *) ; ¬	~ s t u v w x y z	blank 0 1 2 3 4 5 6 7 8 9
Zone 6	Zone C	
/ ¦ , (comma) % _ (underscore) > ?	& { A B C D E F G H I	
Zone 7	Zone D	
: # @ ' (apostrophe) = "	– (minus) } J K L M N O P Q R	
Zone 8		
a b c d e f g h i		

Character Grouping by Digit (D)		
Digit 0	Digit 6	Digit C

blank
&
- (minus)
}
{
\\
0
	f
	o
	w
	F
	O
	W
	6
		<
		*
		%
		@

Digit 1	Digit 7	Digit D
/ a j ~ A J l	g p x G P X 7	( ) _ (underscore) ' (apostrophe)
Digit 2	Digit 8	Digit E
b k s B K S 2	h q y H Q Y 8	+ ; > =
Digit 3	Digit 9	Digit F
c l t C L T 3	i r z I R Z 9	| ¬ ? "
Digit 4	Digit A	
d m u D M U 4	¢ ! ¦ :	
Digit 5	Digit B	
e n v E N V 5	. $, #	

Figure 7-15. Characters Interpreted as Having the Same Zone or Digit

Field Description Entries

Note: The field description entries (columns 43 through 74) must begin one line below the file and record identification entries (columns 7 through 42) for each file.

COLUMN 43 (PACKED OR BINARY FIELD)

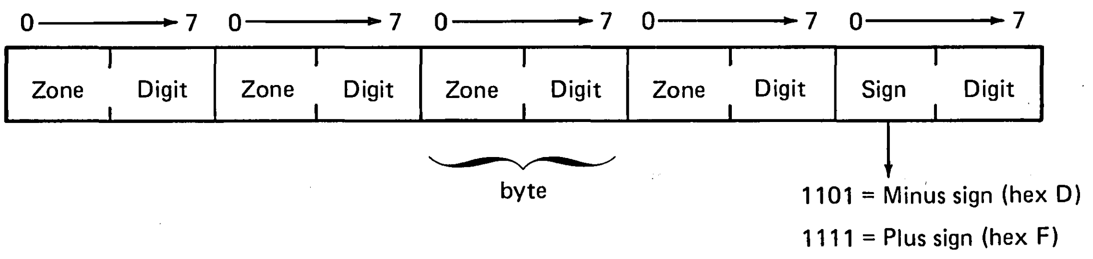
Entry	Explanation
Blank	Field is in zoned decimal format or is alphameric. (This column must be blank for CONSOLE files.)
P	Field named in columns 53 through 58 is in packed decimal format on the disk.
B	Field named in columns 53 through 58 is in binary format on the disk.

Use column 43 to indicate that a numeric field is in packed decimal or binary format. Only disk files support packed decimal or binary fields for read or write operations. Numeric data fields in packed decimal or binary format are converted by the system to the zoned decimal format before they are processed. This conversion ignores decimal points.

Any array that is in packed or binary format should have a P or B in this column. The from and to columns should then define the positions the array occupies in the record in the packed or binary format. The zoned decimal length of each array element is defined on the extension specifications.

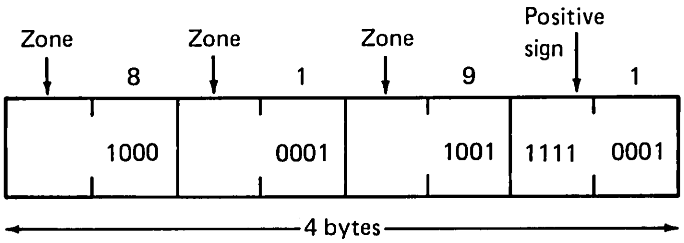
Zoned Decimal Format (Blank)

Zoned decimal format means that each byte of storage, whether on disk or in the computer, can contain 1 character. That character can be a decimal number or an alphabetic or special character. In the zoned decimal format, each byte of storage is divided into a 4-bit zone portion and a 4-bit digit portion. The zoned decimal format looks like this:



Note: RPG II does not perform data verification on numeric data. The value of the digit portion of a character is assumed to be the numeric value of that character.

The zone portion of the low-order byte indicates whether the decimal number is positive or negative. In zoned decimal format, each digit in a decimal number includes a zone portion; however, only the low-order zone portion serves as the sign. The decimal number 8,191 looks like this in zoned decimal format:

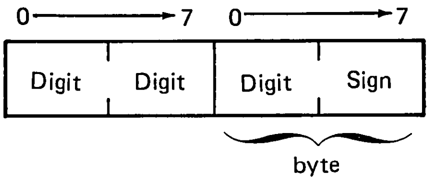


Once data is read into the computer, it must be represented in the zoned decimal format before it can be processed. Thus, data can be stored on disk and read into the computer in the zoned decimal format, thereby eliminating the need to convert the field. However, storing numeric data (decimal numbers) on disk in either the packed decimal or the binary format provides more efficient use of disk storage space.

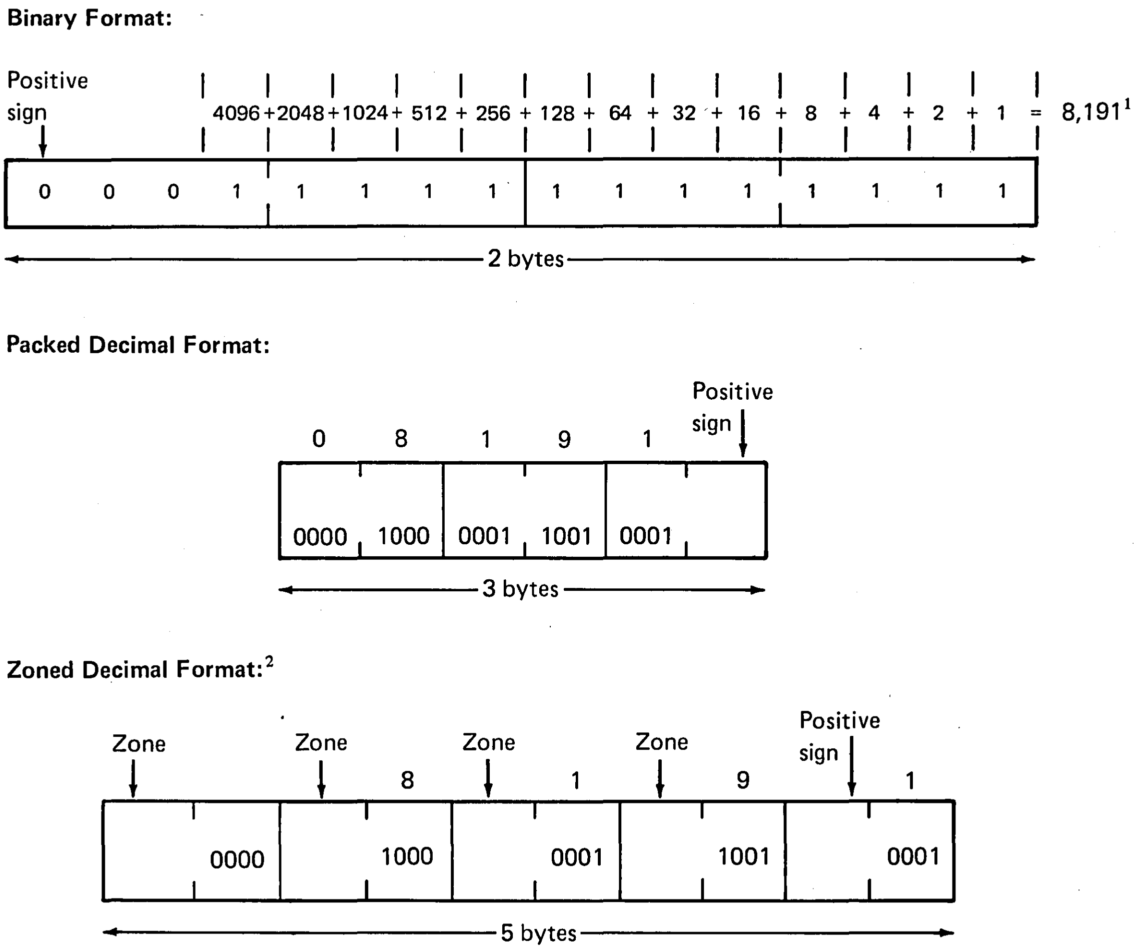
Packed Decimal Format (P)

Packed decimal format means that a byte of disk storage (except for the low-order byte) can contain two decimal numbers. Because many of the fields in a disk file contain decimal numbers, you can conserve disk space by storing these fields in the packed decimal format.

In the packed decimal format, each byte of disk storage, except the low-order byte, is divided into two 4-bit digit portions. The rightmost portion of the low-order byte contains the sign (plus or minus) for that field. The packed decimal format looks like this:



The sign portion of the low-order byte indicates whether the numeric value represented in the digit portions is positive or negative. In the packed decimal format, the sign is included for each decimal number; however, the zone portion is not given for each digit in the number. Compare how the decimal number 8,191 is represented in packed decimal format with its zoned decimal representation shown before (see Figure 7-16).



Binary Format:

Packed Decimal Format:

Zoned Decimal Format:[2]

[1] To obtain the numeric value of a positive binary number, add the values of the bits that are on (1); the sign bit is not included. To obtain the numeric value of a negative binary number, add the values of the bits that are off (0) plus one; the sign bit is not included.

[2] If 8.191 is read into storage as a zoned decimal field, it occupies 4 bytes. However, if it is converted to packed decimal format, it occupies 3 bytes; then when it is converted back to zoned decimal format, it occupies 5 bytes.

Figure 7-16. Binary, Packed Decimal, and Zoned Decimal Representation of 8,191

Because data must be represented in zoned decimal format once it is inside the computer, you must give the RPG II program an indication when input fields are in another format. Entering a P in column 43 indicates that the input field is in the packed decimal format and that the system must convert this field to the required zoned decimal format.

When a packed decimal field is converted to a zoned decimal field, the zoned decimal field always contains an odd number of bytes. If a zoned decimal field with an even number of bytes is converted to a packed decimal field and then converted back to a zoned decimal field, the resulting zoned decimal field also contains an odd number of bytes.

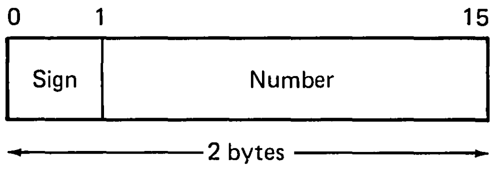
Packed fields can be up to 8 bytes long. The following chart shows the packed equivalents for zoned decimal fields up to 15 bytes long:

Zoned Decimal Length in Bytes	Packed Length in Bytes
15	
14	8
13	
12	7
11	
10	6
9	
8	5
7	
6	4
5	
4	3
3	
2	2
1	1

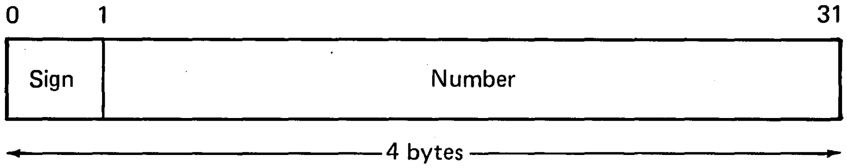
Binary Format (B)

Binary format means that 2 bytes of disk storage can contain a four-digit number, and that 4 bytes of disk storage can contain a nine-digit number. The binary format allows you to save even more disk storage space than you can save using the packed decimal format. In the binary format, each field on disk must be either 2 or 4 bytes long.

Each 2-byte binary field consists of a 1-bit sign followed by a 15-bit numeric value. In binary format, a decimal number as high as 9,999 requires only 2 bytes of disk storage. For each 2-byte binary field stored on disk, the RPG II compiler automatically sets aside 4 bytes of storage to accommodate the field when it is unpacked. A 2-byte field in binary format looks like this:



Each 4-byte binary field consists of a 1-bit sign followed by a 31-bit numeric value. In binary format, a decimal number as high as 999,999,999 requires only 4 bytes of disk storage. For each 4-byte binary field stored on disk, the RPG II compiler automatically sets aside 9 bytes of storage to accommodate the field when it is converted. A 4-byte field in binary format looks like this:



In each case, the sign portion of the high-order byte indicates whether the numeric value is positive (sign bit off) or negative (sign bit on). Positive numbers are represented in true binary notation with a 0 bit in the sign position. Negative numbers are represented in twos-complement notation with a 1 bit in the sign position. The bits between the sign position and the leftmost significant bit of the integer are always the same as the sign bit. When the number is positive, all bits to the left of the most significant bit, including the sign bit, are 0s. When the number is negative, all bits to the left of the most significant bit, including the sign bit, are 1s. Notice that, in the binary format, the zone position of the decimal number is not given. Compare how the decimal number 8,191 is represented in binary format with packed and zoned decimal representation (see Figure 7-16).

Because data must be represented in zoned decimal format once it is inside the computer, you must give the RPG II program an indication when input fields are in another format. Entering a B in column 43 indicates that the input field is in the binary format and that the system must convert this field to the required zoned decimal format.

Note: Although packed and binary fields require less disk storage space, the conversion routines needed to handle such data increase the object program size.

COLUMNS 44-51 (FIELD LOCATION)

Entry	Explanation
1-9999	Beginning of a field (from) or end of a field (to). See Chapter 12 for CONSOLE file considerations. For a WORKSTN file, the from and to positions refer to the location of the fields in the input record and not to their location in the displayed format.

Use columns 44 through 51 to describe the location on the record of the field named in columns 53 through 58. Enter the number of the record position in which the field begins in columns 44 through 47. Enter the number of the record position in which the field ends in columns 48 through 51. The entries must end in columns 47 and 51. Leading zeros can be omitted.

Define a single-position field by entering the same number in both the from (columns 44 through 47) and to (columns 48 through 51) positions. If a field of more than one position is defined, the number entered in columns 44 through 47 must be smaller than the number entered in columns 48 through 51.

The maximum field length for a zoned decimal numeric field is 15 positions (8 if the field is packed and 4 if it is blnary). The maximum field length for an alphameric field is 256 characters, and the maximum length for a data structure is 9,999 characters.

COLUMN 52 (DECIMAL POSITIONS)

Entry	Explanation
Blank	Alphameric field
0-9	Number of decimal positions in numeric field

Use column 52 to indicate the number of positions to the right of the decimal in any numeric field named in columns 53 through 58. Column 52 must contain an entry when the field named in columns 53 through 58 is numeric. To define a field as numeric with no decimal position, enter a 0. If a field is to be used in arithmetic operations or is to be edited, it must be numeric. If the number of decimal positions specified for a field exceeds the length of that field, the number of decimal positions is assumed equal to the length of the field.

COLUMNS 53-58 (FIELD NAME)

Entry	Explanation
1-6 alphameric characters	Field name, array name, or array element
PAGE, PAGE1-PAGE7	Special words

Use columns 53 through 58 to name a field, array, or array element found on your input records. When referencing an array, additional entries may be needed in these columns (see *Array Name and Index* in Chapter 14, *Tables and Arrays*). Use this name throughout the program whenever you refer to this field. Indicate the names of the fields for all types of records using a separate line for each field. However, name only the fields that you use. For example, if you use only the first 10 positions of a record that is 96 positions long, define positions 1 through 10 on the input specifications.

For CONSOLE files, whole array names must be entered in one of the following ways:

- Define the whole array as a subfield within a field.
- Define each element of the array with an index and place this entry in columns 53 through 58 of the input specifications. The index must be an integer value.

Field Names

A field name can be from 1 to 6 characters long and must begin in column 53. The first character must be an alphabetic character. The remaining characters can be any combination of alphabetic and numeric characters (special characters are not allowed). Blanks cannot appear between characters in the name.

All fields in one type of record should have different names. If two or more fields on the same record type have the same name, only the field described last is used. However, fields from different record types can have the same name if the fields are the same length and contain the same type of data. This applies even if the fields are in different locations in each record type.

Numeric fields can have a maximum length of 15 digits. Alphameric fields can have a maximum length of 256 characters (66 for CONSOLE files). A data structure can have a maximum length of 9,999 characters. Subfields can have a maximum length of 256 characters for alphameric subfields and 15 digits for numeric subfields.

If a data structure is specified, only field record relation indicators (columns 63 and 64) can be specified. Entries for control level indicators (columns 59 and 60), match field values (columns 61 and 62), and field indicators (columns 65 through 70) are not allowed. A data structure name cannot be specified as a subfield in a data structure.

Fields that are used in arithmetic operations (see Chapter 10, *Operation Codes*) or fields that are edited or zero suppressed (see *Column 38* and *Columns 45-70* in Chapter 9, *Output Specifications*) must be defined as numeric. Therefore, column 52 must have a decimal position entry (0 through 9).

Field Names in OR Relationship

If two or more record types contain identical fields, you must describe each field. To eliminate duplicate coding of identical fields from different record types, use the OR relationship (see Figure 7-17). A maximum of 20 OR lines can be used for each record sequence group if no AND lines are specified.

An OR relationship means that the fields named can be found in either of the record types. You can use OR lines when:

- Two or more record types have the same fields in the same positions (Figure 7-17).
- Two or more record types have some fields that are identical and some fields that differ in location, length, or type of data. See *Columns 63-64* in this chapter for sample coding of such record types.

Nrite OR in columns 14 and 15 to indicate an OR line. f there are several AND or OR lines, field description nes start after the last record identification line.

Special Words (PAGE, PAGE1-PAGE7)

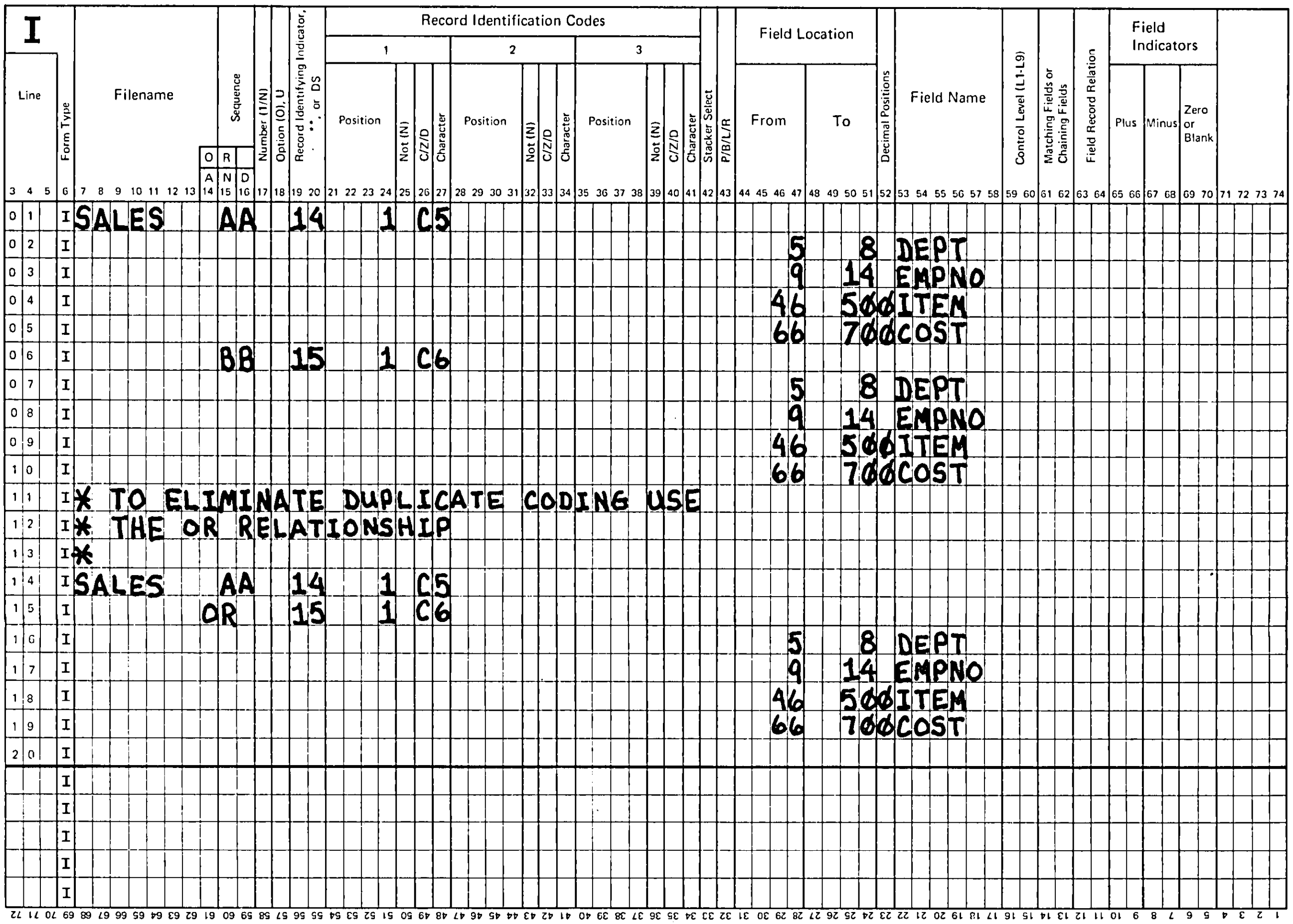
If a printed report has several pages that are to be numbered, use the special word PAGE to indicate that page numbering is to be done. When you use a PAGE entry on the output specifications, page numbering automatically starts with 1 (see *Columns 32-37* in Chapter 9, *Output Specifications*).

To start at a page number other than 1, enter that page number in a field of an input record and name that field PAGE in columns 53 through 58. The number entered in the PAGE field should be one less than the starting page number. If numbering starts with 24, enter a 23 in the PAGE field. The PAGE field can be 1 to 15 digits long, but must have zero decimal positions (see Figure 7-18). If a PAGE field is used but it is not defined, the PAGE field is assumed to be 4 digits long with zero decimal positions. Any entry in the PAGE field should be right-justified, such as 0023.

Page numbering can be restarted during a program run when a number is specified in a PAGE field of any input record. The PAGE field can be defined as a numeric field, 1 to 15 digits in length, with zero decimal positions, and used in calculations like any other field.

The eight possible PAGE entries (PAGE, PAGE1, PAGE2, PAGE3, PAGE4, PAGE5, PAGE6, and PAGE7) are provided for numbering different page types in the output file or for numbering the pages for different printer files.

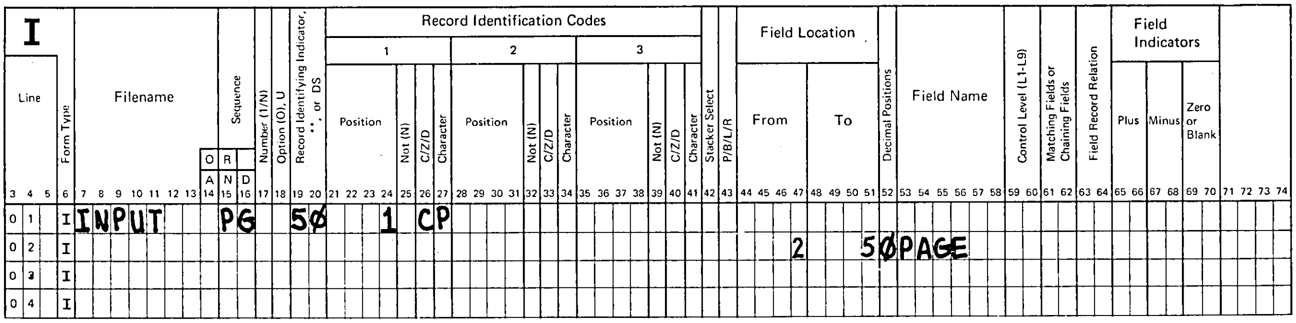
Input Specifications



```
01 ISALES   AA  14   1 C5
02 I                                        5   8 DEPT
03 I                                        9  14 EMPNO
04 I                                       46  5ØØITEM
05 I                                       66  7ØØCOST
06 I        BB  15   1 C6
07 I                                        5   8 DEPT
08 I                                        9  14 EMPNO
09 I                                       46  5ØØITEM
10 I                                       66  7ØØCOST
11 I* TO ELIMINATE DUPLICATE CODING USE
12 I* THE OR RELATIONSHIP
13 I*
14 ISALES   AA  14   1 C5
15 I        OR  15   1 C6
16 I                                        5   8 DEPT
17 I                                        9  14 EMPNO
18 I                                       46  5ØØITEM
19 I                                       66  7ØØCOST
20 I
   I
   I
   I
   I
   I
```

Figure 7-17. Record Types with Identical Fields

Input Specifications



```
01 IINPUT   PG  5Ø   1 CP
02 I                                        2   5ØPAGE
03 I
04 I
```

Figure 7-18. Page Record Description

COLUMNS 59-60 (CONTROL LEVEL)

Entry	Explanation
L1-L9	Any control level indicator. Control level indicators cannot be used with chained, demand, or WORKSTN files or with a data structure.

Use columns 59 and 60 to assign control level indicators to input fields. Use control level indicators to specify when calculation or output operations are to be performed. You can assign a control level indicator to any field; this field is then known as a control field. The program checks the field for a change in information. When the information changes, a control break occurs. All records having the same information in the control field are known as a control group.

Whenever a record containing a control field is read, the data in the control field is compared with data in the same control field from the previous record. When a control break occurs, the control level indicator turns on. Operations conditioned by the control level indicator are then performed (see *Columns 7-8* in Chapter 8, *Calculation Specifications* or *Columns 23-31* in Chapter 9, *Output Specifications*).

There are nine different control levels (L1 through L9). When a control level indicator turns on, all control level indicators lower than it also turn on. For example, if control level indicator 3 turns on, control level indicators 1 and 2 also turn on.

The indicators are ranked in order of importance. The larger numbers rank higher than smaller numbers. L4 has a higher rank than L1. The importance of a control field in relation to other fields determines how you assign indicators. For example, the type of data that demands a subtotal should have a lower control level indicator than data that needs a final total. A field containing department numbers should have a higher control level indicator than a field containing employee numbers (see Figure 7-19).

Because control level indicator L0 is always on, it cannot be assigned to a control field. Nevertheless, you can use it to condition operations (see *Columns 7-8* in Chapter 8, *Calculation Specifications*).

Normally, control level indicators are used to:

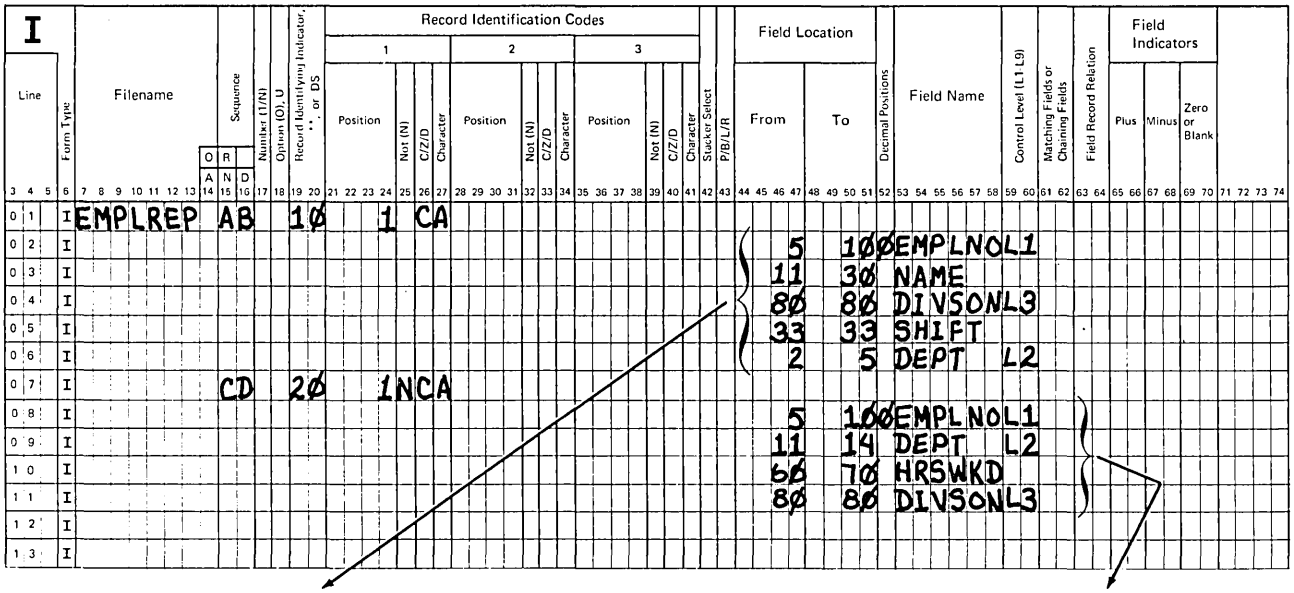
- Condition certain total calculations to be performed when the information in the control field changes.
- Condition certain total output operations to be done after totals are accumulated for one control group.
- Condition certain detail calculation or output operations to be done on the record that causes a change in a control field (first record of a new control group).

Assigning Control Level Indicators

The following considerations apply to assigning control level indicators:

- If the same control level indicator is used in different record types or in different files, the control fields associated with that control level indicator must be the same length and same type (alphabetic or numeric). See Figure 7-19.
- In the same record type, record positions in control fields assigned different control level indicators can overlap (Figure 7-20). However, the total number of positions assigned as control fields must not be greater than 144. In Figure 7-20, for example, 15 positions have been assigned to control levels.
- Field names are ignored in control level operations. Therefore, fields from different record types that have been assigned the same control level indicator can have the same name.
- Control levels need not be written in any sequence. An L2 entry can appear before L1. Also, there can be gaps in the control levels assigned.
- When numeric control fields with decimal positions are compared to determine whether a control break has occurred, they are always treated as if they have no decimal positions. For instance, 3.46 is considered equal to 346.
- If a field is specified as numeric, only the digit portion determines whether a control break has occurred. This means that a field is always considered to be positive. For example, -5 is considered equal to +5.
- All control fields given the same control level indicator are considered numeric if any one of those control fields is described as numeric (column 52 has an entry). Therefore, when numeric control fields are compared to determine whether the information has changed, only the digit portion of each character is compared.

Input Specifications



I																															
							Record Identification Codes														Field Location								Field Indicators		
							1				2				3																
Line	Form Type	Filename	Sequence OR AND	Number (1/N)	Option (O), U	Record Identifying Indicator, **, or DS	Position	Not (N)	C/Z/D	Character	Position	Not (N)	C/Z/D	Character	Position	Not (N)	C/Z/D	Character	Stacker Select	P/B/L/R	From	To	Decimal Positions	Field Name	Control Level (L1-L9)	Matching Fields or Chaining Fields	Field Record Relation	Plus	Minus	Zero or Blank	
3 4 5	6	7 8 9 10 11 12 13 14	15 16	17	18	19 20	21 22 23 24	25	26	27	28 29 30 31	32	33	34	35 36 37 38	39	40	41	42	43	44 45 46 47	48 49 50 51	52	53 54 55 56 57 58	59 60	61 62	63 64	65 66	67 68	69 70	71 72 73 74
0 1	I	EMPLREP	AB			1Ø	1		C	A																					
0 2	I																				5	1Ø	Ø	EMPLNO	L1						
0 3	I																				11	3Ø		NAME							
0 4	I																				8Ø	8Ø		DIVSON	L3						
0 5	I																				33	33		SHIFT							
0 6	I																				2	5		DEPT	L2						
0 7	I		CD			2Ø	1	N	C	A																					
0 8	I																				5	1Ø	Ø	EMPLNO	L1						
0 9	I																				11	14		DEPT	L2						
1 0	I																				66	7Ø		HRSWKD							
1 1	I																				8Ø	8Ø		DIVSON	L3						
1 2	I																														
1 3	I																														

Three control level indicators are assigned. The names of the control fields (DIVSON, DEPT, EMPLNO) give an indication of their relative importance. The division (DIVSON) is the most important group. It is given the highest control level indicator used (L3). The department (DEPT) ranks below the division. L2 is assigned to it. The employee field has the lowest control level indicator (L1) assigned.

The same control level indicators can be used for different record types. Notice, however, that the control fields having the same indicators are the same length. EMPLNO, in both cases, is 6 characters in length, DEPT is 4, and DIVSON is 1.

Figure 7-19. Control Level Indicators (Two Record Types)

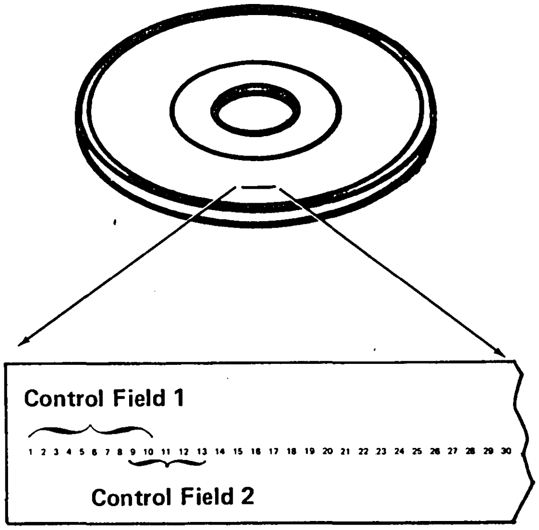


Figure 7-20. Overlapping Control Fields in a Disk Record

- Control fields are initialized to hexadecimal zeros.
- A control break can occur after the first record containing a control field is read. The control fields in this record are compared to an area in storage that contains hexadecimal zeros. Because fields from two different records are not being compared, total calculations and total output operation are bypassed for this cycle. A control break occurs then, but it is not considered to be a true control break.
- If different record types in a file do not have the same number of control fields, unwanted control breaks can occur. See Figure 7-21 for an example of how to avoid unwanted control breaks.
- A control field cannot be specified as binary (B in column 43). However, it can be specified as packed decimal (P in column 43).

Split Control Fields

If a control field is made up of more than one field of a record, it is then known as a split control field. A split control field is created when the same indicator is assigned to two or more connected or unconnected fields on the same record type.

All fields in one record that have the same control level indicators are combined by the program in the order specified by the input specifications and treated as one control field (see Figure 7-22). Some special rules for split control fields are:

- For one control level indicator, a field can be split in some record types and not in others if the field names are different. However, the length of the field, whether split or not, must be the same in all record types.
- The length of the portions of a split control field can vary for different record types if the field names are different. However, the total length of the portions must always be the same.
- No other specification lines can come between lines that describe split control fields.
- If one section of a split control field is numeric, the whole field is considered numeric.
- A numeric split control field can have more than 15 characters if any one portion of the split field does not exceed 15 characters and the sum of all control fields is not greater than 144 characters.
- A split control field cannot be made up of a packed decimal field and a zoned decimal field. Both portions of the control field must be packed or both must be zoned decimal.

Note: Additional rules applying to control level indicators when used with indicators in the field record relation columns are discussed in *Columns 63-64 (Field Record Relation).*

COLUMNS 61-62 (MATCHING FIELDS)

Entry	Explanation
M1-M9	Any matching level

Use columns 61 and 62 to specify match fields and sequence checking. Match fields and sequence checking cannot be specified for chained files, demand files, WORKSTN files, or a data structure.

An entry in columns 61 and 62 indicates:

- Match fields and sequence checking when you have two or more input or update files with match fields
- Sequence checking only when you have just one input or update file

The match levels are ranked in order of importance, with M1 being the least significant.

Match Fields

In multifile processing, specify match fields to compare records from two or more input or update files to determine which record is to be selected for processing. You can use one field, many fields, or an entire record to match records. Whenever the contents of the match field from the primary file record are the same as the contents of the match field from a secondary file record, the matching record (MR) indicator turns on. The MR indicator can then be used to condition those operations that are to be done only when records match (see *Columns 9-17* in Chapter 8, *Calculation Specifications* and *Columns 23-31* in Chapter 9, *Output Specifications*).

As many as nine match fields can be indicated when you use the values M1 through M9. M1 through M9 only identify the fields by which the records are matched; they are not indicators, but they cause the MR indicator to turn on.

For a complete description of how to assign match fields and how records are selected for processing, see Chapter 11, *Multifile Processing.*

	(L2)	
A	Salesman Number	Salesman Name
1	2 3	4 16

Salesman Record

	(L2)	(L1)	
B	Salesman Number	Item Number	Amount
1	2 3	4 6	7 9

Item Record

Different record types normally contain the same number of control fields. However, some applications require a different number of control fields in some records.

The salesman records contain only the L2 control field. The item records contain both L1 and L2 control fields. With normal RPG II coding, an unwanted control break is created by the first item record following the salesman record. This is recognized by an L1 control break immediately following the salesman record and results in an asterisk being printed on the line below the salesman record.

```
01     JOHN SMITH                 Unwanted
                            *     control
           100          3         break
           100          2
                        5   *
           101          4
                        4   *
                        9   **


02     JANE DOE                   Unwanted
                            *     control
           100          6         break
           100          2
                        8   *
           101          3
                        3   *
                       11   **

                       20
```

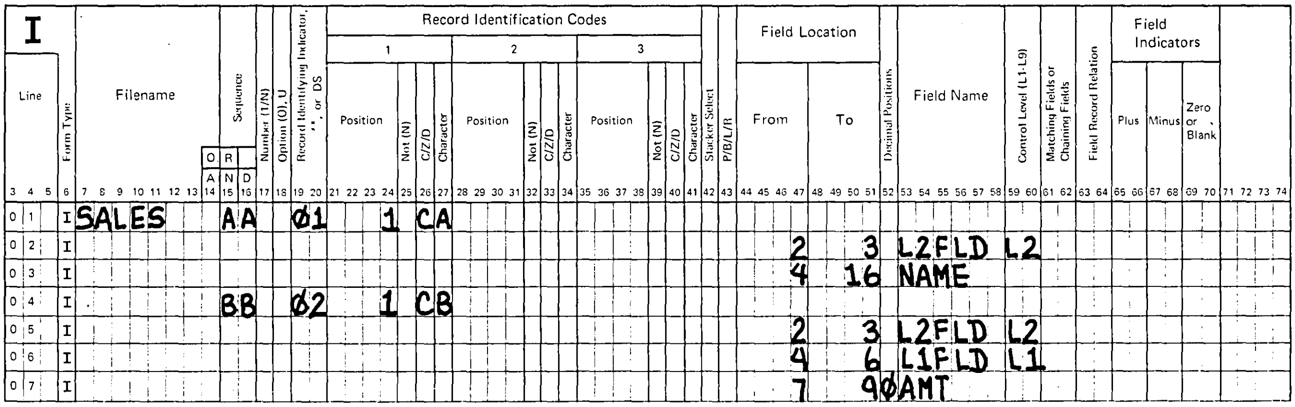
Output Showing Unwanted Control Level Break

```
01     JOHN SMITH

           100          3
           100          2
                        5   *
           101          4
                        4   *
                        9   **


02     JANE DOE

           100          6
           100          2
                        8   *
           101          3
                        3   *
                       11   **

                       20
```

Corrected Output

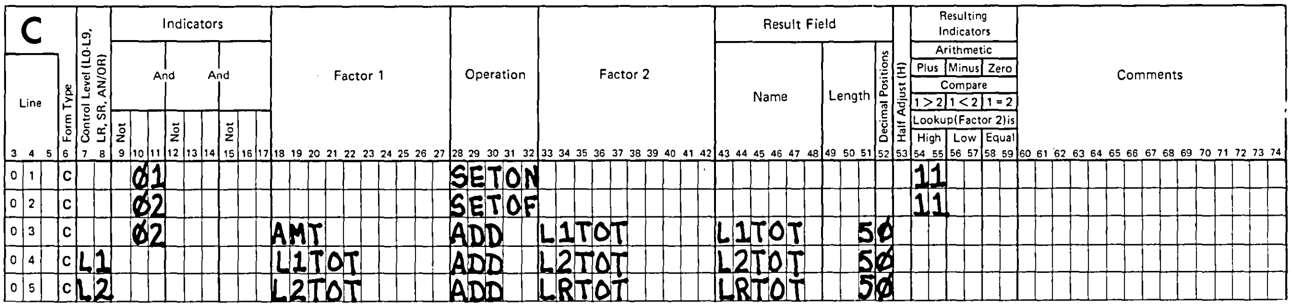
Figure 7-21 (Part 1 of 2). Unwanted Control Breaks

Input Specifications



Line	Form Type	Filename	Sequence	Record Identifying Indicator	Position	C/Z/D	Character	From	To	Decimal Positions	Field Name	Control Level (L1-L9)
01	I	SALES	AA	Ø1	1	C	A					
02	I							2	3		L2FLD	L2
03	I							4	16		NAME	
04	I		BB	Ø2	1	C	B					
05	I							2	3		L2FLD	L2
06	I							4	6		L1FLD	L1
07	I							7	9	Ø	AMT	

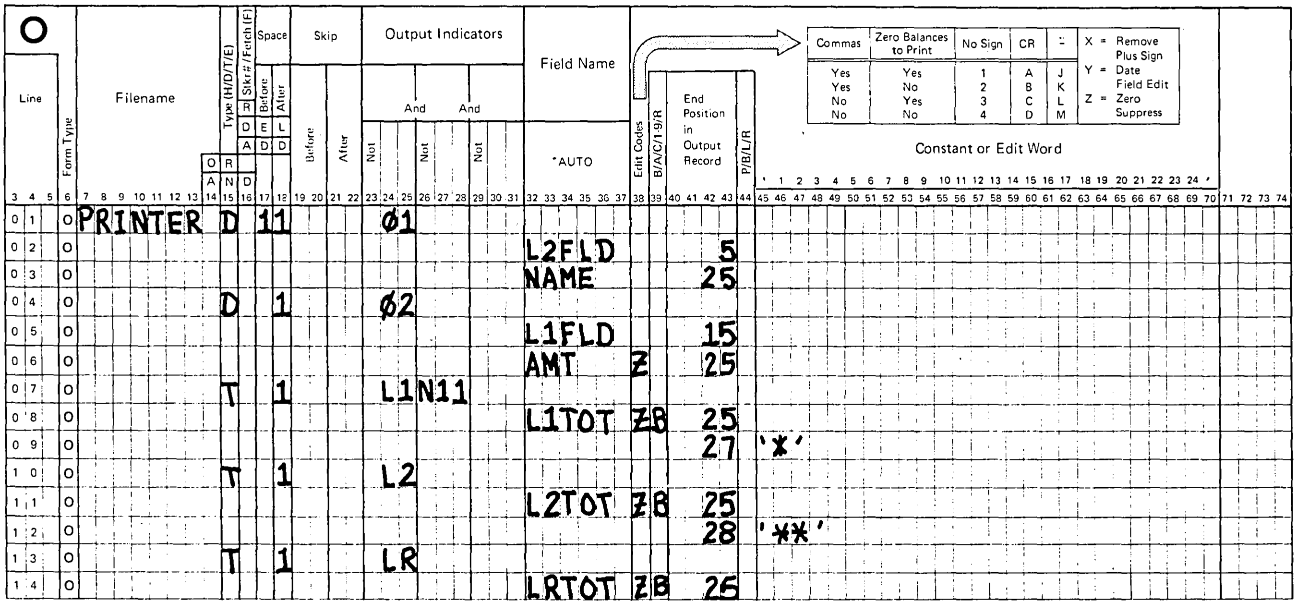
Calculation Specifications



Line	Form Type	Control Level	Indicators	Factor 1	Operation	Factor 2	Result Field Name	Length	Decimal Positions	Resulting Indicators Plus
01	C		Ø1		SETON					11
02	C		Ø2		SETOF					11
03	C		Ø2	AMT	ADD	L1TOT	L1TOT	5	Ø	
04	C	L1		L1TOT	ADD	L2TOT	L2TOT	5	Ø	
05	C	L2		L2TOT	ADD	LRTOT	LRTOT	5	Ø	

This coding prevents the unwanted control break. Line 01 of the calculation specifications sheet sets on indicator 11 when the salesman record is read. When the next item record causes an L1 control break, no total output is printed because indicator 11 is on (line 07 of output specifications sheet). Detail calculations are then processed for the item record, and line 02 of the calculation specifications sheet sets indicator 11 off. This allows the normal L1 control break to occur.

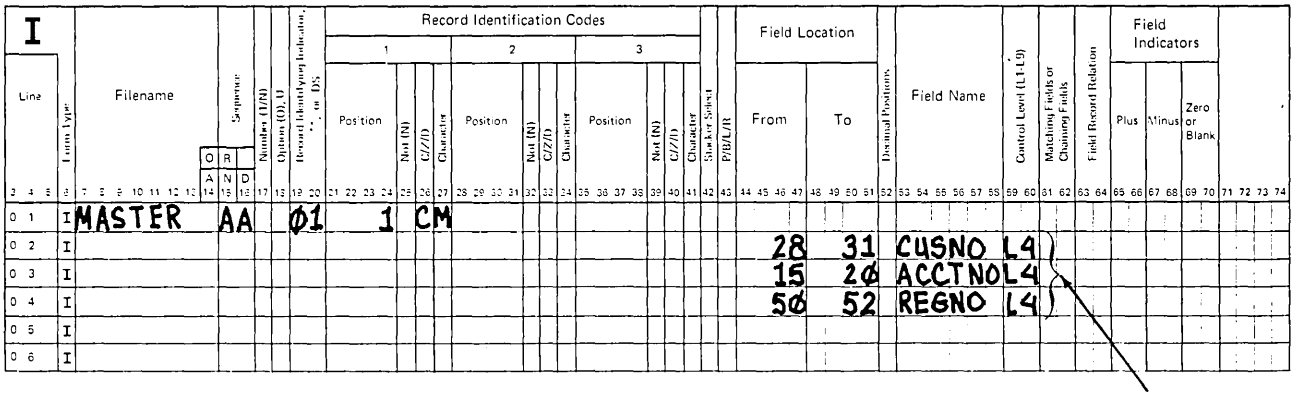
Output Specifications



Line	Form Type	Filename	Type (H/D/T/E)	Space Before	Space After	Output Indicators	Field Name	Edit Codes	Blank After	End Position in Output Record	Constant or Edit Word
01	O	PRINTER	D	1	1	Ø1					
02	O						L2FLD			5	
03	O						NAME			25	
04	O		D		1	Ø2					
05	O						L1FLD			15	
06	O						AMT	Z		25	
07	O		T		1	L1 N11					
08	O						L1TOT	Z	B	25	
09	O									27	'*'
10	O		T		1	L2					
11	O						L2TOT	Z	B	25	
12	O									28	'**'
13	O		T		1	LR					
14	O						LRTOT	Z	B	25	

Figure 7-21 (Part 2 of 2). Unwanted Control Breaks

Input Specifications



Line	Form Type	Filename	Sequence	Record Identifying Indicator	Position	Character	From	To	Field Name	Control Level (L1-L9)
0 1	I	MASTER	AA	Ø1	1	CM				
0 2	I						28	31	CUSNO	L4
0 3	I						15	2Ø	ACCTNO	L4
0 4	I						5Ø	52	REGNO	L4
0 5	I									
0 6	I									

All portions of a split control field must be assigned the same control level indicator.

Figure 7-22. Split Control Fields

Sequence Checking

To check the data in the fields of a record in one input or update file for a special sequence, assign a value of M1 through M9 to the field to be checked. As many as nine fields can be checked. The sequence (ascending or descending) of the record file must be specified in column 18 of the file description specifications (see Chapter 3). See Figure 7-23 for an example of sequence checking.

To check the sequence of record types in a file, see *Columns 15-16 (Sequence)* in this chapter.

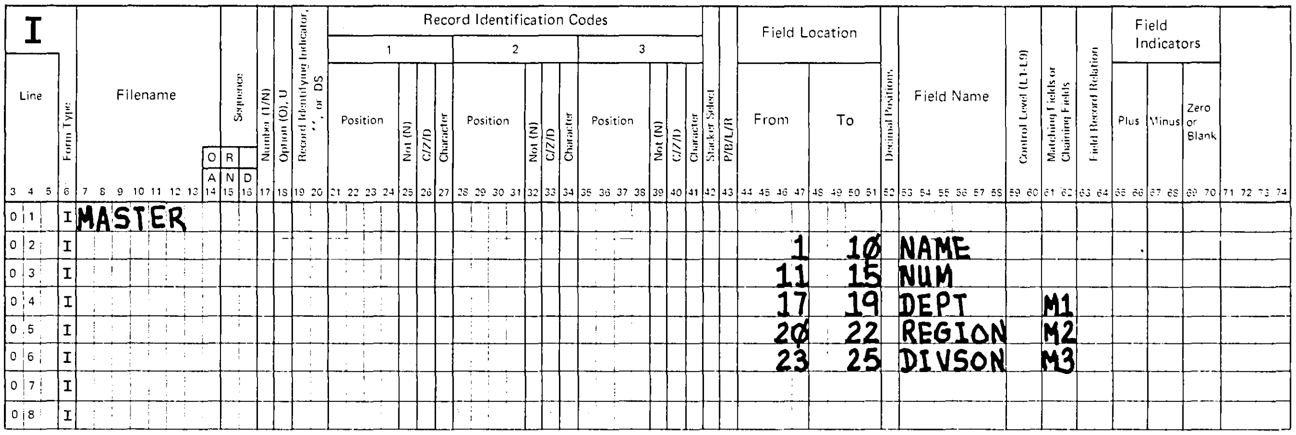
COLUMNS 63-64 (FIELD RECORD RELATION)

Entry	Explanation
Blank	Columns must be blank for CONSOLE files.
01-99	Record identifying indicator assigned to a record type, or an indicator set on elsewhere in the program.
L1-L9	Control level indicator previously used.
MR	Matching record indicator.
U1-U8	External indicator previously set.
H1-H9	Halt indicator previously used.

The following general rules apply to the use of columns 63 and 64:

- All fields, including match or control fields, that have no field record relation indicator should be described before those that do.
- All fields having the same field record relation indicator should be defined on consecutive specification lines for more efficient use of storage. These fields can, however, be entered in any order.
- All portions of a split control field must be assigned the same field record relation indicator and must be defined on consecutive specification lines (see Figure 7-24). For more information on split control fields, see *Columns 59-60 (Control Level).*
- When used with control or match fields, the field record relation indicator must match a record identifying indicator for this file, and the match or control fields must be grouped according to the field record relation indicator. The field record relation indicator for control or match fields can only be 01 through 99 or H1 through H9.
- When any match value (M1 through M9) is specified for a field without a field record relation indicator, all match values used must be specified once without a field record relation indicator. If all match fields are not common to all records, a dummy match field should be used (see Figure 7-25).

Input Specifications



```
  01 IMASTER
  02 I                                        1  10 NAME
  03 I                                       11  15 NUM
  04 I                                       17  19 DEPT    M1
  05 I                                       20  22 REGION  M2
  06 I                                       23  25 DIVSON  M3
  07 I
  08 I
```

An input file called MASTER is to be sequence-checked through three fields. Data from two records is shown below:

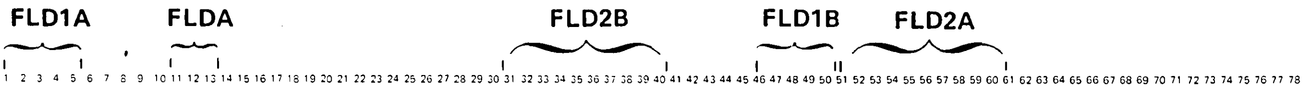
Data from First Record		Data from Second Record	
DEPT	008	DEPT	003
REGION	051	REGION	025
DIVSON	003	DIVSON	005

In sequence checking, all fields are treated as one continuous field. Thus, the match fields look like:

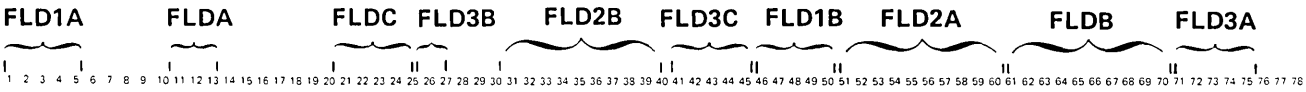
	M3	M2	M1
Record 1	003	051	008
Record 2	005	025	003

The match field from record 1 is compared with the match field from record 2. If the file is specified to be in ascending sequence, the records are in order because 005025003 is higher than 003051008. However, if the file is specified as having a descending sequence, record 2 is out of order.

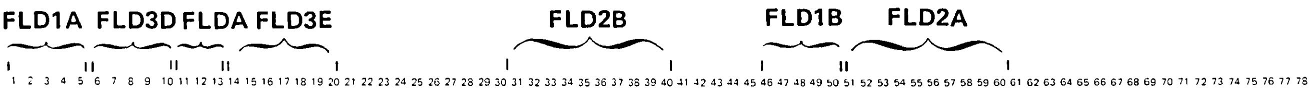
Figure 7-23. Match Fields (Sequence Checking Within a File)



Record identification code = 1



Record identification code = 2



Record identification code = 3

The record identified by a 1 in position 95 has two split control fields:

FLD1A and FLD1B
FLD2A and FLD2B

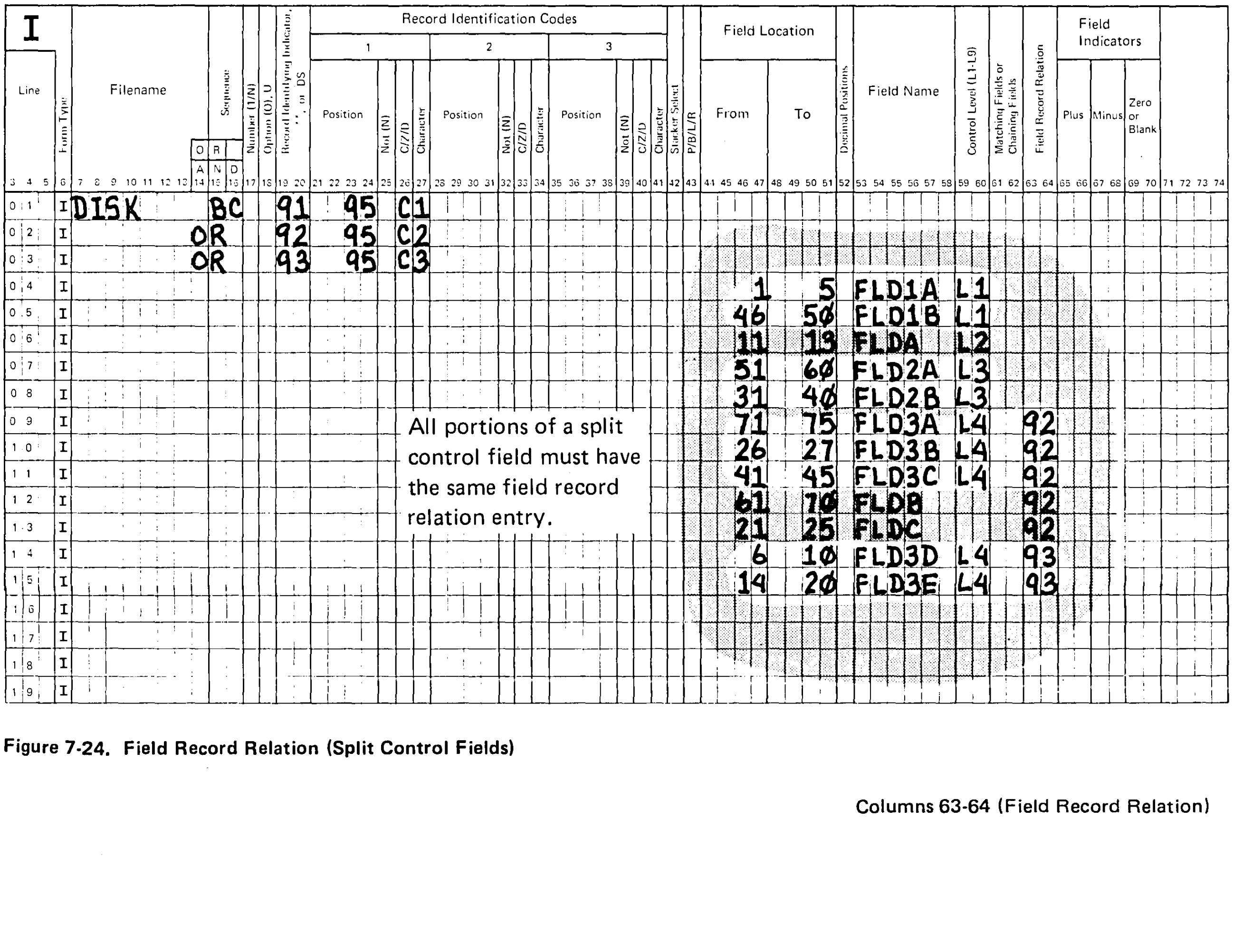
The record with a 2 in position 95 has three split control fields:

FLD1A and FLD1B
FLD2A and FLD2B
FLD3A, FLD3B, and FLD3C

The third record type, identified by the 3 in position 95, also has three split control fields:

FLD1A and FLD1B
FLD2A and FLD2B
FLD3D and FLD3E

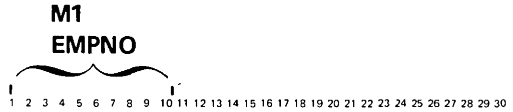
All portions of the split control field must be assigned the same control level indicator and all must have the same field record relation entry.



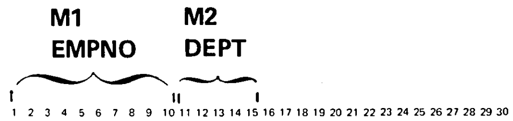
Line	Form Type	Filename	Sequence / AND-OR	Record Identifying Indicator	Position (1)	Not	C/Z/D	Character	From	To	Decimal Positions	Field Name	Control Level (L1-L9)	Matching Fields or Chaining Fields	Field Record Relation	Plus	Minus	Zero or Blank
01	I	DISK	BC	91	95		C	1										
02	I		OR	92	95		C	2										
03	I		OR	93	95		C	3										
04	I								1	5		FLD1A	L1					
05	I								46	5Ø		FLD1B	L1					
06	I								11	13		FLDA	L2					
07	I								51	6Ø		FLD2A	L3					
08	I								31	4Ø		FLD2B	L3					
09	I								71	75		FLD3A	L4		92			
10	I								26	27		FLD3B	L4		92			
11	I								41	45		FLD3C	L4		92			
12	I								61	7Ø		FLDB			92			
13	I								21	25		FLDC			92			
14	I								6	1Ø		FLD3D	L4		93			
15	I								14	2Ø		FLD3E	L4		93			
16	I																	
17	I																	
18	I																	
19	I																	

All portions of a split control field must have the same field record relation entry.

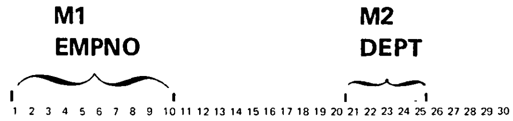
Figure 7-24. Field Record Relation (Split Control Fields)



Record identifying indicator 01

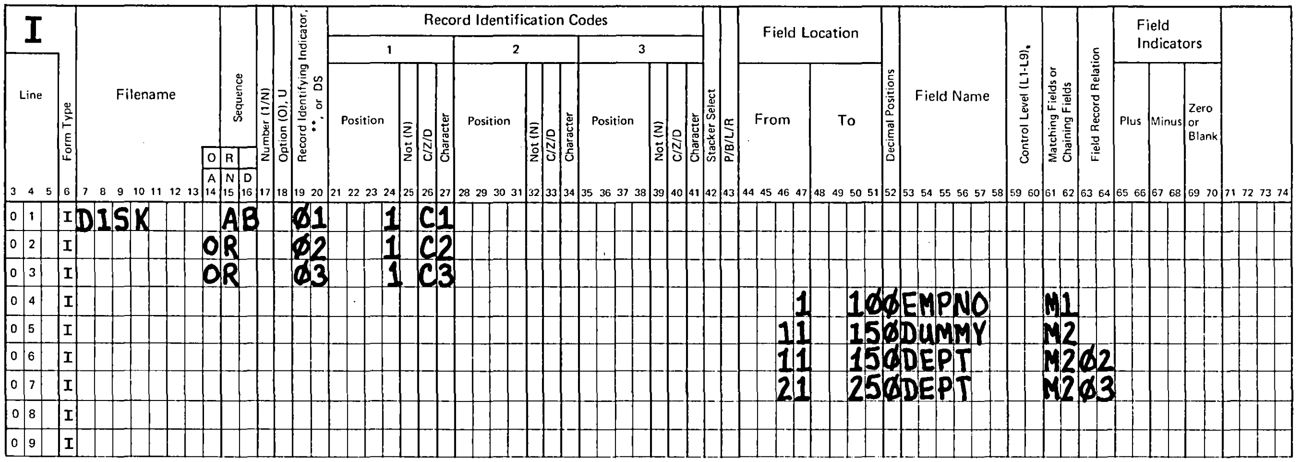


Record identifying indicator 02



Record identifying indicator 03

Three different record types are found in the input file. All three contain a match field in positions 1 through 10. Two of them have a second match field. Because M1 is found on all record types, it can be specified without an entry in columns 63 and 64. If one match value (M1 through M9) is specified without field record relation entries, all match values must be specified once without field record relation entries. Because the value M1 is specified without field record relationship, an M2 value must also be specified once without field record relationship. The M2 field is not on all record types; thus a dummy M2 field must be specified next. The dummy field can be given any unique name, but its specified length must be equal to the length of the true M2 field. The M2 field is then related to the record types on which it is found by field record relation entries (lines 06 and 07).



I

Line	Form Type	Filename	Sequence	Number (1/N)	Option (O), U	Record Identifying Indicator, **, or DS	1 Position	Not (N)	C/Z/D	Character	2 Position	Not (N)	C/Z/D	Character	3 Position	Not (N)	C/Z/D	Character	Stacker Select	P/B/L/R	From	To	Decimal Positions	Field Name	Control Level (L1-L9)	Matching Fields or Chaining Fields	Field Record Relation	Plus	Minus	Zero or Blank
01	I	DISK	AB			Ø1	1		C	1																				
02	I		OR			Ø2	1		C	2																				
03	I		OR			Ø3	1		C	3																				
04	I																				1	1Ø	Ø	EMPNO		M1				
05	I																				11	15	Ø	DUMMY		M2				
06	I																				11	15	Ø	DEPT		M2	Ø2			
07	I																				21	25	Ø	DEPT		M2	Ø3			
08	I																													
09	I																													

Figure 7-25. Dummy Match Fields

Record Identifying Indicators (01-99)

Use a record identifying indicator (01 through 99) in columns 63 and 64 to relate a field to a particular record type.

When several record types are specified in an OR relationship, all fields that do not have a field record relation indicator in columns 63 and 64 are associated with all record types in the OR relationship. To relate a field to just one record type, enter the record identifying indicator assigned to that record type in columns 63 and 64 (see Figure 7-26).

An indicator (01 through 99) that was previously defined in the program can also be used in columns 63 and 64 to condition movement from the input area to the storage area. Control fields, which are specified by a control level indicator (L1 through L9) in columns 59 and 60, and match fields, which are specified by a match value (M1 through M9) in columns 61 and 62, can also be related to a particular record type in an OR relationship by a field record relation indicator. Control fields or match fields in the OR relationship that do not have a field record relation indicator are used with all record types in the OR relationship.

When two control fields have the same control level indicator or two match fields have the same matching level value, a field record relation indicator can be assigned to just one of the match fields. In this case, only the field with the field record relation indicator is used when that indicator is on. If none of the field record relation indicators are on for that control field or match field, the field without a field record relation indicator is used. Control fields and match fields can only have 01 through 99 or H1 through H9 as the entry in columns 63 and 64.

Control Level (L1-L9), Matching Record (MR), and External (U1-U8) Indicators

Columns 63 and 64 can also be used to specify that the program accept and use data from a particular field only when a certain condition occurs (such as matching records, a control break, or an external indicator is on). Indicate the conditions under which the program accepts data from a field by specifying indicator L1 through L9, MR, or U1 through U8 in columns 63 and 64. Data from the field named in columns 53 through 58 is accepted only when the field record relation indicator is on.

External indicators are primarily used when file conditioning is specified in columns 71 and 72 of the file description specifications. However, they can be used even though file conditioning is not specified.

Halt Indicators (H1-H9)

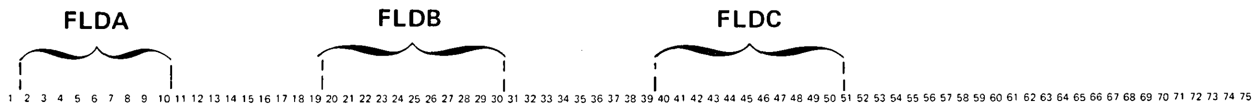
A halt indicator (H1 through H9) in columns 63 and 64 relates a field to a record that is in an OR relationship and also has a halt indicator specified in columns 19 and 20.

COLUMNS 65-70 (FIELD INDICATORS)

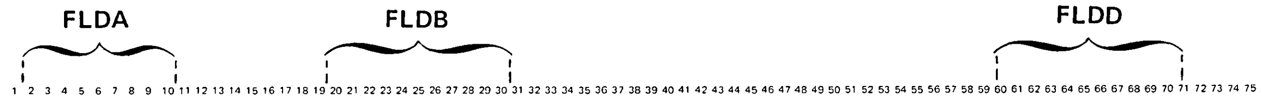
Entry	Explanation
01-99	Numeric indicator
H1-H9	Halt indicator (when checking for an error condition in the data)

Use columns 65 through 70 to check the condition of the numeric fields. Use columns 69 and 70 to check the condition of an alphameric field. These columns cannot be used for a data structure. The three conditions are:

- Plus (columns 65 and 66). Any valid indicator entered in columns 65 and 66 turns on if the numeric field named in columns 53 through 58 is greater than zero.
- Minus (columns 67 and 68). Any valid indicator entered in columns 67 and 68 turns on if the numeric field in columns 53 through 58 is less than zero.
- Zero or blank (columns 69 and 70). Any valid indicator entered in columns 69 and 70 turns on if a numeric field named in columns 53 through 58 is all zeros or if an alphameric field is all blanks. A numeric field that is all blanks turns on an indicator specified for zeros. However, if an alphameric field is all zeros, the field does not turn on the indicator specified for all blanks.

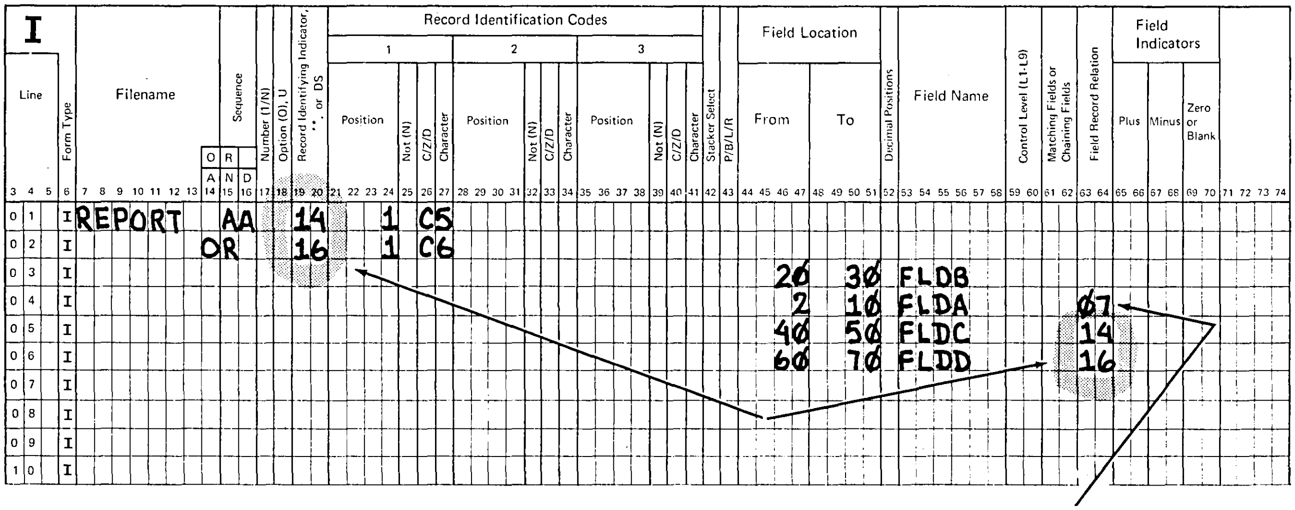


Record identification code = 5



Record identification code = 6

The file contains two different types of records, one identified by ε 5 in position 1 and the other by a 6 in position 1. FLDC is related by record identifying indicator 14 to the record type identified by a 5 in position 1. FLDD is related to the record type having a 6 in position 1 by record identifying indicator 16. This means that FLDC is found on only one type of record (that identified by 5 in position 1) and FLDD is found only on the other type. FLDA is conditioned by indicator 07, which was previously defined elsewhere in the program. FLDB is found on both types because they are not related to any one type by a record identifying indicator.



I Line	Form Type	Filename	Sequence	Number (1/N)	Option (O), U	Record Identifying Indicator, **, or DS	Record Identification Codes 1 Position	Not (N)	C/Z/D	Character	2 Position	Not (N)	C/Z/D	Character	3 Position	Not (N)	C/Z/D	Character	Stacker Select	P/B/L/R	Field Location From	To	Decimal Positions	Field Name	Control Level (L1-L9)	Matching Fields or Chaining Fields	Field Record Relation	Field Indicators Plus	Minus	Zero or Blank	
3 4 5	6	7 8 9 10 11 12 13	14 15 16	17	18	19 20	21 22 23 24	25	26	27	28 29 30 31	32	33	34	35 36 37 38	39	40	41	42	43	44 45 46 47	48 49 50 51	52	53 54 55 56 57 58	59 60	61 62	63 64	65 66	67 68	69 70	71 72 73 74
01	I	REPORT	AA			14	1		C	5																					
02	I		OR			16	1		C	6																					
03	I																				2Ø	3Ø		FLDB							
04	I																				2	1Ø		FLDA			Ø7				
05	I																				4Ø	5Ø		FLDC			14				
06	I																				6Ø	7Ø		FLDD			16				
07	I																														
08	I																														
09	I																														
10	I																														

This indicator was specified elsewhere in the program, and FLDA is made available for processing only when indicator 07 is set on elsewhere in the program.

Figure 7-26. Field Record Relation

Assigning Indicators in Columns 65-70

The following considerations apply to numeric indicators and halt indicators:

- Indicators for plus, minus, zero, or blank are off at the beginning of the program. They are not turned on until the condition (plus, minus, zero, or blank) is satisfied by the field being tested on the record just read.
- Columns 65 through 70 must be blank when table or array names are specified in input specifications. However, an entry can be made for an array element.
- A numeric input field can be assigned two or three field indicators. However, only the indicator that signals the result of the test on that field turns on; the others remain off.
- If the same field indicator is assigned to fields in different record types, its status is always based on the last record type selected.
- When different field indicators are assigned to fields in different record types, a field indicator turned on remains on until another record of that type is read. Similarly, a field indicator assigned to more than one field within a single record type always reflects the status of the last field defined.

Field indicators assigned in these columns can also be set on or set off by SETON or SETOF operations in the calculation specifications.

Numeric Indicators (01-99)

Use numeric indicators 01 through 99 to test a field for a condition of either plus, minus, zero, or blank. The indicator specified turns on if the condition is true; it remains off or turns off if the condition is not true. Usually these same indicators are used to control certain calculation or output operations. See *Columns 9-17 (Indicators)* in Chapter 8, *Calculation Specifications* or *Columns 23-31 (Output Indicators)* in Chapter 9, *Output Specifications.*

Halt Indicators (H1-H9)

Specify any halt indicator (H1 through H9) in columns 65 through 70 to check for an error condition in your data. For example, if a field should not be zero, specify a halt indicator to check for that zero condition. If a zero field is found, the halt indicator turns on and the program stops after the record with the zero field has been processed.

Indicators H1 through H9 cause the program to halt after the cycle that caused the indicator to turn on is complete (all calculations and output for that cycle are complete). The operator can restart the system by responding to the system halt.

COLUMNS 71-74

Columns 71 through 74 are not used. Leave them blank.

COLUMNS 75-80 (PROGRAM IDENTIFICATION)

See *Common Entries* in Chapter 1.

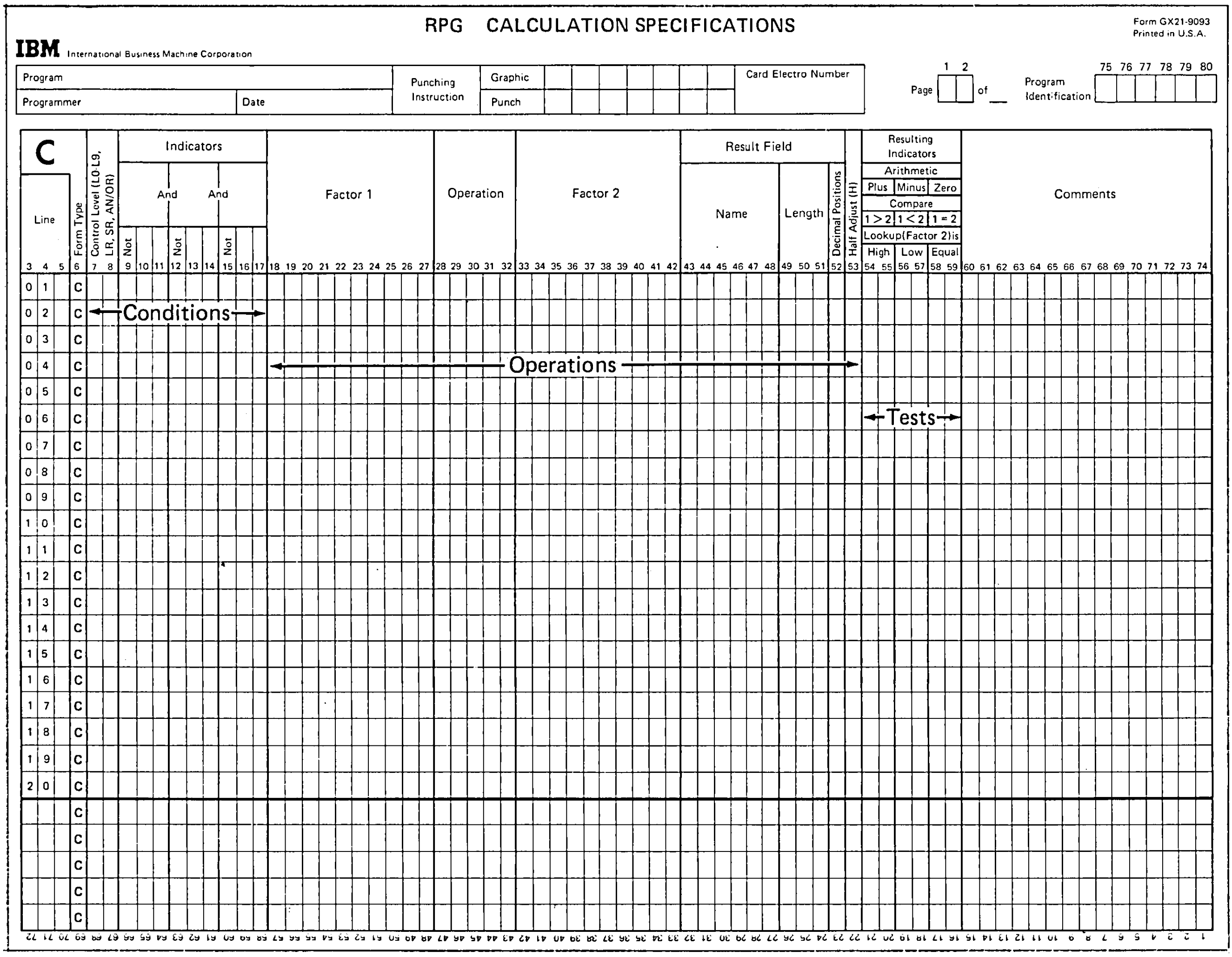
Chapter 8. Calculation Specifications

Calculation specifications describe the calculations you want performed on your data and the order in which you want them performed. Each calculation specifications statement can be divided into three parts:

- When the operation is to be performed (columns 7 through 17). The indicators entered in these columns determine under what conditions the specified operation is to be done.
- What kind of operation is to be performed (columns 18 through 53). Entries in these fields describe the kind of operation to be done and specify the data upon which the operation is to be performed.
- What tests are to be made on the results of the operation (columns 54 through 59). The indicators entered in these columns signal the result of the operation and can be used to condition other operations.

Calculation specifications must be specified in the following order: detail, total, subroutine.

Write these specifications on the RPG Calculation Specifications sheet (see Figure 8-1).



RPG CALCULATION SPECIFICATIONS

Form GX21-9093
Printed in U.S.A.

IBM International Business Machine Corporation

Program		Punching Instruction	Graphic
Programmer	Date		Punch

Card Electro Number

Page 1 2 of ___

Program Identification 75 76 77 78 79 80

C Line (3 4 5)	Form Type (6)	Control Level (L0-L9, LR, SR, AN/OR) (7 8)	Indicators: Not, And Not, And Not (9–17)	Factor 1 (18–27)	Operation (28–32)	Factor 2 (33–42)	Result Field Name (43–48)	Result Field Length (49–51)	Decimal Positions (52)	Half Adjust (H) (53)	Resulting Indicators: Arithmetic Plus Minus Zero; Compare 1>2 1<2 1=2; Lookup(Factor 2)is High Low Equal (54–59)	Comments (60–74)
01	C											
02	C	←Conditions→										
03	C											
04	C			←Operations→								
05	C											
06	C										←Tests→	
07	C											
08	C											
09	C											
10	C											
11	C											
12	C											
13	C											
14	C											
15	C											
16	C											
17	C											
18	C											
19	C											
20	C											
	C											
	C											
	C											
	C											
	C											

Figure 8-1. RPG Calculation Specifications

COLUMNS 1-2 (PAGE)

See *Common Entries* in Chapter 1.

COLUMNS 3-5 (LINES)

See *Common Entries* in Chapter 1.

COLUMN 6 (FORM TYPE)

A C must appear in column 6 to identify this line as a calculation specifications statement.

COLUMNS 7-8 (CONTROL LEVEL)

Entry	**Explanation**
Blank	Calculation operation is done at detail calculation time for each program cycle if the indicators in columns 9 through 17 allow it; or calculation is part of a subroutine.
L0	Calculation operation is done at total calculation time for each program cycle after total calculation processing has started. *Note:* If no control levels are specified on the input specifications, total calculation time processing starts during the second program cycle. If control levels are specified on the input specifications, total calculation time processing starts during the program cycle after the first record containing control fields is processed or at LR time. Totals are always processed at LR time.
L1-L9	Calculation operation is done when the appropriate control break occurs at total calculation time.
LR	Calculation operation is done after the last record has been processed.
SR	Calculation operation is part of a subroutine. A blank entry is also valid for calculations that are part of a subroutine.
AN, OR	Establishes AND and OR relationships between lines of indicators.

Use columns 7 and 8:

- To perform total calculation operations when the appropriate control break occurs.
- To perform calculation operations that are done only after the last record has been read.
- To indicate that an operation is part of a subroutine. However, columns 7 and 8 can also be blank for calculations that are part of a subroutine.
- To specify that certain lines of indicators are in an AND/OR relationship.

Control Level Indicators (L0, L1-L9)

The L0 indicator is on during the entire program. You need never assign this indicator, but you can use it to condition operations, especially when no control fields have been assigned. When a control break occurs, all operations conditioned by control level indicators are done before those that are not conditioned. If no control field is assigned, but total calculations are to be done and total output records are to be written, use the L0 indicator to condition those operations (see Figure 8-2).

Use a control level indicator (L1 through L9) to specify that the operation described on the same specification line is done only when that indicator is on. A control level indicator turns on when information in a control field changes. See *Columns 59-60 (Control Level)* in Chapter 7, *Input Specifications.*

A control break for a certain level turns on all lower control level indicators. Thus, if indicators L3, L2, and L1 are used in a program and L3 turns on, L1 and L2 also turn on. All operations conditioned by L3, L2, and L1 are done.

However, when a control level indicator used as a record identifying indicator turns on to indicate the type of record read or when the SETON operation turns on a control level indicator, only that one control level indicator turns on. All lower level indicators remain unchanged.

Note: In one program cycle, all operations conditioned by control level indicators in columns 7 and 8 are done at total calculation time. Operations that are conditioned by control level indicators in columns 9 through 17 are done at detail calculation time immediately following the control break (see *Relationship Between Columns 7-8 and Columns 9-17* in this chapter).

Last Record Indicator (LR)

For a primary file, RPG II sets on the last record (LR) indicator after the last record is read and processed (end of file has occurred). For a WORKSTN file specified as a primary file, end of file, which causes RPG II to set on the LR indicator, occurs:

- When all display stations have been released (by an R in column 16 of the output specifications or by the REL operation code) if the program does not have an NEP attribute.
- When all display stations have been released and the operator has entered the STOP SYSTEM command if the program has an NEP attribute.

However, under the following conditions, the programmer must set on the LR indicator:

- If the program contains no primary file
- If KEYBORD is specified as the device for a primary input file

If certain operations are to be done only after the last record is read, condition these operations with the LR indicator. Specify the operations conditioned by LR after all calculations conditioned by L0 through L9 (columns 7 and 8) or after detail calculations if there are no total calculations. The last record turns on the LR indicator and all other control level indicators specified (L1 through L9).

The input records have ITEM and COST fieıαs and a one-position record identification field. The records are grouped in ascending sequence by district; that is, the district 1 records as a group are followed by a blank record, and the district 2 records as a group are followed by a blank record.

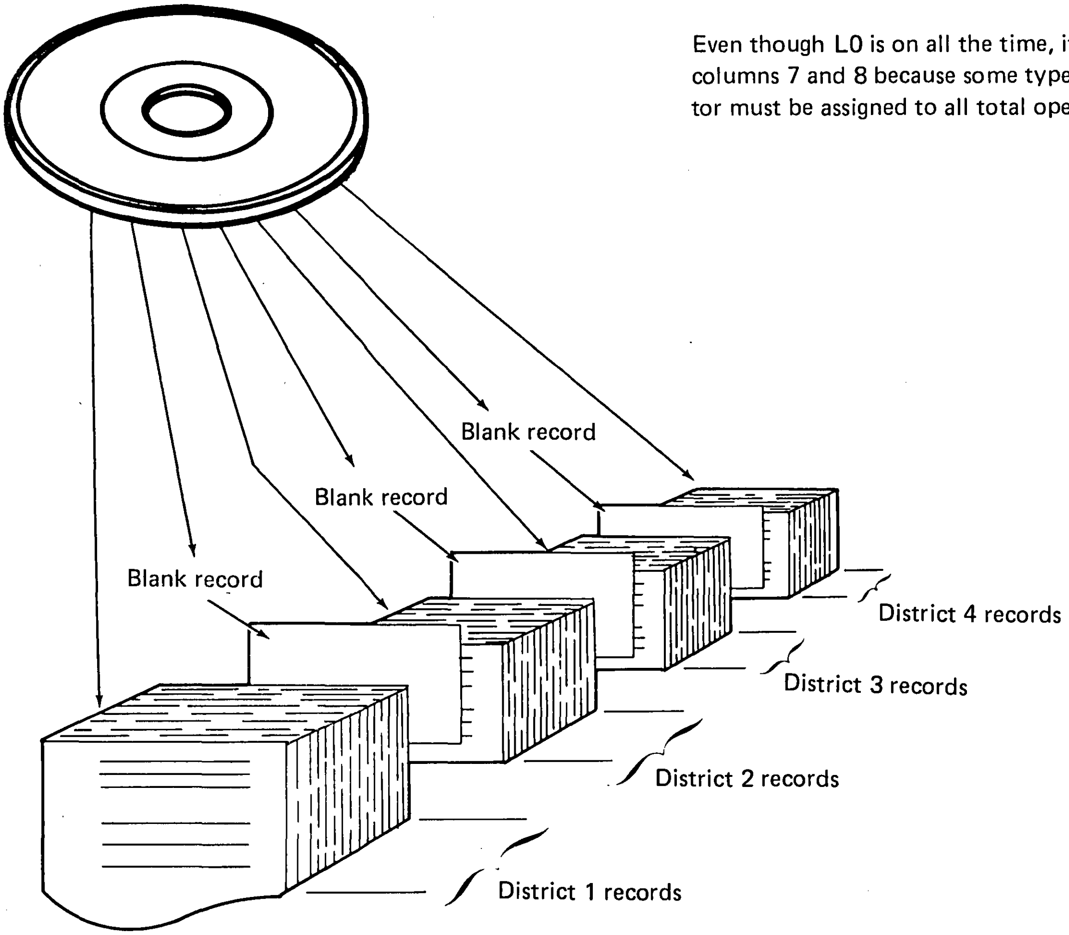


Figure 8-2 (Part 1 of 2). Use of the L0 Indicator

No field can serve as a control field because the district number is not on the records. Instead of a control field, the blank record is used to signal a new district. When the blank record is read, indicator 02 turns on. The blank record tells the program that total calculations and total output operations must be done. However, no total operations can be performed unless they are conditioned by some kind of control level indicator.

Even though L0 is on all the time, it must be used in columns 7 and 8 because some type of control level indicator must be assigned to all total operations.

Input Specifications

Line	Form Type	Filename	Sequence	Record Identifying Indicator	Position	C/Z/D	Character	Field Location From	To	Decimal Positions	Field Name
01	I	INPUT	AA	Ø1	1	C	1				
02	I							1Ø	15	2	COST
03	I							2Ø	3Ø		ITEM
04	I		BB	Ø2	1	C					
05	I										

The program shows how total operations can be performed even though there is no control field (no L1 through L9 indicators).

The program requires:

- A list of items sold in each district
- A total of all sales for each district
- A grand total of all sales in all districts

Calculation Specifications

Line	Form Type	Control Level	Indicators	Factor 1	Operation	Factor 2	Result Field Name	Length	Decimal Positions
01	C		Ø1	COST	ADD	DISTOT	DISTOT	6	2
02	C								
03	C								
04	C								
05	C	LØ	Ø2	DISTOT	ADD	GDTOT	GDTOT	6	2
06	C	LR		DISTOT	ADD	GDTOT	GDTOT		
07	C								

Output Specifications

Line	Form Type	Filename	AND/OR	Type (H/D/T/E)	Space Before	Space After	Output Indicators	Field Name	Edit Codes	Blank After	End Position in Output Record	Constant or Edit Word
01	O	OUT		D		1	Ø1					
02	O							ITEM			3Ø	
03	O							COST	1		5Ø	
04	O			T	2	2	Ø2					
05	O		OR				LR					
06	O							DISTOT	1	B	5Ø	
07	O										51	'*'
08	O			T		2	LR					
09	O							GDTOT	1		5Ø	
10	O										52	'**'
11	O											

Format of the Printed Report

J102	4.50	
J202	3.75	
K450	2.98	
B231	9.08	
	20.31	*
G10H	92.79	
G10K	98.89	
A126	4.29	
	195.97	*
	216.28	**

Figure 8-2 (Part 2 of 2). Use of the L0 Indicator

Subroutine Lines (SR)

An SR entry in columns 7 and 8 indicates that this specification line is part of a subroutine (see *Subroutine Operations* in Chapter 10, *Operation Codes*). The SR entry is not required for a calculation specification line that is part of a subroutine; blanks in columns 7 and 8 are also valid.

Subroutine lines must be specified last.

AND/OR Lines (AN, OR)

Use columns 7 and 8 to specify that lines of indicators are in an AND/OR relationship. When you use the AND/OR relationship, many lines of indicators can be grouped together to condition an operation. A maximum of seven OR lines or seven AND lines or any combination thereof can condition an operation.

The first line of such a group contains blanks in columns 7 and 8 or an L0 through L9, LR, or SR entry if the group of lines is conditioned by a control level indicator or is part of a subroutine. (This entry on the first line applies to all AND/OR lines that follow.) All lines after the first line in the group must have an AN or OR entry in columns 7 and 8. The last line of the group contains the operation and the necessary operands. All lines except the last line in the group must contain blanks in columns 18 through 59 (see Figure 8-3).

COLUMNS 9-17 (INDICATORS)

Entry	Explanation
Blank	Operation is performed on every program cycle.
01-99	Field indicators, record identifying indicators, or resulting indicators assigned elsewhere in the program.
KA-KN, KP-KY	Command key indicators assigned elsewhere.
L1-L9	Control level indicators assigned elsewhere.
LR	Last record indicator.
MR	Matching record indicator.
H1-H9	Halt indicators assigned elsewhere.
U1-U8	External indicators previously set.
OA-OG, OV	Overflow indicator previously assigned.

Use columns 9 through 17 to assign indicators that control the conditions under which an operation is done. You can use from one to three separate fields (columns 10 and 11, 13 and 14, and 16 and 17) on each line, one for each indicator. If the indicator must be off to condition the operation, place an N before the appropriate indicator (columns 9, 12, 15).

The indicators specified in columns 9 through 17 on one specification line are in an AND relationship with each other. The indicators on one line or indicators in grouped lines plus the control level indicator (if used in columns 7 and 8) must all be exactly as specified before the operation is done (see Figure 8-4).

An indicator that is specified in columns 9 through 17 of a calculation specification can also be entered as a resulting indicator on the same line. If the indicator in columns 9 through 17 is on, the calculation is performed. If the calculation is executed, the present setting of the indicator is determined.

Field Indicators (01-99)

Use any field indicators that are specified in columns 65 through 70 on the input specifications to condition an operation that is to be done only after the status of a field has been checked and has met certain conditions (see Figure 8-5).

Command Key Indicators (KA-KN, KP-KY)

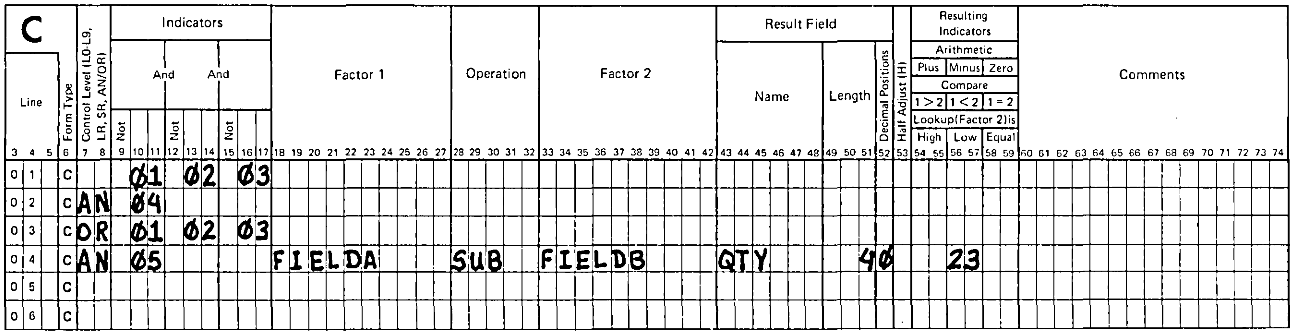
Use any command key indicators specified in columns 54 through 59 of the calculation specifications for a SET or SETOF operation. See *SET* or *SETOF* in Chapter 10, *Operation Codes*, for complete information on each operation.

All command keys are defined for a WORKSTN file. When RPG makes the data read from a display station available for processing, all command key indicators are set off. If a command key was pressed when the data was read into the program, the corresponding command key indicator is set on.

Record Identifying Indicators (01-99)

Use any record identifying indicators that are specified in columns 19 and 20 of the input specifications to condition an operation that is to be done only for a certain type of record (see Figure 8-6).

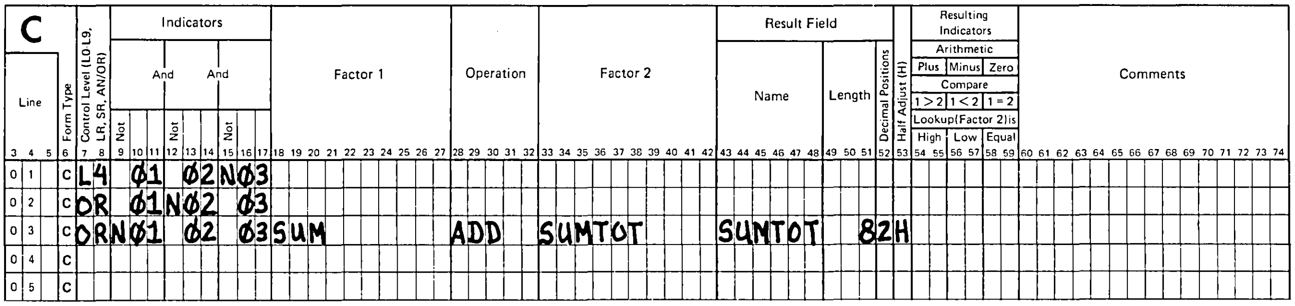
Calculation Specifications



C Line	Form Type	Control Level (L0-L9, LR, SR, AN/OR)	Indicators						Factor 1	Operation	Factor 2	Result Field				Resulting Indicators			Comments
					And		And					Name	Length	Decimal Positions	Half Adjust (H)	Arithmetic Plus Compare 1>2 Lookup(Factor 2)is High	Minus 1<2 Low	Zero 1=2 Equal	
			Not		Not		Not												
3 4 5	6	7 8	9	10 11	12	13 14	15	16 17	18 19 20 21 22 23 24 25 26 27	28 29 30 31 32	33 34 35 36 37 38 39 40 41 42	43 44 45 46 47 48	49 50 51	52	53	54 55	56 57	58 59	60 61 62 63 64 65 66 67 68 69 70 71 72 73 74
0 1	C			Ø1		Ø2		Ø3											
0 2	C	AN		Ø4															
0 3	C	OR		Ø1		Ø2		Ø3											
0 4	C	AN		Ø5					FIELDA	SUB	FIELDB	QTY	4	Ø			23		
0 5	C																		
0 6	C																		

AN and OR entries group lines of indicators. When indicators 01, 02, 03, and 04 are on, or when indicators 01, 02, 03, and 05 are on, the calculation is performed.

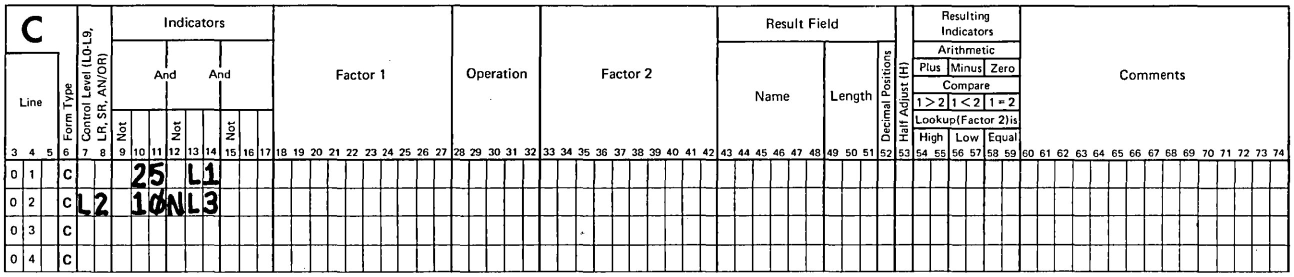
Calculation Specifications



C Line	Form Type	Control Level (L0-L9, LR, SR, AN/OR)	Indicators						Factor 1	Operation	Factor 2	Result Field				Resulting Indicators			Comments
					And		And					Name	Length	Decimal Positions	Half Adjust (H)	Arithmetic Plus Compare 1>2 Lookup(Factor 2)is High	Minus 1<2 Low	Zero 1=2 Equal	
			Not		Not		Not												
3 4 5	6	7 8	9	10 11	12	13 14	15	16 17	18 19 20 21 22 23 24 25 26 27	28 29 30 31 32	33 34 35 36 37 38 39 40 41 42	43 44 45 46 47 48	49 50 51	52	53	54 55	56 57	58 59	60 61 62 63 64 65 66 67 68 69 70 71 72 73 74
0 1	C	L4		Ø1		Ø2	N	Ø3											
0 2	C	OR		Ø1	N	Ø2		Ø3											
0 3	C	OR	N	Ø1		Ø2		Ø3	SUM	ADD	SUMTOT	SUMTOT	8	2	H				
0 4	C																		
0 5	C																		

Three conditions cause the L4 total calculations to be performed: 01 and 02 are on, but not 03; or 01 and 03 are on, but not 02; or 02 and 03 are on, but not 01.

Figure 8-3. Use of AND/OR Lines for Indicators



C			Indicators						Factor 1	Operation	Factor 2	Result Field				Resulting Indicators			Comments
			And		And											Arithmetic			
																Plus	Minus	Zero	
																Compare			
																1 > 2	1 < 2	1 = 2	
																Lookup(Factor 2)is			
Line	Form Type	Control Level (L0-L9, LR, SR, AN/OR)	Not		Not		Not					Name	Length	Decimal Positions	Half Adjust (H)	High	Low	Equal	
3 4 5	6	7 8	9	10 11	12	13 14	15	16 17	18 19 20 21 22 23 24 25 26 27	28 29 30 31 32	33 34 35 36 37 38 39 40 41 42	43 44 45 46 47 48	49 50 51	52	53	54 55	56 57	58 59	60 61 62 63 64 65 66 67 68 69 70 71 72 73 74
0 1	C			25		L1													
0 2	C	L2		1Ø	N	L3													
0 3	C																		
0 4	C																		

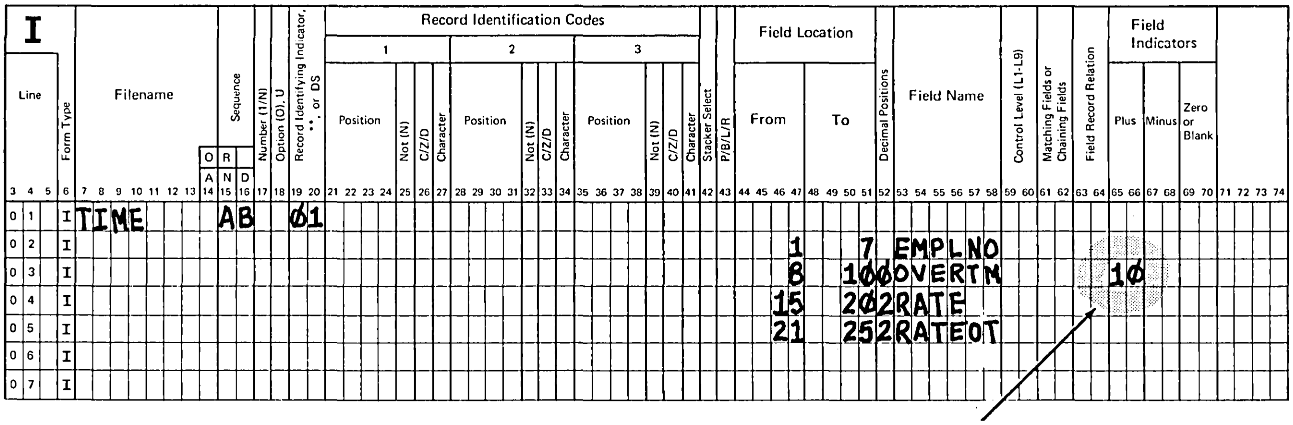
Assume that indicator 25 represents a record type and that a control level 2 break occurred when record type 25 was read. L1 and L2 are both on. All operations conditioned by the control level indicators in columns 7 and 8 are performed before operations conditioned by control level indicators in columns 9 through 17. Thus, the operation in line 02 occurs before the operation in line 01. The operation in line 01 is done on the first record of the new control group indicated by 25, whereas the operation in line 02 is a total operation done for all records of the previous control group.

The operation in line 02 can be done when the L2 indicator is on provided the other conditions are met. Indicator 10 must be on. The L3 indicator must not be on.

The operation conditioned by both L2 and NL3 is done only when a control level 2 break occurs. These two indicators are used together because this operation is not to be done when a control level 3 break occurs, even though L2 is also on.

Figure 8-4. Conditioning Operations (Control Level Indicators)

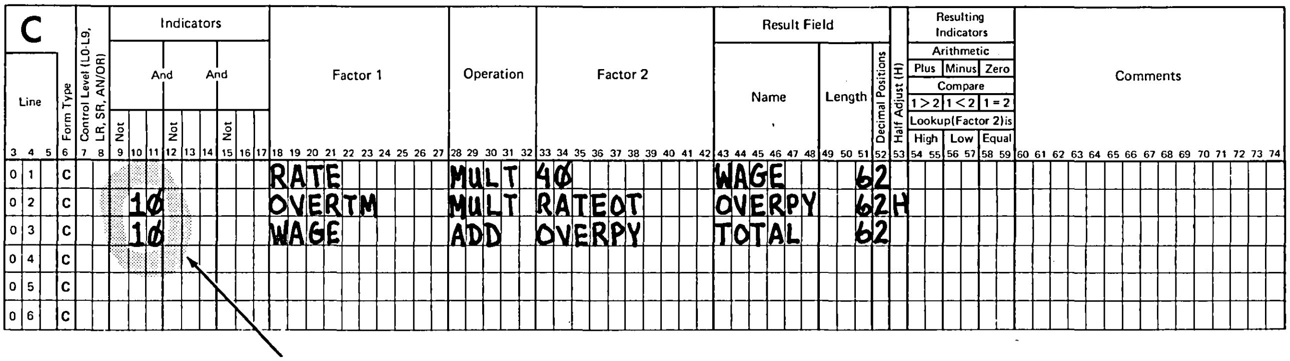
Input Specifications



Line	Form Type	Filename	Sequence	Number (1/N)	Option (O), U	Record Identifying Indicator, **, or DS	Record Identification Codes	Stacker Select	P/B/L/R	Field Location From	Field Location To	Decimal Positions	Field Name	Control Level (L1-L9)	Matching Fields or Chaining Fields	Field Record Relation	Field Indicators Plus	Field Indicators Minus	Field Indicators Zero or Blank
01	I	TIME	AB			Ø1													
02	I									1	7		EMPLNO						
03	I									8	1Ø	Ø	OVERTM				1Ø		
04	I									15	2Ø	2	RATE						
05	I									21	25	2	RATEOT						
06	I																		
07	I																		

Field indicators can be used to condition operations. Assume the program is to find weekly earnings including overtime. The overtime field is checked to determine whether any overtime was put in. If the employee has worked overtime, the field is positive and indicator 10 turns on. In all cases the weekly regular wage is calculated. However, overtime pay is calculated only if indicator 10 is on (calculation lines 02 and 03).

Calculation Specifications

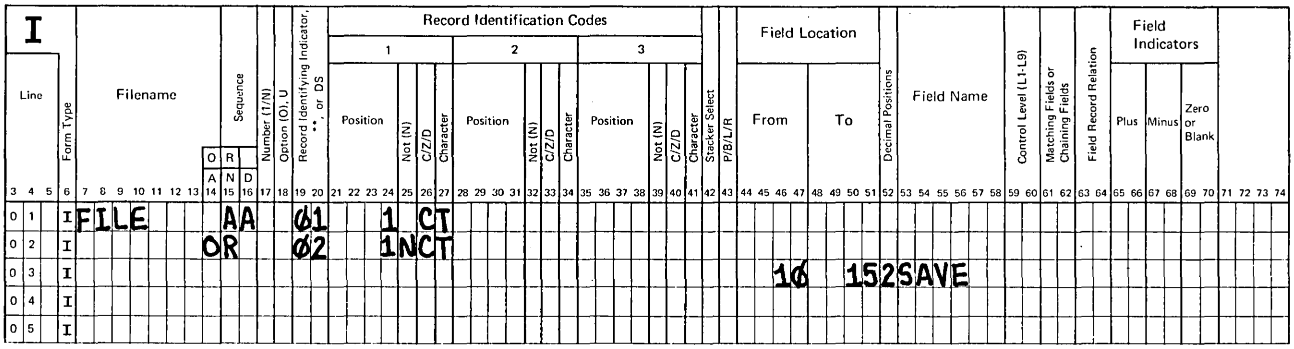


Line	Form Type	Control Level (L0-L9, LR, SR, AN/OR)	Indicators	Factor 1	Operation	Factor 2	Result Field Name	Length	Decimal Positions	Half Adjust (H)	Resulting Indicators	Comments
01	C			RATE	MULT	4Ø	WAGE	6	2			
02	C		1Ø	OVERTM	MULT	RATEOT	OVERPY	6	2	H		
03	C		1Ø	WAGE	ADD	OVERPY	TOTAL	6	2			
04	C											
05	C											
06	C											

Field indicator 10 was assigned on the input specifications. It is being used here to condition calculation operations.

Figure 8-5. Conditioning Operations (Field Indicators)

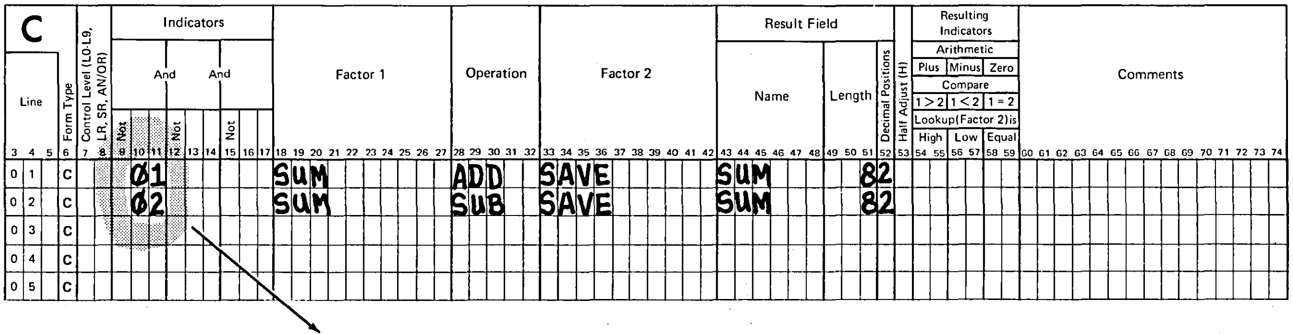
Input Specifications



```
01 IFILE    AA  Ø1   1 CT
02 I       OR   Ø2   1NCT
03 I                                       1Ø  152SAVE
04 I
05 I
```

A record identifying indicator is used to condition an operation. When a record is read that has a T in position 1, the 01 indicator turns on. If this indicator is on, the field named SAVE is added to SUM. When a record having no T in position 1 is read, the 02 indicator is on. The subtract operation, conditioned by 02, is then done instead of the add operation.

Calculation Specifications



```
01 C   Ø1      SUM       ADD  SAVE      SUM     82
02 C   Ø2      SUM       SUB  SAVE      SUM     82
03 C
04 C
05 C
```

Record identifying indicators 01 and 02 are assigned on the input specifications. They are used here to condition calculation operations.

Figure 8-6. Conditioning Operations (Resulting Indicators)

Resulting Indicators (01-99)

Use any resulting indicators specified in columns 54 through 59 on the calculation specifications to condition operations according to the results of calculation operations. See *Columns 54-59 (Resulting Indicators)* in this chapter.

Control Level Indicators (L1-L9)

Use any control level indicators that are specified in columns 59 and 60 of the input specifications or in columns 54 through 59 of the calculation specifications. If control level indicators are used in these columns but not in columns 7 and 8, the operation is performed at detail calculation time on the first record of a new control group or whenever the indicators are on.

Last Record Indicator (LR)

To condition operations to be performed at end of job, use the last record (LR) indicator in columns 9 through 17 only if LR is turned on during calculations. All operations to be performed at end of job should be conditioned by LR in columns 7 and 8.

Matching Record Indicator (MR)

Use the matching record (MR) indicator to condition an operation that is to be done only when matching records are found. See *Columns 61-62 (Matching Fields)* in Chapter 7, *Input Specifications* for more information on matching fields.

Halt Indicators (H1-H9)

Use any halt indicators that are specified in columns 65 through 70 on the input specifications or in columns 54 through 59 on the calculation specifications to prevent the operation from being done when a specified error condition is found in the input data or during calculations. See *Columns 19-20 (Record Identifying Indicator)* in Chapter 7, *Input Specifications.* Using a halt indicator is necessary because the record that causes the halt condition is completely processed before the program stops. Thus, if the operation is performed on an error condition, the results are in error. A halt indicator can also be used to condition an operation that is to be done only when an error occurs.

Note: The system message displayed on a halt can be overridden by a user message member. (See *User Message Member* in Chapter 10, *Operation Codes.*)

External Indicators (U1-U8)

Use any external indicator previously specified to condition which operations should be done and which files should be used for a specific job. If a file is conditioned by an external indicator, any calculations that are to be performed only on that file should be conditioned by the same external indicator.

Overflow Indicators (OA-OG, OV)

Use any overflow indicators that are specified in columns 33 and 34 of the file description specifications to condition operations that are to be done when the last line to be printed on a page is reached. See *Overflow Indicators* in Chapter 9, *Output Specifications,* for more information.

Relationship Between Columns 7-8 and Columns 9-17

In one program cycle, all operations conditioned by control level indicators in columns 7 and 8 (total time) are done before operations conditioned by control level indicators in columns 9 through 17 (see Figure 8-4).

When a control level indicator is used in columns 9 through 17 and columns 7 and 8 are not used (detail time), the operation conditioned by the indicator is done only on the record that causes a control break or any higher-level control break.

When a control level indicator (L1 through L9) is specified in columns 7 and 8 (total time) and MR is specified in columns 9 through 17, MR indicates the matching condition of the previous record and not the one just read that caused the control break. After all operations conditioned by control level indicators (specified in columns 7 and 8 of the calculation specifications) are done, MR then indicates the matching condition of the record just read.

COLUMNS 18-27 (FACTOR 1) AND COLUMNS 33-42 (FACTOR 2)

Use columns 18 through 27 and 33 through 42 to name the fields or to give the actual data (literals) on which an operation is to be performed. See Figure 8-7 for a summary of the operation codes.

The entries you can use for factor 1 and factor 2 are:

- The name of any field that has been defined
- Any alphameric or numeric literal
- Any subroutine, table, array name, or array element
- Any date field name (UDATE, UMONTH, UDAY, UYEAR)
- The special names PAGE, PAGE1, PAGE2, PAGE3, PAGE4, PAGE5, PAGE6, or PAGE7
- Any figurative constant (*BLANK, *BLANKS, *ZERO, *ZEROS)

The following restrictions apply to entries in factor 1 and factor 2:

- A data structure name *cannot* be specified in factor 1 or factor 2.
- A data structure subfield name can be used in factor 1 or factor 2; however, overlapping subfields in a data structure cannot be used in the same calculation. A subfield is considered to be an overlapping subfield if its from or to position occurs within the from and to positions of another subfield within the same data structure. If factor 1, factor 2, or the result field references a subfield in a data structure that is an array or array element with a variable index, the entire array is used to determine whether overlap exists. The same array name can be referenced in the appropriate factors of a calculation specification without violating the overlap rule. See Figure 8-8 for examples of the overlap rule.
- Figurative constants cannot be used with move zone operations, bit operations, or the SET, KEY, SQRT, or DEBUG operation codes.

The following entry can be made for factor 1 only:

- A label for a TAG, BEGSR, or ENDSR operation

The following entries can be made for factor 2 only:

- A label for a GOTO or EXSR operation
- A filename for a SET, CHAIN, DEBUG, READ, FORCE, ACQ, REL, or NEXT operation
- A subroutine name for an EXIT operation
- An array name for an SORTA operation.

An entry in factor 1 must begin in column 18; an entry in factor 2 must begin in column 33.

Entries for factor 1 and factor 2 depend upon the operation code used in columns 28 through 32. Some operations require entries in both factors, some require entries in only one, and some require no entries at all. See *Columns 28-32 (Operation)* for more information on operation codes. For information on how to name a subroutine, see *Subroutine Operations* in Chapter 10, *Operation Codes.*

Literals

A literal is the actual data used in an operation rather than the field name representing that data. A literal can be either alphameric or numeric.

Alphameric Literals

Consider the following rules when using an alphameric literal (see Figure 8-9):

- Any combination of characters can be used in an alphameric literal. Blanks are also valid.
- The maximum length of an alphameric literal is 8 characters.
- Alphameric literals must be enclosed in apostrophes (').
- An apostrophe required as part of a literal is represented by two apostrophes. For example, the literal O'CLOCK is coded as 'O''CLOCK'.
- Alphameric literals cannot be used for arithmetic operations.

Numeric Literals

Consider the following rules when using a numeric literal (see Figure 8-9):

- A numeric literal consists of any combination of the digits 0 through 9. A decimal point or sign can also be included.
- The sign (+ or -), if present, must be the leftmost character. An unsigned literal is treated as a positive number.
- The maximum total length of a numeric literal is 10 characters including the sign and decimal point.
- Blanks cannot appear in a numeric literal.
- Numeric literals must not be enclosed in apostrophes (').
- Numeric literals are used in the same way as a numeric field.

Figurative Constants

The figurative constants *BLANK, *BLANKS, *ZERO, and *ZEROS can be specified as literals. The following rules apply for figurative constants:

- The figurative constants *BLANK and *BLANKS can only be used with alphameric fields.
- The figurative constants *ZERO and *ZEROS can be used with either alphameric or numeric fields.
- The length of the figurative constant is assumed to be equal to the length of the other factor field, if present. Otherwise, the length of the figurative constant is assumed to be equal to the length of the result field.
- Figurative constants are considered to be elementary items, and, if used in conjunction with an array, act like a field. For example:

  ```
  MOVE *ZEROS  ARR
  ```

  If ARR has 4-character elements, each element of ARR contains '0000' after the move is executed.
- After a figurative constant has its length set, the figurative constant's logical placement in the collating sequence can be altered by specifying an alternate collating sequence.

	Control Level Indicators	Conditioning Indicators				Resulting Indicators		
	Columns					Columns		
Operation Code	7-8	9-17	Factor 1	Factor 2	Result Field	54-55	56-57	58-59
ACQ	O	O	R	R			O	
ADD	O	O	O	R	R	O	O	O
BEGSR	SR or blank		R					
BITOF	O	O		R	R			
BITON	O	O		R	R			
CHAIN	O	O	R	R		O		
COMP	O	O	R	R		O[3]	O[3]	O[3]
DEBUG	O	O	O	R	O			
DIV	O	O	O	R	R	O	O	O
ENDSR	SR or blank		O	O				
EXCPT	O	O						
EXIT	O	O		R				
EXSR	O	O		R				
FORCE		O		R				
GOTO	O	O		R			O	
KEYnn[1]	O	O	O		R	O	O	O
LOKUP (Array)	O	O	R	R		O[4]	O[4]	O[4]
LOKUP (Table)	O	O	R	R	O	O[4]	O[4]	O[4]
MHHZO	O	O		R	R			
MHLZO	O	O		R	R			
MLHZO	O	O		R	R			
MLLZO	O	O		R	R			

Figure 8-7 (Part 1 of 2). Operation Codes

	Control Level Indicators	Conditioning Indicators				Resulting Indicators		
	Columns					Columns		
Operation Code	7-8	9-17	Factor 1	Factor 2	Result Field	54-55	56-57	58-59
MOVE	O	O		R	R			
MOVEA	O	O		R	R			
MOVEL	O	O		R	R			
MULT	O	O	O	R	R	O	O	O
MVR	O	O			R	O	O	O
NEXT	O	O	R	R			O	
POST	O	O	R		R		O	
READ	O	O		R			O[2]	O
REL	O	O	R	R			O	
RLABL					R			
SETnn[1]	O	O	O	O		O	O	O
SETOF	O	O				O[3]	O[3]	O[3]
SETON	O	O				O[3]	O[3]	O[3]
SETLL	O	O	R	R				
SHTDN	O	O				R		
SORTA	O	O		R				
SQRT	O	O		R	R			
SUB	O	O	O	R	R	O	O	O
TAG	O		R					
TESTB	O	O		R	R	O[3]	O[3]	O[3]
TESTZ	O	O			R	O[3]	O[3]	O[3]
TIME	O	O			R			
XFOOT	O	O		R	R	O	O	O
Z-ADD	O	O		R	R	O	O	O
Z-SUB	O	O		R	R	O	O	O

[1] The nn entries in columns 31 and 32 are for message indicator numbers. If the result field of a SET operation contains the keyword ERASE, factor 2 must contain the name of the CONSOLE file. Otherwise, factor 2 and the result field must be blank.

[2] Columns 56 and 57 can contain an indicator when the READ operation is used with a WORKSTN device.

[3] At least one resulting indicator must be specified in columns 54 through 59.

[4] At least one resulting indicator must be specified in columns 54 through 59, but no more than two can be used.

Fields without entries must be blank.

O	=	Optional
R	=	Required
SR	=	The only allowable nonblank characters in columns 7 and 8 for the BEGSR and ENDSR operation codes.

Figure 8-7 (Part 2 of 2). Operation Codes

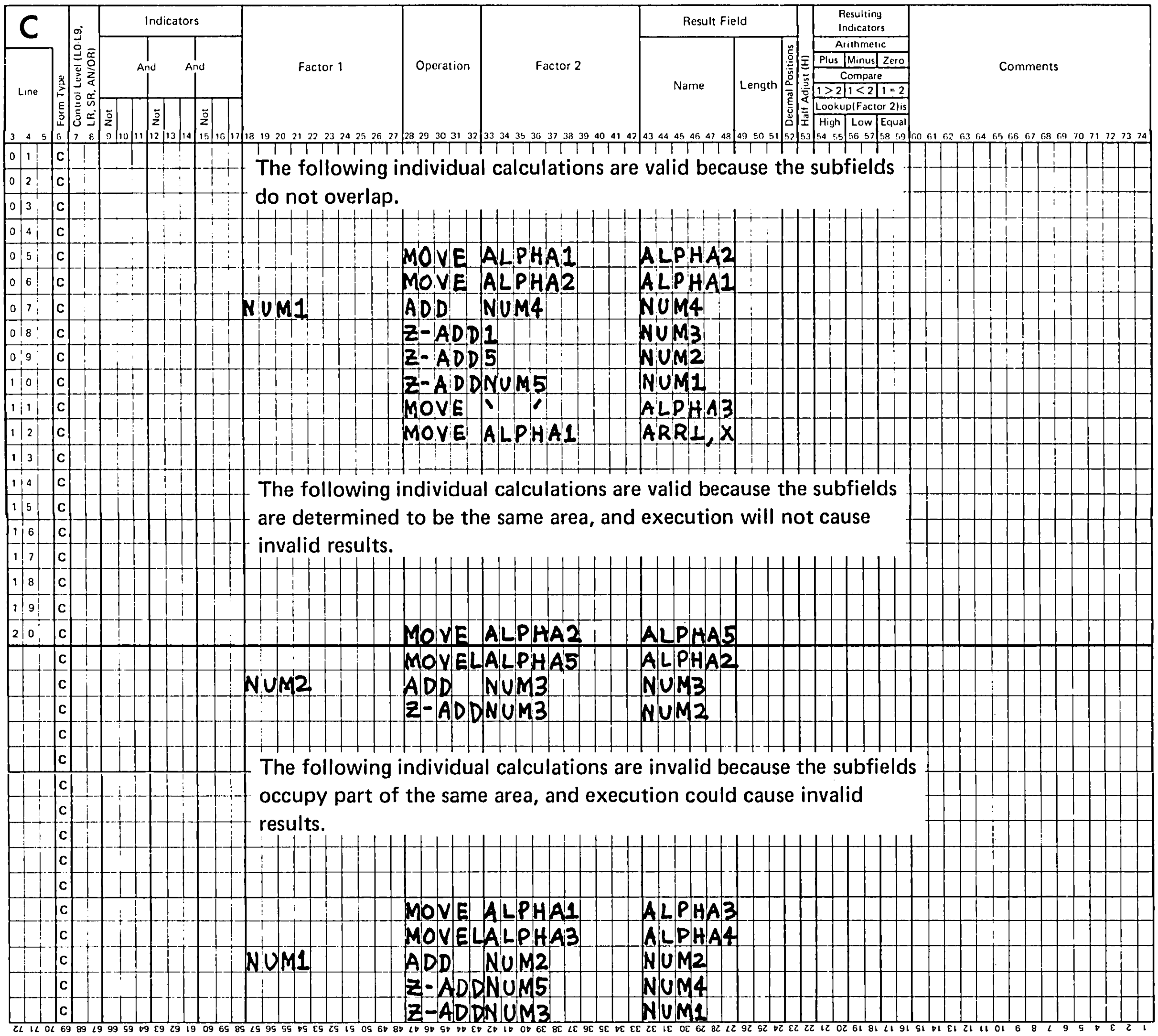
Input Specifications

Line	Form Type	Filename	Sequence	Number (1/N)	Option (O), U	Record Identifying Indicator, **, or DS	Record Identification Codes 1	Record Identification Codes 2	Record Identification Codes 3	Stacker Select	P/B/L/R	Field Location From	Field Location To	Decimal Positions	Field Name	Control Level (L1-L9)	Matching Fields or Chaining Fields	Field Record Relation	Field Indicators Plus	Field Indicators Minus	Field Indicators Zero or Blank
01	I	DATADS				DS															
02	I											1	5		ALPHA1						
03	I											6	1Ø		ALPHA2						
04	I											4	8		ALPHA3						
05	I											7	9		ALPHA4						
06	I											6	1Ø		ALPHA5						
07	I											11	14	Ø	NUM1						
08	I											11	15	Ø	NUM2						
09	I											11	15	Ø	NUM3						
10	I											15	2Ø	1	NUM4						
11	I											17	18	Ø	NUM5						
12	I											21	5Ø		ARR1						
13	I											41	7Ø		ARR2						
14	I																				
15	I																				
16	I																				
17	I																				
18	I																				
19	I																				
20	I																				
	I																				
	I																				
	I																				
	I																				
	I																				

The data structure DATADS contains subfields and arrays that are defined as overlapping (that is, occupying part of the same area). ARR1 (on line 12) has six elements, and each element is five positions long for a total length of 30. ARR2 (on line 13) has five elements, and each element is six positions long for a total length of 30.

Figure 8-8 (Part 1 of 3). Examples of Valid and Invalid Calculations with Overlapping Subfields in a Data Structure

Calculation Specifications


C
Line
Form Type
Control Level (L0-L9, LR, SR, AN/OR)
Indicators
And
And
Not
Not
Not
Factor 1
Operation
Factor 2
Result Field
Name
Length
Decimal Positions
Half Adjust (H)
Resulting Indicators
Arithmetic
Plus
Minus
Zero
Compare
1 > 2
1 < 2
1 = 2
Lookup(Factor 2)is
High
Low
Equal
Comments

The following individual calculations are valid because the subfields do not overlap.

```
  01 C
  02 C
  03 C
  04 C
  05 C                     MOVE ALPHA1    ALPHA2
  06 C                     MOVE ALPHA2    ALPHA1
  07 C           NUM1      ADD  NUM4      NUM4
  08 C                     Z-ADD1         NUM3
  09 C                     Z-ADD5         NUM2
  10 C                     Z-ADDNUM5      NUM1
  11 C                     MOVE '  '      ALPHA3
  12 C                     MOVE ALPHA1    ARR1,X
  13 C
```

The following individual calculations are valid because the subfields are determined to be the same area, and execution will not cause invalid results.

```
  14 C
  15 C
  16 C
  17 C
  18 C
  19 C
  20 C                     MOVE ALPHA2    ALPHA5
     C                     MOVELALPHA5    ALPHA2
     C           NUM2      ADD  NUM3      NUM3
     C                     Z-ADDNUM3      NUM2
     C
```

The following individual calculations are invalid because the subfields occupy part of the same area, and execution could cause invalid results.

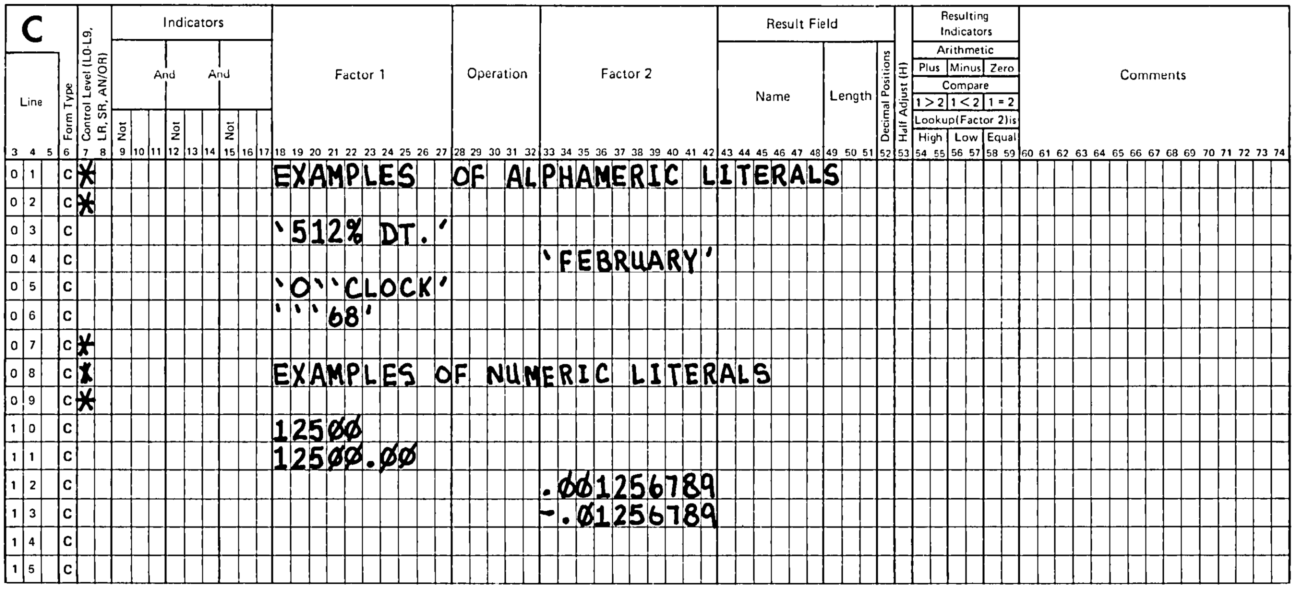
```
     C
     C
     C
     C
     C
     C                     MOVE ALPHA1    ALPHA3
     C                     MOVELALPHA3    ALPHA4
     C           NUM1      ADD  NUM2      NUM2
     C                     Z-ADDNUM5      NUM4
     C                     Z-ADDNUM3      NUM1
```

Figure 8-8 (Part 2 of 3). Examples of Valid and Invalid Calculations with Overlapping Subfields in a Data Structure

Line	Form Type	Control Level (L0-L9, LR, SR, AN/OR)	Indicators	Factor 1	Operation	Factor 2	Result Field Name	Length	Decimal Positions	Half Adjust (H)	Resulting Indicators	Comments
0 1	C		The following individual calculations involve the same array and are									
0 2	C		valid calculations.									
0 3	C											
0 4	C											
0 5	C				MOVE	ARR1,1	ARR1,5					
0 6	C				MOVE	ARR1,X	ARR1,2					
0 7	C				MOVE	ARR2,X	ARR2,Y					
0 8	C											
0 9	C		The following individual calculations are valid because the array									
1 0	C		elements associated with the constant indexes do not overlap.									
1 1	C											
1 2	C				MOVE	ARR1,1	ARR2,1					
1 3	C				MOVEL	ARR1,2	ARR2,5					
1 4	C											
1 5	C		The following individual calculations are invalid because the array									
1 6	C		elements associated with the constant indexes overlap, or variable									
1 7	C		indexes are specified and the entire array is required to determine									
1 8	C		overlap.									
1 9	C											
2 0	C											
	C				MOVE	ARR2,1	ARR1,5					
	C				MOVE	ARR1,X	ARR2,X					
	C				MOVE	ARR1,X	ARR2,1					
	C				MOVE	ARR1,1	ARR2,X					
	C											

Figure 8-8 (Part 3 of 3). Examples of Valid and Invalid Calculations with Overlapping Subfields in a Data Structure

Calculation Specifications



```
01 C*          EXAMPLES  OF ALPHAMERIC LITERALS
02 C*
03 C           '512% DT.'
04 C                          'FEBRUARY'
05 C           'O''CLOCK'
06 C           '''68'
07 C*
08 C*          EXAMPLES OF NUMERIC LITERALS
09 C*
10 C           125ØØ
11 C           125ØØ.ØØ
12 C                          .ØØ1256789
13 C                          -.Ø1256789
14 C
15 C
```

Figure 8-9. Alphameric and Numeric Literals

COLUMNS 28-32 (OPERATION)

Use columns 28 through 32 to specify the kind of operation to be performed using factor 1, factor 2, and/or the result field. The operation code must begin in column 28. A special set of operation codes must be used to indicate the type of operation to be performed. Every operation code used requires certain entries on the same specification line. See Figure 8-7 for a summary of all the operation codes and the entries required for each code. For further information on the operation codes, see Chapter 10, *Operation Codes*.

The program performs the operations in the order specified on the calculation specifications sheet.

COLUMNS 31-32

Entry	Explanation
Blank or 01-99	Message identification code (MIC) of user message member to be displayed for SET or KEY operations unless overridden by a factor 1 entry

Use columns 31 and 32 for all KEY operations and for SET operations in which command key indicators are specified in columns 54 through 59, unless an entry is made in factor 1. Entries in columns 31 and 32 are ignored when factor 1 is specified on the same line as the SET or KEY operation.

The same combination of MICs should not be assigned to more than one KEY or SET operation except when the SET operation immediately precedes a KEY operation conditioned by the same indicators (columns 9 through 17) and the special SET-KEY combination is used. See *SET* and *KEY* in Chapter 10, *Operation Codes*, for complete information.

COLUMNS 43-48 (RESULT FIELD)

Entry	Explanation
ERASE	Erase the CONSOLE file buffer by using the SET operation code.
Field name, table name, array name, array element, data structure subfield name, or data structure name	The field specified contains the result of, or is the object of, the operation specified in columns 28 through 32. A data structure name can be specified as a result field only if the operation code in columns 28 through 32 is RLABL or POST.
INxx (xx = any RPG II indicator)	The indicator to be transferred to an external subroutine in an RLABL operation.

ERASE

Enter ERASE in columns 43 through 48 to blank or erase the entire buffer for the CONSOLE file. The filename of the CONSOLE file must be entered in columns 33 through 42. ERASE indicates to the system that the buffer should be set to blanks just before getting a record at the beginning of the next RPG II cycle.

Because the buffer is not erased until the beginning of the next RPG II cycle, processing of the current record continues after the ERASE operation is encountered. If the ERASE operation is executed because of invalid input data, you should insert code in your program to avoid further calculations and to return to the start of the RPG II cycle. A correct form of the record containing the invalid input data and any records that were entered after that record can then be reentered.

Field Name, Table Name, Array Name, Array Element, or Data Structure

Use columns 43 through 48 to name the field, data structure subfield, table, array, array element, or data structure that holds the result of the operation specified in columns 28 through 32, or that is the field upon which an operation is performed. Use the name of a field, table, array, array element, data structure, or data structure subfield that has already been defined either by the input, extension, or calculation specifications; or define a new field by entering a field name that is not already used. Any field defined in the result field is created when the program is compiled. The result field can be either numeric or alphameric. A field used in arithmetic operations (see *Columns 28-32 (Operation)*) or numeric compare operations or a field edited or zero suppressed by output specifications must be numeric.

A data structure name can be used as the result field only if the operation specified in columns 28 through 32 is RLABL or POST. Overlapping subfields in a data structure cannot be used in the same calculation. If factor 1, factor 2, or the result field references a subfield in a data structure that is an array or array element with a variable index, the entire array is used to determine whether overlap exists. The same array name can be referenced in the appropriate factors of a calculation specification without violating the overlap rule.

The result field name must begin with an alphabetic character in column 43 and contain no blanks or special characters.

If columns 43 through 48 contain the name of a field that is not defined elsewhere, columns 49 through 52 should also contain entries.

If the field is defined elsewhere, entries in columns 49 through 52 are not necessary but, if specified, must agree with the previous definition of that field.

COLUMNS 49-51 (FIELD LENGTH)

Entry	Explanation
1-256	Result field length

Use columns 49 through 51 to specify the length of the result field. If the result field is defined elsewhere, no entry is required for the length. However, if the length is specified, it must be the same as the previously defined length, with the same number of decimal positions. If the result field is a new field, consider the form your data is in because the result field must be large enough to hold the largest possible result. If the result field is too small, significant digits can be lost.

For example, to add field A (8 characters long, four decimal positions) to field B (10 characters long, six decimal positions), the result field, field C, must be large enough to contain 11 characters:

9999.0000	Field A
0001.111111	Field B
10000.111111	Field C (result field)

In this example, field C must be defined as 11 characters long with six decimal positions. Some of the numbers to the right of the decimal could be lost without changing the meaning of the result greatly. However, if field C was defined as 10 characters long with six decimal positions, a significant digit to the left of the decimal would be lost. Field C in this case would be 0000.111111; the meaning of the result has greatly changed.

Figure 8-10 shows how the contents of a result field can change after a multiplication operation, depending on the decimal position (column 52) and field length (columns 49 through 51) specifications. The result field for a multiply operation should be as long as the sum of the lengths of the two factor fields.

Numeric fields have a maximum length of 15 digits. Alphameric fields can be up to 256 characters long.

If the result field contains the name of a table or array, an entry in these columns is optional. If used, the entry must agree with the length described by the extension specifications.

Multiplication: 98.76 x 1.234 = 121.86984

Decimal Positions for Result Field (column 52)	Result Field Length (columns 49-51)									
	10	9	8	7	6	5	4	3	2	1
9	1.869840000	.869840000	▨	▨	▨	▨	▨	▨	▨	▨
8	21.86984000	1.86984000	.86984000	▨	▨	▨	▨	▨	▨	▨
7	121.8698400	21.8698400	1.8698400	.8698400	▨	▨	▨	▨	▨	▨
6	0121.869840	121.869840	21.869840	1.869840	.869840	▨	▨	▨	▨	▨
5	00121.86984	0121.86984	121.86984	21.86984	1.86984	.86984	▨	▨	▨	▨
4	000121.8698	00121.8698	0121.8698	121.8698	21.8698	1.8698	.8698	▨	▨	▨
3	0000121.869	000121.869	00121.869	0121.869	121.869	21.869	1.869	.869	▨	▨
2	00000121.86	0000121.86	000121.86	00121.86	0121.86	121.86	21.86	1.86	.86	▨
1	000000121.8	00000121.8	0000121.8	000121.8	00121.8	0121.8	121.8	21.8	1.8	.8
0	0000000121	000000121	00000121	0000121	000121	00121	0121	121	21	1

▨	Not permitted
░	Permitted but inaccurate
□	Recommended

Figure 8-10. Result Field Contents Based on Various Field Length and Decimal Position Specification

COLUMN 52 (DECIMAL POSITIONS)

Entry	Explanation
Blank	Alphameric or numeric result field is described elsewhere, or the newly defined result field is alphameric.
0-9	Number of decimal positions in a newly defined numeric result field.

Use column 52 to indicate the number of positions to the right of the decimal in a numeric result field. If the numeric result field contains no decimal positions, enter a 0 (zero). This column must be blank if the result field is alphameric. This column can be left blank if the result field is numeric but was described by input or calculations specifications. In this case, field length (columns 49 through 51) must also be left blank.

The number of decimal positions must never be greater than the length of the field. The number can, however, be larger or smaller than the number of decimal positions that actually result from an operation. If the number of decimal positions specified is greater than the number of decimal places that actually result from an operation, zeros are filled in to the right. If the number specified is smaller than the number that results from the operation, the rightmost digits are dropped.

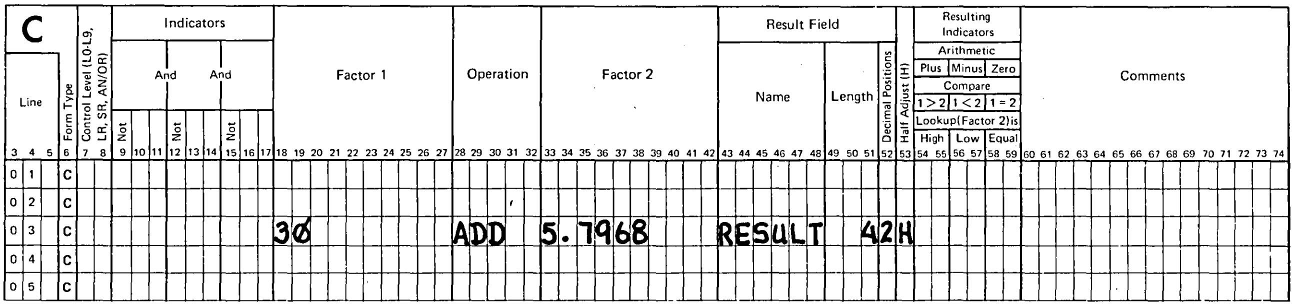
COLUMN 53 (HALF ADJUST)

Entry	Explanation
Blank	Do not half adjust
H	Half adjust

Use column 53 to indicate that the contents of the result field are to be half adjusted (rounded). Half adjusting is done by adding the single digit to the right of the last decimal position specified to the same position in the result field. All decimal positions to the right of the position specified for that field are then dropped (see Figure 8-11).

The half-adjust entry is allowed only with arithmetic operations. See *Columns 28-32 (Operation).* However, half adjusting cannot be specified for an MVR operation or for a DIV operation followed by an MVR operation.

Calculation Specifications



Line (3-5)	Form Type (6)	Factor 1 (18-27)	Operation (28-32)	Factor 2 (33-42)	Result Field Name (43-48)	Length (49-51)	Decimal Positions (52)	Half Adjust (H) (53)
01	C							
02	C							
03	C	3Ø	ADD	5.7968	RESULT	4	2	H
04	C							
05	C							

This calculation line shows a result field being half adjusted to two decimal positions (2 in column 52 and H in column 53).

Second Position
↓

35.7968	Result of an add operation.
6	Add the digit to the right of the last decimal position specified to the same position in the result field.
35.80xx	Drop all decimal positions to the right at the position specified.
35.80	Result after half adjusting.

Figure 8-11. Half Adjust

COLUMNS 54-59 (RESULTING INDICATORS)

Entry	Explanation
01-99	Any two-digit number
KA-KN, KP-KY	Any command key indicator (allowed only with SET or SETOF operation)
H1-H9	Any halt indicator
L1-L9	Any control level indicator
LR	Last record indicator
OA-OG, OV	Any overflow indicator
U1-U8	Any external indicator

Columns 54 through 59 have three purposes:

- To test the value of the result field after an arithmetic operation or to test the result of a CHAIN, KEY, LOKUP, COMP, READ, TESTB, TESTZ, ACQ, REL, NEXT, POST, or SHTDN operation. See Chapter 10, *Operation Codes*, for more information on each specific operation.
- To specify which command keys can be pressed for a SET operation.
- To specify which indicators are to be turned on or off by the SETON and SETOF operations.

Test Results

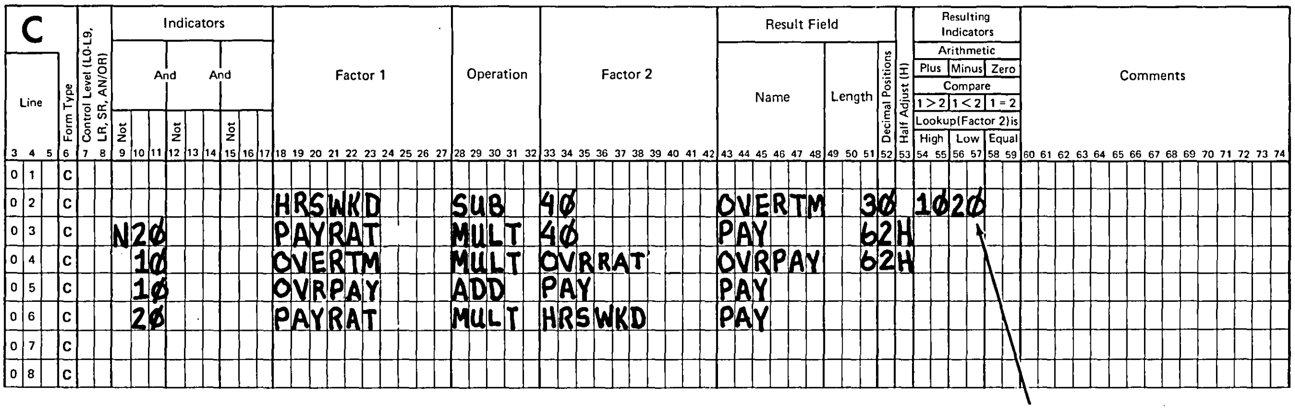
An indicator can be used in columns 54 through 59 to test the value of the result field, or to indicate an end-of-file condition, a no-record-found condition or an exception/error condition. Normally, only indicators 01 through 99 and H1 through H9 are used for testing. The indicator specified turns on only if the result field satisfies the condition being tested for. If the condition tested for is not met, the indicator is turned off. This indicator can then be used to condition following calculations or output operations (see Figure 8-12). If the same indicator is used to test the result of more than one operation, the last operation performed determines the setting of the indicator.

Three fields (columns 54 and 55, 56 and 57, and 58 and 59) can be used for testing the results. Each field is used to test for different conditions. You can specify testing for any or all conditions at the same time.

Columns 54-55 (Plus or High): Use an indicator in these columns when testing:

- Whether the result field in an arithmetic operation is positive
- Whether factor 1 is higher than factor 2 in a compare operation
- Whether factor 2 is higher than factor 1 in a table or array LOKUP operation
- Whether a CHAIN operation is not successful
- Whether each bit named in factor 2 is off for a TESTB operation
- Whether the character tested in a TESTZ operation is one of the characters &, A through I
- Whether the numeric field entered in a KEY operation is positive
- Whether the system operator has requested shutdown

Calculation Specifications



Line	Form Type	Control Level (L0-L9, LR, SR, AN/OR)	Indicators	Factor 1	Operation	Factor 2	Result Field Name	Length	Decimal Positions	Half Adjust (H)	Resulting Indicators High (Plus, 1>2)	Low (Minus, 1<2)	Equal (Zero, 1=2)	Comments
01	C													
02	C			HRSWKD	SUB	4Ø	OVERTM	3	Ø		1Ø	2Ø		
03	C		N2Ø	PAYRAT	MULT	4Ø	PAY	6	2	H				
04	C		1Ø	OVERTM	MULT	OVRRAT	OVRPAY	6	2	H				
05	C		1Ø	OVRPAY	ADD	PAY	PAY							
06	C		2Ø	PAYRAT	MULT	HRSWKD	PAY							
07	C													
08	C													

Two indicators are used to test for the different conditions in a subtract operation. These indicators are used to condition the calculations that must be performed for a payroll job. Indicator 10 turns on if the hours worked (HRSWKD) are greater than 40 and is then used to condition all operations necessary to find overtime pay. Indicator 20 turns on if HRSWKD is less than 40. It is also used to condition other operations. In line 03 if indicator 20 is not on (the employee worked 40 or more hours), regular pay based on a 40-hour week is calculated. In line 06 if indicator 20 is on (employee worked less than 40 hours), pay based on less than a 40-hour week is calculated.

Figure 8-12. Conditioning Operation (Resulting Indicators)

Columns 56-57 (Minus or Low): Use an indicator in these columns when testing:

- Whether the result field in an arithmetic operation is negative
- Whether factor 1 is lower than factor 2 in a compare operation
- Whether factor 2 is lower than factor 1 in a table or array LOKUP operation
- Whether the bits named in factor 2 are of mixed status (some bits on, some bits off) for a TESTB operation
- Whether the character tested in a TESTZ operation is one of the characters - J through R
- Whether the numeric field entered in a KEY operation is negative
- Whether the ACQ, REL, NEXT, READ, or POST operation to a WORKSTN file is not successful

Columns 58-59 (Zero or Equal): Use an indicator in these columns when testing:

- Whether the result field in an arithmetic operation is zero
- Whether factor 1 is equal to factor 2 in a compare operation
- Whether factor 2 is equal to factor 1 in a table or array LOKUP operation
- Whether an end-of-file condition is reached for the demand file that is read by the READ operation
- Whether each bit named in factor 2 is on for a TESTB operation
- Whether the character tested in a TESTZ operation is any character other than &, A through I, or -, J through R
- Whether the numeric field entered in a KEY operation is zero or whether an alphameric field is blank

Allowing Command Keys To Be Pressed (SET)

Columns 54 through 59 can contain command key indicators (KA through KN, KP through KY) for a SET operation.

When a SET operation occurs, only the command keys in columns 54 through 59 for that SET operation can be pressed at that time. From one to three command keys can be entered for each SET operation. If one or two command keys are specified, they can appear in any of the three sets of columns. See *SET* in Chapter 10, *Operation Codes,* for complete information on this operation.

Setting Indicators (SETON, SETOF)

The operation codes SETON or SETOF can be used to turn indicators on or off. See *SETON* and *SETOF Operations* in Chapter 10, *Operation Codes,* for more information on these operations. Any indicators to be turned on or off by the SETON or SETOF operation codes can be specified in any of the three resulting indicator fields (see Figure 8-13). The headings for columns 54 through 59 have no meaning for SETON or SETOF operations.

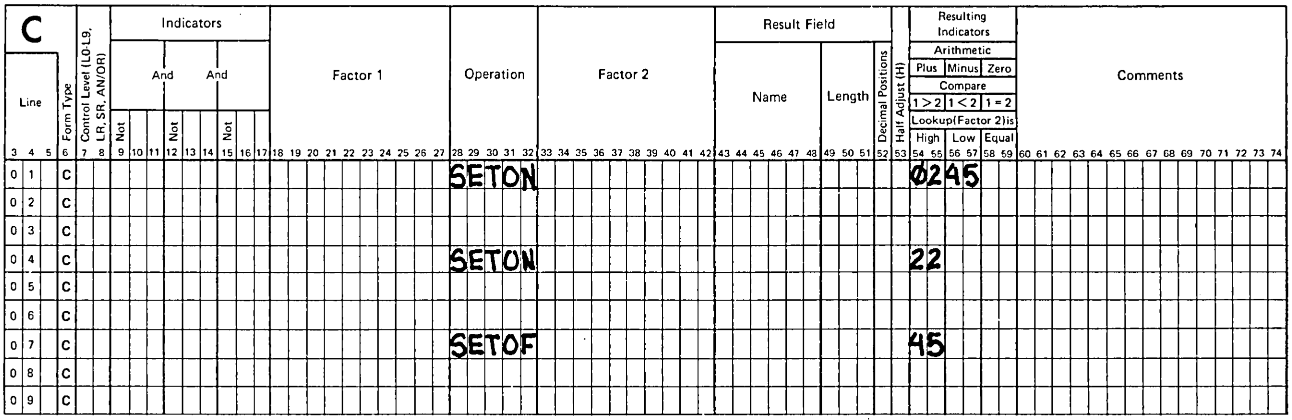
COLUMNS 60-74 (COMMENTS)

Use columns 60 through 74 to enter any meaningful comments that will help you understand the purpose of each specification line. Comments are not instructions to the RPG II program; they serve only as a means of documenting your program.

COLUMNS 75-80 (PROGRAM IDENTIFICATION)

See *Common Entries* in Chapter 1.

Calculation Specifications



Line (3-5)	Form Type (6)	Control Level (L0-L9, LR, SR, AN/OR) (7-8)	Indicators (9-17)	Factor 1 (18-27)	Operation (28-32)	Factor 2 (33-42)	Result Field Name (43-48)	Length (49-51)	Decimal Positions (52)	Half Adjust (H) (53)	Resulting Indicators Plus / 1>2 / High (54-55)	Minus / 1<2 / Low (56-57)	Zero / 1=2 / Equal (58-59)	Comments (60-74)
01	C				SETON						Ø2	45		
02	C													
03	C													
04	C				SETON						22			
05	C													
06	C													
07	C				SETOF						45			
08	C													
09	C													

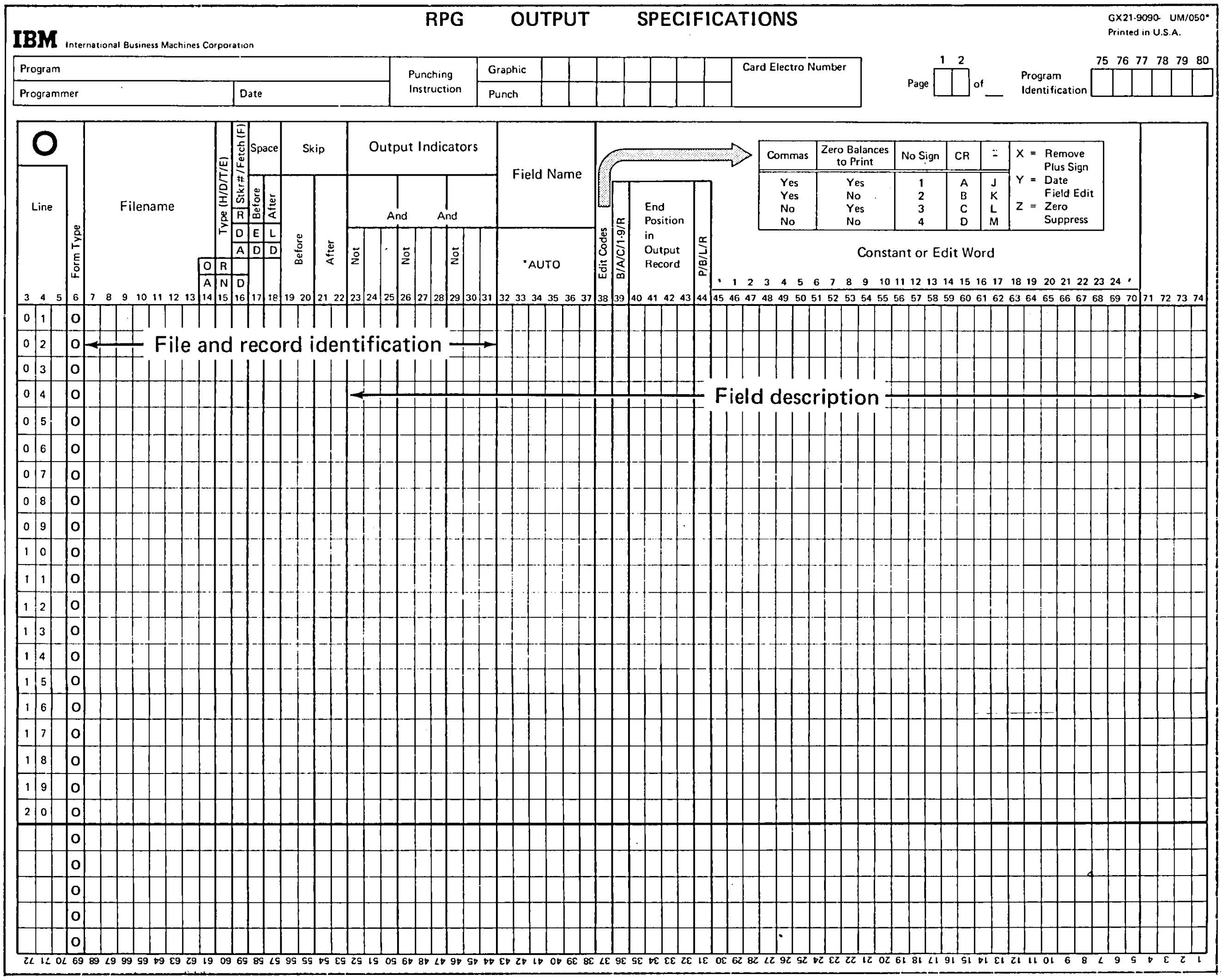
Figure 8-13. Setting Indicators

Chapter 9. Output Specifications

Output specifications describe the records and fields in the output file and the conditions under which output operations are performed. These specifications can be divided into two general categories:

- File and record identification entries (columns 7 through 31) that describe the output file, the records, and the indicators that condition the output.
- Field description entries (columns 23 through 74) that describe the position and format of data on the output record. These entries must begin one line below the file and record identification entries.

Write these specifications on the RPG Output Specifications sheet (see Figure 9-1).



IBM International Business Machines Corporation

RPG OUTPUT SPECIFICATIONS

GX21-9090- UM/050*
Printed in U.S.A.

Program

Programmer

Date

Punching Instruction

Graphic

Punch

Card Electro Number

Page 1 2 of

Program Identification 75 76 77 78 79 80

O

Line

Form Type

Filename

Type (H/D/T/E)

Stkr #/Fetch (F)

Space Before After

Skip Before After

Output Indicators And And Not Not Not

OR

AND

ADD

DEL

Field Name

*AUTO

Edit Codes

B/A/C/1-9/R

End Position in Output Record

P/B/L/R

Commas	Zero Balances to Print	No Sign	CR	-
Yes	Yes	1	A	J
Yes	No	2	B	K
No	Yes	3	C	L
No	No	4	D	M

X = Remove Plus Sign
Y = Date Field Edit
Z = Zero Suppress

Constant or Edit Word

' 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 '

01 O ← File and record identification →
02 O
03 O
04 O ← Field description →
05 O
06 O
07 O
08 O
09 O
10 O
11 O
12 O
13 O
14 O
15 O
16 O
17 O
18 O
19 O
20 O
O
O
O
O
O

Figure 9-1 RPG Output Specifications

COLUMNS 1-2 (PAGE)

See *Common Entries* in Chapter 1.

COLUMNS 3-5 (LINE)

See *Common Entries* in Chapter 1.

COLUMN 6 (FORM TYPE)

An O must appear in column 6 to identify this line as an output specifications statement.

COLUMNS 7-14 (FILENAME)

Entry	Explanation
A valid filename	Same filename that appears on the file description specifications for the output, combined, update, or add file

Use columns 7 through 14 to identify the output file being described. The filename must begin in column 7.

The filename need be specified only on the first line. However, if another output file is specified and further specifications are then required for the first file, the first filename must be repeated in columns 7 through 14 (see Figure 9-2).

COLUMNS 14-16 (AND/OR)

Entry	Explanation
AND or OR	AND/OR indicates a relationship between lines of output indicators.

Use columns 14 through 16 to specify AND/OR lines for output operations. For further information, see *AND and OR Lines* under *Columns 23-31 (Output Indicators),* in this chapter.

COLUMN 15 (TYPE)

Entry	Explanation
H	Heading records
D	Detail records
T	Total records
E	Exception records (lines to be written during calculation time)

Use column 15 to indicate the type of record to be written. Column 15 must have an entry for every output record.

Describe output files by entering the records for each file in this order: heading, detail, total, and exception (see Figure 9-2). Or enter all record types for all output files in this order: heading, detail, total, and exception (see Figure 9-2).

Heading Records (H)

Heading records usually contain constant identifying information such as column headings, page number, and date.

Detail Records (D)

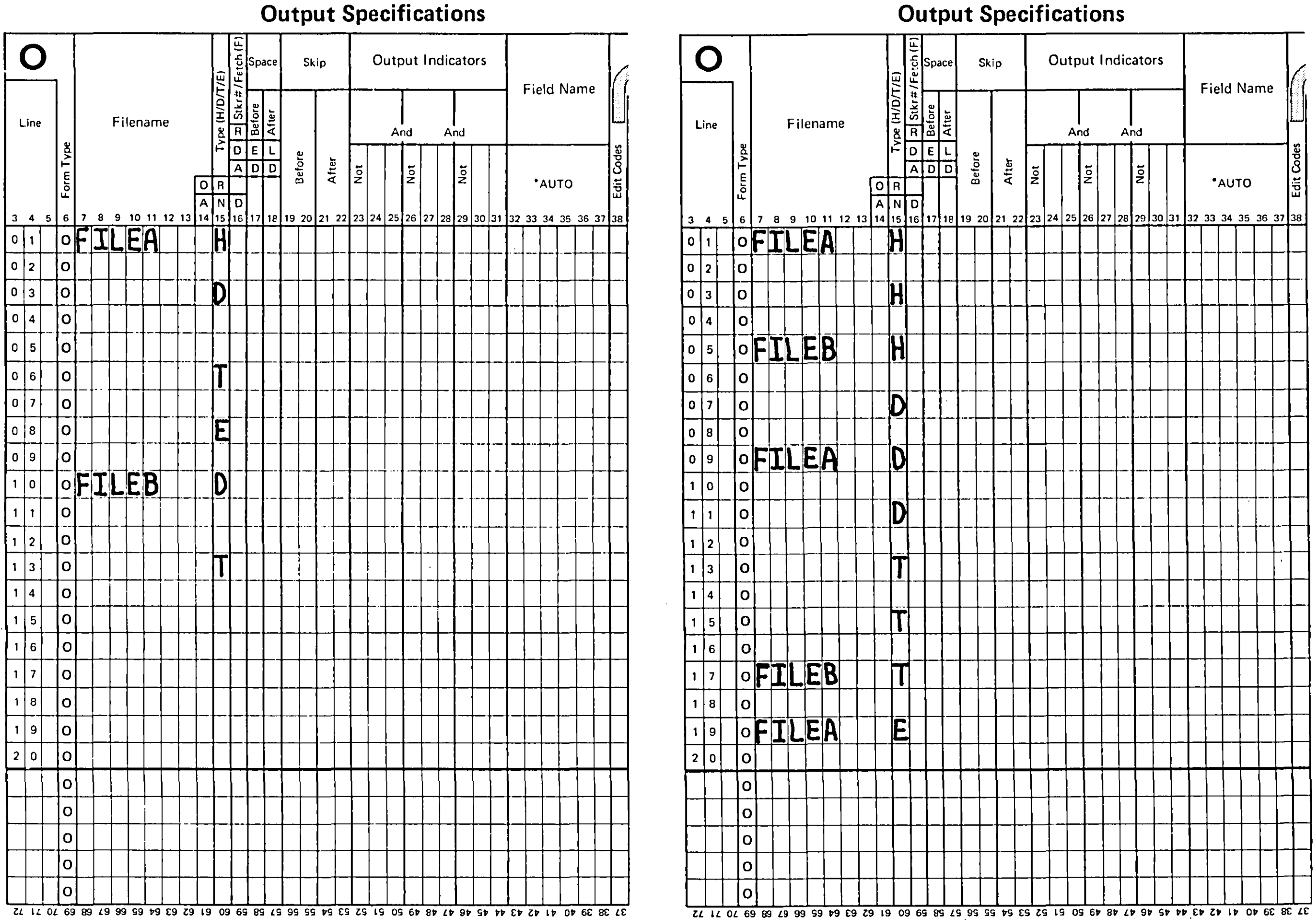
Detail records usually contain data that comes directly from the input record or is the result of calculations performed on data from the input record.

Total Records (T)

Total records usually contain data that is the end result of specific calculations on several detail records. Total output cannot be specified for primary or secondary update files. Records can be added to indexed primary and secondary files at total time if add is specified (A in column 66) on the file description specifications.

Exception Records (E)

Exception records are written during calculation time. Exception records can be specified only when the operation code EXCPT is used. See Chapter 10, *Operation Codes,* for further information on the EXCPT operation code.



Output Specifications

Line	Form Type	Filename	Type (H/D/T/E)
01	O	FILEA	H
02	O		
03	O		D
04	O		
05	O		
06	O		T
07	O		
08	O		E
09	O		
10	O	FILEB	D
11	O		
12	O		
13	O		T
14	O		
15	O		
16	O		
17	O		
18	O		
19	O		
20	O		
	O		
	O		
	O		
	O		
	O		

Output Specifications

Line	Form Type	Filename	Type (H/D/T/E)
01	O	FILEA	H
02	O		
03	O		H
04	O		
05	O	FILEB	H
06	O		
07	O		D
08	O		
09	O	FILEA	D
10	O		
11	O		D
12	O		
13	O		T
14	O		
15	O		T
16	O		
17	O	FILEB	T
18	O		
19	O	FILEA	E
20	O		
	O		
	O		
	O		
	O		
	O		

Figure 9-2. Order of Output Record Types

COLUMNS 16-18 (ADD/DEL)

Entry	Explanation
ADD	Add a record to an indexed, direct, or sequential file defined as an input, output, or update file.
DEL	Delete the last record read on the identified update file.

ADD

When ADD is specified in columns 16 through 18 to add a record to an indexed, direct, or sequential file, column 66 of the file description specifications must contain an A for the file to which records are being added. The output device for this file must be DISK.

The ADD entry must not be used in an OR line. An ADD entry in columns 16 through 18 of the previous line also applies to the record in the OR relationship. (For a detailed description of adding records to file, see *Column 66 (File Addition)* in Chapter 3.)

DEL

If a record is to be deleted from a file, the file must be defined as delete-capable when it is built (for further information on defining a delete-capable file, see *FILE Statement* in the *System Support Reference Manual*). If you attempt to delete a record from a file that is not delete-capable, an execution-time error message is displayed.

DEL must be specified in columns 16 through 18 of the main output record line. DEL applies to all the OR extensions to the main line. When records are deleted from a file, the file must be defined as an update file (column 15 of the file description specifications contains U).

Note: Record deletion is not dependent on the file organization and mode of processing entries.

Records are not physically removed from a file when they are deleted. Deleted records are filled with hex FFs. If a direct file load of a delete-capable file is executed, the entire file is initialized to hex FFs (for further information on direct file loading of delete-capable files, see *Direct Files* in Chapter 3).

When a file containing deleted records is processed sequentially or consecutively (primary, secondary, or demand files) a deleted record is not returned to the program when it is accessed. It is bypassed, and the next record is read. This process is repeated until a nondeleted record is found or the end of the file is reached. When a file containing deleted records is processed randomly using CHAIN, the no-record-found indicator is turned on when a deleted record is accessed. If this indicator is not specified in columns 54 and 55 of the calculation specification specifying the CHAIN operation, an error message is displayed.

COLUMN 16 (FETCH OVERFLOW OR RELEASE)

Entry	Explanation
F	Fetch overflow routine
R	Release the device (display station or SSP-ICF session) after output

Fetch Overflow

Use column 16 to specify fetch overflow for a printer file only. Column 16 of each OR line must contain an F if the overflow routine is to be used for each record in the OR relationship. Fetch overflow cannot be used when an overflow indicator is specified in columns 23 through 31 on the same specification line. If this occurs, the overflow routine is not fetched. Specifying fetch overflow allows you to alter the basic RPG II overflow logic (see *Columns 33-34* in Chapter 3, *File Description Specifications*). You can advance forms when total, detail, or exception records are printed instead of waiting for the usual time. The fetched overflow routine does not automatically cause forms to advance; that is, the entry in columns 21 and 22 of the output specifications must contain a two-digit entry that is less than the number of the line the printer is currently on. Fetching the overflow routine can prevent printing over the page perforation and can ensure use of as much of the page as possible.

The fetch overflow specification causes the system to check whether the overflow indicator assigned to the printer file is on before printing total, detail, or exception records. The system tests the indicator each time an F is encountered in column 16 of the output specifications. If the overflow indicator is on, the overflow routine is fetched and the following operations are done:

1. All total lines conditioned by the overflow indicator are printed.

2. Forms advance to a new page when a skip to a line number less than the line number the printer is currently on is specified in a line conditioned by an overflow indicator.

3. Heading lines and detail lines conditioned by the overflow indicator are printed.

4. The line that fetched the overflow routine is printed.

5. Any detail, exception, and total lines left to be printed for that output cycle are printed.

Use fetch overflow when printing a particular line causes overflow, and not enough space is left on the page to print the remaining detail, exception, or total output lines. To determine when to fetch the overflow routine, study all possible overflow situations. By counting lines and spaces, you can calculate what happens if overflow occurs on each detail and total line.

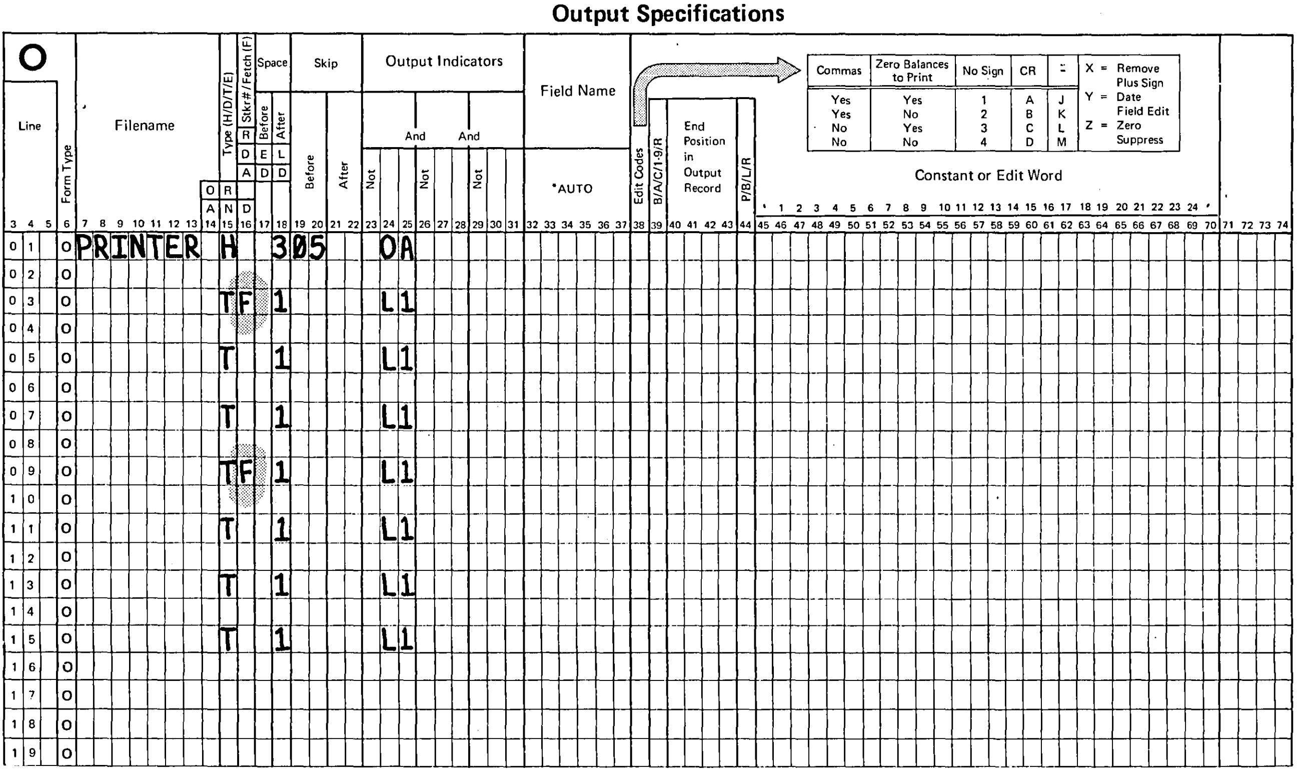
Figure 9-3 shows an example of specifying fetch overflow.

Release

A device can be released from the program after output to that device has occurred. To release the device, enter an R in column 16. OR lines can be specified; however, column 16 must contain an R for each OR line. The device is released when that output specification is encountered during the output operations. If a format name is specified on a field description for the record that contains an R in column 16, the format is written and then the device is released.

RPG II sets on the LR indicator when all devices have been released if the WORKSTN file is a primary file and the program does not have an NEP attribute. If the program is an NEP, all devices must have been released and the system operator must enter the STOP SYSTEM command before RPG II sets on the LR indicator.

Note: For WORKSTN files, a device can be either a display station or an SSP-ICF session.



Output Specifications

Commas	Zero Balances to Print	No Sign	CR	–	X = Remove Plus Sign Y = Date Field Edit Z = Zero Suppress
Yes	Yes	1	A	J	
Yes	No	2	B	K	
No	Yes	3	C	L	
No	No	4	D	M	

Line	Form Type	Filename	Type (H/D/T/E)	Stkr #/Fetch (F)	Space Before	Space After	Skip Before	Skip After	Output Indicators	Field Name	Edit Codes	B/A/C/1-9/R	End Position in Output Record	P/B/L/R	Constant or Edit Word
01	O	PRINTER	H		3	Ø	5		OA						
02	O														
03	O		T	F		1			L1						
04	O														
05	O		T			1			L1						
06	O														
07	O		T			1			L1						
08	O														
09	O		T	F		1			L1						
10	O														
11	O		T			1			L1						
12	O														
13	O		T			1			L1						
14	O														
15	O		T			1			L1						
16	O														
17	O														
18	O														
19	O														

Figure 9-3. Uses of Fetch

COLUMNS 17-22 (SPACING AND SKIPPING)

Column 17 (Space Before)

Entry	Explanation
0-3	Number of lines to be spaced before a line is printed for a CRT file or a printer file.

Column 18 (Space After)

Entry	Explanation
0-3	Number of lines to be spaced after a line is printed for a CRT file or printer file.

Columns 19-20 (Skip Before)

Entry	Explanation
01	Display screen is blanked immediately for a CRT file.
01-99	Skip to lines 01 to 99 before printing for printer files.
A0-A9	Skip to lines 100 to 109 before printing for printer files.
B0-B2	Skip to lines 110 to 112 before printing for printer files.

Columns 21-22 (Skip After)

Entry	Explanation
01-99	Skip to lines 01 to 99 after printing for printer files.
A0-A9	Skip to lines 100 to 109 after printing for printer files.
B0-B2	Skip to lines 110 to 112 after printing for printer files.

Use columns 17 through 22 to specify line spacing and skipping for printer and CRT files. Spacing refers to advancing one line at a time, and skipping refers to jumping from one print line to another.

Figure 9-4 shows the possible spacing and skipping entries for these files. If an incorrect entry is made in these columns, the compiler drops the entry and assumes a blank specification. If columns 17 through 22 are blank, single spacing occurs after each line is printed. Different spacing and skipping can be specified for OR lines. If there are no spacing or skipping entries for the OR line, spacing and skipping are done according to the specifications for the line preceding the OR line. No spacing or skipping can be specified on AND lines.

To prevent OCL statements from printing on the same page as output data, specify a skip to a new page at the beginning of each job. For example, if the last output line for the job is printed on line 01 of a page and no spacing or skipping is specified, the system begins printing OCL statements for the next job on the next line. To avoid this, specify a space-1-after.

Spacing and Skipping for Printer Files

Line spacing and skipping can be specified both before and after printing of a line. If both spacing and skipping are specified on the same line, they occur in this order:

1. Skip before
2. Space before
3. Skip after
4. Space after

Specifying spacing (column 18) and skipping (columns 21 and 22) after printing a line saves time because the system does not have to wait for the forms to advance before printing can be done.

With spacing, the maximum number of blank lines that can occur between two lines of print is five. If six spaces are specified (three after the preceding print line and three before the current print line), the printer spaces six lines and begins printing on the sixth line.

Spacing or skipping to the overflow line or past the overflow line turns the overflow indicator on. Skipping past the overflow line to a line of the next page, however, does not turn the overflow indicator on. Under this condition, use a SETON operation to turn on the overflow indicator to condition overflow operations.

Skipping is usually done when a new page is needed. A skip to a lower line number means advance to a new page. Skipping can also be specified when more than five blank lines are required between two lines of print. The entry for skipping must be a two-digit number that indicates the number of the next line to be printed. If a skip is specified to the same line number that the forms are positioned on, the forms do not move.

Files	Space Before	Space After	Skip Before[1]	Skip After[1]
	Column 17	Column 18	Columns 19-20	Columns 21-22
Printer	0-3	0-3	01-99, A0-A9, B0-B2	01-99, A0-A9, B0-B2
CRT	0-3	0-3	01[2]	No entry

[1] The skip entries you specify in columns 19 through 22 must not exceed the form length specified in line counter specifications, or must not exceed 66 if no line counter specifications are supplied.

[2] Only allowable entry is 01, which causes the screen to be erased.

Figure 9-4. Possible Spacing and Skipping Entries

Spacing and Skipping for CRT Files

The following rules apply to spacing and skipping for CRT files:

- A space-before entry (0-3) can be specified in column 17.
- A space-after entry (0-3) can be specified in column 18.
- If a CRT file has a record length of 40 or less, the space-before and space-after entries cannot both be 3.
- A skip-before entry, to line 01 only, can be in columns 19 and 20. This entry immediately clears the display screen.
- A skip-after entry (columns 21 and 22) must not be specified for CRT files.

COLUMNS 23-31 (OUTPUT INDICATORS)

Entry	**Explanation**
01-99	Any resulting indicator, field indicator, or record identifying indicator previously specified.
KA-KN, KP-KY	Any command key indicator previously specified in a SET operation or used with a WORKSTN file.
L0-L9	Any control level indicators previously specified.
H1-H9	Any halt indicators previously specified.
U1-U8	Any external indicator set prior to program execution.
OA-OG, OV	Any overflow indicator previously assigned to this file.
MR	Matching record indicator.
LR	Last record indicator.
1P	First page indicator. The 1P indicator cannot be specified for a WORKSTN file.

Use output indicators to specify the conditions under which a line or field is written as output.

When an indicator is to condition an entire output line, enter it on the line that specified the type of record (see Figure 9-5). When an indicator is to condition when a field is to be written, enter it on the same line as the field name (see Figure 9-5).

One indicator can be specified in each of the three separate output indicator fields (columns 23 through 25, 26 through 28, and 29 through 31). If these indicators are on, the output operation is done. An N in the column preceding each indicator (column 23, 26, or 29) means that the output operation is done only if the indicator is not on (a negative indicator). No output line should be conditioned by all negative indicators; at least one of the indicators should be positive. If all negative indicators condition a heading or detail operation, the operation is performed at the beginning of the program cycle when the first page (1P) lines are written.

If no output indicators are specified, the line is output every time that record is checked for output. If no output indicators are specified on a heading or detail line, that record is also produced as output at the beginning of the program cycle.

AND and OR Lines

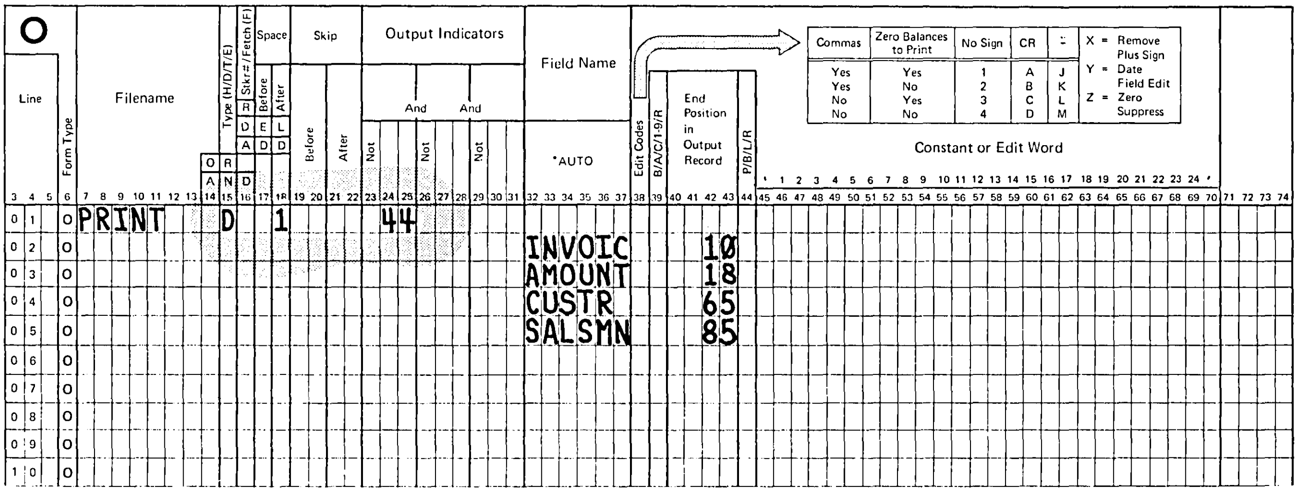
Use an AND line if more than three indicators are needed to condition an output operation. Enter the word AND in columns 14 through 16 of each additional line. The condition for all indicators in an AND relationship must be satisfied before the output operation is done. A maximum of 20 AND lines can be used for an output operation if no OR lines are used.

Output indicators can also be in an OR relationship. If one or the other condition is met, the output operation is done. A maximum of 20 OR lines can be used for an output operation if no AND lines are used.

If AND and OR lines are combined, the total number of AND and OR lines for an output operation cannot exceed 20.

AND and OR lines can be used to condition entire output lines, but they must not be used to condition fields (see Figure 9-6). However, you can condition an output field with more than three indicators by using the SETON operation in calculations. For example, if indicators 10, 12, 14, 16, and 18 are used to condition an output field named PAY, in the calculations you can set on indicator 20 if indicators 10, 12, and 14 are on. Then condition the output field PAY on indicators 20, 16, and 18 in the output specifications.

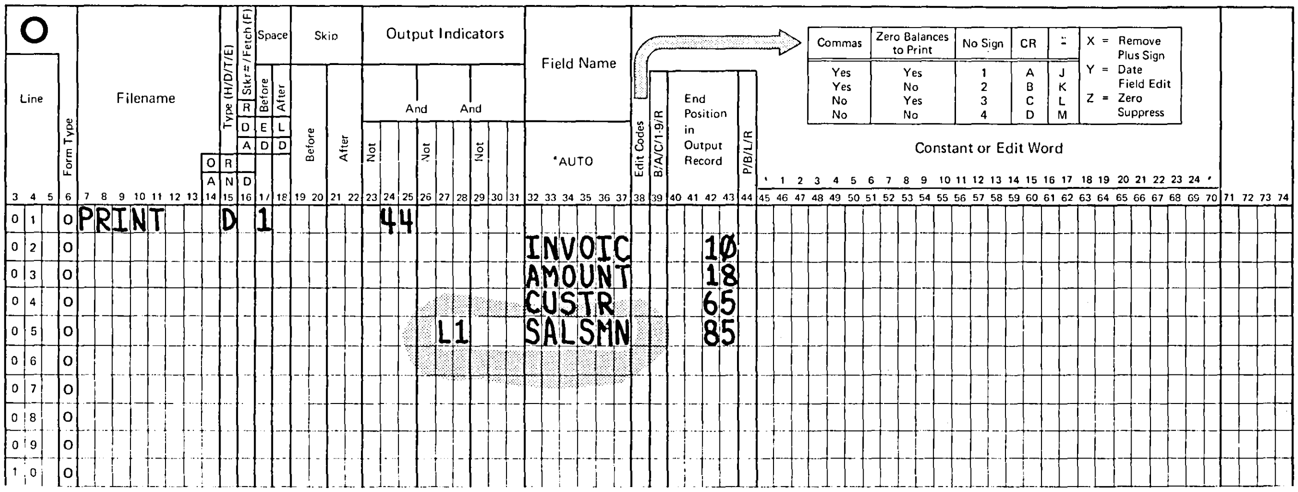
Output Specifications



Line	Form Type	Filename	Type (H/D/T/E)	Space	Output Indicators	Field Name	End Position in Output Record
01	O	PRINT	D	1	44		
02	O					INVOIC	10
03	O					AMOUNT	18
04	O					CUSTR	65
05	O					SALSMN	85
06	O						
07	O						
08	O						
09	O						
10	O						

One indicator is used to condition an entire line of printing. When 44 is on, the fields named INVOIC, AMOUNT, CUSTR, and SALSMN are all printed.

Output Specifications

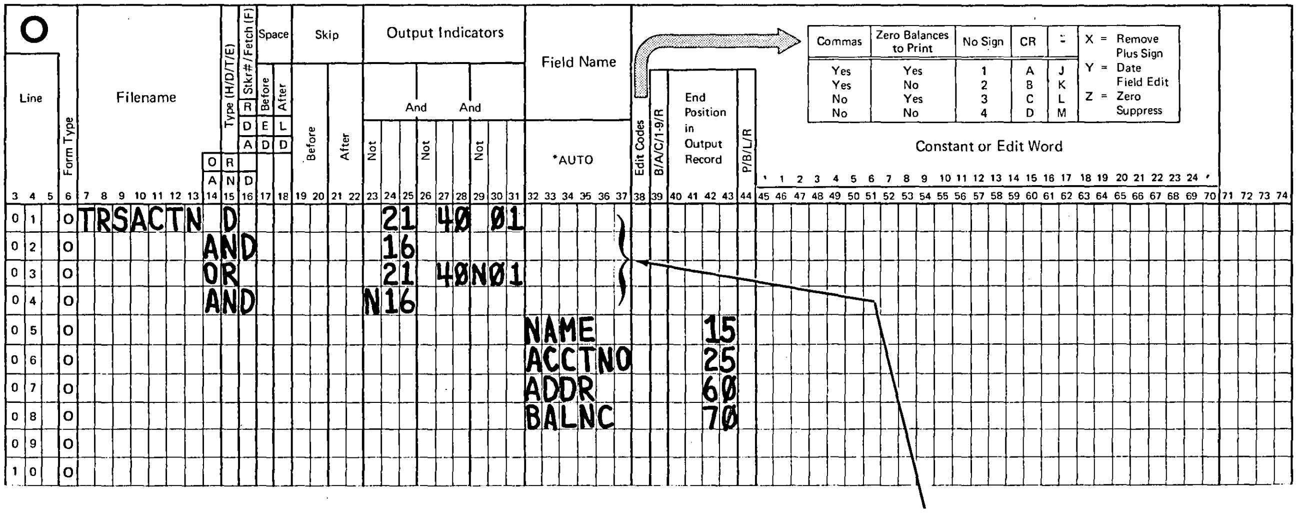


Line	Form Type	Filename	Type (H/D/T/E)	Space	Output Indicators	Field Name	End Position in Output Record
01	O	PRINT	D	1	44		
02	O					INVOIC	10
03	O					AMOUNT	18
04	O					CUSTR	65
05	O				L1	SALSMN	85
06	O						
07	O						
08	O						
09	O						
10	O						

A control level indicator is used to condition when one field should be printed. When indicator 44 is on, fields INVOIC, AMOUNT, and CUSTR are always printed. However, SALSMN is printed for the first record of a new control group only if 44 and L1 are on.

Figure 9-5. Output Indicators

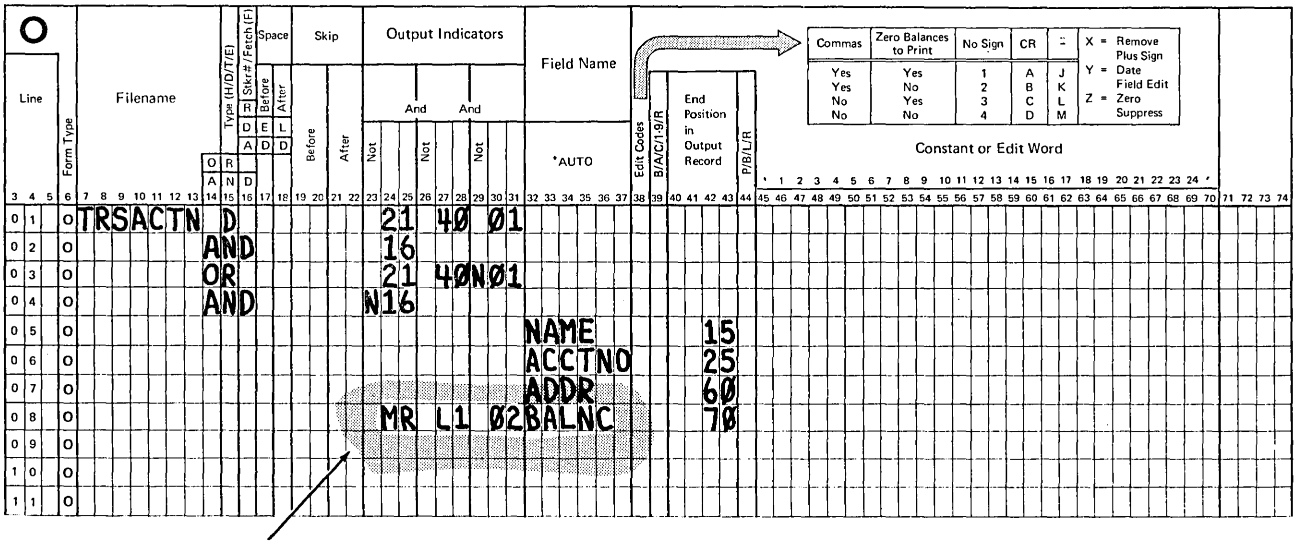
Output Specifications



```
01 OTRSACTN D        21 4Ø Ø1
02 O       AND       16
03 O       OR        21 4ØNØ1
04 O       AND      N16
05 O                         NAME      15
06 O                         ACCTNO    25
07 O                         ADDR      6Ø
08 O                         BALNC     7Ø
09 O
10 O
```

The detail line is printed if either of two sets of conditions is met. If 21, 40, 01, and 16 are all on, the line is printed; if 21 and 40 are on and 01 and 16 are off, the line is also printed.

Output Specifications



```
01 OTRSACTN D        21 4Ø Ø1
02 O       AND       16
03 O       OR        21 4ØNØ1
04 O       AND      N16
05 O                         NAME      15
06 O                         ACCTNO    25
07 O                         ADDR      6Ø
08 O                 MR L1 Ø2BALNC     7Ø
09 O
10 O
11 O
```

A maximum of three indicators can be used to condition a field.

Figure 9-6. Output Indicators in AND and OR Lines

The use of any of the LO through L9 indicators in an OR relationship with an LR indicator can result in the specified operation being done twice when LR is on. One operation is performed during LR processing and the other at detail or total time. Figure 9-7 shows how to correctly use the LO through L9 indicators in an OR relationship.

Command Key Indicators (KA-KN, KP-KY)

Use command key indicators in columns 23 through 31 to condition output operations; however, any command keys entered in these columns must also be specified in columns 54 through 59 of the calculation specifications for a SET or SETOF operation, or used for a WORKSTN file. All command keys (KA through KN, KP through KY) can be used for a WORKSTN file. When the operator presses a command key, the data keyed at the display station is returned to the program, the corresponding command key indicator turns on, and all other command key indicators turn off. You can then use the command key indicators to condition calculation and output operations.

See Chapter 10, *Operation Codes*, for complete information on the SET or SETOF operation.

Overflow Indicators (OA-OG, OV)

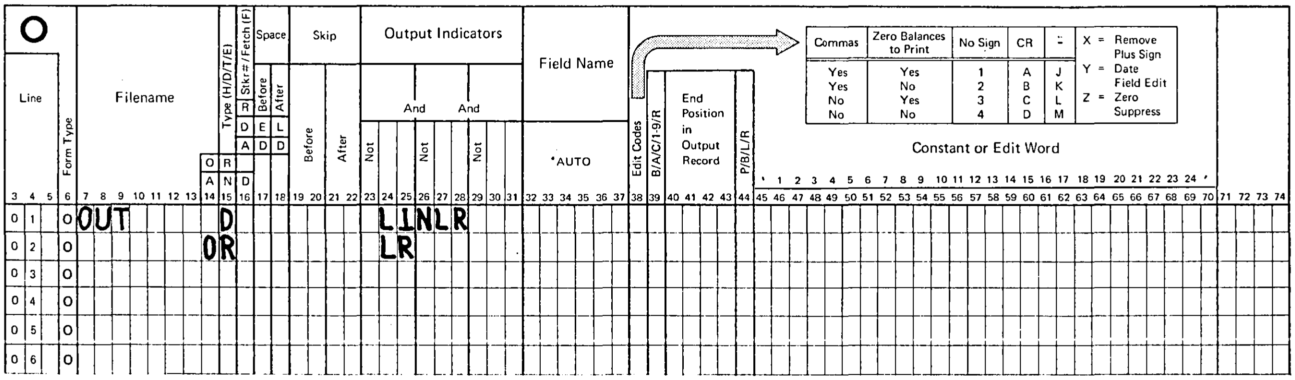
Overflow indicators condition output operations for a printer file. These operations are done only after the overflow line (end of page) is sensed. To use an overflow indicator in the output specifications, the same overflow indicator must be assigned to the printer file on the file description specifications. If no overflow indicator is assigned on the file description specifications, the compiler automatically handles overflow (see *Columns 33-34* in Chapter 3, *File Description Specifications*).

Assigning Overflow Indicators

When assigning overflow indicators in the output specifications, consider the following:

- Spacing past the overflow line turns the overflow indicator on.
- Skipping past the overflow line to any line on the same page turns the overflow indicator on.
- Skipping past the overflow line to any line on the new page does not turn the overflow indicator on.
- A skip to a new page specified on a line not conditioned by an overflow indicator turns the overflow indicator off before the forms advance to a new page.
- Control level indicators can be used with an overflow indicator so that each page will contain information from only one control group (see Figure 9-8).
- An overflow indicator can appear on either AND or OR lines. However, only one overflow indicator can be associated with one group of output indicators.
- When the overflow indicator is used in an AND relationship with a record identifying indicator, unusual results are often obtained because the record type might not be the one read when overflow occurred. Thus, the record identifying indicator is not on, and all lines conditioned by both overflow and record identifying indicators do not print.
- An overflow indicator cannot condition an exception line (E in column 15) but can condition fields within the exception record.
- Overflow indicators can be turned on and off by the operation codes SETON and SETOF.

Output Specifications



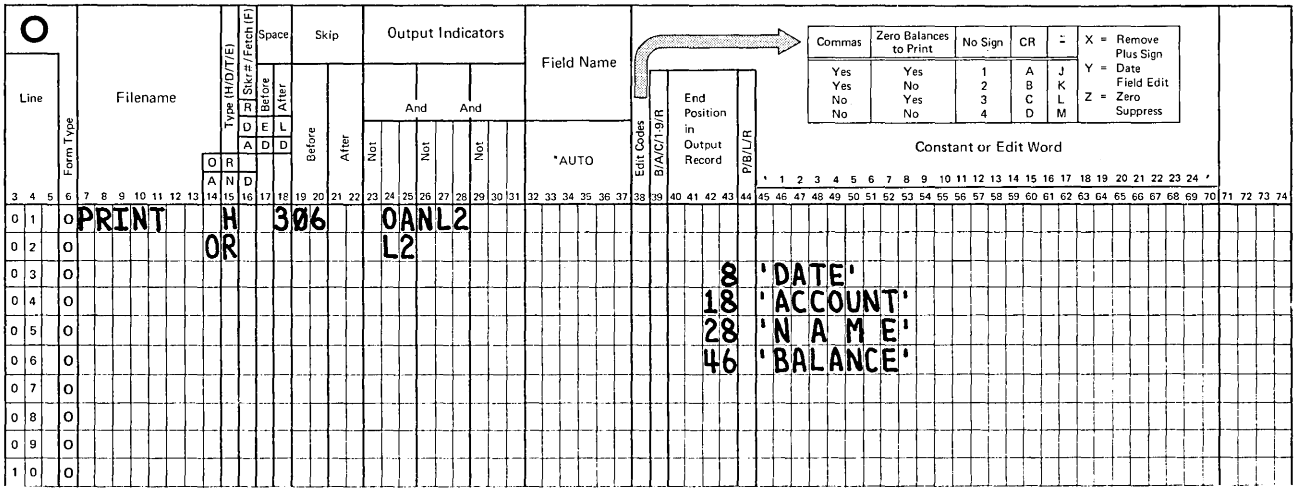
Line	Form Type	Filename	AND/OR	Type (H/D/T/E)	Output Indicators
0 1	O	OUT		D	L1NLR
0 2	O		OR		LR
0 3	O				
0 4	O				
0 5	O				
0 6	O				

Commas	Zero Balances to Print	No Sign	CR	-	
Yes	Yes	1	A	J	X = Remove Plus Sign
Yes	No	2	B	K	Y = Date Field Edit
No	Yes	3	C	L	Z = Zero Suppress
No	No	4	D	M	

Space
Skip
Before
After
Field Name
*AUTO
Edit Codes
B/A/C/1-9/R
End Position in Output Record
P/B/L/R
Constant or Edit Word

Figure 9-7. Correct Use of Indicators LO through L9 in OR Relationship

Output Specifications



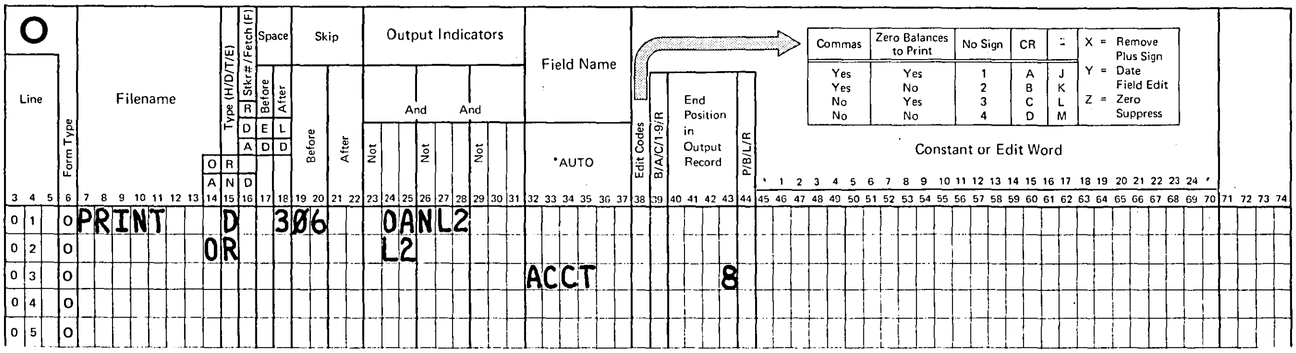
Commas	Zero Balances to Print	No Sign	CR	–	X = Remove Plus Sign
Yes	Yes	1	A	J	Y = Date Field Edit
Yes	No	2	B	K	Z = Zero Suppress
No	Yes	3	C	L	
No	No	4	D	M	

Line	Form Type	Filename	Type (H/D/T/E)	Space Before	Skip Before	Output Indicators	Field Name	End Position in Output Record	Constant or Edit Word
01	O	PRINT	H	3	06	OANL2			
02	O	OR				L2			
03	O							8	'DATE'
04	O							18	'ACCOUNT'
05	O							28	'N A M E'
06	O							46	'BALANCE'
07	O								
08	O								
09	O								
10	O								

This is the coding necessary for printing headings on every page: first page, every overflow page, and each new page to be started because of a change in control fields (L2 is on). Line 01 allows the headings to be printed at the top of a new page (skip to 06) only when an overflow occurs (OA is on and L2 is not on).

Line 02 allows printing of headings on the new page only at the beginning of a new control group (L2 is on). This way, duplicate headings caused by both L2 and OA being on at the same time do not occur. Line 02 allows headings to be printed on the first page after the first record is read because the first record always causes a control break (L2 turns on) if control fields are specified on the record.

Output Specifications



Line	Form Type	Filename	Type (H/D/T/E)	Space Before	Skip Before	Output Indicators	Field Name	End Position in Output Record	Constant or Edit Word
01	O	PRINT	D	3	06	OANL2			
02	O	OR				L2			
03	O						ACCT	8	
04	O								
05	O								

This is the necessary coding for the printing of certain fields on every page; a skip to 06 is done either on an overflow condition or on a change in control level (L2). The NL2 indicator in line 01 prevents the line from printing and skipping twice in the same cycle.

Figure 9-8. Using Control Level Indicators with an Overflow Indicator

Figure 9-9 shows the setting of overflow indicators during the normal overflow routine and during a fetched overflow routine for both normal output and exception output. The left portion of the graph shows the general program cycle. The solid black lines indicate that the indicator is on, and the dashes show a connection between the end of one cycle and the start of the next.

First Page Indicator (1P)

The first page (1P) indicator allows printing either on the first page only or on every page if used with an overflow indicator (see Figure 9-10). The line conditioned by the 1P indicator must contain constant information used as headings or fields for reserved words such as PAGE and UDATE. The constant information is specified on the output specifications in columns 45 through 70.

Use the 1P indicator only with heading or detail output lines. It cannot be used to condition total or exception output lines, in an AND relationship with control level indicators, or to condition output for a WORKSTN file.

See *Column 41 (1P Forms Position)* in Chapter 2, *Control Specifications*, for information on forms alignment of the first page.

Halt Indicators (H1-H9)

If certain error conditions occur, you might not want output operations performed. Use halt indicators to prevent the data that caused the error from being used (see Figure 9-11).

External Indicators (U1-U8)

You may want to condition certain output records on external conditions. If so, use the external indicator to condition those records.

COLUMNS 32-37 (FIELD NAME)

In columns 32 through 37, use one of the following methods to specify each field that is to be written out:

- Any field name or data structure name previously used in this program
- The special words PAGE, PAGE1 through PAGE7, *PLACE, UDATE, UDAY, UMONTH, or UYEAR
- A table name, array name, or array element

Field Names

The field names used must be the same as the field names on the input specifications (columns 53 through 58) or the calculation specifications (columns 43 through 48). Do not enter a field name if a constant is used in columns 45 through 70. If a field name is entered in columns 32 through 37, columns 7 through 22 must be blank.

Fields can be listed on the specifications sheet in any order because the sequence in which they appear on the output record is determined by the entry in columns 40 through 43. However, the fields are usually listed sequentially. If fields overlap, the last field specified is the only field completely written.

The sign (+ or -) of a numeric field is in the units position (rightmost digit). The units position prints as a letter unless the field is edited. See *Column 38 (Edit Codes)*.

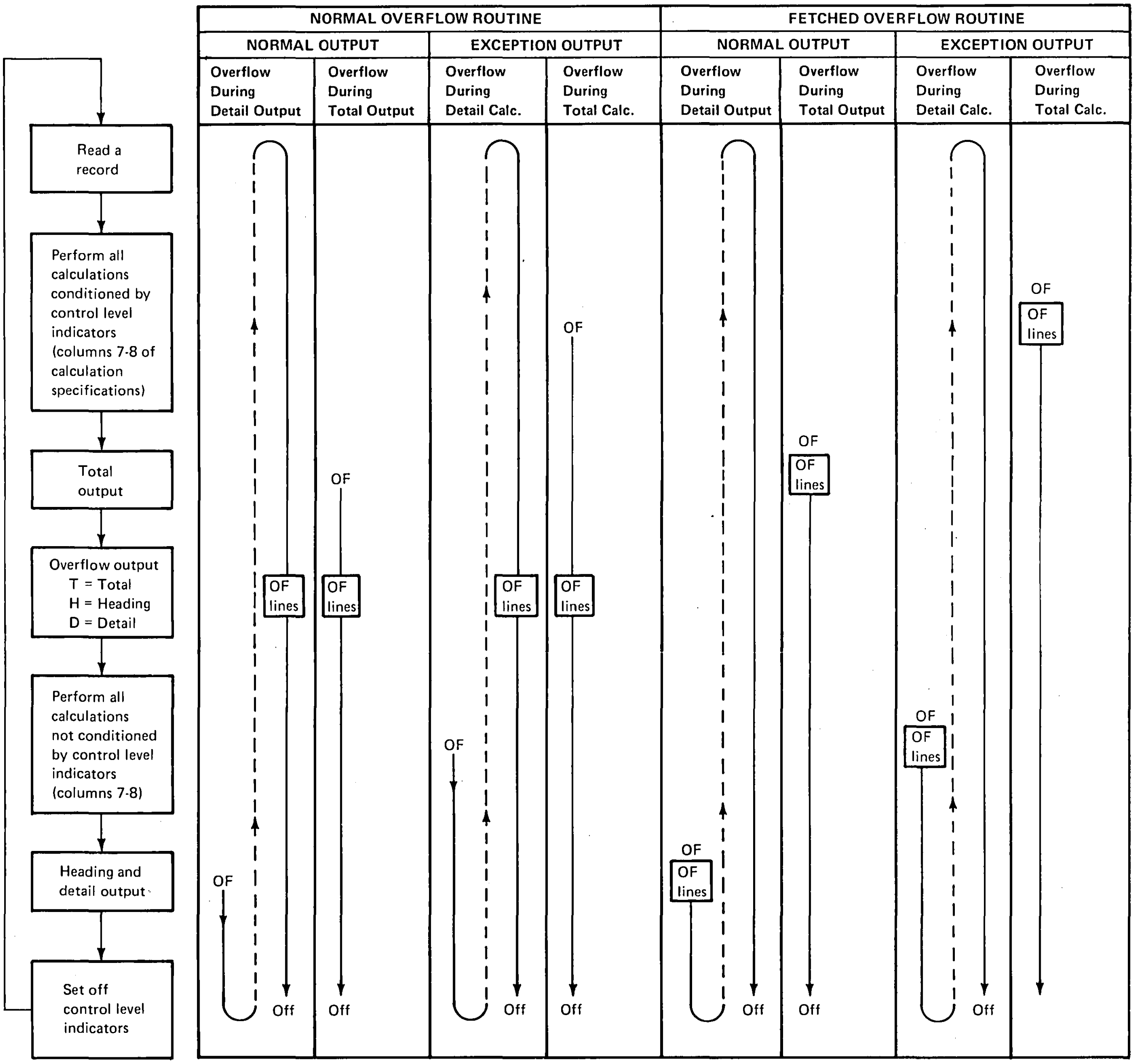
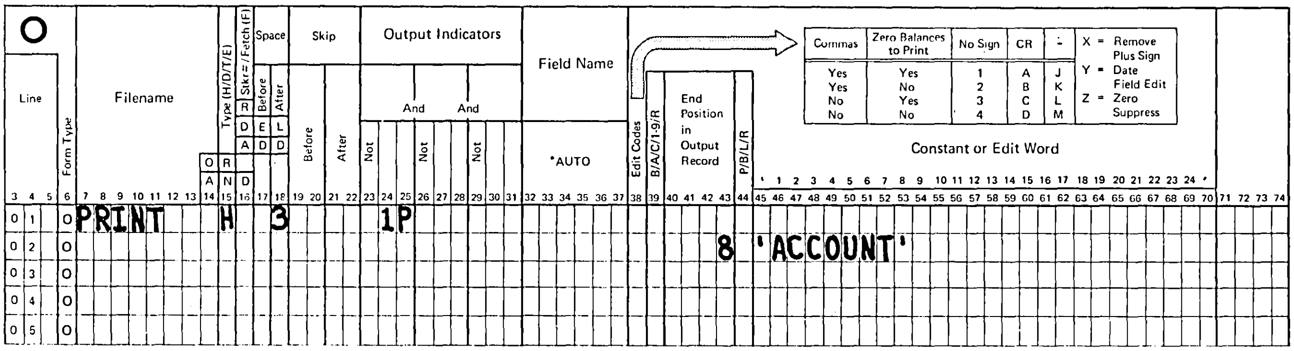

Read a record
Perform all calculations conditioned by control level indicators (columns 7-8 of calculation specifications)
Total output
Overflow output
T = Total
H = Heading
D = Detail
Perform all calculations not conditioned by control level indicators (columns 7-8)
Heading and detail output
Set off control level indicators
NORMAL OVERFLOW ROUTINE
FETCHED OVERFLOW ROUTINE
NORMAL OUTPUT
EXCEPTION OUTPUT
NORMAL OUTPUT
EXCEPTION OUTPUT
Overflow During Detail Output
Overflow During Total Output
Overflow During Detail Calc.
Overflow During Total Calc.
Overflow During Detail Output
Overflow During Total Output
Overflow During Detail Calc.
Overflow During Total Calc.
OF
OF lines
Off

Figure 9-9. Overflow Printing: Setting of the Overflow Indicator

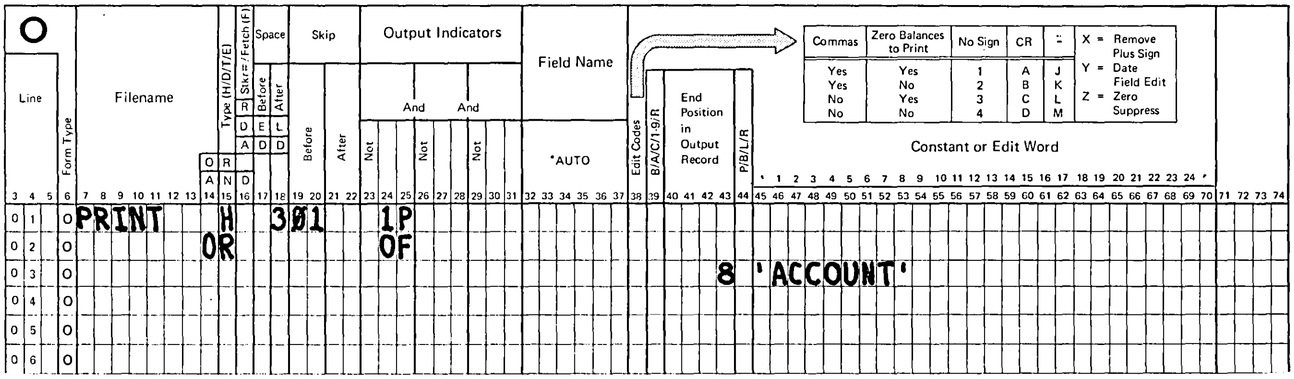
Output Specifications



```
  01 OPRINT   H  3     1P
  02 O                                    8 'ACCOUNT'
  03 O
  04 O
  05 O
```

The 1P indicator is used when headings are to be printed on the first page only.

Output Specifications

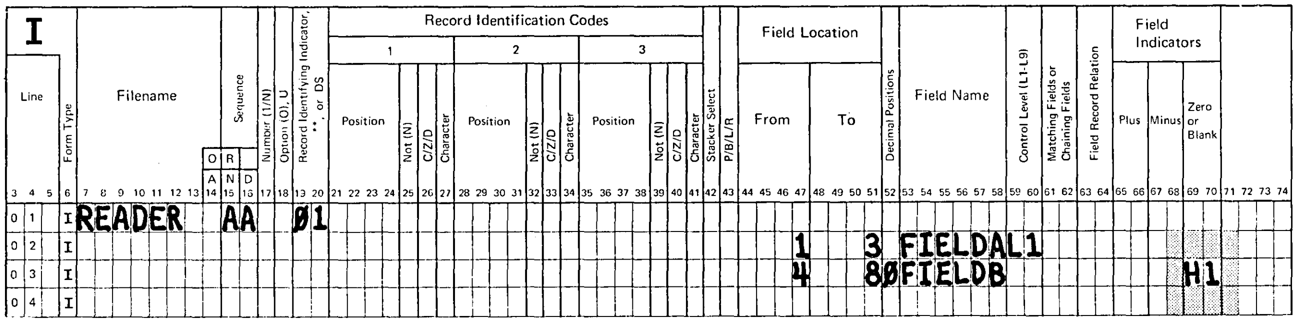


```
  01 OPRINT   H  3Ø1   1P
  02 O       OR        OF
  03 O                                    8 'ACCOUNT'
  04 O
  05 O
  06 O
```

The 1P indicator and an overflow indicator can be used to print headings on every page.

Figure 9-10. 1P Indicators

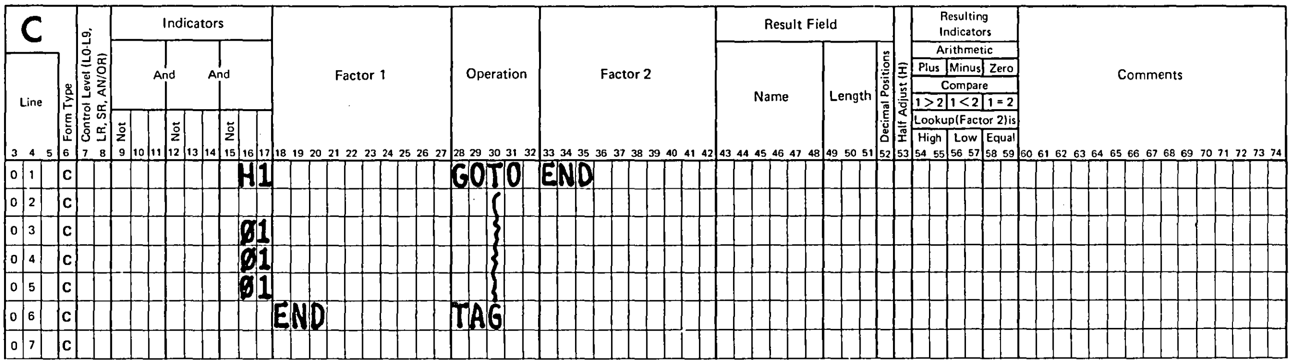
Input Specifications



```
01 IREADER  AA  Ø1
02 I                                        1   3 FIELDAL1
03 I                                        4  8ØFIELDB          H1
04 I
```

When an error condition (zero in FIELDB) is found, the halt indicator turns on.

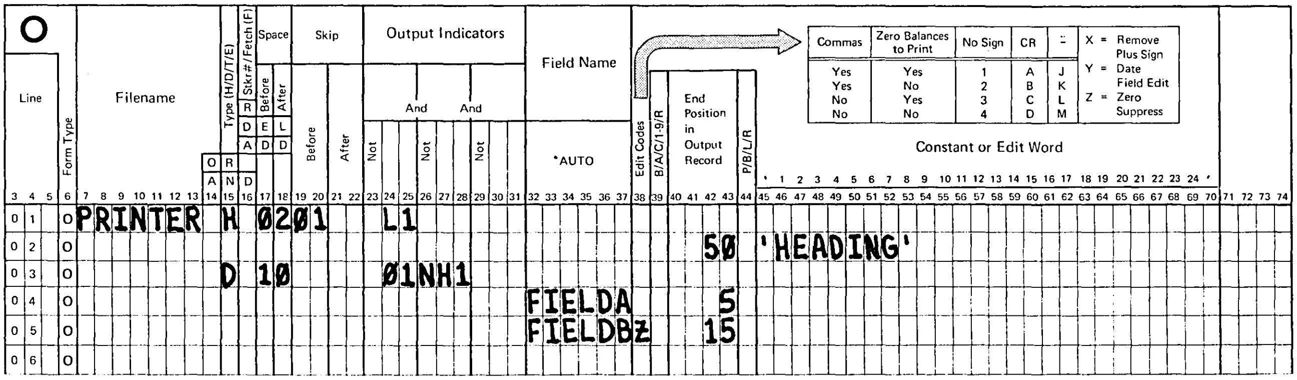
Calculation Specifications



```
01 C         H1          GOTO END
02 C
03 C         Ø1
04 C         Ø1
05 C         Ø1
06 C           END       TAG
07 C
```

When H1 is on, all calculations are bypassed.

Output Specifications



```
01 OPRINTER H Ø2Ø1   L1
02 O                                   5Ø 'HEADING'
03 O        D 1Ø     Ø1NH1
04 O                         FIELDA     5
05 O                         FIELDBZ   15
06 O
```

FIELDA and FIELDB are printed only if H1 is not on. Therefore, if H1 is on, FIELDA and FIELDB are not printed. Use this general format when you do not want information that is in error to be printed.

Figure 9-11. Preventing Fields From Printing

Special Words

Page Numbering (PAGE, PAGE1-PAGE7)

PAGE is a special word that, when used, causes automatic numbering of the pages. Enter the word PAGE or PAGE1 through PAGE7 in these columns if the pages are to be numbered. When a PAGE field is named in these columns without being defined elsewhere, it is assumed to be a four-digit, numeric field with zero decimal positions. Leading zeros are suppressed automatically. A PAGE field can also be defined on input or calculations specifications as a numeric field from 1 to 15 digits long, with zero decimal positions.

The page number starts with 0001 unless otherwise specified, and 1 is automatically added for each new page. See *Columns 53-58 (Field Name)* in Chapter 7, *Input Specifications*, for information concerning page numbering that starts at a number other than 1.

Page numbering can be restarted at any point in a job. To do this, set the PAGE field to zero before it is printed by specifying either blank after in column 39 or an output indicator. If the status of the indicator is as specified, the PAGE field is reset to zero and 1 is added to the PAGE field before it is printed (see Figure 9-12).

The eight possible PAGE entries (PAGE, PAGE1 through PAGE7) may be needed for numbering different types of output pages or for numbering pages for different printer files.

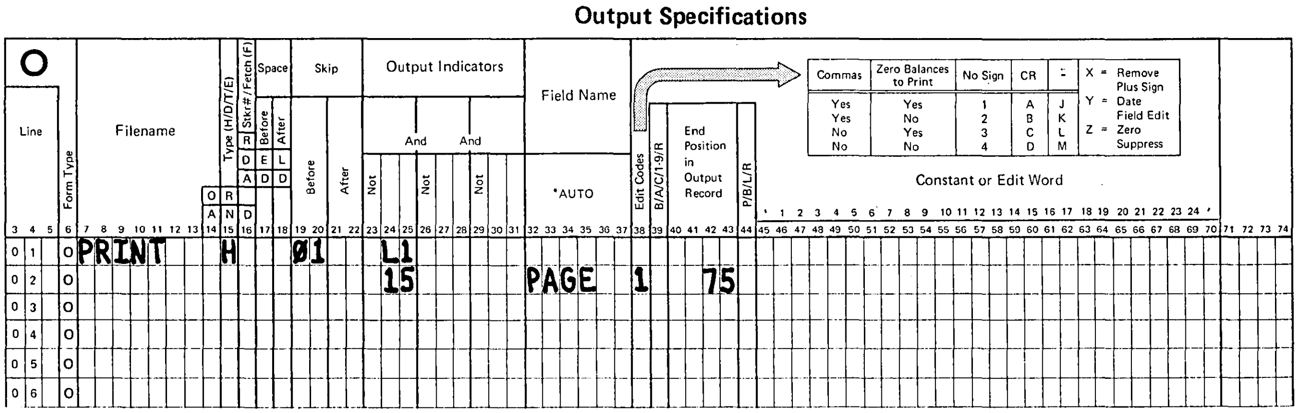
*Repeating Output Fields (*PLACE)*

*PLACE is a special RPG II word that allows you to write the same fields in several locations on one record without naming the fields and giving their end position each time the fields are to be written. The fields repeated by means of *PLACE are written ending in the position specified in columns 40 through 43 of the same specifications line. For example, if FIELDS A, B, and C appear twice on one record, the fields can be specified in two ways:

- Define each field and its corresponding end position each time the field is to be printed or written on disk (see Figure 9-13).
- Use the special word *PLACE (see Figures 9-13 and 9-14).

Both coding methods shown in Figure 9-13 produce a record that looks like this:

Ending Record Positions	10	20	30	40	50	60	75
Fields	FIELDA	FIELDB	FIELDC	FIELDA	FIELDB	FIELDC	FIELDD



Output Specifications

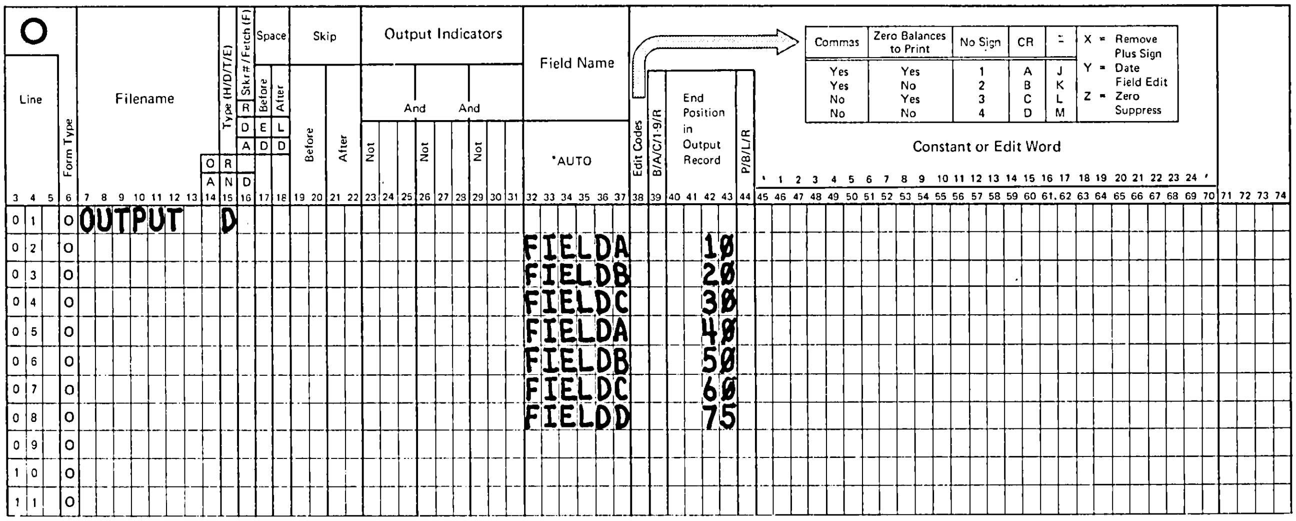
O
Line
Form Type
Filename
Type (H/D/T/E)
Stkr #/Fetch (F)
Space
Before
After
Skip
Before
After
Output Indicators
And
And
Not
Not
Not
Field Name
*AUTO
Edit Codes
B/A/C/1-9/R
End Position in Output Record
P/B/L/R
Constant or Edit Word
Commas
Zero Balances to Print
No Sign
CR
-
X = Remove Plus Sign
Y = Date Field Edit
Z = Zero Suppress

Line	Form Type	Filename	Type	Skip Before	Output Indicators	Field Name	Edit Codes	End Position in Output Record
01	O	PRINT	H	Ø1	L1			
02	O				15	PAGE	1	75
03	O							
04	O							
05	O							
06	O							

When indicator 15 is on, the PAGE field is reset to zero and a 1 is added before the field is printed. When 15 is off, a 1 is added to the contents of the PAGE field before it is printed.

Figure 9-12. Resetting the PAGE Fields to Zero

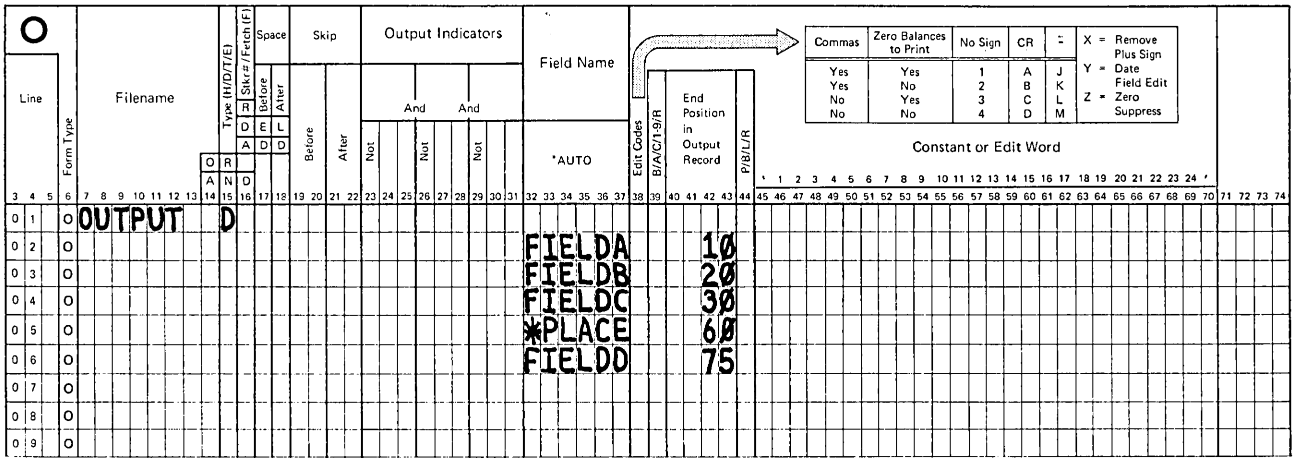
Output Specifications



Line	Form Type	Filename	Type (H/D/T/E)	Field Name	End Position in Output Record
01	O	OUTPUT	D		
02	O			FIELDA	1Ø
03	O			FIELDB	2Ø
04	O			FIELDC	3Ø
05	O			FIELDA	4Ø
06	O			FIELDB	5Ø
07	O			FIELDC	6Ø
08	O			FIELDD	75
09	O				
10	O				
11	O				

To repeat an output field, each field can be defined each time it is to be printed or written to disk.

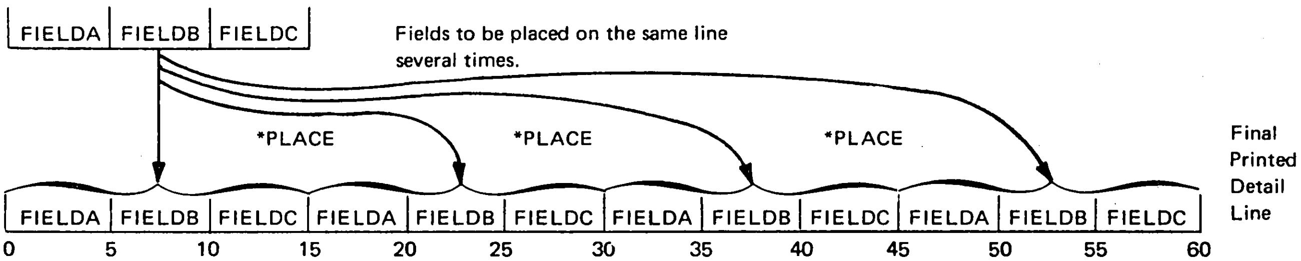
Output Specifications



Line	Form Type	Filename	Type (H/D/T/E)	Field Name	End Position in Output Record
01	O	OUTPUT	D		
02	O			FIELDA	1Ø
03	O			FIELDB	2Ø
04	O			FIELDC	3Ø
05	O			*PLACE	6Ø
06	O			FIELDD	75
07	O				
08	O				
09	O				

Or the special word *PLACE can be used to repeat a group of fields.

Figure 9-13. Writing Fields Twice on the Same Record

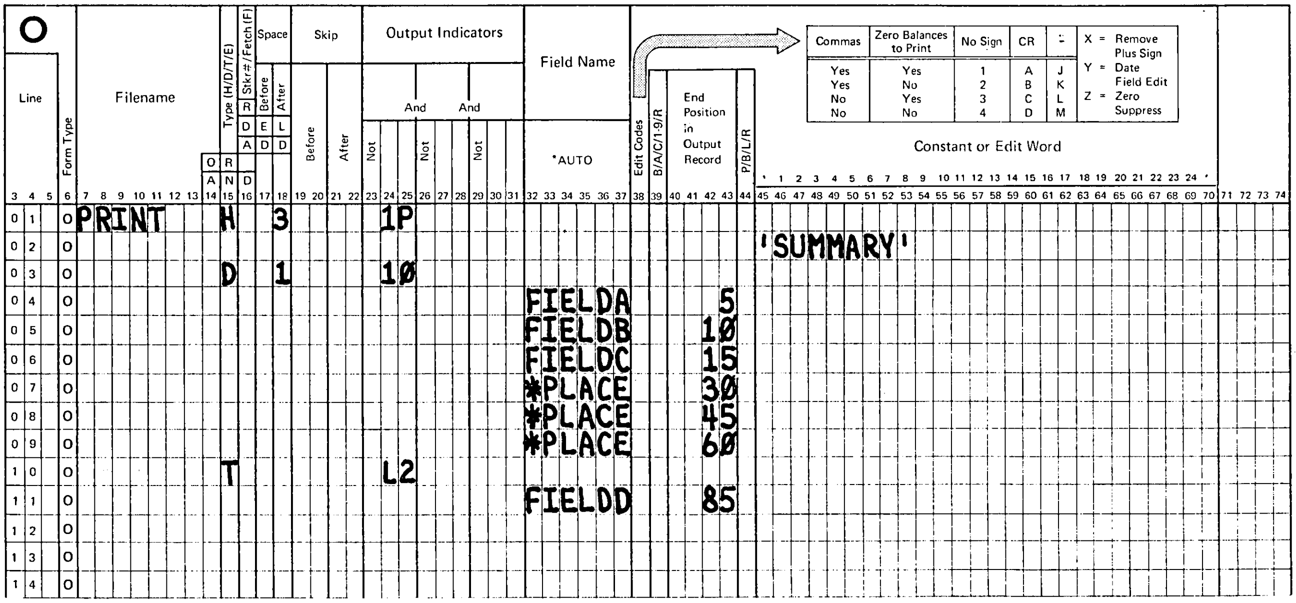


*PLACE can also be used to print the same group of fields several times on the same line. FIELDS A, B, and C are to be printed four times on one line as shown above. They are printed once when they are named and once for every *PLACE entry.

*PLACE is specified after the fields that are to be printed several times on the same line (below). All fields to which *PLACE applies appear on the same record. FIELD D, which appears on the total record, is not affected by *PLACE.

Notice also that an end position is given for every *PLACE. FIELDS A, B, and C have a total length of 15 characters. Thus, the end positions given for the *PLACE entries allow room for the printing of 15 characters. This eliminates any overlapping.

Output Specifications



O

Commas	Zero Balances to Print	No Sign	CR	-	X = Remove Plus Sign Y = Date Field Edit Z = Zero Suppress
Yes	Yes	1	A	J	
Yes	No	2	B	K	
No	Yes	3	C	L	
No	No	4	D	M	

Line (3-5)	Form Type (6)	Filename (7-14)	Type (H/D/T/E) (15)	Space Before (17)	Space After (18)	Skip Before (19-20)	Skip After (21-22)	Output Indicators (23-31)	Field Name (32-37)	Edit Codes (38)	B/A/C/1-9/R (39)	End Position in Output Record (40-43)	P/B/L/R (44)	Constant or Edit Word (45-70)	71-74
01	O	PRINT	H	3				1P							
02	O													'SUMMARY'	
03	O		D	1				1Ø							
04	O								FIELDA			5			
05	O								FIELDB			1Ø			
06	O								FIELDC			15			
07	O								*PLACE			3Ø			
08	O								*PLACE			45			
09	O								*PLACE			6Ø			
10	O		T					L2							
11	O								FIELDD			85			
12	O														
13	O														
14	O														

Figure 9-14. *PLACE

When you specify *PLACE, all fields named for each record type (H/D/T/E) are written as usual in the location specified. The entry *PLACE then causes all of these fields to be written ending at the position specified in columns 40 through 43 of the *PLACE specification. When you specify *PLACE, remember:

- *PLACE must be specified after the field names that are to be written in different positions in one record (see Figure 9-14).
- *PLACE causes all fields within a record type to be written, not just the field name on the line immediately above the *PLACE entry.
- *PLACE must appear on a separate specification line each time a field or a group of fields is to be written.
- An end position no greater than 256 must be specified for every *PLACE line. Allow enough space for all fields to be written (see Figure 9-14); otherwise, overlapping occurs.
- Multiple or successive *PLACE entries can be specified if the fields preceding the first *PLACE specification are to be repeated more than once.
- The leftmost position of the fields to be written by the *PLACE specification is always assumed to be position 1.
- The high end position to be used by *PLACE cannot be defined by a whole array. If a whole array does have the highest end position of all fields preceding the *PLACE, a field must be defined that has an end position greater than the end position of the whole array. This field can be a one-position blank constant.
- Additional fields or constants can be specified after the *PLACE specification and are not affected by any preceding *PLACE specification.

Note: Attempts to use the *PLACE function for other than its defined purpose may produce unpredictable results.

Date Fields (UDATE, UMONTH, UDAY, UYEAR)

To have the date printed on a report or program listing, use special words UDATE, UMONTH, UDAY, or UYEAR. The date fields are established at job setup time. UDATE contains the program date that may not be the same as the date in the result field of the TIME operation. The result field of the TIME operation contains the system date. See the *System Support Reference Manual* for a complete discussion of the system date, program date, and the DATE OCL statement. The following rules apply to date fields:

- UDATE prints a 6-character numeric date field in one of three formats:

  Month/day/year

  Year/month/day

  Day/month/year

  Use columns 19 and 20 of the control specifications to specify the date format and the editing to be done. If columns 19 and 20 are blank, the date format is determined by the contents of column 21.
- Use UDAY for the day only, UMONTH for the month only, and UYEAR for the year only.
- These fields cannot be changed by any operations specified in the program. Thus, these fields are generally used only in compare and test operations.

COLUMN 38 (EDIT CODES)

Use column 38 to:

- Suppress leading zeros in a numeric field
- Omit a sign from the low-order position of a numeric field
- Punctuate a numeric field without establishing an edit word

A table summarizing the edit codes that can be used in column 38 is printed above columns 45 through 70 on the output specifications sheet. If column 38 contains an edit code, columns 45 through 70 must be blank except for the following condition: if asterisk fill (that is, the leading zeros of the field are to be replaced with asterisks) or a floating currency symbol is required, enter '*' or the currency symbol (enclosed in apostrophes) in columns 45 through 47. When an edit code is used to punctuate an array, two spaces are automatically inserted between fields of the array to the left of each element. Only zoned decimal numeric fields can be edited.

Note: Editing fields of a nonprinter file must be done with caution. If you do edit fields of a nonprinter file, you must be aware of the contents of the edited fields and the effects of any operations you want to do on them. For example, if you add an unedited field to an edited field, the results will be erroneous.

Figure 9-15 shows the edit codes and the options they provide. Figure 9-16 illustrates how data looks when it is edited by edit codes. Each code punctuates the field a little differently. All codes suppress leading zeros, except when J is entered in column 21 of the control specifications. In this instance, all zero balances and balances with zero values to the left of the decimal comma are always written with one leading zero (such as 0,00 or 0,04). If an edit code is specified on the output specifications and the edit code is to write zero balances, a zero balance field always has a zero to the left of the decimal comma. The edit code cannot suppress this zero.

Figure 9-17 shows editing for date fields.

Normally, when an edit code is used in column 38, an edit word cannot be defined in columns 45 through 70; however, there are two exceptions:

- If asterisks are to replace leading zeros, enter '*' in columns 45 through 47 of the line containing the edit code.
- If the currency symbol is to appear before the first digit in the field (floating currency symbol), enter the currency symbol (enclosed in apostrophes) in columns 45 through 47 of the line containing the edit code.

Asterisk fill and the floating currency symbol are not allowed with X, Y, and Z edit codes.

The currency symbol can appear before the asterisk fill (fixed currency symbol). This is done in the following manner:

1. Place the currency symbol constant one space before the beginning of the edited field by entering the currency symbol constant on a separate specification line.

2. Place '*' in columns 45 through 47 of the line containing the edit code.

Note: If the currency symbol is not the dollar sign ($), the currency symbol must be entered in column 18 of the control specifications.

Figure 9-18 shows the effect the different edit codes have on the same field with a specified end position for output.

COLUMN 39 (BLANK AFTER)

Entry	Explanation
Blank	Field is not reset.
B	Field specified in columns 32 through 37 is reset to blank or zero after the output operation is complete.

Use column 39 to reset a numeric field to zeros or an alphameric field to blanks. If the field is conditioned by indicators in columns 23 through 31, the blank after is also conditioned. This column must be blank for look-ahead and UDATE fields.

Resetting fields to zeros is useful when totals are accumulated and written for each control group in a program. After the total is accumulated and written for one control group, the total field can be reset to zeros before accumulation begins on the total for the next control group.

If blank after (column 39) is specified for a field to be written more than once, the B should be entered on the last line specifying output for that field.

When blank after is specified with a table name, the field that is blanked contains the last element found by a successful LOKUP, or the first element of the table if no LOKUP or no successful LOKUP occurred. Blank after can be specified with an array element or a whole array.

COLUMNS 40-43 (END POSITION IN OUTPUT RECORD)

Entry	Explanation
1-4096	End position for disk or SPECIAL file
1-4075	End position for BSCA file
1-1919	End position for WORKSTN file
1-132	End position for 132-position printer
1-79	End position for CRT file
K1-K8	Length of format name for a WORKSTN file

Use columns 40 through 43 to define the end position of a field or constant on the output record. All entries in these columns must end in column 43. Enter only the position of the rightmost character in the field or constant.

Note: If columns 40 through 43 are left blank, the field or constant is placed in the output record immediately following the field specified in the previous output specification for that record. If no previous field specification exists for the record, the high-order position of the field is placed in position 1. A blank end position with *PLACE causes the *PLACE to be ignored.

				Entry in Column 21 of Control Specifications			
Edit Code	**Commas**	**Decimal Point**	**Sign for Negative Balance**	**D or Blank**	**I**	**J**	**Zero Suppress**
1	Yes	Yes	No sign	.00 or 0	,00 or 0	0,00 or 0	Yes
2	Yes	Yes	No sign	Blanks	Blanks	Blanks	Yes
3		Yes	No sign	.00 or 0	,00 or 0	0,00 or 0	Yes
4		Yes	No sign	Blanks	Blanks	Blanks	Yes
A	Yes	Yes	CR	.00 or 0	,00 or 0	0,00 or 0	Yes
B	Yes	Yes	CR	Blanks	Blanks	Blanks	Yes
C		Yes	CR	.00 or 0	,00 or 0	0,00 or 0	Yes
D		Yes	CR	Blanks	Blanks	Blanks	Yes
J	Yes	Yes	- (minus)	.00 or 0	,00 or 0	0,00 or 0	Yes
K	Yes	Yes	- (minus)	Blanks	Blanks	Blanks	Yes
L		Yes	- (minus)	.00 or 0	,00 or 0	0,00 or 0	Yes
M		Yes	- (minus)	Blanks	Blanks	Blanks	Yes
X[1]							
Y[2]							Yes
Z							Yes

[1] The X code performs no editing.

[2] The Y code suppresses the leftmost zero only. The Y code edits a three-to six-digit field according to the following pattern:

nn/n
nn/nn
nn/nn/n
nn/nn/nn

Figure 9-15. Edit Codes

Edit Code	Positive Number— 2 Decimal Positions	Positive Number— 0 Decimal Positions	Negative Number— 3 Decimal Positions	Negative Number— 0 Decimal Positions	Zero Balance— Two Decimal Positions			Zero Balance— 0 Decimal Positions
					Entry in Column 21 of Control Specifications			
					D or Blank	I	J	
Unedited	1234567	1234567	00012[1]	00012[1]	000000	000000	000000	000000
1	12,345.67	1,234,567	.120	120	.00	,00	0,00	0
2	12,345.67	1,234,567	.120	120				
3	12345.67	1234567	.120	120	.00	,00	0,00	0
4	12345.67	1234567	.120	120				
A	12,345.67	1,234,567	.120CR	120CR	.00	,00	0,00	0
B	12,345.67	1,234,567	.120CR	120CR				
C	12345.67	1234567	.120CR	120CR	.00	,00	0,00	0
D	12345.67	1234567	.120CR	120CR				
J	12,345.67	1,234,567	.120-	120-	.00	,00	0,00	0
K	12,345.67	1,234,567	.120-	120-				
L	12345.67	1234567	.120-	120-	.00	,00	0,00	0
M	12345.67	1234567	.120-	120-				
X	1234567	1234567	00012[1]	00012[1]	000000	000000	000000	000000
Y								0/00/00
Z	1234567	1234567	120	120				

[1] The EBCDIC values of negative decimal numbers do not print as numerics. The alpha equivalent is printed if column 45 of the control specifications contains a blank (that is, the program halts for unprintable characters), or if the operator takes a continue option to the halt. A minus zero (hex D0) prints as a blank, a minus 1 (hex D1) as J, a minus 2 (hex D2) as K, and so on.

Figure 9-16. Examples of Edit Code Usage

		Control Specification					
UDATE	Edit Code	Contents of Column 19	Contents of Column 20	Contents of Column 21			
				Blank	D	I	J
Jan 30, 1978	Y	Blank	Blank	1/30/78	30/01/78	30.01.78	30.01.78
			–	1-30-78	30-01-78	30-01-78	30-01-78
		M	Blank	1/30/78	1/30/78	1/30/78	1/30/78
			–	1-30-78	1-30-78	1-30-78	1-30-78
		D	Blank	30.01.78	30.01.78	30.01.78	30.01.78
			–	30-01-78	30-01-78	30-01-78	30-01-78
		Y	Blank	78.01.30	78.01.30	78.01.30	78.01.30
			–	78-01-30	78-01-30	78-01-30	78-01-30

Figure 9-17. Date Fields

	Negative Number – 2 Decimal Positions End Position Specified as 10									
Edit	Output Print Positions									
Code	3	4	5	6	7	8	9	10	11	
Unedited				0	0	4	1	K[1]		
1					4	.	1	2		
2					4	.	1	2		
3					4	.	1	2		
4					4	.	1	2		
A			4	.	1	2	C	R		
B			4	.	1	2	C	R		
C			4	.	1	2	C	R		
D			4	.	1	2	C	R		
J				4	.	1	2	-		
K				4	.	1	2	-		
L				4	.	1	2	-		
M				4	.	1	2	-		
X				0	0	4	1	K[1]		
Y			0	/	4	1	/	2		
Z						4	1	2		

[1] K represents a negative 2.

Figure 9-18. Effect of Edit Codes on End Position

COLUMN 44 (PACKED OR BINARY FIELD)

Entry	Explanation
Blank	Field is zoned decimal numeric data or alphameric data. Leave this column blank for nondisk files.
P	Field is to be written on disk in packed decimal format.
B	Field is to be written on disk in binary format.

Use column 44 to specify whether a numeric field (decimal number) is to be written on disk in packed decimal or binary format. Packed decimal and binary fields cannot be displayed or printed; these fields can be written only on disk. Column 44 must be blank for *PLACE.

After decimal numbers are processed, they can be left in the zoned decimal format. However, for more efficient use of disk space, convert decimal numbers into packed decimal or binary format. When binary output is specified, a numeric field one to four digits long (zoned decimal in storage) is converted into a 2-byte binary field when it is written on disk; a numeric field five to nine digits long is converted into a 4-byte binary field. When packed decimal output is specified, a byte of disk storage (except for the low-order byte) can contain two decimal numbers. See *Column 43 (Packed or Binary Field)* in Chapter 7, *Input Specifications*, for a description of how data fields are represented in zoned decimal, packed decimal, and binary formats.

Notes:
1. Although packed and binary fields require less disk storage space, the conversion routines needed to handle such data increase the object program size (and execution time).
2. Key fields cannot contain any hex FF characters. Therefore, if the key field is a binary field, you must be sure that no hex FF characters appear in the key field.

COLUMNS 45-70 (CONSTANT OR EDIT WORD)

Use columns 45 through 70 to specify a constant, the format name for a WORKSTN file, or an edit word. If you are using edit codes, you can also use columns 45 through 47 to specify a floating currency symbol or asterisk fill.

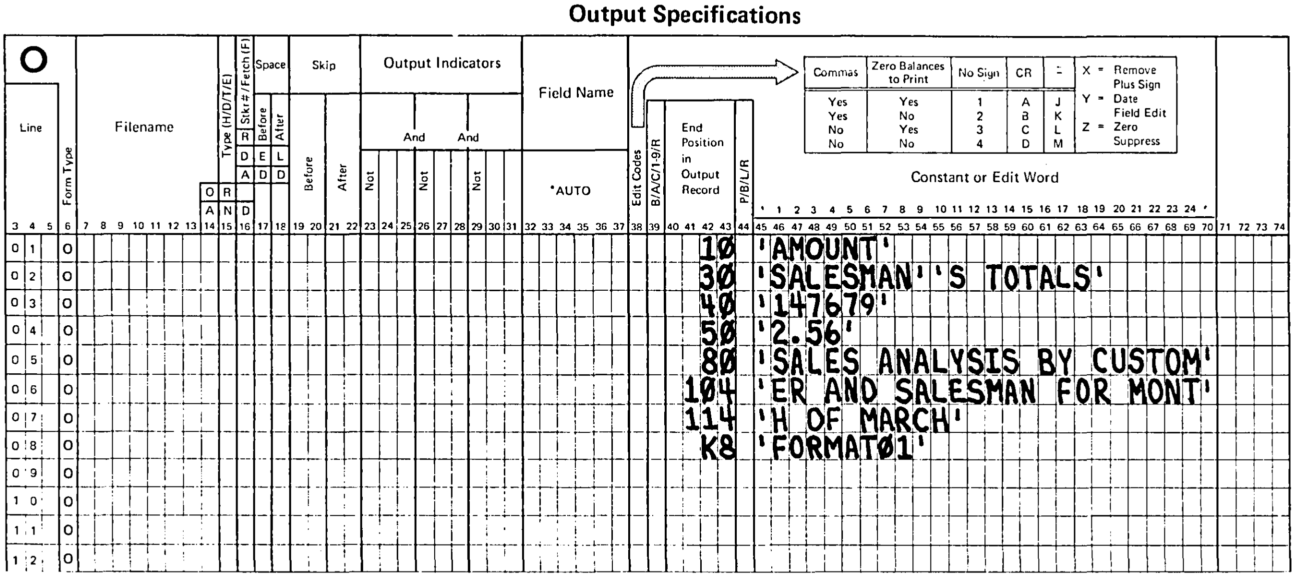
Constants

A constant is any unchanging information that is to appear on a report. Constants are usually words used for report headings or column headings.

The following rules apply to constants (see Figure 9-19 for examples):

- Field name (columns 32 through 37) must be blank.
- A constant must be enclosed in apostrophes. Enter the leading apostrophe in column 45.
- An apostrophe in a constant must be represented by two apostrophes. For example, if the word *you're* appears in a constant it must be coded as 'YOU''RE'.
- Numeric data can be used as a constant.
- Up to 24 characters of constant information can be placed in one line. Additional lines can be used, but each line must be treated as a separate line of constants. The end position is specified in columns 40 through 43. If no end position is specified, the constant is placed in the output record immediately following the field or constant specified in the previous output specification line for that record (see *Columns 40-43, End Position in Output Record*, in this chapter).

Output Specifications



Commas	Zero Balances to Print	No Sign	CR	-
Yes	Yes	1	A	J
Yes	No	2	B	K
No	Yes	3	C	L
No	No	4	D	M

X = Remove Plus Sign
Y = Date Field Edit
Z = Zero Suppress

Line	Form Type	End Position in Output Record	Constant or Edit Word
01	O	1Ø	'AMOUNT'
02	O	3Ø	'SALESMAN''S TOTALS'
03	O	4Ø	'147679'
04	O	5Ø	'2.56'
05	O	8Ø	'SALES ANALYSIS BY CUSTOM'
06	O	1Ø4	'ER AND SALESMAN FOR MONT'
07	O	114	'H OF MARCH'
08	O	K8	'FORMATØ1'
09	O		
10	O		
11	O		
12	O		

Figure 9-19. Examples of Output Constants

Format Name

The name of the display screen format that is used by the WORKSTN file must be specified in columns 45 through 54. One format name is required for each output record for the WORKSTN file; the specification of more than one format name per record is not allowed. The format name must be enclosed in apostrophes. This is the same name that is specified in columns 7 through 14 of the S specification line on the display screen format specifications. You must also enter Kn in columns 42 and 43, where n is the length of the format name. For example, if the format name is FORM1, enter K5 in columns 42 and 43.

For more information on the display screen format, see Chapter 13, *WORKSTN File Considerations and Sample Programs.*

Note: The output specifications line containing the format name cannot be conditioned by any indicators.

Edit Codes

If column 38 contains an edit code, columns 45 through 70 must be blank except for the following condition: if asterisk fill or a floating currency symbol is required, enter '*' or the currency symbol (enclosed in apostrophes) in columns 45 through 47. When '*' is entered in columns 45 through 47, asterisks replace all leading zeros (**NN). When the currency symbol is entered in columns 45 through 47, the currency symbol appears before the first digit in the field ($N.NN). For more information on edit codes, see *Column 38 (Edit Codes)* in this chapter.

Note: Asterisk fill and the floating currency symbol cannot be used with the X, Y, and Z edit codes.

Edit Words

An edit word allows more flexibility in punctuating a numeric field than an edit code. You specify directly:

- If commas, decimal points, and zero suppression are needed
- If the negative sign should print
- If the output is dollars and cents, whether a currency symbol and leading asterisks should be used

The following rules apply to edit words:

- Column 38 (edit codes) must be blank.
- Columns 32 through 37 (field name) must contain the name of a numeric field.
- Columns 40 through 43 must contain the end position of the field in the output record.
- An edit word must be enclosed in apostrophes. Enter the leading apostrophe in column 45. The edit word itself must begin in column 46.
- Any printable character is valid, but certain characters in certain positions have special uses. See *Editing Considerations* in the following text.
- An edit word cannot be longer than 24 characters.
- The number of replaceable characters in the edit word must equal the length of the field to be edited. See *Editing Considerations* in the following text.
- All leading zeros are suppressed unless a zero or asterisk is specified in the edit word. The leftmost zero or asterisk indicates the last leading zero in the field to be replaced by a blank or asterisk.
- Any zeros or asterisks following the leftmost zero or asterisk are treated as constants; they are not replaceable characters.

Editing Considerations

When using an edit word, leave exactly enough room on the printed form for the edited word. If the field to be edited is 7 characters long on the input record, consider whether seven positions allow enough space for it to print on the report. If the field is edited, it might contain more than 7 characters.

When computing the length of an edited output field, determine how many of the editing characters are replaceable. The number of replaceable characters in the edit word must equal the length of the field to be edited. A replaceable character is a character in the edit word that does not require a position in the output file. The replaceable characters are:

0 (if used for zero suppression)

* (if used for asterisk fill)

ƀ (blank)

the currency symbol (if it appears immediately to the left of zero suppress; that is, a floating currency symbol)

A fixed currency symbol, decimal points, floating currency symbol, commas, ampersands (representing blanks), negative signs (- or CR), and constant information all require space in the output file.

Note: There are two exceptions to the rule that the number of replaceable characters in the edit word must equal the length of the field to be edited. The exceptions are:

- An extra space must be left in the edit word for the floating currency symbol to ensure a print position for the currency symbol if the output field is full:

Unedited Field	Edit Word	Edited Field	Unedited Field Length	Replaceable Characters in Edit Word
72432N	'b̸b̸,b̸$0.b̸b̸&-'	$7,243.25b̸-	6	7

- An extra space can be left in the edit word if the first character in the edit word is a zero. In this case, the field to be edited is not zero suppressed by the leading zero, but all other specified editing is performed:

Unedited Field	Edit Word	Edited Field	Unedited Field Length	Replaceable Characters in Edit Word
00746J	'0ƀƀƀ,ƀƀƀ'	007,461	6	7

Formatting Edit Words

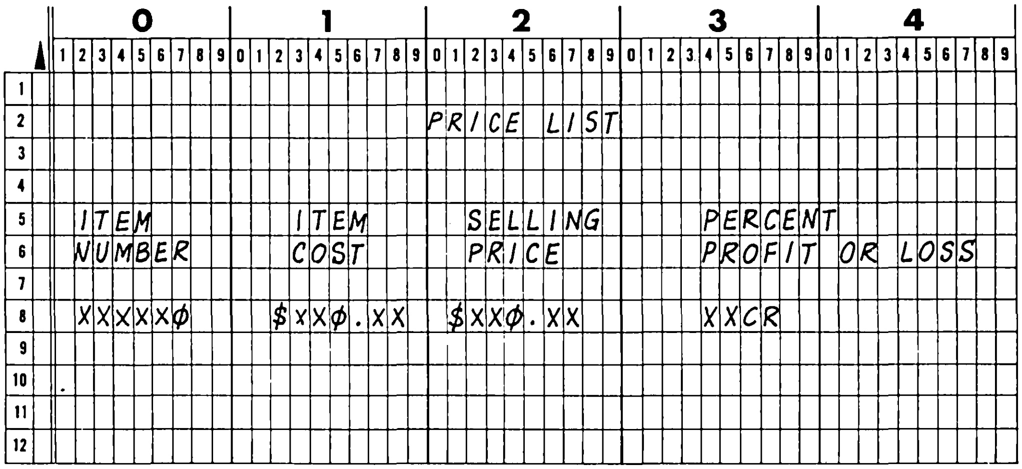
The printer spacing chart can help you format edit words. Figure 9-20 shows how an output line can be formatted on this chart. The Xs and zeros show field positions. A zero indicates where zero suppression stops. An X indicates that any number can appear in the position. Use blanks in place of the Xs when writing the edit words.

If it is necessary to show a negative number, a sign must be included in the edit word. Use either the minus sign (-) or the letters CR. These print only for a negative number; however, the character positions they require must be included when you enter the end position of the field on the output specifications. Figure 9-20 shows that for the field PERCPL, CR is to be printed for a negative balance. Assume the field PERCPL contains the negative data 2N (which is -25). The printed output is: 25CR. If PERCPL is positive, CR does not print and the same field appears as 25.

Unedited Data

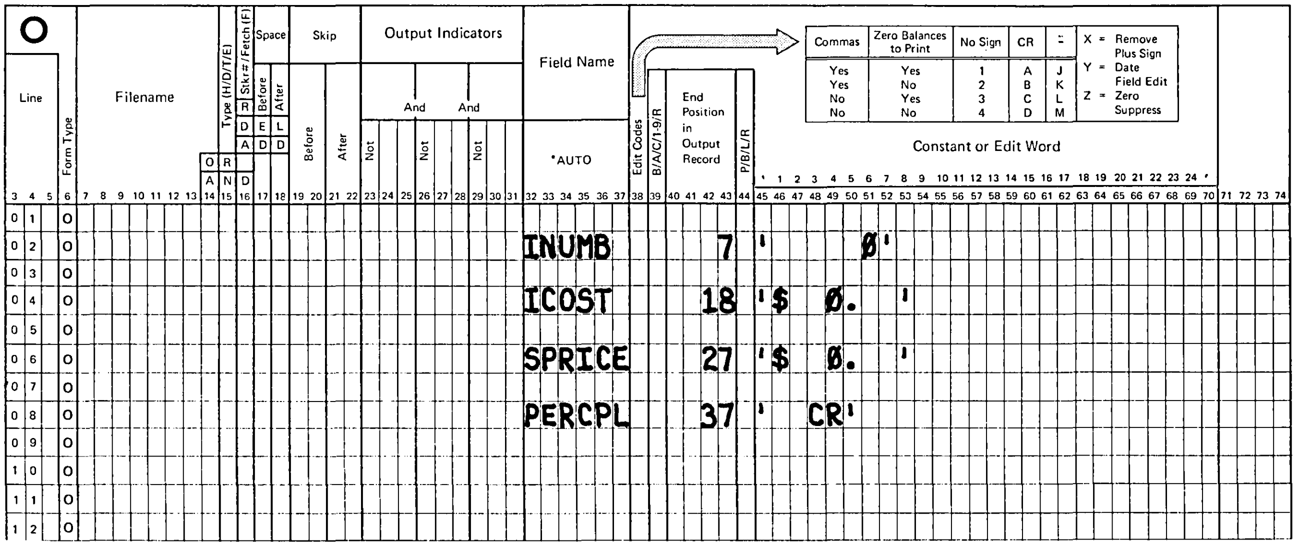
Item number	000241
Item cost	02000
Selling price	02200
% profit or loss	25

Printer Spacing Chart



```
                   PRICE LIST


 ITEM       ITEM     SELLING     PERCENT
NUMBER     COST      PRICE       PROFIT OR LOSS

 XXXXXØ    $XXØ.XX  $XXØ.XX      XXCR
```

Output Specifications



Line	Form Type	Field Name	End Position in Output Record	Constant or Edit Word
01	O			
02	O	INUMB	7	`'     Ø'`
03	O			
04	O	ICOST	18	`'$  Ø.  '`
05	O			
06	O	SPRICE	27	`'$  Ø.  '`
07	O			
08	O	PERCPL	37	`'  CR'`
09	O			
10	O			
11	O			
12	O			

Edited Output

241 $ 20.00 $ 22.00 25

Figure 9-20. Using the Printer Spacing Chart to Format Data

A minus sign can also indicate a negative number. To leave a space between the number and the negative sign, place an ampersand (&) in the edit word before the minus sign. PERCPL then prints as 25ƀ-.

To print a currency symbol to the left of a field, enter the currency symbol next to the first apostrophe and then include the necessary blanks and punctuation in the edit word. A currency symbol in this position is called a fixed currency symbol. The SPRICE field shown in Figure 9-21, line 01, can look like any of the following (N stands for any number):

```
$NNN.NN

$  NN.NN

$    N.NN

$      .NN
```

If you do not want blank spaces between the currency symbol and the first digit when zero suppression occurs, you can indicate asterisks (*) fill. Instead of using 0 to indicate zero suppression, use the asterisk (*) to indicate that all extra spaces should be filled with asterisks. The SPRICE field in Figure 9-21, line 03, can look like any of the following (N stands for any number):

```
$NNN.NN
$*NN.NN
$**N.NN
$***.NN
```

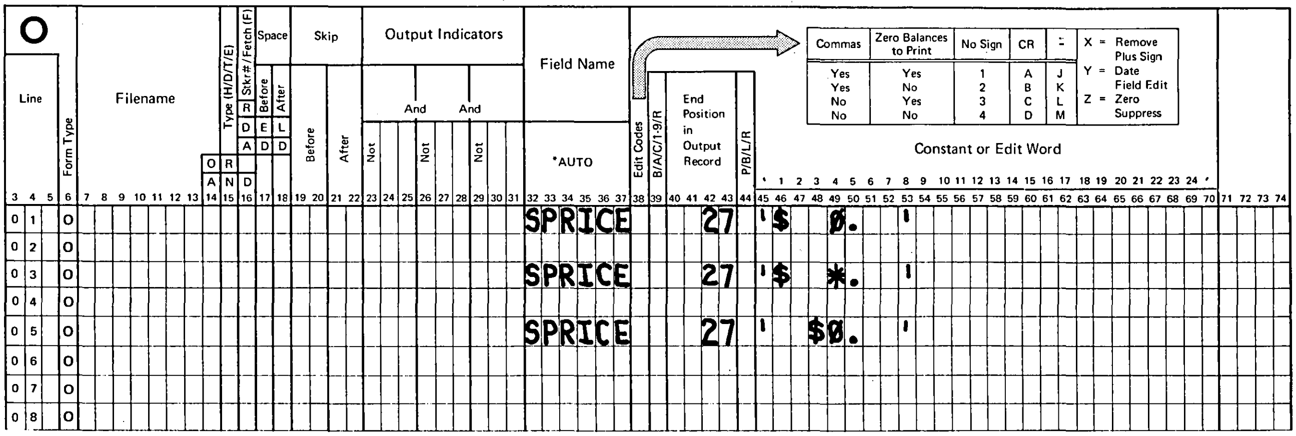
To have the currency symbol always print next to the leftmost digit, place the currency symbol next to the zero suppress 0 in the edit word. A currency symbol that changes positions depending upon the number of positions that are zero suppressed is known as a floating currency symbol. When printed, the SPRICE field in Figure 9-21, line 05, can look like any of the following:

```
$NNN.NN
 $NN.NN
  $N.NN
   $.NN
```

An extra space must be left in the edit word for the floating currency symbol to ensure a print position for the currency symbol if the output field is full.

Note: If the currency symbol is not the dollar sign ($), the currency symbol must be entered in column 18 of the control specification.

Output Specifications



O

Commas	Zero Balances to Print	No Sign	CR	-	
Yes	Yes	1	A	J	X = Remove Plus Sign
Yes	No	2	B	K	Y = Date Field Edit
No	Yes	3	C	L	Z = Zero Suppress
No	No	4	D	M	

Line	Form Type	Field Name	End Position in Output Record	Constant or Edit Word
01	O	SPRICE	27	'$ Ø. '
02	O			
03	O	SPRICE	27	'$ *. '
04	O			
05	O	SPRICE	27	' $Ø. '
06	O			
07	O			
08	O			

Filename
Type (H/D/T/E)
Stkr #/Fetch (F)
Space
Before
After
Skip
Before
After
Output Indicators
And
And
Not
Not
Not
OR
AND
ADD
DEL
*AUTO
Edit Codes
B/A/C/1-9/R
P/B/L/R

Figure 9-21. Different Edit Words Used on the Same Field

Examples of Edit Words

Figure 9-22 shows examples of edit words. For all examples, column 38 (edit codes) of the output specification is blank. The symbol ƀ indicates where blank spaces appear in the edited data field. Zeros are not slashed because there should be no confusion with the letter O.

Examples 1 through 9 are sample edit words for some of the most frequently used output formats. Examples 10 through 71 show several possible edit words and the resulting edited data field.

The following paragraphs provide an additional explanation of each example shown in Figure 9-22:

1. A decimal point appears between dollars and cents; commas appear every three positions in the dollar portion of the field. The symbol CR appears in the edited data field when the data is negative; otherwise, it is replaced by blanks. Because zero suppression occurs through the units position (zero in the edit word just left of the decimal point), blanks replace leading zeros and constants until a significant digit or until the specified zero is encountered. If there are no significant digits, the zero is replaced with a blank; however, the decimal point and the digits to its right always appear in the edited data.

2. Leading zeros and commas are replaced by blanks through the position of the zero suppression 0 (column 54). Thus, if the entire data field is zero, a zero appears only in the low-order position of the edited data. A minus sign appears in the edited data if the field is negative; if not, the minus sign is replaced by a blank. The constant ON HAND always appears in the edited data as it is specified in the edit word, regardless of whether the minus sign appears as specified or as a blank.

3. Because the zero suppression 0 appears in the tens position of the edit word, leading zeros and constants are retained starting with the units position. Because the dollar sign is placed just left of the zero suppression 0, it is a floating dollar sign. In an edited data field, the floating dollar sign always appears to the immediate left of the first digit or retained constant. An extra position is allowed in the high-order portion of the edit word to accommodate the floating dollar sign. The minus sign appears if the field is negative; the asterisk always appears as a constant because a zero is specified to the left of it.

4. This example is similar to example 3 except that (1) zero suppression occurs up to the decimal point, (2) CR indicates a negative value, and (3) two asterisks print at the end of the edited data. In the edited data shown, the dollar sign has *floated* to the left to precede the first significant digit. If the unedited data were all zeros, the output record would appear as $.00ƀƀ**. An extra position is allowed in the leftmost portion of the edit word for the dollar sign.

5. This example is similar to example 4 except that (1) no symbol indicates a negative value, and (2) the edit word includes a fixed dollar sign. Because the dollar sign is placed in the leftmost position of the word and is not followed immediately by a zero suppression 0, it is a fixed dollar sign. The fixed dollar sign always appears in the leftmost position of the edited data field.

6. A space can be left in the edited data field between a fixed dollar sign and the first digit, even when the entire field contains significant digits. An ampersand (&) in an edit word appears as a blank in the edited field. The minus sign appears in the edited data if the field is negative; the constant GROSS always appears in the edited data.

7. If a zero or asterisk is not specified, zero suppression can occur throughout the field; thus, edited data begins with the first significant digit. If the unedited data field contained only zeros, the entire edit word except the minus sign is replaced by blanks when the field is edited.

8. Zero suppression can occur throughout the field. The symbol CR appears in the edited data field when the field contains a minus zero.

9. When asterisk fill is indicated, asterisks replace all positions in the edit word to the left of the first significant digit. If the asterisk is in the rightmost position of the edit word, the entire edited field contains asterisks when the data is all zeros.

10, 11, and 12. When no edit word is used, the data in the output record has the same format as the unedited data. The low-order position of the output field is printed as an alphabetic character (J through R) if the source data field is negative.

13, 14, and 15. When a blank edit word is used, all leading zeros are replaced with blanks, and any sign in the low-order position of the unedited field is removed when the data is edited. Negative values are not identified.

16. Using a zero suppression 0 in the rightmost position of the edit word is the same as using a blank edit word (examples 13, 14, and 15).

17. Although the zero suppression 0 appears in the high-order position of the edit word, suppression of the first leading zero cannot be avoided. See the note under *Editing Considerations* for a discussion of this exception.

18 and 19. An ampersand appears as a blank in the edited data. The symbol CR appears in the edited data if the field is negative; CR is replaced by blanks if the field is positive. The constant NET always appears in the edited data field.

20. An ampersand appears as a blank in the edited data. A minus sign, instead of CR, indicates negative values.

21 and 22. NET CR indicates when the edited data field is negative. Therefore, when the edited field is positive, CR is replaced with blanks.

23. The constant PROFIT appears in the edited data field. Negative values are not identified.

24 and 25. This example is similar to example 20, except that a fixed dollar sign is shown. An extra position is added to the edit word to allow for the dollar sign.

26. When the dollar sign appears to the immediate left of the zero suppression 0, it becomes a floating dollar sign, even when the dollar sign is entered in the leftmost position of the edit word.

27 and 28. The floating dollar sign is shown for different numbers of leading zeros. An extra position must be allowed in the high-order portion of the edit word for the dollar sign.

29 and 30. Even if there is no zero in the edit word, the minus sign appears in the edited field when the contents of the data field are minus zero. The constant TOTAL always appears in the edited field.

31. Some zeros can appear in the edited field when the entire field is zero. In this example zero suppression occurs through the position of the zero in the edit word (column 54), thus leaving two positions in which zeros can appear in the edited field.

32. This example is similar to example 31, except that a floating dollar sign replaces the last suppressed zero.

33. Because the dollar sign is adjacent to the zero in the low-order position, it is a floating dollar sign. The floating dollar sign appears in the low-order position of an all-zero data field. This gives full protection with a floating dollar sign, even when all leading zeros are suppressed.

34. Because the asterisk appears in the low-order position of the edit word, asterisks appear throughout the edited field when the contents of that field are zero. This gives full protection with an asterisk, even when all leading zeros are suppressed.

35. Asterisk protection occurs only to a certain position; thereafter, any additional leading zeros appear in the edited field.

36 and 37. Asterisk protection and zero suppression are indicated for the leftmost position only. The asterisk is replaced by a significant digit if one occurs in that position. Negative values are not identified.

38. Asterisk protection and zero suppression are indicated for the entire field. Asterisks are replaced by significant digits. Negative values are not identified.

39 and 40. Blanks replace punctuation and zeros to the left of the first significant digit. The decimal point is lost when there are fewer than three significant digits. The CR symbol is printed for an all-zero negative field; the constants NET and -NET always appear in the edited field.

41. The ampersand, which appears in the edited field as a blank, makes it possible to keep the dollar sign fixed while limiting zero suppression to the minimum one position. All punctuation is retained regardless of the number of leading zeros because the zero in the edit word is placed to the left of the first comma.

Edit Word 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70	Example Number	Source Data	Appears in Output Record as:
`' ,    ,  0.  &CR'`	1	0000000005 -	ƀƀƀƀƀƀƀƀƀƀ.05ƀCR
`' ,    , 0 -&ON&HAND'`	2	00000000	ƀƀƀƀƀƀƀƀƀ0ƀƀONƀHAND
`'   ,   ,$0 .  -*'`	3	0000000005 +	ƀƀƀƀƀƀƀƀƀ$0.05ƀ*
`'   ,   , $0.  CR**'`	4	0034567890 -	ƀƀƀ$345,678.90CR**
`'$  ,   ,  0.  '`	5	0000000000	$ƀƀƀƀƀƀƀƀƀƀ.00
`'$&  ,   ,  0 .  &-&GROSS'`	6	1234567890 -	$ƀ12,345,678.90ƀ-ƀGROSS
`'   ,   ,   .  -'`	7	00000000123 -	ƀƀƀƀƀƀƀƀƀƀ1.23-
`' ,    ,   .  CR'`	8	0000000000 -	ƀƀƀƀƀƀƀƀƀƀƀƀƀƀCR
`' ,    , * .  &-'`	9	0000135792 -	*****1,357.92ƀ-
	10	0000135678	0000135678
	11	0000135678 +	0000135678
	12	0000135678 -	000013567Q
`'          '`	13	0000000000	ƀƀƀƀƀƀƀƀƀƀ
`'          '`	14	0000135678 +	ƀƀƀƀ135678
`'          '`	15	0000135678 -	ƀƀƀƀ135678
`'         0'`	16	0000135678 -	ƀƀƀƀ135678
`'0         '`	17	0000135678 +	ƀ000135678
`'          &CR&NET'`	18	0000135678 +	ƀƀƀƀ135678ƀƀƀƀNET
`'          &CR&NET'`	19	0000135678 -	ƀƀƀƀ135678ƀCRƀNET
`'          &-&&NET'`	20	0000135678 -	ƀƀƀƀ135678ƀ-ƀƀNET
`'          &NET&CR'`	21	0000135678	ƀƀƀƀ135678ƀNETƀƀƀ
`'          &NET&CR'`	22	0000135678 -	ƀƀƀƀ135678ƀNETƀCR
`'          &&PROFIT'`	23	0000135678	ƀƀƀƀ135678ƀƀPROFIT
`'$          &-&NET'`	24	0000135678 +	$ƀƀƀƀ135678ƀƀƀNET
`'$          &-&NET'`	25	0000135678 -	$ƀƀƀƀ135678ƀ-ƀNET
`'$0         -&NET'`	26	0000135678	ƀ$000135678ƀƀNET
`'       $0  &CR'`	27	0000135678 -	ƀƀƀƀ$135678ƀCR
`'       $0  &CR'`	28	1234567809 -	$1234567809ƀCR
`'          &-&TOTAL'`	29	0000000000 -	ƀƀƀƀƀƀƀƀƀƀƀ-ƀTOTAL
`'         0&-&TOTAL'`	30	0000000000 -	ƀƀƀƀƀƀƀƀƀƀƀ-ƀTOTAL
`'$       0  &CR&GROSS'`	31	0000000000 +	$ƀƀƀƀƀƀƀ00ƀƀƀƀGROSS
`'       $0  &CR&GROSS'`	32	0000000000 -	ƀƀƀƀƀƀƀƀ$00ƀCRƀGROSS
`'         $0'`	33	0000000000 -	ƀƀƀƀƀƀƀƀƀƀ$

Figure 9-22 (Part 1 of 2). Example of Edit Words

Edit Word 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70	Example Number	Source Data	Appears in Output Record as:
' *&CR'	34	0000000000 -	**********ƀCR
' * &CR'	35	0000000000 -	********00ƀCR
'* '	36	0000135678 -	*000135678
'* '	37	1234567890 +	1234567890
' *'	38	0000135678 -	****135678
' , , . &CR&&NET'	39	0000135678 -	ƀƀƀƀƀ1,356.78ƀCRƀƀNET
' , , . &CR&-NET'	40	0000135678	ƀƀƀƀƀ1,356.78ƀƀƀƀ-NET
'$&Ø , , . &NET'	41	0000135678 +	$ƀƀ0,001,356.78ƀNET
' , ,$Ø . &NET'	42	0000000005	ƀƀƀƀƀƀƀƀƀ$0.05ƀNET
' , , $Ø. '	43	0000000005	ƀƀƀƀƀƀƀƀƀƀ$.05
' , , $Ø. -'	44	1234567890 -	$12,345,678.90-
' , , $Ø. CR'	45	0001356789 -	ƀƀƀƀ$13,567.89CR
' , , *. *CR**'	46	0000135678 +	*****1,356.78ƀƀƀ**
' , Ø. &CR*'	47	00000000 -	ƀƀƀƀƀƀƀ.00ƀCR*
' , , *. -'	48	0000000000 -	**********.00-
' , $Ø, . -SALES'	49	0000001234	ƀƀƀƀƀƀ$,012.34ƀSALES
'$& , , Ø. CR'	50	1234567890 -	$ƀ12,345,678.90CR
' , , , -OLD&BALNCE'	51	1234567890 -	1,234,567,890-OLDƀBALNCE
' , , , Ø -OLD&BALNCE'	52	0000000000 +	ƀƀƀƀƀƀƀƀƀƀƀƀ0ƀOLDƀBALNCE
' , , DOLLARS CENTS'	53	0000135678	ƀƀƀƀƀ1,356DOLLARS78CENTS
' , DOLLARS CENTS'	54	000000 +	ƀƀƀƀƀƀƀƀƀƀƀƀƀƀCENTS
' , Ø DOLLARS CENTS&CR'	55	000000	ƀƀƀƀ0DOLLARS00ƀƀƀƀƀƀƀƀ
' Ø LBS.& OZ.TARE&-'	56	000002 +	ƀƀƀ0LBS.ƀ02ƀƀƀƀƀƀƀƀ
' ØLBS. OZ.TARE&-'	57	000002 -	ƀƀƀƀLBS.02OZ.TAREƀ-
'Ø - - '	58	095140036	ƀ95-14-0036
'Ø HRS. MINS.&O''CLOCK'	59	0042	ƀ0HRS.42MINS.ƀO'CLOCK
' , Ø. '	60	000000	ƀƀƀƀƀ.00
' , .Ø '	61	000000	ƀƀƀƀƀƀƀ0
' , $,Ø . '	62	00123456	ƀƀƀ1$,234.56
' , , Ø * '	63	0000000000	ƀƀƀƀƀƀƀƀƀ0*00
'Ø, ,Ø '	64	001234	ƀ,012,034
' , *, * '	65	0000001234	******,012*34
'& ,* Ø, '	66	013579	***130,579
' - - &LATER'	67	093066	ƀ9-30-66ƀLATER
' & & &LATER'	68	093066	ƀ9ƀ30ƀ66ƀLATER
' / / '	69	100166	10/01/66
' , , . -'	70	000000015 -	ƀƀƀƀƀƀƀƀƀƀ15-
' , , Ø . -'	71	000000005	ƀƀƀƀƀƀƀƀ0.05ƀ

Figure 9-22 (Part 2 of 2). Example of Edit Words

42 through 45. The floating dollar sign is placed so that at least the decimal point is retained regardless of the number of leading zeros. The extra position appears in the leftmost position of the edit word to allow for the floating dollar sign.

46. Asterisk protection and zero suppression occur up to the decimal point. The decimal point is retained regardless of the number of leading zeros. Asterisks replace punctuation when leading zeros are suppressed. The second asterisk appears only when the edited data field is negative; the third and fourth asterisks always appear in the edited field.

47. This example shows a standard programming technique for retaining the decimal point while suppressing all leading zeros. The edited data shown is a minus zero value.

48. Asterisk protection and zero suppression occur up to the decimal point. The decimal point is retained regardless of the number of leading zeros. A minus sign appears in the edited data if the field is negative.

49. A constant (in this case, a comma) follows the dollar sign in the edited data if the floating dollar sign and the zero suppression 0 immediately precede a constant. This occurs if there is a number of leading zeros. The comma following a dollar sign looks awkward; however, a decimal point following the dollar sign does not (see example 43).

50. A space can be inserted between a fixed dollar sign and the first data digit when all digits in the field are significant. An ampersand in an edit word appears as a space in the edited data field.

51. In this quantity field, all leading zeros, including the units position, are suppressed (compare with example 52).

52. A single zero is shown in the edited data field when the data field contains only zeros.

53 through 57. Constants in the edit word are handled the same as punctuation marks. That is, only constants to the right of the first significant digit or the zero suppression 0 appear in the edited data. Examples 55 and 56 show how more edit word constants, other than the CR or minus, can appear as blanks in a positive field. Examples 55 through 57 also show the effect that the position of the zero suppression 0 has on constants. In example 56, an ampersand placed after the first constant provides a space following that constant in the edited data.

58. A hyphen (-) is used within the edit word to edit a social security number. In this example, the initial zero is suppressed. However, to include the initial zero in the edited data, leave an extra position in the edit word. See the note under *Editing Considerations* for a discussion of this exception.

59. Several constants can be used in an edit word.

60 and 61. This example shows the effect that the position of the zero suppression 0 has on the decimal point (or any other constants) and following zeros.

62. This example shows that a dollar sign separated from the zero suppression 0, even if only by a comma, is not a floating dollar sign, but a constant.

63 through 66. Any zero or asterisk to the right of the high-order zero or asterisk is a constant, not a zero suppression 0 or asterisk-protection symbol. Examples 65 and 66 also show that asterisk protection replaces not only blanks but also other constants to the left of the first significant digit.

67 through 69. These are three examples of editing a date field. Since month numbers have at most one leading zero, it is not necessary to specify a zero suppression 0. Example 68 shows the use of an ampersand to retain a blank space in the edited data.

70. This example shows what happens to the decimal point when no zero suppression 0 is specified for a field that has fewer than three significant digits. This applies if the field is more than three digits long.

71. This example shows how to retain the decimal point in a data field that has fewer than three significant digits. This applies if the field is more than three digits long.

COLUMNS 71-74

Columns 71 through 74 are not used. Leave them blank.

COLUMNS 75-80 (PROGRAM IDENTIFICATION)

See *Common Entries* in Chapter 1.

Chapter 10. Operation Codes

The RPG II language allows you to perform many different types of operations on your data. Operation codes, which are entered on the calculation specifications, indicate the operation to be performed. Usually these codes are abbreviations of the name of the operation.

Many operation codes can be placed into categories. The first part of this chapter includes general information about these categories. The latter part of the chapter describes each operation code in alphabetical order and shows one or more examples for most of the operations. Figure 10-1 is a summary of the specifications for each operation code.

	Control Level Indicators	Conditioning Indicators				Resulting Indicators		
	Columns					Columns		
Operation Code	7-8	9-17	Factor 1	Factor 2	Result Field	54-55	56-57	58-59
ACQ	O	O	R	R			O	
ADD	O	O	O	R	R	O	O	O
BEGSR	SR or blank		R					
BITOF	O	O		R	R			
BITON	O	O		R	R			
CHAIN	O	O	R	R		O		
COMP	O	O	R	R		O[3]	O[3]	O[3]
DEBUG	O	O	O	R	O			
DIV	O	O	O	R	R	O	O	O
ENDSR	SR or blank		O	O				
EXCPT	O	O						
EXIT	O	O		R				
EXSR	O	O		R				
FORCE		O		R				
GOTO	O	O		R				
KEYnn[1]	O	O	O		R	O	O	O
LOKUP (Array)	O	O	R	R		O[4]	O[4]	O[4]
LOKUP (Table)	O	O	R	R	O	O[4]	O[4]	O[4]
MHHZO	O	O		R	R			
MHLZO	O	O		R	R			
MLHZO	O	O		R	R			
MLLZO	O	O		R	R			

Figure 10-1 (Part 1 of 2). Summary of Operation Code Specifications

	Control Level Indicators	Conditioning Indicators				Resulting Indicators		
	Columns					Columns		
Operation Code	7-8	9-17	Factor 1	Factor 2	Result Field	54-55	56-57	58-59
MOVE	O	O		R	R			
MOVEA	O	O		R	R			
MOVEL	O	O		R	R			
MULT	O	O	O	R	R	O	O	O
MVR	O	O			R	O	O	O
NEXT	O	O	R	R			O	
POST	O	O	R		R		O	
READ	O	O		R			O[2]	O
REL	O	O	R	R			O	
RLABL					R			
SETnn[4]	O	O	O	O		O	O	O
SETOF	O	O				O[3]	O[3]	O[3]
SETON	O	O				O[3]	O[3]	O[3]
SETLL	O	O	R	R				
SHTDN	O	O				R		
SORTA	O	O		R				
SQRT	O	O		R	R			
SUB	O	O	O	R	R	O	O	O
TAG	O		R					
TESTB	O	O		R	R	O[3]	O[3]	O[3]
TESTZ	O	O			R	O[3]	O[3]	O[3]
TIME	O	O			R			
XFOOT	O	O		R	R	O	O	O
Z-ADD	O	O		R	R	O	O	O
Z-SUB	O	O		R	R	O	O	O

[1] The nn entries in columns 31 and 32 are for message indicator numbers. If the result field of a SET operation contains the keyword ERASE, factor 2 must contain the name of the CONSOLE file. Otherwise, factor 2 and the result field must be blank.

[2] Columns 56 and 57 can contain an indicator when the READ operation is used with a WORKSTN device.

[3] At least one resulting indicator must be specified in columns 54 through 59.

[4] At least one resulting indicator must be specified in columns 54 through 59, but no more than two can be used.

Fields without entries must be blank.

O	=	Optional
R	=	Required
SR	=	The only allowable nonblank characters in columns 7 and 8 for the BEGSR and ENDSR operation codes.

Figure 10-1 (Part 2 of 2). Summary of Operation Code Specifications

ARITHMETIC OPERATIONS

Arithmetic operations can be performed only on numeric fields or literals. The result field must also be numeric. Decimal alignment is performed for all arithmetic operations. Even though truncation can occur, the position of the decimal point in the result field is not affected. For arithmetic operations in which all three fields are used:

- Factor 1, factor 2, and the result field can be three different fields.
- Factor 1, factor 2, and the result field can all be the same field.
- Factor 1 and factor 2 can be the same field but different from the result field.
- Either factor 1 or factor 2 can be the same as the result field.

The length of any field specified in an arithmetic operation cannot exceed 15 characters. If the result exceeds 15 characters, characters are dropped from either or both ends depending on the location of the decimal point. The results of all operations are signed (+ or -). Any data placed in the result field replaces the data that was there previously.

MOVE OPERATIONS

Move operations (MOVE and MOVEL) transfer all or part of factor 2 to the result field. Factor 2 remains unchanged. Factor 1 must be blank, and no resulting indicators can be specified in columns 54 through 59.

The move operations can be used to change numeric fields to alphameric fields and alphameric fields to numeric fields. To change a numeric field to an alphameric field, enter the name of the numeric field in factor 2 and specify an alphameric result field. To change an alphameric field to a numeric field, enter the name of the alphameric field in factor 2 and specify a numeric result field.

When an alphameric field is moved into a numeric result field, the digit portion of each character is converted to its corresponding numeric character and then moved to the result field. Blanks are transferred as zeros. For the MOVE operation, the zone portion of the rightmost alphameric character is converted to its corresponding sign and is moved to the rightmost position of the numeric field where it becomes the sign of the field. For the MOVEL operation, the zone portion of the rightmost character of factor 2 is converted and used as the sign of the result field whether or not the rightmost character is included in the move operation.

When move operations are specified to move data into numeric fields, the decimal positions specified for the factor 2 field are ignored. For example, if the data 1.00 is moved into a numeric field with one decimal position, the result is 10.0.

MOVE ZONE OPERATIONS

The move zone operations move only the zone portion of a character.

A minus (-) sign in a move zone operation does not yield a negative character in the result field because a minus sign is represented by a hex 60 internally and a D zone is required for a negative character. Characters J through R have D zones and can be used to obtain a negative value (J = hex D1, ..., R = hex D9).

Note: Whenever the word *high* is used in a move zone operation, the field involved must be alphameric; whenever *low* is used, the field involved can be either alphameric or numeric.

COMPARE AND TESTING OPERATIONS

The compare and testing operations test fields for certain conditions. Resulting indicators assigned in columns 54 through 59 turn on according to the results of the operation. No fields are changed by these operations.

BIT OPERATIONS

The three operation codes BITON, BITOF, and TESTB set and test individual bits. Use the individual bits as switches in a program in order to save storage for binary type switches.

When you use the BITON, BITOF, and TESTB operations, any field named in factor 2 or result field must be a one-position alphameric field. A field is considered alphameric if there are no entries in the decimal positions column of the input or calculation specifications. The field specified as factor 2 or as the result field can be an array element if each element in the array is a one-position alphameric element.

SETON AND SETOF OPERATIONS

The operation codes SETON and SETOF turn indicators on or off. Any indicator to be turned on or off is specified in columns 54 through 59. The headings for these columns (plus or high, minus or low, zero or equal) have no meaning in these operations. When setting indicators, remember:

- The following indicators cannot be turned on by the SETON operation: 1P, MR, L0, KA through KN, KP through KY.
- The following indicators cannot be turned off by the SETOF operation: 1P, MR, L0, and LR.
- If the LR indicator is turned on by a SETON operation that is conditioned by a control level indicator (columns 7 and 8 of the calculation specifications), processing stops after all total output operations are finished. If it is turned on by a SETON operation at detail time (not conditioned by a control level indicator in columns 7 and 8), processing stops after the next total output operation is completed.
- If the halt indicators (H1 through H9) are set on and not turned off before the detail output operations are complete, the system stops. The operator can continue processing by responding to the halt for every halt indicator that is on.
- Setting L1 through L9 on or off does not automatically set any lower control level indicators.
- Indicators L1 through L9 and the record identifying indicators are always turned off after the next detail output operations are completed regardless of the previous SETON or SETOF operation.
- Whenever a new record is read, record identifying indicators (01 through 99) and field indicators are set to reflect conditions on the new record. The setting from any previous SETON or SETOF operation does not apply then.
- If a numeric indicator (01 through 99) is set on and not changed in other calculations, it remains on until it is set off by another calculation specification.

BRANCHING WITHIN RPG II

Operations are normally performed in the order in which they appear on the calculation specifications. There may be times, however, when the operations should be performed in a different order, such as:

- Several operations should be skipped when certain conditions occur.
- Certain operations should be performed for several, but not all, record types.
- Several operations should be repeated.

SUBROUTINE OPERATIONS

The operation codes BEGSR, ENDSR, and EXSR are used only for subroutines. In an RPG II program, a subroutine is a group of calculation specification statements that can be executed several times in one program cycle. A subroutine must be written on the calculation specifications after all other calculation operations for a program. Subroutine specification lines must be identified by SR or blanks in columns 7 and 8; therefore, individual operations within a subroutine cannot be conditioned by control level indicators in columns 7 and 8. Within a subroutine, SR or blanks in columns 7 and 8 can be intermixed.

LINKING TO EXTERNAL SUBROUTINES

To provide linkage from RPG II to an assembler language subroutine, use the EXIT and RLABL operations. Control cannot be transferred from one user assembler subroutine to another user assembler subroutine. The EXIT and RLABL operation codes also provide linkage to the IBM-supplied message retrieve subroutine (SUBR23).

Note: User-written subroutines should be placed in #LIBRARY or #RPGLIB (the library the RPG compiler resides in), not in the same library as the RPG II source program.

WORKSTN OPERATIONS

The operation codes ACQ and REL are used only with the WORKSTN file.

For these operations, the operation code must be entered in columns 28 through 32. Factor 1 specifies either the name of a 2-character field that contains the device identification or a 2-character alphameric literal that is the device identification. Factor 2 specifies the name of the WORKSTN file for which the operation is requested. Columns 56 and 57 can contain a resulting indicator, which is set on if an exception/error occurs.

Note: For WORKSTN files, a device can be either a display station or an SSP-ICF session.

PROGRAMMED CONTROL OF INPUT AND OUTPUT

The normal program cycle can be altered to allow input and output operations during calculations. (See *RPG II Program Cycle* in Chapter 1 for a brief description of the program cycle.) The following operations provide this capability:

- EXCPT (Exception Output)
- READ (Read)
- FORCE (Force)
- NEXT (Next)
- CHAIN (Chain)
- KEY (Key)
- SET (Set)
- SETLL (Set Lower Limits)

Alphabetized Operation Codes

The following section of this chapter discusses, in alphabetical order, individual operation codes.

ACQ (ACQUIRE)

Indicators				Result	Resulting Indicators		
7-8	9-17	Factor 1	Factor 2	Field	54-55	56-57	58-59
Opt	Opt	Required	Required	Blank	Blank	Optional	Blank

The ACQ operation acquires the device specified in factor 1 for the program. If the device is available, ACQ attaches it to the program. If it is not available or is already attached to the program, an error occurs. If an indicator is specified in columns 56 and 57, the indicator is set on. If no indicator is specified, but the INFSR subroutine is specified, the INFSR automatically receives control when an exception/error occurs. If no indicator or INFSR subroutine is specified, the program halts when an exception/error occurs.

No input or output operation occurs when the ACQ operation is executed.

ADD (ADD)

Indicators				Result	Resulting Indicators		
7-8	9-17	Factor 1	Factor 2	Field	54-55	56-57	58-59
Opt	Opt	Optional	Required	Required	Optional	Optional	Optional

Factor 2 is added to factor 1. The sum is placed in the result field. Factor 1 and factor 2 are not changed by the operation. If factor 1 is not present, factor 2 is added to the result field, and the sum is placed in the result field.

BEGSR (BEGIN SUBROUTINE)

Indicators				Result	Resulting Indicators		
7-8	9-17	Factor 1	Factor 2	Field	54-55	56-57	58-59
Opt: SR	Blank	Required	Blank	Blank	Blank	Blank	Blank

The BEGSR operation serves as the beginning point of the subroutine. Factor 1 must contain the name of the subroutine. The control level entry (columns 7 and 8) can be SR or remain blank. Columns 9 through 17 must not contain any conditioning indicators. The subroutine name can be from 1 to 6 characters long and must begin in column 18 with an alphabetic character. The remaining characters can be any combination of alphabetic or numeric characters. However, special characters are not allowed and blanks cannot appear between characters in the name.

Every subroutine must have a unique name. This name cannot be used as the label of a TAG or ENDSR operation.

BITOF (SET BITS OFF)

Indicators				Result	Resulting Indicators		
7-8	9-17	Factor 1	Factor 2	Field	54-55	56-57	58-59
Opt	Opt	Blank	Required	Required	Blank	Blank	Blank

The BITOF operation causes bits identified in factor 2 to be set off (be set to 0) in the field named as the result field. Factor 2 is always a source of bits for the result field. The result field is the field in which the bits are set off.

Factor 2 can contain:

- *Bit numbers 0-7:* From 1 to 8 bits can be set off per operation. The bits to be set off are identified by the numbers 0 through 7 (0 is the leftmost bit). The bit numbers must be enclosed in apostrophes and the entry must begin in column 33. For example, to set off bits 0, 2, and 5, enter '025' in factor 2.

- *Field name:* The name of a one-position alphameric field, table element, or array element can be specified in factor 2. In this case, the bits that are on in the field, table element, or array element are set off in the result field; bits that are off are not affected.

See Figure 10-2 for a summary of BITOF operations.

The operation code BITOF must appear in columns 28 through 32. Conditioning indicators can be used in columns 7 through 17. However, factor 1, decimal positions, half-adjust, and the resulting indicator columns must be blank.

BITON (SET BITS ON)

Indicators				Result	Resulting Indicators		
7-8	9-17	Factor 1	Factor 2	Field	54-55	56-57	58-59
Opt	Opt	Blank	Required	Required	Blank	Blank	Blank

The BITON operation causes bits identified in factor 2 to be set on (be set to 1) in the field named as the result field. Factor 2 is always a source of bits for the result field. The result field is the field in which the bits are set on.

Factor 2 can contain:

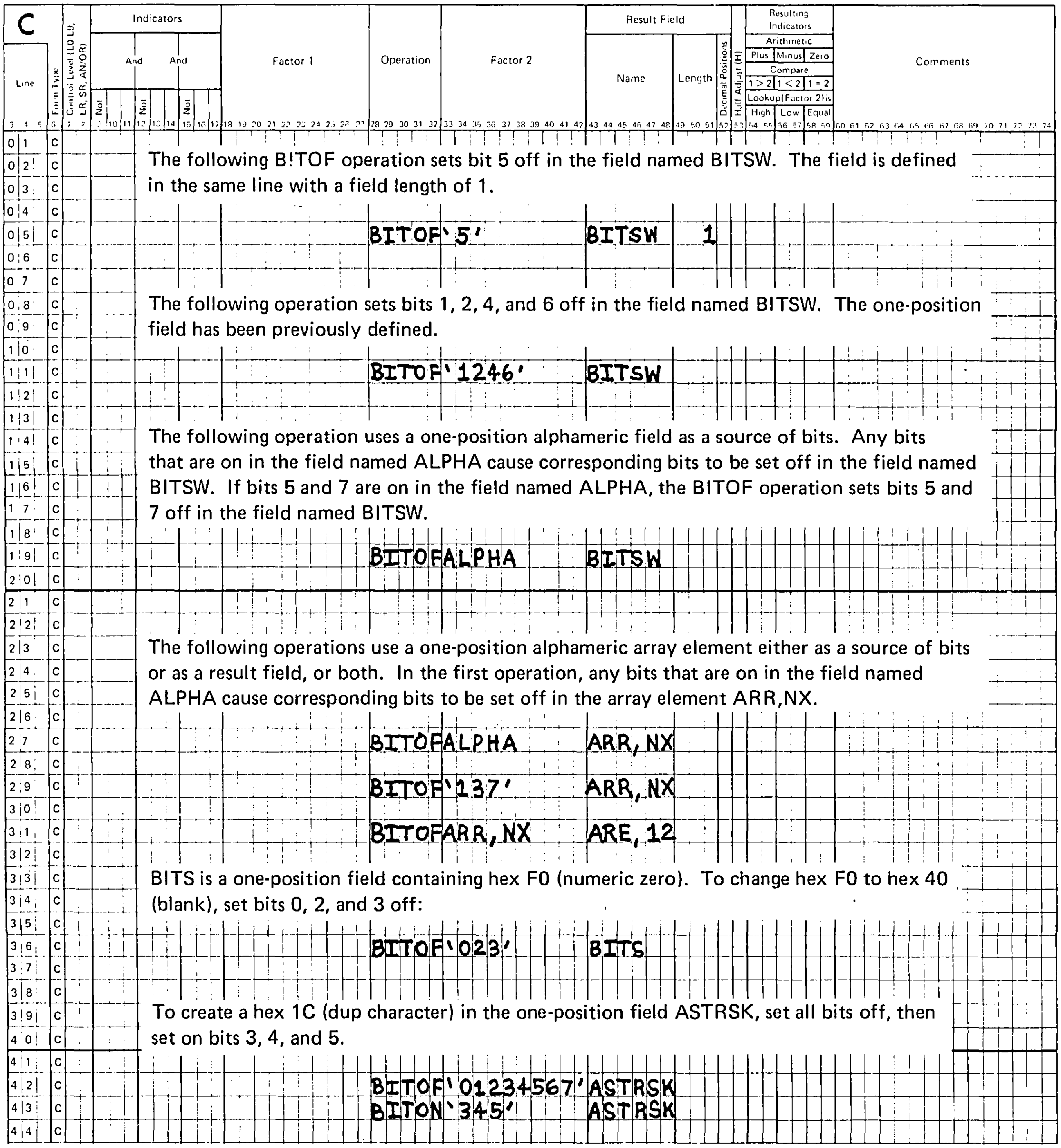
- *Bit numbers 0-7:* From 1 to 8 bits can be set on per operation. The bits to be set on are identified by the numbers 0 through 7 (0 is the leftmost bit). The bit numbers must be enclosed in apostrophes and the entry must begin in column 33. For example, to set on bits 0, 2, and 5, enter '025' in factor 2.

- *Field name:* The name of a one-position alphameric field, table element, or array element can be specified in factor 2. In this case, the bits that are on in the field, table element, or array element are set on in the result field; bits that are off are not affected.

See Figure 10-3 for a summary of BITON operations.

The operation code BITON must appear in columns 28 through 32. Conditioning indicators can be used in columns 7 through 17. However, factor 1, decimal positions, half-adjust, and the resulting indicator columns must be blank.

Calculation Specifications



The following BITOF operation sets bit 5 off in the field named BITSW. The field is defined in the same line with a field length of 1.

Line	Form Type	Factor 1	Operation	Factor 2	Result Field Name	Length
05	C		BITOF	'5'	BITSW	1

The following operation sets bits 1, 2, 4, and 6 off in the field named BITSW. The one-position field has been previously defined.

Line	Form Type	Factor 1	Operation	Factor 2	Result Field Name	Length
11	C		BITOF	'1246'	BITSW	

The following operation uses a one-position alphameric field as a source of bits. Any bits that are on in the field named ALPHA cause corresponding bits to be set off in the field named BITSW. If bits 5 and 7 are on in the field named ALPHA, the BITOF operation sets bits 5 and 7 off in the field named BITSW.

Line	Form Type	Factor 1	Operation	Factor 2	Result Field Name	Length
19	C		BITOF	ALPHA	BITSW	

The following operations use a one-position alphameric array element either as a source of bits or as a result field, or both. In the first operation, any bits that are on in the field named ALPHA cause corresponding bits to be set off in the array element ARR,NX.

Line	Form Type	Factor 1	Operation	Factor 2	Result Field Name	Length
27	C		BITOF	ALPHA	ARR,NX	
29	C		BITOF	'137'	ARR,NX	
31	C		BITOF	ARR,NX	ARE,12	

BITS is a one-position field containing hex F0 (numeric zero). To change hex F0 to hex 40 (blank), set bits 0, 2, and 3 off:

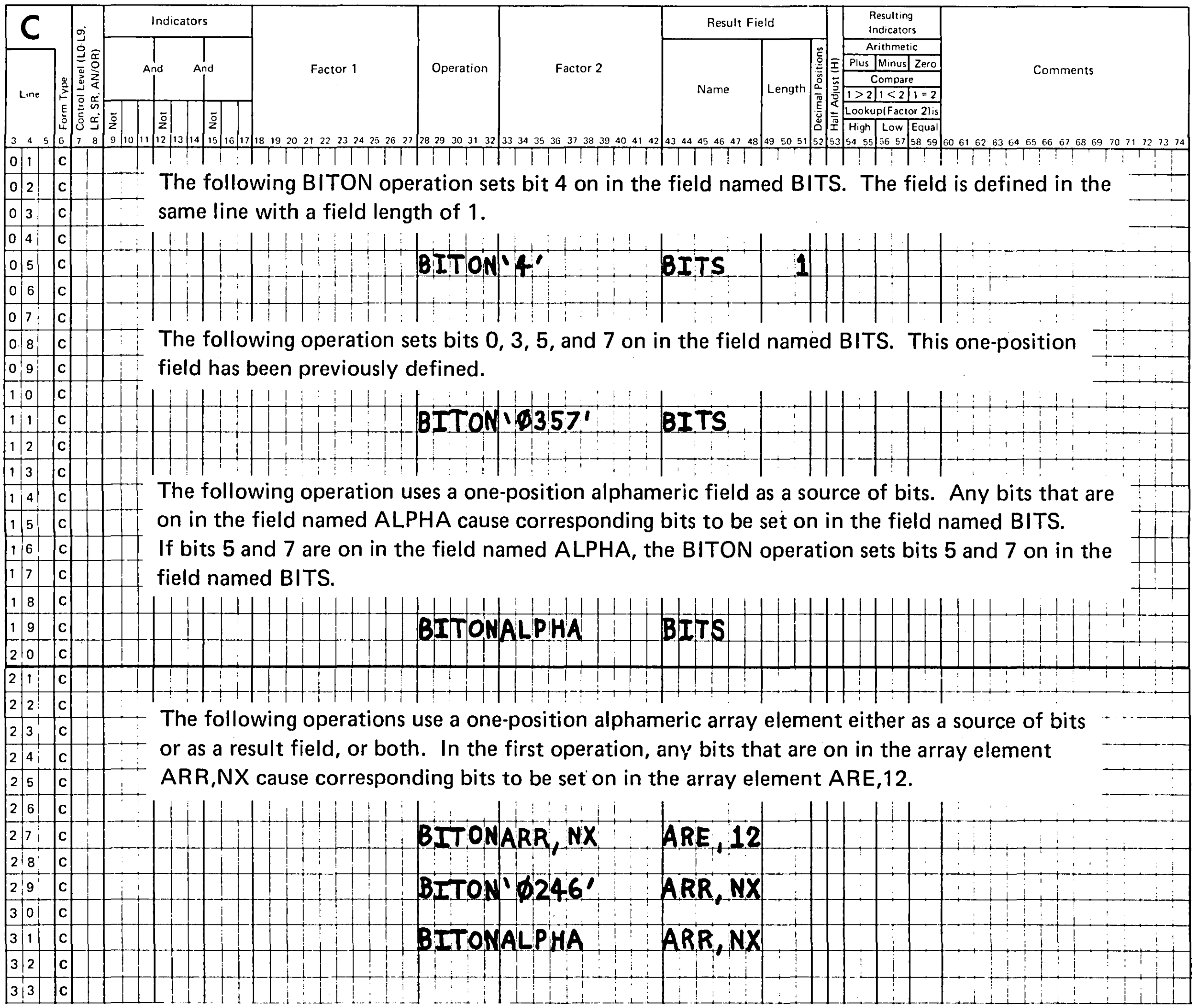
Line	Form Type	Factor 1	Operation	Factor 2	Result Field Name	Length
36	C		BITOF	'023'	BITS	

To create a hex 1C (dup character) in the one-position field ASTRSK, set all bits off, then set on bits 3, 4, and 5.

Line	Form Type	Factor 1	Operation	Factor 2	Result Field Name	Length
42	C		BITOF	'01234567'	ASTRSK	
43	C		BITON	'345'	ASTRSK	

Figure 10-2. Summary of BITOF Operations

Calculation Specifications



The following BITON operation sets bit 4 on in the field named BITS. The field is defined in the same line with a field length of 1.

```
05 C                     BITON'4'       BITS    1
```

The following operation sets bits 0, 3, 5, and 7 on in the field named BITS. This one-position field has been previously defined.

```
11 C                     BITON'Ø357'    BITS
```

The following operation uses a one-position alphameric field as a source of bits. Any bits that are on in the field named ALPHA cause corresponding bits to be set on in the field named BITS. If bits 5 and 7 are on in the field named ALPHA, the BITON operation sets bits 5 and 7 on in the field named BITS.

```
19 C                     BITONALPHA     BITS
```

The following operations use a one-position alphameric array element either as a source of bits or as a result field, or both. In the first operation, any bits that are on in the array element ARR,NX cause corresponding bits to be set on in the array element ARE,12.

```
27 C                     BITONARR,NX    ARE,12
29 C                     BITON'Ø246'    ARR,NX
31 C                     BITONALPHA     ARR,NX
```

Figure 10-3. Summary of BITON Operations

CHAIN (CHAIN)

Indicators				Result	Resulting Indicators		
7-8	9-17	Factor 1	Factor 2	Field	54-55	56-57	58-59
Opt	Opt	Required	Required	Blank	Optional	Blank	Blank

The CHAIN operation causes one record to be read from a disk file during calculations. The CHAIN operation can be used either to read records randomly from an indexed, sequential, or direct file or to load a nondelete-capable direct file (for more information on nondelete-capable direct file loads, see *Direct Files* in Chapter 3).

Enter the operation code CHAIN in columns 28 through 32. Factor 1 defines the relative record number or the key of the record to be selected for processing, and factor 2 names the chained file from which the record is read. This file must be defined by a C entry in column 16 of the file description specifications.

Indicators can be used in columns 7 through 17, but columns 43 through 53 and 56 through 59 must be blank. If the chained file is conditioned by an external indicator on the file description specifications, the CHAIN statement should be conditioned by the same external indicator.

A maximum of 15 chained and/or demand files is allowed per program.

Columns 54 and 55 should specify an indicator. If the record is not found (or, for a direct file load, if the record location does not exist in the file), the indicator turns on. No update is permitted to a chained update file when the specified record is not found; however, adding records to a file is allowed. Duplicate records in the file are possible after an unsuccessful chain to an update-add file if the key field is modified prior to an add to the file.

If an indicator is not specified in columns 54 and 55 and the record is not found, the program halts and an operator action is required. When chaining to a file with packed record keys, the field specified in factor 1 of the CHAIN operation must have a packed length that is the same as the length of the key field in the chained file. Packed key fields can be up to 8 bytes long. The packed field equivalents for zoned decimal fields up to 15 bytes long are shown in a chart under *Packed Decimal Format* in Chapter 7, *Input Specifications.*

Note: If you chain to one or more files during the same RPG II cycle, record identifying indicators assigned to the chained file or files remain on throughout the cycle if the previous chain operations were executed successfully. If you chain to the same file more than once during an RPG II cycle, only the last record processed is updated during output time unless an exception output is associated with each chain operation.

Random Processing

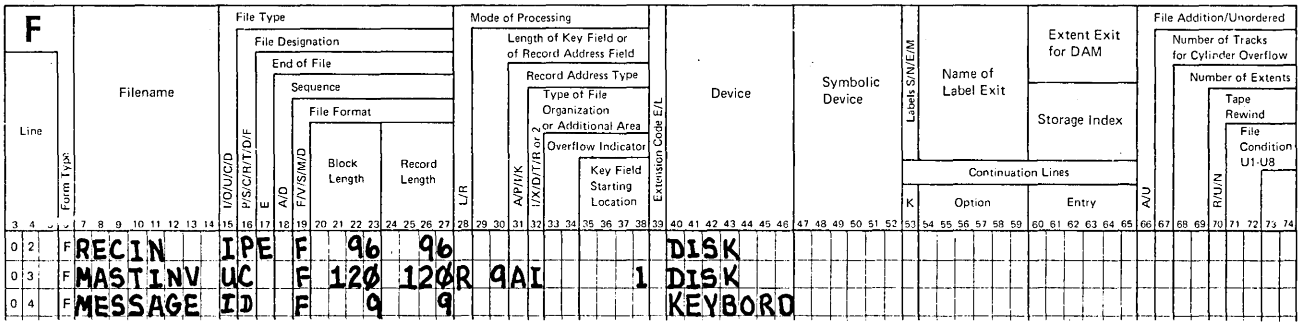
To read a record from a sequential or direct file with the CHAIN operation, the record must be identified by relative record number. To read a record from an indexed file with the CHAIN operation, the record must be identified by a record key. A field can be specified to contain the relative record number or record key.

If the record has been deleted from the file, the no-record-found indicator is turned on. If the no-record found indicator is not specified, a message is displayed.

Factor 1 must contain a relative record number or record key, or the name of a field that contains a relative record number or record key. Factor 2 must contain the name of the file from which the record is read.

Figure 10-4 shows an example of chaining to and updating an indexed file.

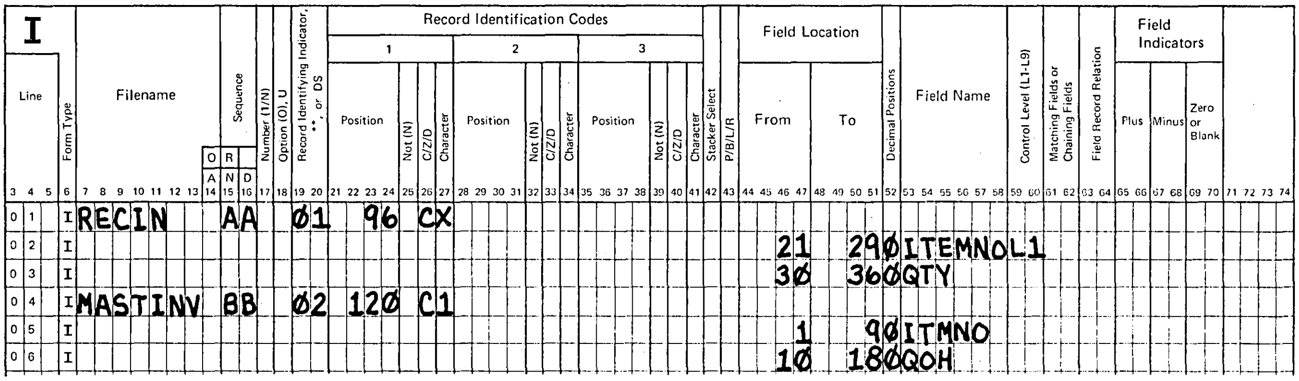
File Description Specifications



```
  02 FRECIN   IPE F  96  96            DISK
  03 FMASTINV UC  F 120 120R 9AI     1 DISK
  04 FMESSAGE ID  F   9   9            KEYBORD
```

RECIN file consists of records sorted by item number, with each record containing a quantity ordered.

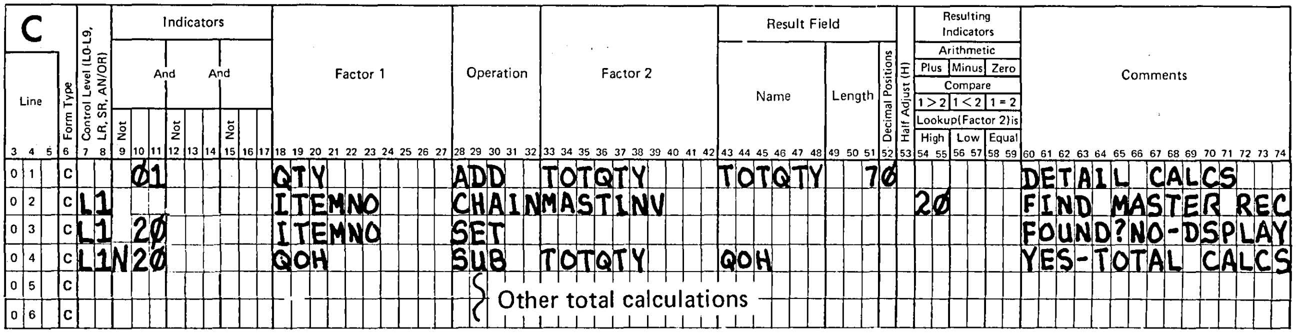
Input Specifications



```
  01 IRECIN   AA  01  96 CX
  02 I                                        21  29ØITEMNOL1
  03 I                                        3Ø  36ØQTY
  04 IMASTINV BB  Ø2 12Ø C1
  05 I                                         1   9ØITMNO
  06 I                                        1Ø  18ØQOH
```

ITEMNO is used as a control field. When all the quantities for one item number are added, a control break occurs. The CHAIN operation then uses ITEMNO to find the master record and update it. If it is not found, a SET operation displays the item number on the screen. Indicator 20 turns on if the record is not found.

Calculation Specifications

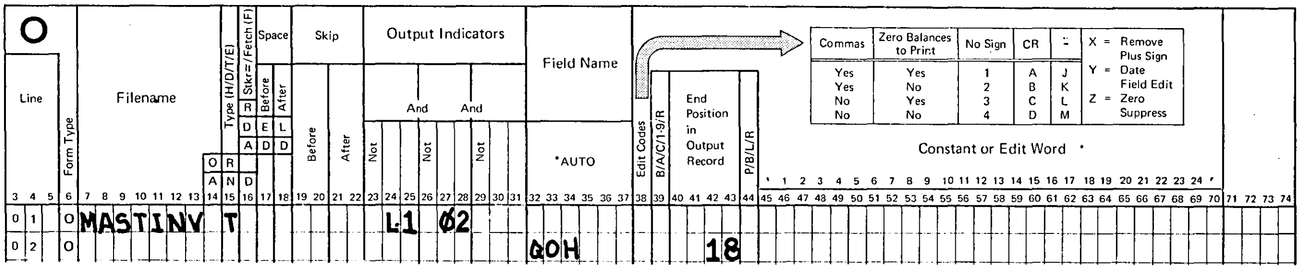


```
  01 C     Ø1      QTY       ADD  TOTQTY    TOTQTY  7Ø       DETAIL CALCS
  02 CL1           ITEMNO    CHAINMASTINV                2Ø    FIND MASTER REC
  03 CL1 2Ø        ITEMNO    SET                                FOUND?NO-DSPLAY
  04 CL1N2Ø        QOH       SUB  TOTQTY    QOH                 YES-TOTAL CALCS
  05 C
                           { Other total calculations
  06 C
```

If the master record is found, the total quantity for the item number is subtracted from the quantity on hand.

After the total calculations, the QOH field in the master record is updated.

Output Specifications



```
  01 OMASTINV T        L1 Ø2
  02 O                         QOH       18
```

Figure 10-4. CHAIN Operation

COMP (COMPARE)

Indicators				Result	Resulting Indicators		
7-8	9-17	Factor 1	Factor 2	Field	54-55	56-57	58-59
Opt	Opt	Required	Required	Blank	1 required		

The COMP operation compares factor 1 with factor 2. As a result of the compare, indicators turn on as follows:

High   Factor 1 is greater than factor 2.

Low   Factor 1 is less than factor 2.

Equal   Factor 1 equals factor 2.

Factor 1 and factor 2 must be both alphameric or both numeric.

At least one resulting indicator must be specified in positions 54 through 59.

The fields are automatically aligned before they are compared. If the fields are alphameric, they are aligned to their leftmost character. If one is shorter, the unused positions are filled with blanks (see Figure 10-5). The maximum field length for alphameric fields to be compared is 256 characters.

If the fields to be compared are numeric, they are aligned according to the decimal point. Any missing digits are filled with zeros (see Figure 10-6). The maximum field length for numeric fields to be compared is 15 digits.

If an alternate collating sequence is specified, alphameric fields are compared according to the alternate sequence.

Figure 10-7 shows some examples of specifications for compare operations.

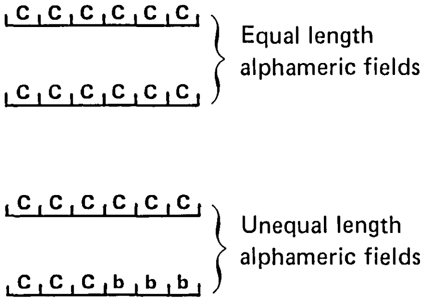

C C C C C C
C C C C C C
Equal length alphameric fields
C C C C C C
C C C b b b
Unequal length alphameric fields

Figure 10-5. Comparison of Alphameric Fields

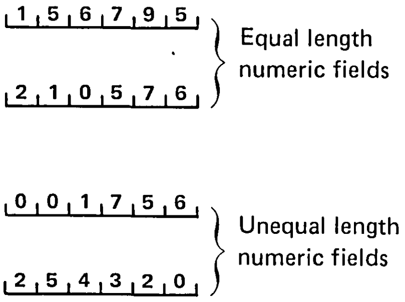
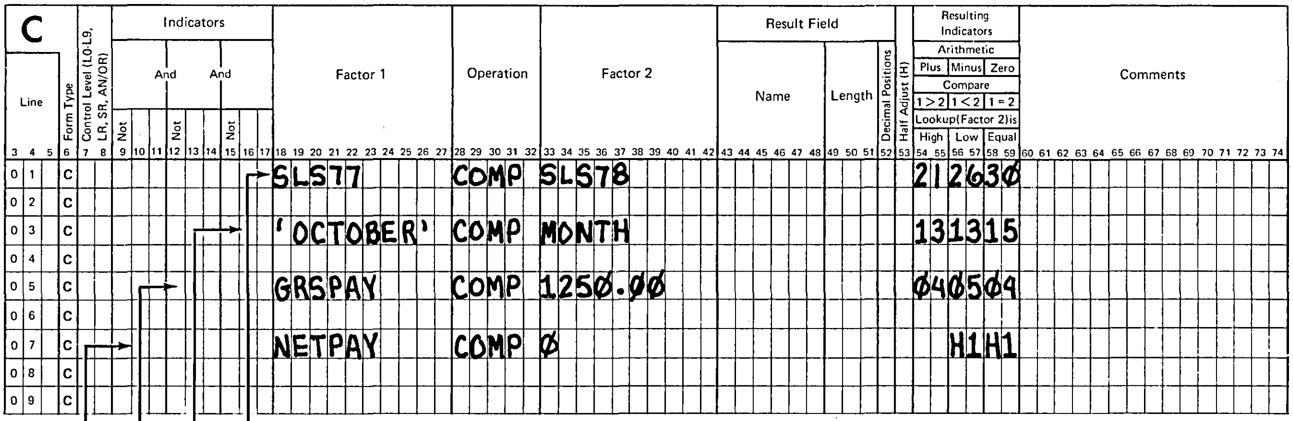


Figure 10-6. Comparison of Numeric Fields

Calculation Specifications



C							Result Field				Resulting Indicators			
											Arithmetic Plus	Minus	Zero	
											Compare 1>2	1<2	1=2	
											Lookup(Factor 2)is High	Low	Equal	
Line	Form Type	Control Level (LO-L9, LR, SR, AN/OR)	Indicators Not And Not And Not	Factor 1	Operation	Factor 2	Name	Length	Decimal Positions	Half Adjust (H)	54 55	56 57	58 59	Comments
3 4 5	6	7 8	9 10 11 12 13 14 15 16 17	18 19 20 21 22 23 24 25 26 27	28 29 30 31 32	33 34 35 36 37 38 39 40 41 42	43 44 45 46 47 48	49 50 51	52	53				60 61 62 63 64 65 66 67 68 69 70 71 72 73 74
0 1	C			SLS77	COMP	SLS78					21	26	3Ø	
0 2	C													
0 3	C			'OCTOBER'	COMP	MONTH					13	13	15	
0 4	C													
0 5	C			GRSPAY	COMP	1250.ØØ					Ø4	Ø5	Ø4	
0 6	C													
0 7	C			NETPAY	COMP	Ø						H1	H1	
0 8	C													
0 9	C													

The contents of the field SLS77 (1977 sales) are compared with the contents of SLS78. If 1977 sales exceed 1978 sales, resulting indicator 21 turns on; if they are less, indicator 26 turns on; if the two years had equal sales, indicator 30 turns on.

The alphameric constant OCTOBER is compared with the contents of the field named MONTH, which must also be defined as alphameric. If the MONTH field does not contain the word OCTOBER, indicator 13 turns on; if it does, indicator 15 turns on.

The contents of the field named GRSPAY, which must be defined as numeric, are decimal-aligned with numeric constant 1250.00. If the value in field GRSPAY is greater than or equal to 1250.00, indicator 04 turns on; if its value is less than 1250.00, indicator 05 turns on.

The contents of the field NETPAY, which must be defined as numeric, are decimal-aligned with numeric constant 0 and then compared to it. If NETPAY is greater than zero, indicator H1 remains off; however if NETPAY is zero or negative, indicator H1 turns on.

Figure 10-7. Compare Operations

DEBUG (DEBUG)

Indicators				Result	Resulting Indicators		
7-8	9-17	Factor 1	Factor 2	Field	54-55	56-57	58-59
Opt	Opt	Optional	Required	Optional	Blank	Blank	Blank

The DEBUG operation is an RPG II function that helps you find errors in a program that is not working properly. Either one or two records containing information helpful for finding programming errors are written to an output file as a result of this operation. All DEBUG output in a program is written to the same file.

The DEBUG operation code can be specified at any point or at several points in the calculation specifications. Whenever the program encounters the DEBUG operation, either one or two records are written depending upon the specifications entered. The first record contains a list of all indicators that are on at the time the DEBUG operation was encountered. The second record, if specified, shows the contents of the field specified in the result field.

Factor 1 can contain a literal or the name of a field to help identify the particular DEBUG operation. The length of the specified field can be from 1 to 8 characters. The contents of the field or the literal are written in the first record. If factor 1 is not used, the RPG II generated statement number of the DEBUG operation code is written in the first record. Factor 2 must contain the name of an output file on which the lines are written and can be any valid output file . The same output filename must appear in factor 2 for all DEBUG statements in a program. The result field can contain the name of a field or array whose contents are to be written in the second record. Any valid indicator can be used in columns 7 through 17. Columns 49 through 59 must be blank.

The DEBUG operation can be used in the calculation specifications only if a 1 is entered in column 15 of the control specifications. If the control specifications entry was not made, the DEBUG operation code and its conditioning indicators are treated as a comment. See *Column 15 (DEBUG)* in Chapter 2, *Control Specifications*, for more information.

Records Written for DEBUG

For a DEBUG operation, the first record is always written and appears in the following format:

Output Positions	**Information**
1-8	DEBUG-
9-16	Literal or contents of field entered in factor 1 (optional) or the statement number of the DEBUG operation code in the program.
17	Blank
18-32	The words INDICATORS ON-
33-any position (depending on length of field)	The names of all indicators that are on, each separated by a blank. More than one record may be needed.

The second record is written only when an entry is made in the result field. The record is written in the following format:

Output Positions	**Information**
1-14	The words FIELD VALUE-
15-any position (depending on length of field)	The contents of the result field (up to 256 characters). If the result field is an array, more than one output record may be needed to contain the array.

DIV (DIVIDE)

Indicators		Factor 1	Factor 2	Result Field	Resulting Indicators		
7-8	9-17				54-55	56-57	58-59
Opt	Opt	Optional	Required	Required	Optional	Optional	Optional

Factor 1 (dividend) is divided by factor 2 (divisor). The quotient (result) is placed in the result field. Factor 1 and factor 2 are not changed. If factor 1 is 0, the result of the divide operation is 0. Factor 2 cannot be 0. If it is, the job stops immediately. The operator can continue processing, however, by responding to the halt. When processing is continued, the result and remainder are set to 0. If factor 1 is not present, the result field is divided by factor 2, and the quotient is placed in the result field.

Any remainder resulting from the divide operation is lost unless the move remainder (MVR) operation is specified as the next operation. If move remainder is the next operation, the result of the divide operation cannot be half-adjusted (rounded).

ENDSR (END SUBROUTINE)

Indicators		Factor 1	Factor 2	Result Field	Resulting Indicators		
7-8	9-17				54-55	56-57	58-59
Opt: SR	Blank	Optional	Optional	Blank	Blank	Blank	Blank

The ENDSR operation defines the end of the subroutine; therefore, it must be the last statement in the subroutine. Factor 1 can contain a name that can be used as a point to which a GOTO operation within the subroutine can branch. The control level entry (columns 7 and 8) can be SR or remain blank. Columns 9 through 17 must not contain any conditioning indicators.

The ENDSR operation ends the subroutine and automatically causes a branch back to the statement that follows the EXSR operation unless the subroutine is the INFSR (exception/error processing) subroutine. For the INFSR subroutine, an optional factor 2 entry on the ENDSR operation specifies the return point for the subroutine. The valid entries for factor 2 for the INFSR subroutine are described in Chapter 13 under *WORKSTN Exception/Error Handling*. For all other subroutines, factor 2 must not contain an entry.

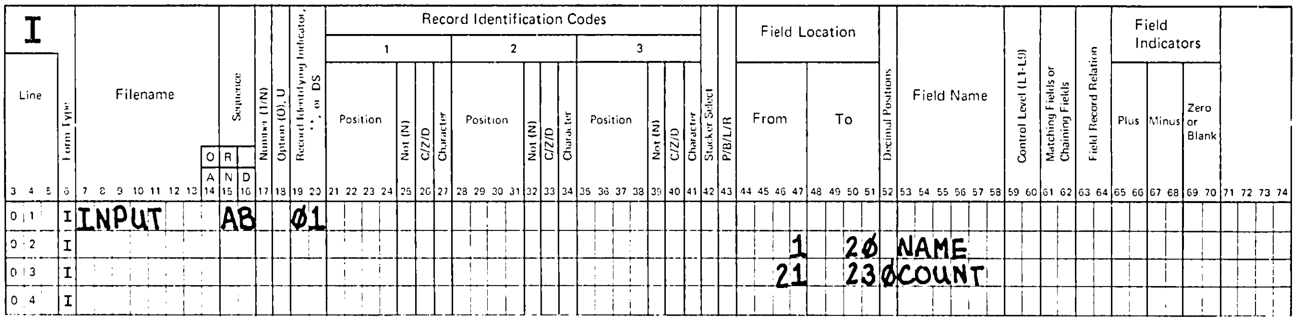
EXCPT (EXCEPTION OUTPUT)

Indicators		Factor 1	Factor 2	Result Field	Resulting Indicators		
7-8	9-17				54-55	56-57	58-59
Opt	Opt	Blank	Blank	Blank	Blank	Blank	Blank

The EXCPT operation allows records to be written during calculations. Use the EXCPT operation primarily to write a variable number of similar or identical records (either detail or total) in one program cycle. (Normally only the exact number of records specified by the output specifications are written in one program cycle.) For example, EXCPT can be used to produce a variable number of identical mailing labels, to write out contents of a table, or to produce a number of records having the same information.

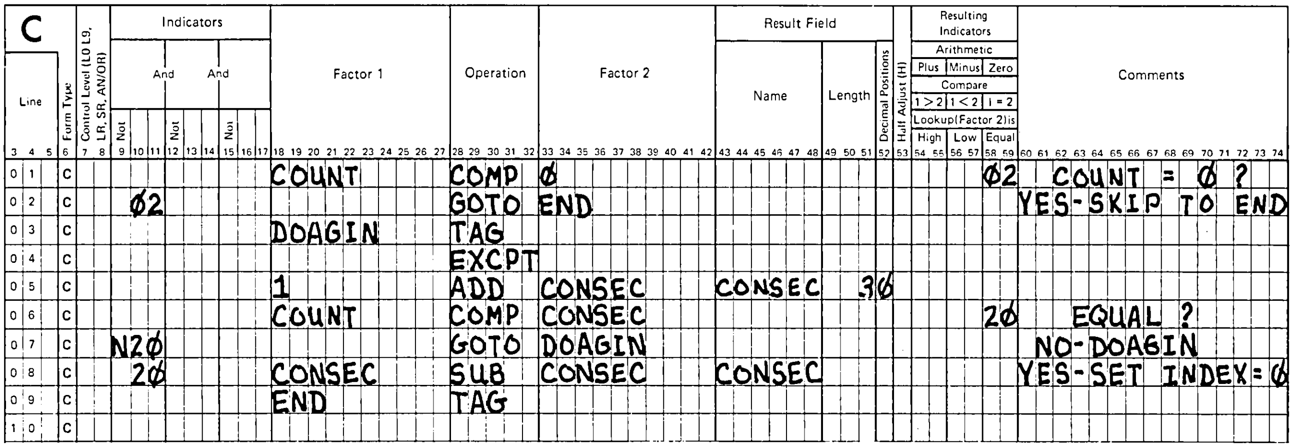
To use this operation, enter EXCPT in columns 28 through 32. Indicators can be used in columns 7 through 17; however, all other columns must be blank. The lines that are to be written during calculation time are indicated by an E in column 15 of the output specifications. Figure 10-8 shows the use of the EXCPT operation to produce a variable number of identical records on a printer file.

Input Specifications



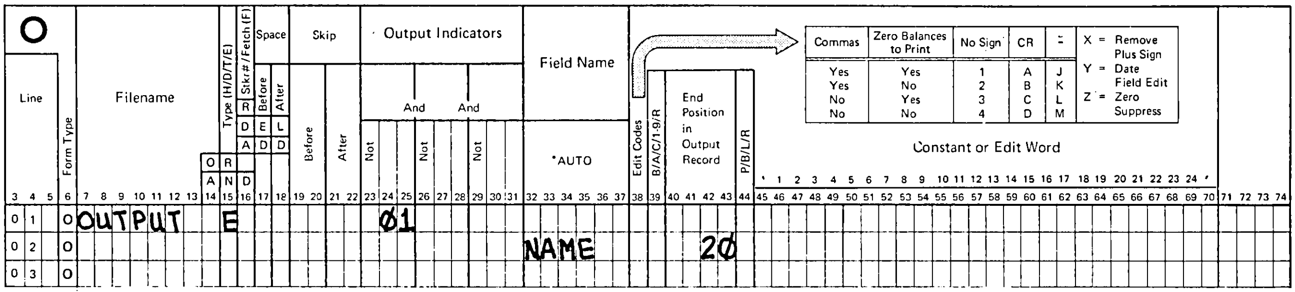
Line	Form Type	Filename	Sequence	Record Identifying Indicator	Field Location From	Field Location To	Decimal Positions	Field Name
01	I	INPUT	AB	Ø1				
02	I				1	2Ø		NAME
03	I				21	23	Ø	COUNT
04	I							

Calculation Specifications



Line	Form Type	Indicators	Factor 1	Operation	Factor 2	Result Field Name	Length	Decimal Positions	Resulting Indicators (Equal)	Comments
01	C		COUNT	COMP	Ø				Ø2	COUNT = Ø ?
02	C	Ø2		GOTO	END					YES-SKIP TO END
03	C		DOAGIN	TAG						
04	C			EXCPT						
05	C		1	ADD	CONSEC	CONSEC	3	Ø		
06	C		COUNT	COMP	CONSEC				2Ø	EQUAL ?
07	C	N2Ø		GOTO	DOAGIN					NO-DOAGIN
08	C	2Ø	CONSEC	SUB	CONSEC	CONSEC				YES-SET INDEX=Ø
09	C		END	TAG						
10	C									

Output Specifications



Line	Form Type	Filename	Type (H/D/T/E)	Output Indicators	Field Name	End Position in Output Record
01	O	OUTPUT	E	Ø1		
02	O				NAME	2Ø
03	O					

Records in the input file have two fields, NAME and COUNT. The NAME field is to be entered into a certain number of records. That number is indicated in the COUNT field.

The first calculation line compares COUNT to zero. If COUNT is equal to zero, indicator 02 turns on and the GOTO operation skips further calculations. If the COUNT field is one or more, the EXCPT operation is performed, and the exception record indicated by the E in column 15 of the output specifications sheet is printed.

The field CONSEC keeps track of the number of records printed. Each time an exception record is printed, one is added to CONSEC. CONSEC is then compared with COUNT, the field that tells how many records should be printed. If they are not equal (indicator 20 is not on), a branch is taken back to DOAGIN. Another record is printed. One is added to CONSEC, and CONSEC is compared to COUNT. If these fields are equal, another input record is read. If not, the same operations are done again. Whenever CONSEC equals COUNT, enough records are printed. CONSEC is then subtracted from itself, making it zero. This last operation is necessary so that an accurate count can be kept for the next record.

Figure 10-8. EXCPT Operation that Produces a Variable Number of Identical Records

EXIT (EXIT TO AN EXTERNAL SUBROUTINE)

Indicators				Result	Resulting Indicators		
7-8	9-17	Factor 1	Factor 2	Field	54-55	56-57	58-59
Opt	Opt	Blank	Required	Blank	Blank	Blank	Blank

The EXIT operation designates the point in the calculation specifications when control is to be transferred from RPG II to an assembler language subroutine.

The rules for use of the EXIT operation on the calculation specifications are as follows:

Columns	Entry
Operation (28-32)	EXIT
Factor 1 (18-27)	Blank
Factor 2 (33-42)	The name of the subroutine to which control is to be passed. The name must consist of 5 or 6 characters, the first 4 of which are SUBR. The remaining characters must be alphabetic for user-written subroutines. (Numeric characters are reserved for IBM-supplied subroutines.) The module name and entry point name must be the same.
Result field (43-48)	Blank
Resulting indicators (54-59)	Blank

The EXIT operation can be controlled by a control level indicator (columns 7 and 8) and conditioning indicators (columns 9 through 17). If no control level indicator is used, the EXIT operation occurs at detail calculation time.

The position of the EXIT operation in the calculation specifications of the RPG II program determines when the actual subroutine execution occurs (see Figure 10-9).

To specify linkage to a non-I/O subroutine for a SPECIAL file, use the EXIT operation. You must keep track of the EXIT that is taken because index register 2 does not point to the DTF on an EXIT operation.

Note: The maximum number of user-written assembler subroutines that can be used in a program is 2256.

Position	Execution of Subroutine
First detail line in calculation specifications	Immediately following data routine file, that is, after data is extracted from input record
Last detail line in calculation specifications	Immediately before heading records output time
First total line in calculation specifications	Immediately following input routine (after determination of record type and testing for control level break)
Last total line in calculation specifications	Immediately before total records output time
Any other detail/total line in calculation specifications	Immediately following the previous calculation operation

Figure 10-9. Relationship between Position of EXIT Operation and Execution of Subroutine

EXSR (EXECUTE SUBROUTINE)

Indicators				Result	Resulting Indicators		
7-8	9-17	Factor 1	Factor 2	Field	54-55	56-57	58-59
Opt	Opt	Blank	Required	Blank	Blank	Blank	Blank

The EXSR operation causes the subroutine named in factor 2 to be executed. The EXSR operation can appear anywhere in the program. Whenever it appears, the subroutine is executed. After operations in the subroutine are performed, the operation in the line following the EXSR operation is performed.

The EXSR operation can be conditioned by any indicators; thus, the subroutine is executed only when all conditions are satisfied. Any valid indicator can be used in columns 7 through 17. If no indicators are used, the subroutine is always executed.

Factor 2 must contain the name of the subroutine that is to be executed. This name must appear on a BEGSR operation.

Coding Subroutines

All RPG II operations can be performed within a subroutine, and these operations can be conditioned by any valid indicators in columns 9 through 17. Because SR or blanks must appear in columns 7 and 8, control level indicators cannot be used in these columns. However, AND/OR lines within the subroutine can be indicated in columns 7 and 8.

Fields used in a subroutine can be defined either in the subroutine or in the main program. In either instance, the fields can be used by both the main program and the subroutine.

Any number of subroutines can be included in a program; however, a subroutine cannot contain another subroutine. One subroutine can call another subroutine; that is, a subroutine can contain an EXSR operation code. However, a subroutine cannot call itself.

Subroutines do not have to be specified in the order they are used. Each subroutine must have a unique name and contain a BEGSR and ENDSR operation.

See Figure 10-10 for an example of coding a subroutine.

FORCE (FORCE)

Indicators				Result	Resulting Indicators		
7-8	9-17	Factor 1	Factor 2	Field	54-55	56-57	58-59
Blank	Opt	Blank	Required	Blank	Blank	Blank	Blank

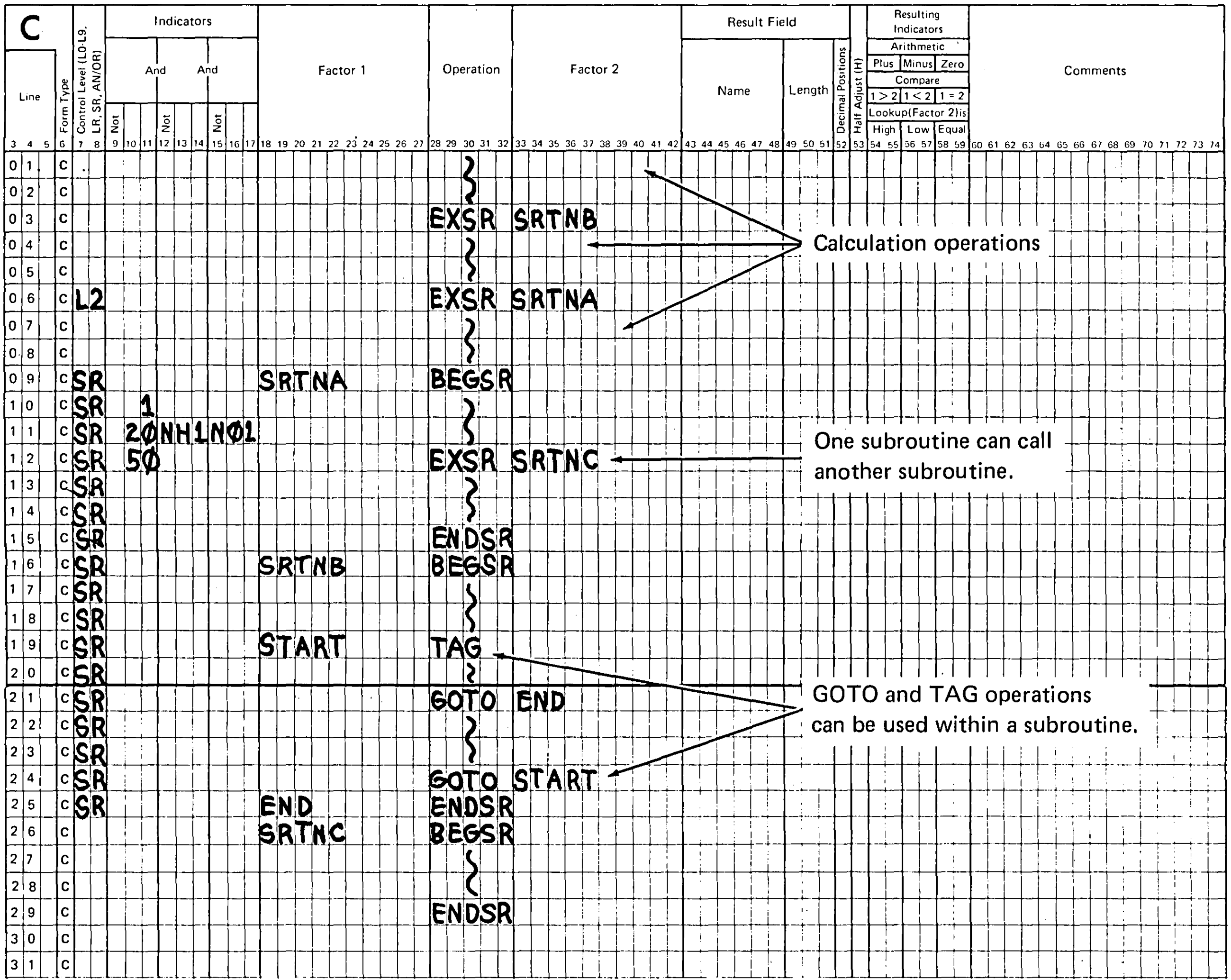
The FORCE operation allows selection of the file from which the next record is to be read. The FORCE operation can be used for primary or secondary input and update files; however, it cannot be used to read from files assigned to a KEYBORD or WORKSTN device.

Factor 2 in a FORCE statement identifies the file from which the next record is to be selected. If the statement is executed, the record is read at the start of the next program cycle. If more than one FORCE statement is executed during the same program cycle, all but the last is ignored. FORCE should not be specified at total time.

FORCE statements override the multifile processing method by which the program normally selects records. However, the first record to be processed is always selected by the normal method. The remaining records can be selected by FORCE statements.

Figure 10-11 shows how the FORCE operation can be used to control input from primary and secondary files.

Calculation Specifications



Line	Form Type	Control Level (LØ-L9, LR, SR, AN/OR)	Indicators	Factor 1	Operation	Factor 2	Result Field Name	Length	Decimal Positions	Half Adjust (H)	Resulting Indicators	Comments
01	C				}							
02	C				}							
03	C				EXSR	SRTNB						
04	C				}							
05	C				}							
06	C	L2			EXSR	SRTNA						
07	C				}							
08	C				}							
09	C	SR		SRTNA	BEGSR							
10	C	SR	1		}							
11	C	SR	2ØNH1NØ1		}							
12	C	SR	5Ø		EXSR	SRTNC						
13	C	SR			}							
14	C	SR			}							
15	C	SR			ENDSR							
16	C	SR		SRTNB	BEGSR							
17	C	SR			}							
18	C	SR			}							
19	C	SR		START	TAG							
20	C	SR			}							
21	C	SR			GOTO	END						
22	C	SR			}							
23	C	SR			}							
24	C	SR			GOTO	START						
25	C	SR		END	ENDSR							
26	C			SRTNC	BEGSR							
27	C				}							
28	C				}							
29	C				ENDSR							
30	C											
31	C											

Calculation operations

One subroutine can call another subroutine.

GOTO and TAG operations can be used within a subroutine.

Figure 10-10. Example of Coding Subroutines

The NBR field of each primary record contains the number of secondary records to be read and written after each primary record is read. If NBR is less than or equal to zero, a halt occurs. No primary or secondary records are read. Processing begins with the next primary record according to normal selection.

If NBR is greater than zero, the field is reduced by 1 and tested (line 02 of the calculation specifications). If the result is not negative, the FORCE operation calls for input on the next program cycle from the secondary file. The primary record is written, and secondary records are read and written until NBR is negative (indicator 03 is on). The FORCE operation in line 04 then calls for input on the next primary file.

File Description Specifications

Line	Form Type	Filename	File Type	File Designation	End of File	Sequence	File Format	Block Length	Record Length	Device
02	F	INPUT1	I	P	E		F	64	64	DISK
03	F	INPUT2	I	S			F	64	64	DISK
04	F	OUTPUT	O				F	64	64	DISK
05	F									

Input Specifications

Line	Form Type	Filename	Sequence	Record Identifying Indicator	Position	Not (N)	C/Z/D	Character	From	To	Field Name
01	I	INPUT1	AA	Ø1	1		C	1			
02	I								2	2Ø	NBR
03	I								3	8	FIELDA
04	I								9	24	FIELDB
05	I								25	3Ø	FIELDC
06	I	INPUT2	BB	Ø2	1		C	1			
07	I								13	18	FIELDA
08	I								2Ø	35	FIELDB
09	I								38	43	FIELDC
10	I										

Calculation Specifications

Line	Form Type	Indicators	Factor 1	Operation	Factor 2	Result Field Name	Resulting Indicators High (Plus)	Low (Minus)	Equal (Zero)	Comments
01	C	Ø1	NBR	COMP	Ø		H1	H1		
02	C	NH1	NBR	SUB	1	NBR		Ø3		
03	C	NØ3NH1		FORCE	INPUT2					NEXT CYCLE SEC
04	C	Ø3NH1		FORCE	INPUT1					NEXT CYCLE PRI
05	C									

Output Specifications

Line	Form Type	Filename	Type (H/D/T/E)	Output Indicators	Field Name	End Position in Output Record
01	O	OUTPUT	D	Ø1		
02	O	OR		Ø2		
03	O				FIELDA	1Ø
04	O				FIELDB	27
05	O				FIELDC	33
06	O					

Commas	Zero Balances to Print	No Sign	CR	–	
Yes	Yes	1	A	J	X = Remove Plus Sign
Yes	No	2	B	K	Y = Date Field Edit
No	Yes	3	C	L	Z = Zero Suppress
No	No	4	D	M	

Figure 10-11. Example of FORCE Operation Controlling Input

GOTO (BRANCH TO)

Indicators				Result	Resulting Indicators		
7-8	9-17	Factor 1	Factor 2	Field	54-55	56-57	58-59
Opt	Opt	Blank	Required	Blank	Blank	Blank	Blank

The GOTO operation allows operations to be skipped by instructing the program to go to (or branch to) another operation. A GOTO operation can be used to specify a branch:

- To a previous or a succeeding specification line
- From a detail calculation line to another detail calculation line
- From a total calculation line to another total calculation line

However, a branch cannot be made from a detail calculation line to a total calculation line or vice versa. Neither can a branch be made from calculations conditioned by L0 through L9 to calculations conditioned by LR or vice versa. (A total calculation line is defined as one that is conditioned by a control level indicator in columns 7 and 8 of the calculation specifications.)

Factor 2 must contain the name of the label to which the program is to branch. This label is entered in factor 1 of a TAG operation. If the GOTO is within the subroutine, the label can be specified on the ENDSR statement in factor 1. The label can be from 1 to 6 characters long and must begin in column 33 with an alphabetic character. The remaining characters can be any combination of alphabetic or numeric characters. Blanks must not appear between characters in the label.

Factor 1 and the result field are not used in this operation. The GOTO operation can be conditioned by any indicators. If no indicators are specified, the operation is always done.

See Figures 10-12 and 10-13 for examples of the GOTO operation.

Calculation Specifications

Line	Form Type	Control Level (L0-L9, LR, SR, AN/OR)	Indicators	Factor 1	Operation	Factor 2	Result Field Name	Length	Decimal Positions	Half Adjust (H)	Resulting Indicators Plus / High (1>2)	Minus / Low (1<2)	Zero / Equal (1=2)	Comments
01	C			FIELDA	SUB	FIELDB	FIELDB	5	2			10		
02	C		10		GOTO	RTN1								
03	C			FIELDA	MULT	4	SAVE	8	2					
04	C				GOTO	RTN1								
05	C			RTN2	TAG									
06	C				Some calculation									
07	C				operations									
08	C													
09	C			RTN1	TAG									
10	C				Some calculation									
11	C				operations									
12	C												15	
13	C		N15		GOTO	RTN2								
14	C				TESTZ		FIELDC				20		20	
15	C		20		GOTO	END								
16	C				MHLZO	FIELDC	FIELDD							
17	C			END	TAG									
18	C													
19	C													

1. If the result of the subtraction in line 01 is minus (indicator 10 is on), a branch is taken to RTN1 (routine 1) named by the TAG operation code in line 09. Notice that both the GOTO (line 02) and TAG (line 09) are not conditioned by control level indicators.

2. If the branch is not taken in line 02, the multiplication in line 03 is performed. Then the branch to RTN1 (line 09) must be taken because this branch is not conditioned by indicators.

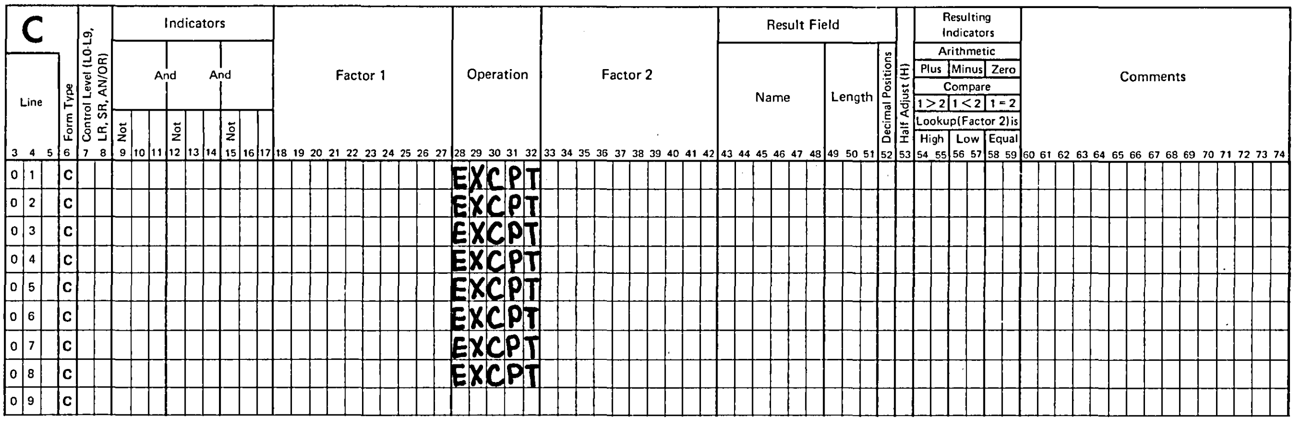
3. Operations in lines 10 through 12 are then done. If the operation in line 12 does not turn indicator 15 on, a branch is taken backwards to RTN2 (line 05).

4. Operations then go in the order specified again from lines 06 through 12. Nothing is done in line 09 because TAG gives only a name. These same operations are performed again and again until indicator 15 does turn on.

5. When indicator 15 is on, the branch to RTN2 is not taken. The TESTZ operation is then performed. If this operation causes indicator 20 to turn on, a branch is taken to line 17 (GOTO END). If indicator 20 is not on, the operation in line 16 is done.

Figure 10-12. Using GOTO and TAG (Skipping Operations)

Calculation Specifications

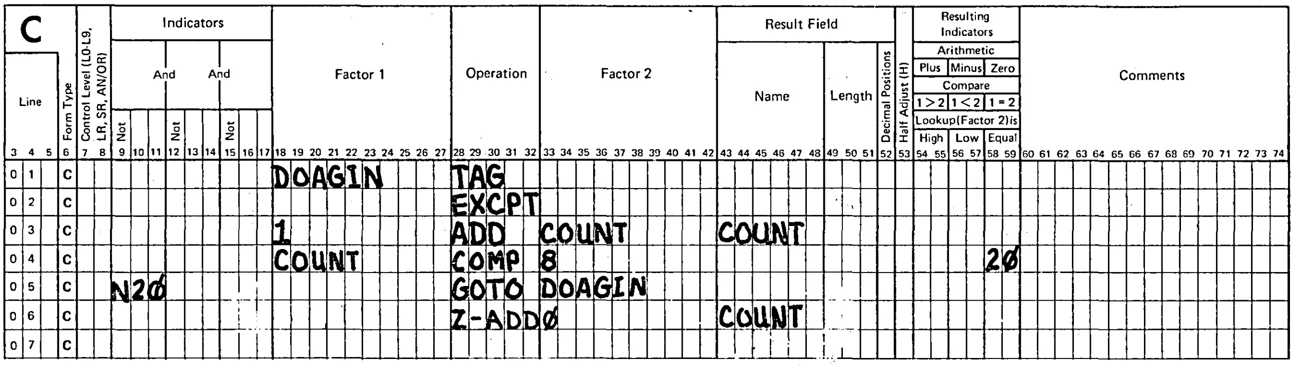


Line	Form Type	Control Level (L0-L9, LR, SR, AN/OR)	Indicators (Not, And Not, And Not)	Factor 1	Operation	Factor 2	Result Field Name	Length	Decimal Positions	Half Adjust (H)	Resulting Indicators: Plus / 1>2 / High	Minus / 1<2 / Low	Zero / 1=2 / Equal	Comments
01	C				EXCPT									
02	C				EXCPT									
03	C				EXCPT									
04	C				EXCPT									
05	C				EXCPT									
06	C				EXCPT									
07	C				EXCPT									
08	C				EXCPT									
09	C													

Assume you want to make eight mailing labels for every customer you have. The customer's name and address are found on an input record. Because you want to write eight labels for each record, you can use exception lines and the operation EXCPT instead of coding eight identical output line specifications. (See *Exception* (EXCPT) in this chapter for further information.)

However, by using branching, you can code it all in five lines as shown below. An EXCPT line is printed out. One is added to COUNT to keep track of how many times the line is printed. Then COUNT is compared to 8. If COUNT does not equal 8, a branch is taken back to the beginning (GOTO DOAGIN). If COUNT equals 8, the branch is not taken. Instead, the COUNT field is set to zero for the next cycle.

Calculation Specifications



Line	Form Type	Control Level (L0-L9, LR, SR, AN/OR)	Indicators (Not, And Not, And Not)	Factor 1	Operation	Factor 2	Result Field Name	Length	Decimal Positions	Half Adjust (H)	Resulting Indicators: Plus / 1>2 / High	Minus / 1<2 / Low	Zero / 1=2 / Equal	Comments
01	C			DOAGIN	TAG									
02	C				EXCPT									
03	C			1	ADD	COUNT	COUNT							
04	C			COUNT	COMP	8							2Ø	
05	C		N2Ø		GOTO	DOAGIN								
06	C				Z-ADD	Ø	COUNT							
07	C													

Figure 10-13. Using GOTO and TAG to Eliminate Duplicate Coding

KEY (KEY)

Indicators				Result	Resulting Indicators		
7-8	9-17	Factor 1	Factor 2	Field	54-55	56-57	58-59
Opt	Opt	Optional	Blank	Required	Optional	Optional	Optional

The KEY operation causes a pause in calculations during which the operator can enter data from the keyboard. All KEY operations are directed to the display station that loaded the program or to the display station assigned to the program by the WORKSTN OCL statement.

To use the KEY operation code, the device name KEYBORD must be specified in columns 40 through 46 of the file description specifications. KEY can be used only with a KEYBORD input file. As the data is keyed by the operator, it is displayed on the screen in one of two formats:

- If the record length is 40 or less, the display consists of six lines, with 40 characters per line, centered both vertically and horizontally on the screen.
- If the record length is greater than 40, the display consists of 12 or 24 lines with 79 characters per line (1 character is reserved for field attributes).

When the KEY operation is used, the contents of the result field are determined by the operator's response. The possible responses are:

- The operator keys the data and presses an entry function key. If not all positions of a field are keyed, numeric fields are right-justified and padded to the left with zeros; alphameric fields are padded to the right with blanks.
- The operator presses only an entry function key, which causes any data in the result field to be changed to zero or blank.
- The operator presses the Dup key and then an entry function key, which does not modify the data in the result field.

Note: The operator can use any one of the following four keys as an entry function key: Field Exit, Field-, Field+, or Enter/Rec Adv. However, if data has been entered into a numeric field, the Enter/Rec Adv key cannot be used as an entry function key.

Bypassing a KEY Operation

When the KEY operation causes a pause in the calculations, the operator can go to the next calculation without keying any data for the current calculation. To do this, the operator simply presses an entry function key, which causes the data in the result field to be changed to zeros or blanks. After each KEY operation (regardless of whether data is entered), the operator must press an entry function key before the next operation can be done. See *SET (Set)* in this chapter for the special situation that allows the SET and KEY operations to be combined so an entry function key has to be pressed only once.

Specifications for a KEY Operation

The following specifications are made for a KEY operation:

Columns 7-8: Enter any valid control level indicator or AN/OR. However, leave these columns blank if the KEY operation is not part of a subroutine or if it is to be performed only at detail time.

Columns 9-17: Enter any valid conditioning indicators, including command key indicators if they have been specified in a SET or SETOF operation. However, leave these columns blank if the KEY operation is to be performed on every program cycle.

Columns 18-27: Enter the constant, literal, field name, or table or array element to be displayed on the display screen.

Columns 28-30: Enter the operation code KEY.

Columns 31-32: Enter the message identification code (MIC) corresponding to the message in the user message member file that is to be displayed on the display screen. This message prompts the operator to perform a KEY operation. Valid entries are 01 through 99. An entry is required in columns 31 and 32 when columns 18 through 27 are blank. If no user message member is specified prior to execution with the MEMBER statement or if there is no message with the associated MIC, the system prompt 'nn-MESSAGE INDICATOR' is displayed, where nn is the contents of columns 31 and 32. If columns 18 through 27 contain an entry by which the keying operation is prompted, the contents of columns 31 and 32 are ignored.

Columns 33-42: Leave these columns blank.

Columns 43-48: Enter the name of the field to be keyed.

Columns 49-51: Enter the length of the keyed field if the field specified in columns 43 through 48 is not defined elsewhere. The maximum length for a numeric field is 15. The maximum length for an alphameric field is 40 if the record length is less than or equal to 40, or is 79 if the record length is greater than 40.

Column 52: Leave this column blank for alphameric fields. For numeric fields, enter the number of decimal positions (0 through 9) in the keyed field if that field is not defined elsewhere.

Column 53: Leave this column blank.

Columns 54-59: Use these columns to test the condition of a numeric field (plus, minus, or zero) or to test an alphameric field for blanks (columns 58 and 59).

Figure 10-14 shows examples of KEY operations.

The KEY operation is normally used with the SET operation. See *SET* (SET) for further information on this topic.

Using KEY Operations in Subroutines

Sometimes it is necessary to write a program that has the KEY operations performed at several different points in the program. Instead of writing these KEY operations and related SET operations every time they are needed, you can write them just once in a subroutine. Then, call the subroutine each time it is needed (see *Subroutine Operations* in this chapter for information on specifying and using subroutines).

User Message Member

The System/34 system support program product lets you create your own message members, which are called user message members. These message members can contain prompts or informational messages to be displayed during your RPG II program.

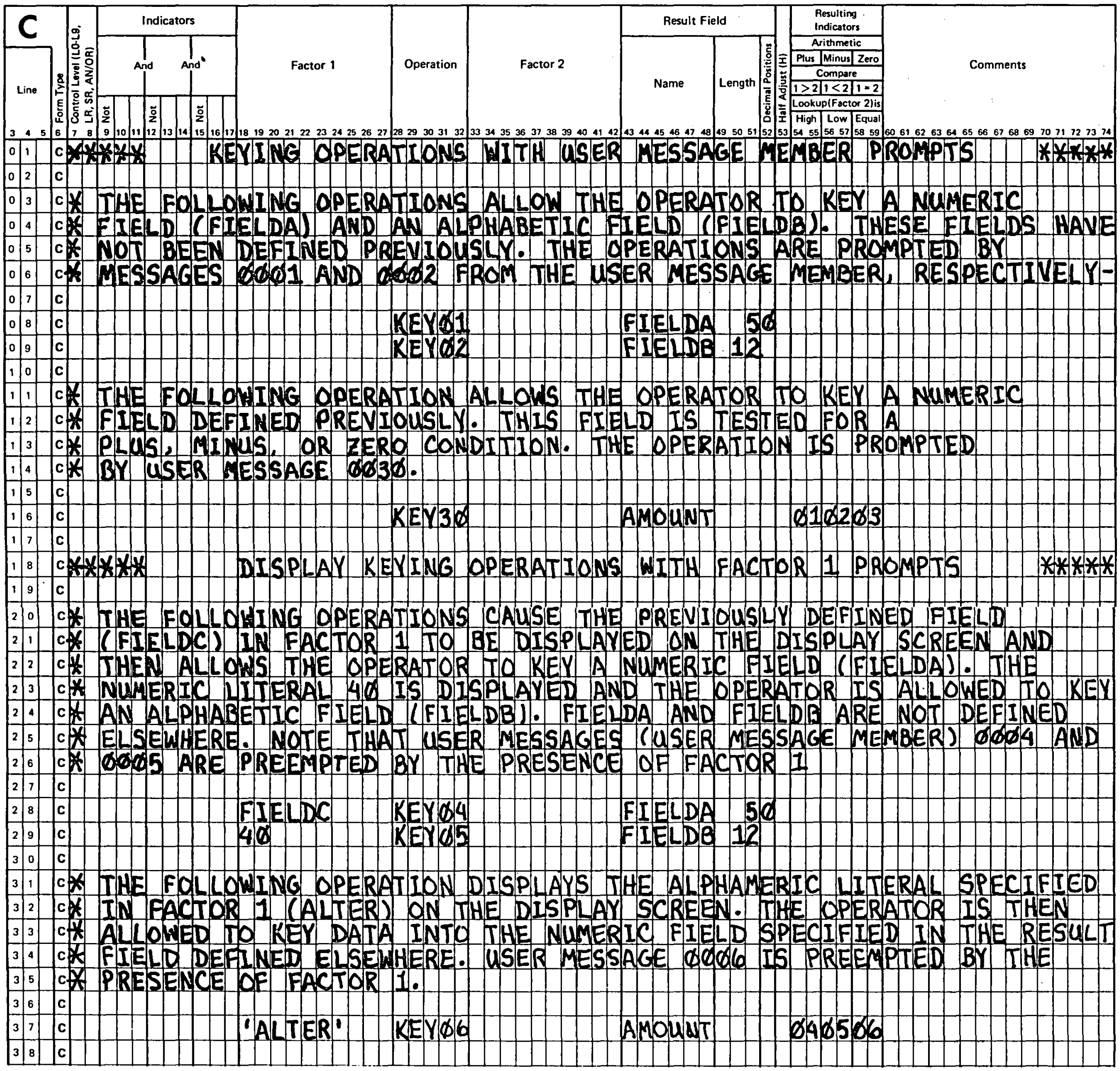
For information on creating message members, see *$MGBLD Utility Program* in the *System Support Reference Manual.* The messages contained in user message members are displayed when you specify the halt indicators (H1 through H9) or the message indicator option of the SET and KEY operation codes. The messages displayed must be formatted so MIC numbers 0001 through 0109 are assigned to the specific function as follows:

MIC	**Function**
0001-0099	Message to be displayed as specified by the nn portion of the SET and KEY operation codes (SETnn, KEYnn, where nn = 01 to 99).
0100	Message to be displayed at the end of an RPG II cycle when the system is finished processing outstanding halt indicators.
0101-0109	Message to be displayed at the end of an RPG II cycle in which the system has encountered H1 through H9 halt indicators (0101 through 0109 correspond to H1 through H9 respectively).

For a message contained in a user message member to be displayed, the message text must exist in an object message member. The message member must be specified in the MEMBER USER1 OCL statement, and the SETnn or KEYnn operation or an H1 to H9 indicator must be used in the program. (See *MEMBER Statement* in the *System Support Reference Manual.*)

Note: The specified user message member remains active until the system processes another MEMBER statement or the display station session is ended (the display station operator signs off). If one of the user message members is active (either USER1 or USER2) and an execution time error is encountered, you will receive a user message rather than the appropriate system message (unless you have copied the system messages into your user message member). For more information on how long a user message member remains active, see *MEMBER Statement* in the *System Support Reference Manual.*

Calculation Specifications



```
01 C*****    KEYING OPERATIONS WITH USER MESSAGE MEMBER PROMPTS    *****
02 C
03 C* THE FOLLOWING OPERATIONS ALLOW THE OPERATOR TO KEY A NUMERIC
04 C* FIELD (FIELDA) AND AN ALPHABETIC FIELD (FIELDB). THESE FIELDS HAVE
05 C* NOT BEEN DEFINED PREVIOUSLY. THE OPERATIONS ARE PROMPTED BY
06 C* MESSAGES ØØØ1 AND ØØØ2 FROM THE USER MESSAGE MEMBER, RESPECTIVELY-
07 C
08 C                     KEYØ1          FIELDA  5Ø
09 C                     KEYØ2          FIELDB 12
10 C
11 C* THE FOLLOWING OPERATION ALLOWS THE OPERATOR TO KEY A NUMERIC
12 C* FIELD DEFINED PREVIOUSLY. THIS FIELD IS TESTED FOR A
13 C* PLUS, MINUS, OR ZERO CONDITION. THE OPERATION IS PROMPTED
14 C* BY USER MESSAGE ØØ3Ø.
15 C
16 C                     KEY3Ø          AMOUNT     Ø1Ø2Ø3
17 C
18 C*****      DISPLAY KEYING OPERATIONS WITH FACTOR 1 PROMPTS     *****
19 C
20 C* THE FOLLOWING OPERATIONS CAUSE THE PREVIOUSLY DEFINED FIELD
21 C* (FIELDC) IN FACTOR 1 TO BE DISPLAYED ON THE DISPLAY SCREEN AND
22 C* THEN ALLOWS THE OPERATOR TO KEY A NUMERIC FIELD (FIELDA). THE
23 C* NUMERIC LITERAL 4Ø IS DISPLAYED AND THE OPERATOR IS ALLOWED TO KEY
24 C* AN ALPHABETIC FIELD (FIELDB). FIELDA AND FIELDB ARE NOT DEFINED
25 C* ELSEWHERE. NOTE THAT USER MESSAGES (USER MESSAGE MEMBER) ØØØ4 AND
26 C* ØØØ5 ARE PREEMPTED BY THE PRESENCE OF FACTOR 1
27 C
28 C           FIELDC    KEYØ4          FIELDA  5Ø
29 C           4Ø        KEYØ5          FIELDB 12
30 C
31 C* THE FOLLOWING OPERATION DISPLAYS THE ALPHAMERIC LITERAL SPECIFIED
32 C* IN FACTOR 1 (ALTER) ON THE DISPLAY SCREEN. THE OPERATOR IS THEN
33 C* ALLOWED TO KEY DATA INTO THE NUMERIC FIELD SPECIFIED IN THE RESULT
34 C* FIELD DEFINED ELSEWHERE. USER MESSAGE ØØØ6 IS PREEMPTED BY THE
35 C* PRESENCE OF FACTOR 1.
36 C
37 C           'ALTER'   KEYØ6          AMOUNT     Ø4Ø5Ø6
38 C
```

Figure 10-14. Possible KEY Operations

The first level message corresponding to the H1 to H9 halt indicators can be described in more detail with an associated second level message member with the same MIC number. The text of a second level message can be up to 225 characters long. Second level messages can only be specified with the H1 to H9 halt indicators, and a MEMBER USER2 OCL statement must be included at execution time. After the halt indicators (H1-H9) turn on, all calculations and detail output operations are performed for the record before processing stops, and a message is issued. If the halt indicators (H1-H9) are turned on during LR time, the program does not halt processing and continues to completion.

Figure 10-15 shows the coding and OCL statements necessary to issue a message. The source member (MESG1 for this example) must be loaded into a library via the source entry utility or the $MAINT utility. The object member message MESG1 must be created prior to execution. For information on loading the message source member, see *Sign-On Procedures* in the *SEU Reference Manual*; for information on creating a message load member, see the *System Support Reference Manual.*

Calculation Specifications

Line (3-5)	Form Type (6)	Control Level (L0-L9, LR, SR, AN/OR) (7-8)	Indicators (9-17)	Factor 1 (18-27)	Operation (28-32)	Factor 2 (33-42)	Result Field Name (43-48)	Length (49-51)	Decimal Positions (52)	Half Adjust (H) (53)	Resulting Indicators Plus / 1>2 / High (54-55)	Minus / 1<2 / Low (56-57)	Zero / 1=2 / Equal (58-59)	Comments (60-74)
01	C				KEYØ1		FLD	4						
02	C			FLD	COMP	'HALT'							H1	
03	C													
04	C													

Messages are received from the system message member because no overriding MEMBER statement was specified:

```
// LOAD USER
// RUN
```

01-MESSAGE INDICATOR is the text issued as the prompt for the KEY operation. If HALT is keyed, the indicator H1 turns on and the RPG II operator message 0101 RPG II INDICATOR H1 IS ON is displayed. If the 0 option is selected, message 0100 ALL HALT INDICATORS PRE–VIOUSLY DISPLAYED is issued.

Messages are retrieved from the user-created message member MESG1 specified in the MEMBER statement:

```
// LOAD USER
// MEMBER USER1-MESG1
// RUN
```

The prompt KEY HALT TO TERMINATE PROGRAM is the text issued for the KEY01 operation. This is the contents of the object member loaded into the user message member specified by the MEMBER statement. Subsequently, the USER 0101 message text that is issued when the literal HALT is keyed is HALT HAS BEEN ENTERED WITH KEY OP.

The second message issued, USER 0100, has not been defined in the user message member; thus it cannot be retrieved, and the message MESSAGE NOT RETRIEVED (SEE MSG MANUAL) is issued.

Figure 10-15. Issuing a Message

Special Combinations of the SET and KEY Operations

Normally, the operator must press an entry function key after each KEY operation or after command keys specified in a SET operation are pressed. However, it is possible to combine these operations so that the operator can press command keys (specified in columns 54 through 59 of a SET operation), key a field (specified in a KEY operation), and only press an entry function key once. This is only possible if:

- The SET operation immediately precedes the KEY operation.
- The SET end KEY operations are conditioned by the same indicators (columns 7 through 17). Indicators for both operations must be specified in the same order.
- The SET and KEY operations contain the same message indicators. If factor 1 is used to display messages, then columns 31 and 32 can be blank to satisfy this requirement.
- Factor 1 for the SET and KEY operations can be the same, different, or missing from one operation. If factor 1 is specified for both the SET and KEY operations, the contents of each factor 1 are displayed.

If the data field is numeric, the operator must first press the specified command key, key the field, and then press the Field Exit, Field +, or Field - key. The Enter/Rec Adv key is not allowed as an entry function key for a numeric field. For an alphameric field, the operator must perform the same sequence of steps if the Field Exit or Field + key is pressed. However, if the Enter/Rec Adv key is used, the operator can press the command key and then key the field, or key the field and then press the command key before pressing the Enter/Rec Adv key.

LOKUP (LOOKUP)

Array LOKUP

Indicators				Result	Resulting Indicators		
7-8	9-17	Factor 1	Factor 2	Field	54-55	56-57	58-59
Opt	Opt	Required	Required	Blank	1 required		

Table LOKUP

Indicators				Result	Resulting Indicators		
7-8	9-17	Factor 1	Factor 2	Field	54-55	56-57	58-59
Opt	Opt	Required	Required	Optional	1 required		

The LOKUP operation causes a search to be made for a particular element in a table or array. The table or array is named in factor 2. Factor 1 is the search word (data for which you want to find a match in the table or array named). Factor 1, the search word, can be:

- An alphameric or numeric constant
- A field name
- An array element
- A table name

When a table is named in factor 1, it refers to the element of the table last selected in a LOKUP operation, not to the whole table.

Resulting indicators are always used with a LOKUP operation. The indicators first specify the type of search to be made and then reflect the result of the search. The specified indicator turns on only if the search is successful.

Resulting indicators specify the type of search and reflect the result of the search in the following manner:

- A resulting indicator assigned to equal (columns 58 and 59) instructs the program to search for an entry in the table or array equal to the search word. The first equal entry found turns on the indicator assigned to equal.

- An indicator assigned to low (columns 56 and 57) instructs the program to locate an entry in the table that is nearest to, yet lower in sequence than, the search word. The first such entry found turns on the indicator assigned to low.

- The indicator assigned to high (columns 54 and 55) instructs the program to find the entry that is nearest to, yet higher in sequence than, the search word. The first higher entry found turns on the indicator assigned to high.

At least one resulting indicator must be assigned, but no more than two can be used. Resulting indicators can be assigned to equal and high or to equal and low. The program searches for an entry that satisfies either condition with equal given precedence; that is, if no equal entry is found, the nearest lower or nearest higher entry is selected. If resulting indicators are assigned both to high and low, the indicator assigned to low is ignored.

When you use the LOKUP operation, remember:

- The search word and each table or array element must have the same length and the same format (alphameric or numeric).

- A search can be made for high, low, high and equal, or low and equal only if the table or array is in sequence.

- No resulting indicator turns on if the search is not successful.

LOKUP With One Table

When searching a single table, factor 1, factor 2, and at least one resulting indicator must be specified. Conditioning indicators (specified in columns 7 through 17) can also be used.

Whenever a table element is found that satisfies the type of search being made (equal, high, low), a copy of that table element is placed in a special storage area. Every time a search is successful, the newly found table element is placed in this area, replacing what was there before. If the search is not successful, no table element is placed in the storage area. Therefore, the contents of the area remain the same as before the unsuccessful search.

Resulting indicators reflect the result of the search. If the indicator is on, reflecting a successful search, a copy of the element searched for is in the special storage area.

LOKUP With Two Tables

When two related tables are used in a search, only one is actually searched (see Figure 10-16). When the search condition (high, low, equal) is satisfied, the corresponding elements from both tables are placed in their respective special storage areas and are made available for use.

Factor 1 must contain the search word and factor 2 must contain the name of the table to be searched. The result field must name the related table from which data is also made available for use. A resulting indicator must also be used. Conditioning indicators can be specified in columns 7 through 17, if needed.

The two tables used should have the same number of entries. If the table that is searched contains more elements than its related table, it is possible to satisfy the search condition. However, there might not be an element in the second table that corresponds to the element found in the search table. Unpredictable results can occur.

Note: If you specify a table name in an operation other than LOKUP before a successful LOKUP occurs, unpredictable results can occur because the contents of the special storage area referenced by the table name are unknown.

Referencing the Table Element Found in a LOKUP Operation

Whenever a table name is used in an operation other than LOKUP, the table name actually refers to the data placed in the special storage area by the last successful search. Thus, when you specify the table name in this fashion, elements from a table can be used in calculation operations.

If the table is used as factor 1 in a LOKUP operation, the contents of the special storage area are used as the search word. In this way an element from a table can itself become a search word.

The table can also be used as the result field in operations other than the LOKUP operation. In this case the contents of the special storage area are changed by the calculation specification. The corresponding table element in the table in main storage is also changed. In this way the contents of the table can be modified by calculation operations (see Figure 10-17).

Figure 10-18 shows a sample LOKUP operation for a table.

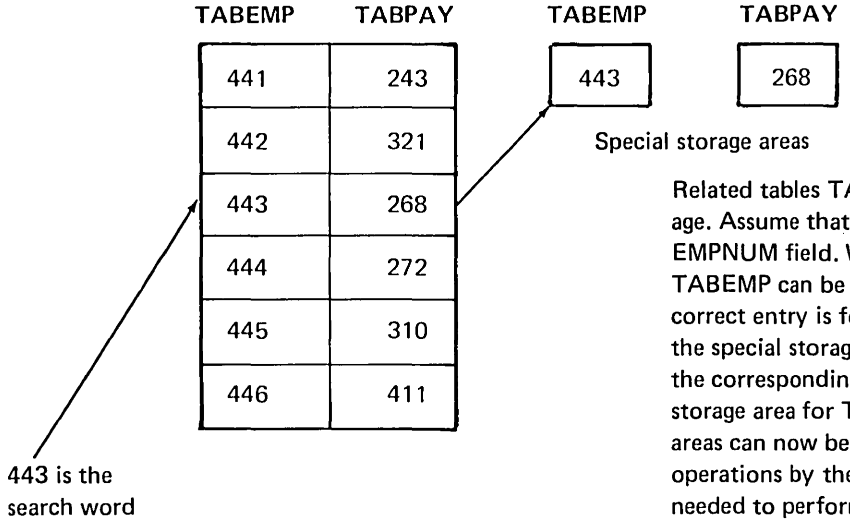
Using the LOKUP Operation with Arrays

The LOKUP specifications for arrays are the same as for tables except that the result field cannot be used. In addition, if the element searched for is found, it is not moved to a special storage area because these areas are used only for tables. The indicators reflect only that the element is in the array; therefore, the programmer does not have ready access to this item.

Figure 10-19 shows two LOKUP operations performed with an array.

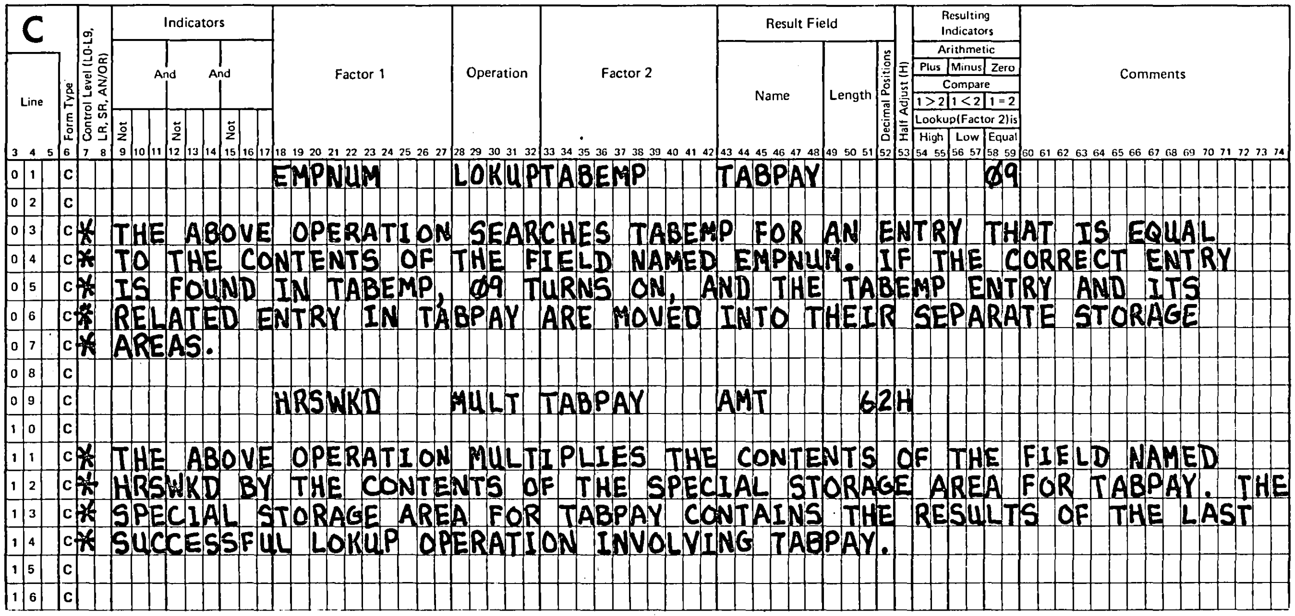
Starting the Search at a Particular Array Element

To save processing time, start the LOKUP search at a particular element in the array. This type of search is indicated by additional entries in columns 33 through 42. Enter the name of the array to be searched in these columns followed by a comma and a numeric literal or by the name of a numeric field (with 0 decimal positions). The numeric literal or numeric field provides the number of the element at which the search is to start (see Figure 10-20). This numeric literal or field is called the index because it points to a certain element in the array. All other columns are used as previously described for the normal lookup operation.



Related tables TABEMP and TABPAY are read into storage. Assume that an input record is read with 443 in the EMPNUM field. With 443 as the search word, the table TABEMP can be searched for an equal entry. When the correct entry is found, the table item 443 is moved into the special storage area for TABEMP. At the same time, the corresponding item 268 is moved into the special storage area for TABPAY. The contents of the storage areas can now be referenced in subsequent calculation operations by the appropriate table name. The coding needed to perform the LOKUP operation also shows how to reference the contents of the special storage area after a successful LOKUP operation.

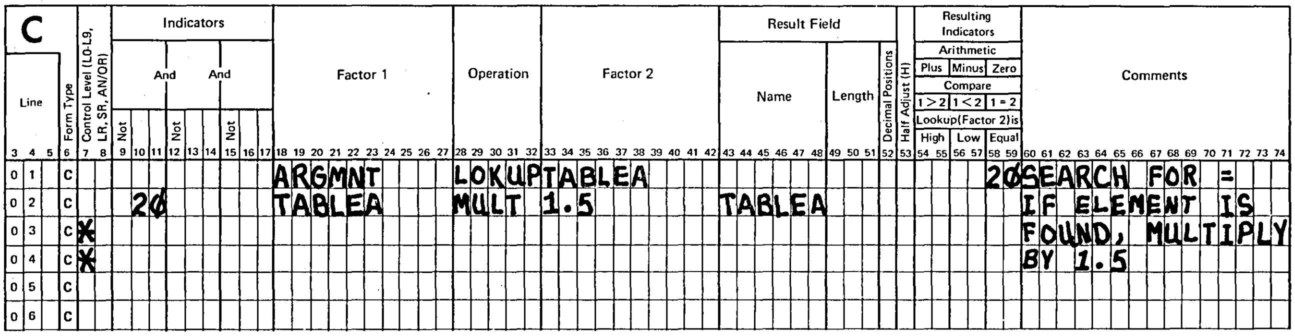
Calculation Specifications


C
Line
Form Type
Control Level (L0-L9, LR, SR, AN/OR)
Indicators
And
And
Not
Not
Not
Factor 1
Operation
Factor 2
Result Field
Name
Length
Decimal Positions
Half Adjust (H)
Resulting Indicators
Arithmetic
Plus
Minus
Zero
Compare
1 > 2
1 < 2
1 = 2
Lookup(Factor 2)is
High
Low
Equal
Comments

```
01 C          EMPNUM    LOKUPTABEMP     TABPAY             Ø9
02 C
03 C* THE ABOVE OPERATION SEARCHES TABEMP FOR AN ENTRY THAT IS EQUAL
04 C* TO THE CONTENTS OF THE FIELD NAMED EMPNUM. IF THE CORRECT ENTRY
05 C* IS FOUND IN TABEMP, Ø9 TURNS ON, AND THE TABEMP ENTRY AND ITS
06 C* RELATED ENTRY IN TABPAY ARE MOVED INTO THEIR SEPARATE STORAGE
07 C* AREAS.
08 C
09 C          HRSWKD    MULT TABPAY     AMT       62H
10 C
11 C* THE ABOVE OPERATION MULTIPLIES THE CONTENTS OF THE FIELD NAMED
12 C* HRSWKD BY THE CONTENTS OF THE SPECIAL STORAGE AREA FOR TABPAY. THE
13 C* SPECIAL STORAGE AREA FOR TABPAY CONTAINS THE RESULTS OF THE LAST
14 C* SUCCESSFUL LOKUP OPERATION INVOLVING TABPAY.
15 C
16 C
```

Figure 10-16. LOKUP Operation for Related Tables

Calculation Specifications



Line	Form Type	Control Level (L0-L9, LR, SR, AN/OR)	Indicators	Factor 1	Operation	Factor 2	Result Field Name	Length	Decimal Positions	Half Adjust (H)	Resulting Indicators High / Low / Equal	Comments
01	C			ARGMNT	LOKUP	TABLEA					Equal: 2Ø	SEARCH FOR =
02	C		2Ø	TABLEA	MULT	1.5	TABLEA					IF ELEMENT IS
03	C	*										FOUND, MULTIPLY
04	C	*										BY 1.5
05	C											
06	C											

Figure 10-17. Referencing the Table Element Found in LOKUP Operation

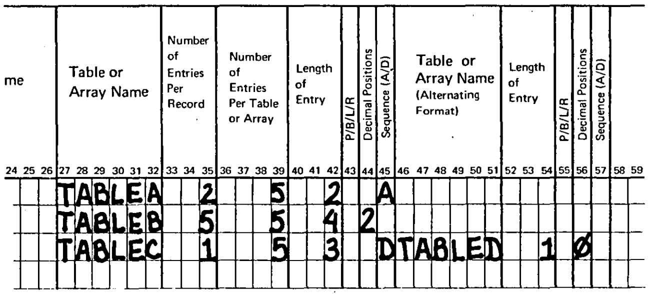
These four tables are used in the LOKUP operations described in Figure 10-18, Part 2.

	First Entry	Second Entry	Third Entry	Fourth Entry	Fifth Entry
Table A	01	05	08	32	96
Table B	06.13	02.12	47.16	28.70	15.16
Table C	WWW	NNN	LLL	GGG	AAA
Table D	7	8	3	2	5

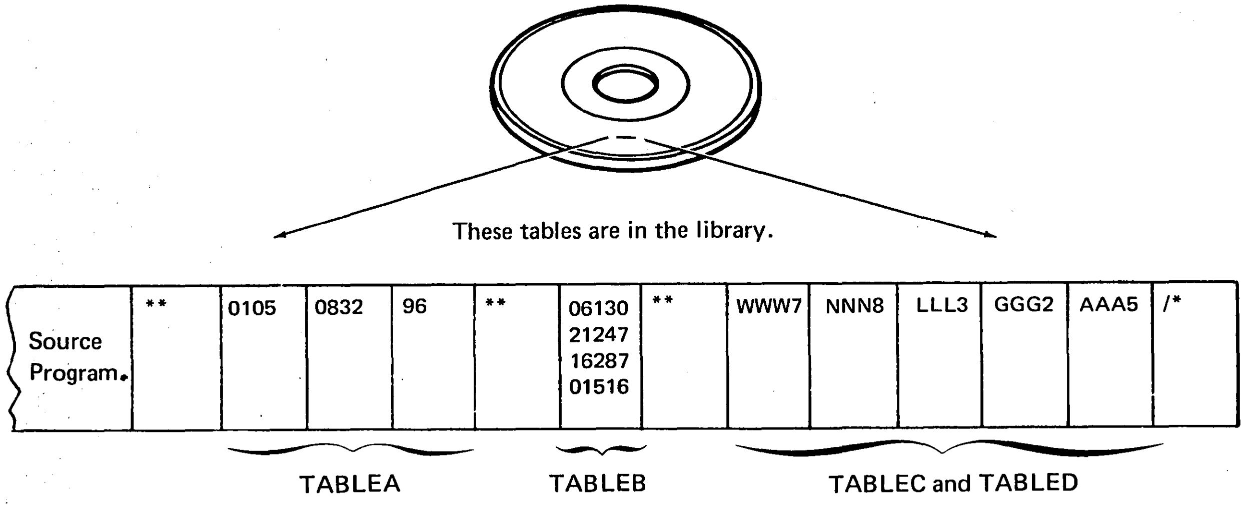
The table input records are loaded into the system in this order at compile time.

Tables A and B are described separately on the extension specifications and are, therefore, entered separately. Tables C and D are related tables and are entered in alternating format on the table input records.

Extension Specifications



me (24-26)	Table or Array Name (27-32)	Number of Entries Per Record (33-35)	Number of Entries Per Table or Array (36-39)	Length of Entry (40-42)	P/B/L/R (43)	Decimal Positions (44)	Sequence (A/D) (45)	Table or Array Name (Alternating Format) (46-51)	Length of Entry (52-54)	P/B/L/R (55)	Decimal Positions (56)	Sequence (A/D) (57)	58 59
	TABLEA	2	5	2			A						
	TABLEB	5	5	4		2							
	TABLEC	1	5	3			D	TABLED	1		Ø		



Source Program.	**	0105	0832	96	**	06130 21247 16287 01516	**	WWW7	NNN8	LLL3	GGG2	AAA5	/*

Figure 10-18 (Part 1 of 2). Example of Table LOKUP Operations

Calculation Specifications

Line	Form Type	Control Level (L0-L9, LR, SR, AN/OR)	Indicators (Not, And, Not, And, Not)	Factor 1	Operation	Factor 2	Result Field Name	Length	Decimal Positions	Half Adjust (H)	Resulting Indicators: Arithmetic Plus / Compare 1>2 / Lookup(Factor 2)is High	Minus / 1<2 / Low	Zero / 1=2 / Equal	Comments
01	C			'Ø8'	LOKUP	TABLEA	TABLEB				Ø1			
02	C			'Ø8'	LOKUP	TABLEA	TABLEB					Ø2		
03	C			'Ø8'	LOKUP	TABLEA	TABLEB						Ø3	
04	C			'Ø8'	LOKUP	TABLEA					Ø4		Ø5	
05	C			'Ø8'	LOKUP	TABLEA						Ø6	Ø7	
06	C			'ØØ'	LOKUP	TABLEA						Ø8		
07	C			'97'	LOKUP	TABLEA					Ø9			
08	C			'Ø9'	LOKUP	TABLEA	TABLEC				1Ø		11	
09	C			'Ø9'	LOKUP	TABLEA	TABLEC					12	13	
10	C			'LLL'	LOKUP	TABLEC	TABLED				14			
11	C			'JJJ'	LOKUP	TABLEC						15		
12	C			'JJJ'	LOKUP	TABLEC							16	
13	C			'JJJ'	LOKUP	TABLEC	TABLED				17			
14	C			2	LOKUP	TABLED							18	
15	C			6	LOKUP	TABLED							19	
16	C													
17	C													

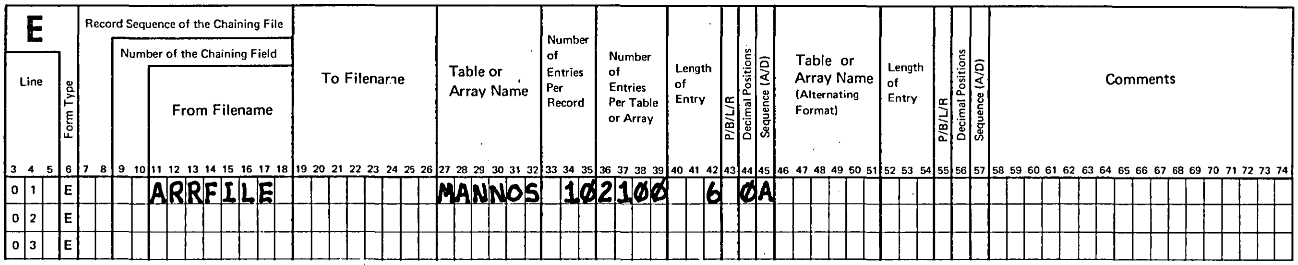
Specifications above are shown for 15 different LOKUP operations for tables A, B, C, and D. Indicators specified in columns 54 through 59 determine the type of search to be made.

Results of the LOKUP operations (in the following table) show whether the element searched for was found, what indicator is on to indicate a successful search, and what item was taken from a related table when one was used.

Specification Line Number	Entry Found	Indicator On	Table Item Satisfying Search Condition	Table Item Used From Related Table
01	Yes	01	32	28.70
02	Yes	02	05	02.12
03	Yes	03	08	47.16
04	Yes	05	08	
05	Yes	07	08	
06	No			
07	No			
08	Yes	10	32	GGG
09	Yes	12	08	LLL
10	Yes	14	NNN	8
11	Yes	15	GGG	
12	No			
13	Yes	17	LLL	3
14	Yes	18	2	
15	No			

Figure 10-18 (Part 2 of 2). Example of Table LOKUP Operations

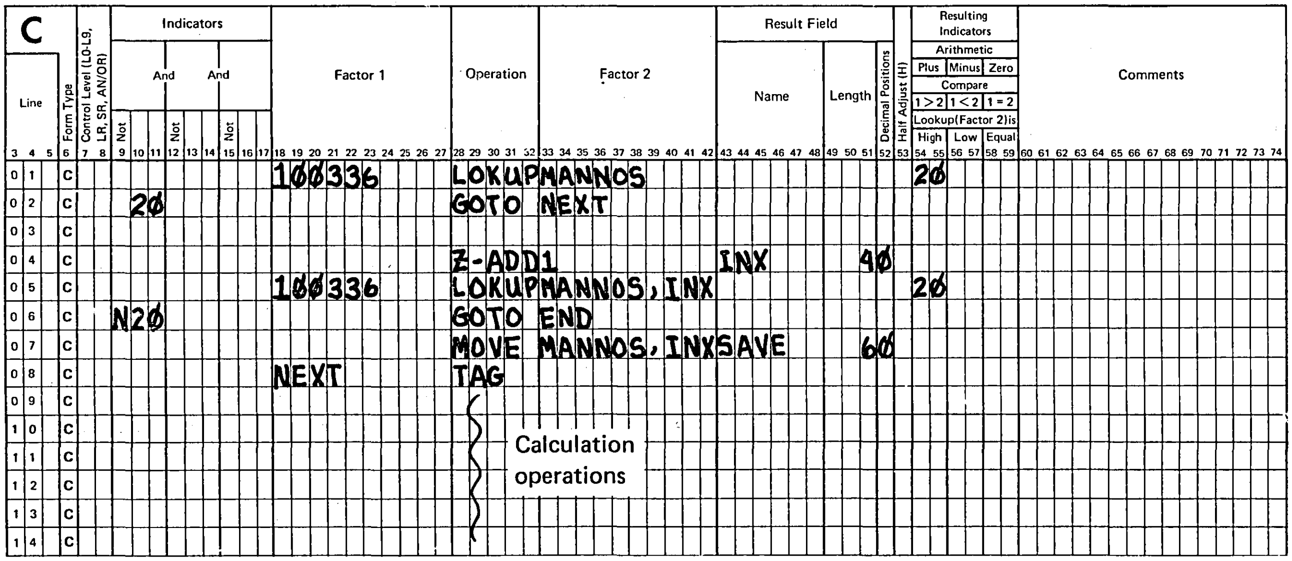
Extension Specifications



Line	Form Type	Record Sequence of the Chaining File	Number of the Chaining Field	From Filename	To Filename	Table or Array Name	Number of Entries Per Record	Number of Entries Per Table or Array	Length of Entry	P/B/L/R	Decimal Positions	Sequence (A/D)	Table or Array Name (Alternating Format)	Length of Entry	P/B/L/R	Decimal Positions	Sequence (A/D)	Comments
3 4 5	6	7 8	9 10	11 12 13 14 15 16 17 18	19 20 21 22 23 24 25 26	27 28 29 30 31 32	33 34 35	36 37 38 39	40 41 42	43	44	45	46 47 48 49 50 51	52 53 54	55	56	57	58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74
0 1	E			ARRFILE		MANNOS	1Ø	21ØØ	6		Ø	A						
0 2	E																	
0 3	E																	

MANNOS, a 2100-element array of employee numbers, is read in at execution time from the file ARRFILE with ten 6-position elements per record; the array elements are in ascending order.

Calculation Specifications



Line	Form Type	Control Level (L0-L9, LR, SR, AN/OR)	Indicators	Indicators And	Indicators And	Factor 1	Operation	Factor 2	Result Field Name	Length	Decimal Positions	Half Adjust (H)	Resulting Indicators High (Plus, 1>2)	Low (Minus, 1<2)	Equal (Zero, 1=2)	Comments
3 4 5	6	7 8	9 10 11	12 13 14	15 16 17	18 19 20 21 22 23 24 25 26 27	28 29 30 31 32	33 34 35 36 37 38 39 40 41 42	43 44 45 46 47 48	49 50 51	52	53	54 55	56 57	58 59	60 61 62 63 64 65 66 67 68 69 70 71 72 73 74
0 1	C					1ØØ336	LOKUP	MANNOS					2Ø			
0 2	C		2Ø				GOTO	NEXT								
0 3	C															
0 4	C						Z-ADD	1	INX	4	Ø					
0 5	C					1ØØ336	LOKUP	MANNOS,INX					2Ø			
0 6	C		N2Ø				GOTO	END								
0 7	C						MOVE	MANNOS,INX	SAVE	6	Ø					
0 8	C					NEXT	TAG									
0 9	C															
1 0	C															
1 1	C															
1 2	C															
1 3	C															
1 4	C															

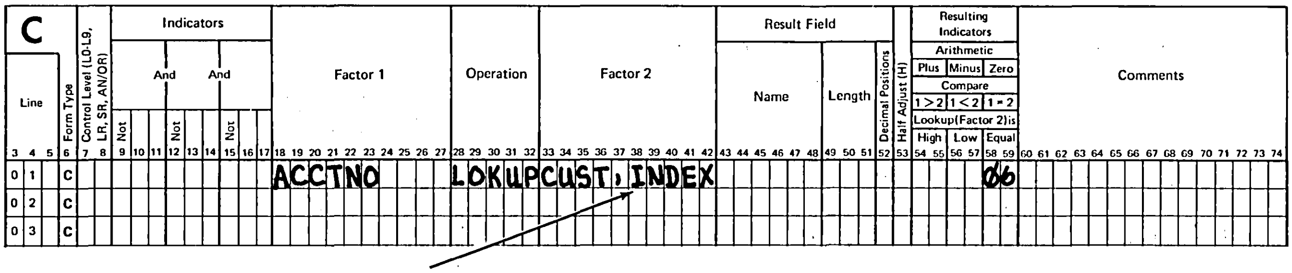
Calculation operations

The first calculation specification is a LOKUP of array MANNOS to find the element nearest to, but higher in sequence than, the search word 100336. If this element is found in the array, indicator 20 turns on and the GOTO in line 02 is performed. Indicator 20 indicates only whether or not the searched for element exists in the array.

The specification on line 05 shows essentially the same LOKUP operation. Indicator 20 turns on when the first element higher in sequence than 100336 is found. However, in this LOKUP operation, the array MANNOS is indexed by the field INX. This index field was set to 1 in line 04, so the LOKUP begins at the first element of MANNOS. If the searched for element is found, the number of this element (not its contents) is placed in the field INX. In this way, the actual element that satisfied the LOKUP can be used in subsequent calculation operations, as in line 07. If no element was found to satisfy the LOKUP, the field INX is reset to 1.

Figure 10-19. LOKUP Operation for an Array

Calculation Specifications



Line	Form Type	Control Level	Indicators	Factor 1	Operation	Factor 2	Result Field Name	Length	Decimal Positions	Half Adjust	Resulting Indicators High 54-55	Low 56-57	Equal 58-59	Comments
01	C			ACCTNO	LOKUP	CUST,INDEX							06	
02	C													
03	C													

The index provides the number of the element at which the search is to start.

An array name plus comma and index that is 10 characters long can be referenced in factor 1 or factor 2 of the calculation specifications only. For all other references, the length is limited to 6 characters.

Figure 10-20. Array Lookup: Starting at a Particular Array Item

The search starts at the specified element and continues until the searched-for element is found or until the end of the array is reached. When a field is specified for the index, an unsuccessful search causes that field to contain the value of 1. If, however, an element is found that satisfies the conditions of the LOKUP operation, the number of that array element (counting from the first element) is placed in the index field. The index field can then be used to reference that array element so it can be used in subsequent calculation operations (see Figure 10-19). However, a numeric literal used as an index is not changed to reflect the result of the search.

Note: If a literal or field index for an array is zero or greater than the number of elements in the array, the following occurs:

- For a literal index, a severe error occurs and compilation ceases.
- For a field index, the job halts, allowing the operator to cancel or continue the program. If the program is continued, the field index is reset to 1.

MHHZO (MOVE HIGH TO HIGH ZONE)

Indicators				Result	Resulting Indicators		
7-8	9-17	Factor 1	Factor 2	Field	54-55	56-57	58-59
Opt	Opt	Blank	Required	Required	Blank	Blank	Blank

The MHHZO operation moves the zone from the leftmost position of factor 2 to the leftmost position of the result field. Factor 2 and the result field must be alphameric.

MHLZO (MOVE HIGH TO LOW ZONE)

Indicators				Result	Resulting Indicators		
7-8	9-17	Factor 1	Factor 2	Field	54-55	56-57	58-59
Opt	Opt	Blank	Required	Required	Blank	Blank	Blank

The MHLZO operation moves the zone from the leftmost position of factor 2 to the rightmost position of the result field. Factor 2 must be alphameric. The result field can be alphameric or numeric.

MLHZO (MOVE LOW TO HIGH ZONE)

Indicators				Result	Resulting Indicators		
7-8	9-17	Factor 1	Factor 2	Field	54-55	56-57	58-59
Opt	Opt	Blank	Required	Required	Blank	Blank	Blank

The MLHZO operation moves the zone from the rightmost position of factor 2 to the leftmost position of the result field. Factor 2 can be numeric or alphameric, but the result field must be alphameric.

MLLZO (MOVE LOW TO LOW ZONE)

Indicators				Result	Resulting Indicators		
7-8	9-17	Factor 1	Factor 2	Field	54-55	56-57	58-59
Opt	Opt	Blank	Required	Required	Blank	Blank	Blank

The MLLZO operation moves the zone from the rightmost position of factor 2 to the rightmost position of the result field. Factor 2 and the result field can be either alphameric or numeric.

Functions of the four move zone operations are shown in Figure 10-21.

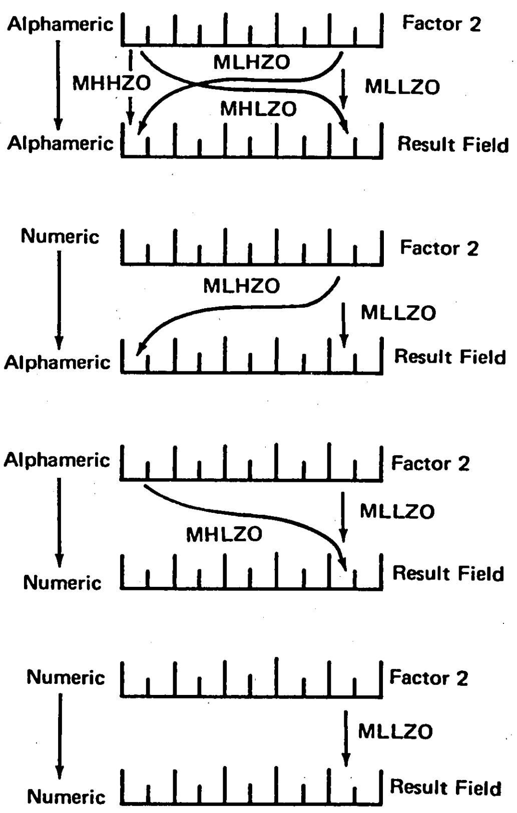


Figure 10-21. Functions of Move Zone Operations

MOVE (MOVE)

Indicators				Result	Resulting Indicators		
7-8	9-17	Factor 1	Factor 2	Field	54-55	56-57	58-59
Opt	Opt	Blank	Required	Required	Blank	Blank	Blank

The MOVE operation transfers characters from factor 2 to the rightmost positions in the result field. Moving starts with the rightmost character of factor 2.

If factor 2 is longer than the result field, the excess leftmost characters of factor 2 are not moved. If the result field is longer than factor 2, the excess leftmost characters in the result field are unchanged.

The MOVE operation is summarized in Figure 10-22.

Factor 2 Shorter Than Result Field

	Factor 2		Result Field	
a. Alphameric	P H 4 S N	Before MOVE	1 2 3 4 5 6 7 8 $\overset{+}{4}$	Alphameric
	P H 4 S N	After MOVE	1 2 3 4 P H 4 S N	
b. Alphameric	P H 4 S N	Before MOVE	1 2 3 4 5 6 7 8 $\overset{+}{4}$	Numeric
	P H 4 S N	After MOVE	1 2 3 4 7 8 4 2 $\overset{-}{5}$	
c. Numeric	1 2 7 8 4 2 5	Before MOVE	1 2 3 4 5 6 7 8 9	Numeric
	1 2 7 8 4 2 5	After MOVE	1 2 1 2 7 8 4 2 5	
d. Numeric	1 2 7 8 4 2 5	Before MOVE	A C F G P H 4 S N	Alphameric
	1 2 7 8 4 2 5	After MOVE	A C 1 2 7 8 4 2 5	

Factor 2 Longer Than Result Field

	Factor 2		Result Field	
a. Alphameric	A C E G P H 4 S N	Before MOVE	5 6 7 8 4	Alphameric
	A C E G P H 4 S N	After MOVE	P H 4 S N	
b. Alphameric	A C E G P H 4 S N	Before MOVE	5 6 7 8 $\overset{+}{4}$	Numeric
	A C E G P H 4 S N	After MOVE	7 8 4 2 $\overset{-}{5}$	
c. Numeric	1 2 7 8 4 2 5	Before MOVE	5 6 7 4 8	Numeric
	1 2 7 8 4 2 5	After MOVE	7 8 4 2 5	
d. Numeric	1 2 7 8 4 2 5	Before MOVE	P H 4 S N	Alphameric
	1 2 7 8 4 2 5	After MOVE	7 8 4 2 5	

Factor 2 and Result Field Same Length

	Factor 2		Result Field	
a. Alphameric	P H 4 S N	Before MOVE	5 6 7 8 4	Alphameric
	P H 4 S N	After MOVE	P H 4 S N	
b. Alphameric	P H 4 S N	Before MOVE	5 6 7 8 4	Numeric
	P H 4 S N	After MOVE	7 8 4 2 $\overset{-}{5}$	
c. Numeric	7 8 4 2 $\overset{-}{5}$	Before MOVE	A L T 5 F	Numeric
	7 8 4 2 $\overset{-}{5}$	After MOVE	7 8 4 2 $\overset{-}{5}$	
d. Numeric	7 8 4 2 $\overset{-}{5}$	Before MOVE	A L T 5 F	Alphameric
	7 8 4 2 $\overset{-}{5}$	After MOVE	7 8 4 2 N	

$\overset{+}{4}$ = letter D

$\overset{-}{5}$ = letter N

Figure 10-22. MOVE Operations

MOVEA (MOVE ARRAY)

Indicators				Result	Resulting Indicators		
7-8	9-17	Factor 1	Factor 2	Field	54-55	56-57	58-59
Opt	Opt	Blank	Required	Required	Blank	Blank	Blank

The MOVEA operation transfers characters from the leftmost positions of factor 2 to the leftmost positions of the result field. Factor 2 and the result field cannot reference the same array even if the array is indexed. All arrays and fields referenced by a MOVEA operation must be alphameric.

The length of the move is determined by the shorter of the lengths of factor 2 and the result field. If factor 2 is longer than the result field, the excess rightmost characters of factor 2 are not moved; if the result field is longer than factor 2, the rightmost characters in the result field are unchanged.

The MOVEA operation makes it possible to:

- Move several contiguous array elements to a single field.
- Move a single field to several contiguous array elements.
- Move contiguous elements of one array to contiguous elements of another array.

Movement of data starts with the first element of an array if the array is not indexed or with the element referenced if the array is indexed. The movement of data is terminated when the last array element is moved or filled or when the number of characters moved equals the length of the shorter field specified by factor 2 and the result field; therefore, the move could terminate in the middle of an array element.

If MOVEA is specified with a figurative constant and an array, the constant fills the entire array. If MOVEA is specified with a figurative constant and an array element, the array is filled with the figurative constant from the referenced element to the end of the array.

Figure 10-23 illustrates the use of the MOVEA operation.

Example: Array-to-array move. No indexing; different length arrays, same element length.

Calculation Specifications

	Operation	Factor 2	Result Field		
			Name	Length	Decimal Positions
26 27	28 29 30 31 32	33 34 35 36 37 38 39 40 41 42	43 44 45 46 47 48	49 50 51	52
	MOVEA	ARRX	ARRY		

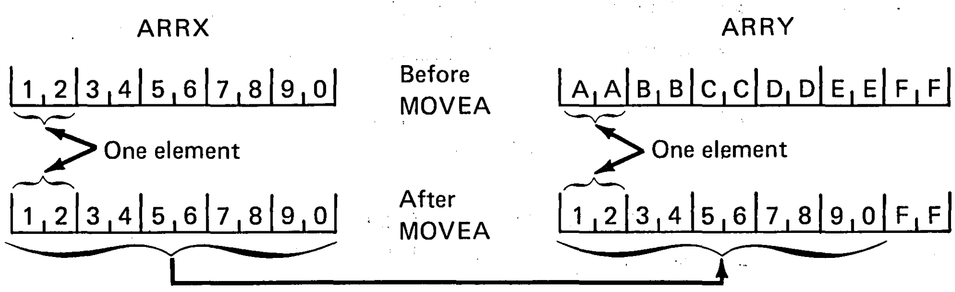
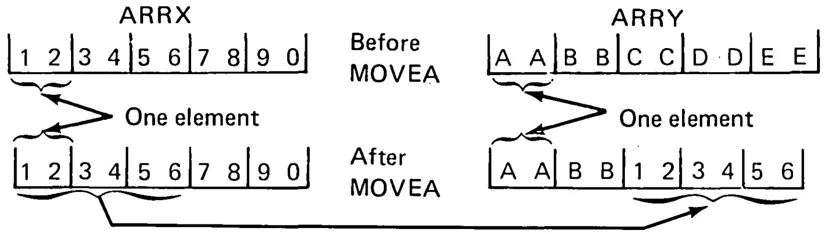


Figure 10-23 (Part 1 of 3). MOVEA Operation

Calculation Specifications

Example: Array-to-array move. Index result field.

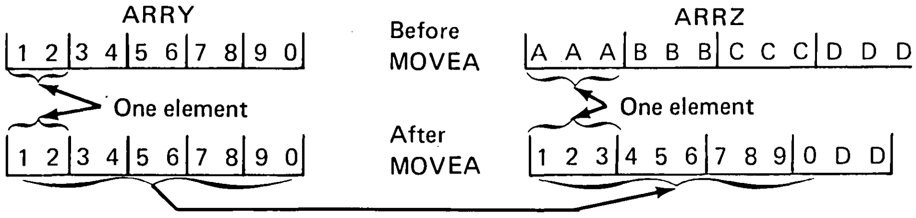
			Result Field	
	Operation	Factor 2	Name	Leng
26 27	28 29 30 31 32	33 34 35 36 37 38 39 40 41 42	43 44 45 46 47 48	49 50
	MOVEA	ARRX	ARRY,3	



Calculation Specifications

Example: Array-to-array move. No indexing, different length array elements.

			Result Field	
	Operation	Factor 2	Name	Leng
26 27	28 29 30 31 32	33 34 35 36 37 38 39 40 41 42	43 44 45 46 47 48	49 50
	MOVEA	ARRY	ARRZ	



Calculation Specifications

Example: Array-to-array move. Index factor 2, different length array elements.

			Result Field	
	Operation	Factor 2	Name	Leng
26 27	28 29 30 31 32	33 34 35 36 37 38 39 40 41 42	43 44 45 46 47 48	49 50
	MOVEA	ARRY,4	ARRZ	

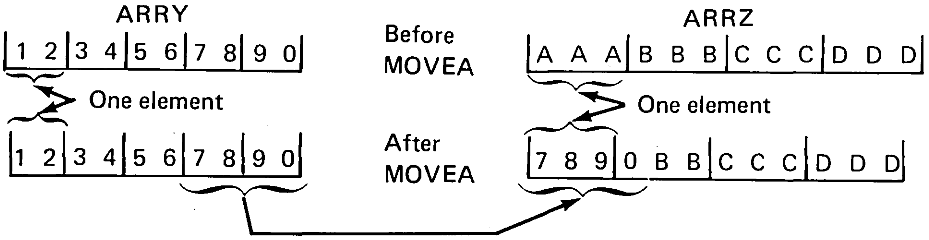
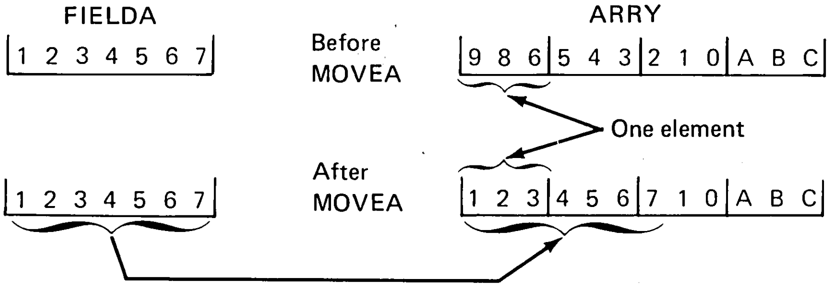


Figure 10-23 (Part 2 of 3). MOVEA Operation

Example: Field-to-array move. No indexing on array.

Calculation Specifications

Operation	Factor 2	Result Field Name	Result Field Leng
28 29 30 31 32	33 34 35 36 37 38 39 40 41 42	43 44 45 46 47 48	49 50
MOVEA	FIELDA	ARRY	



Example: Array-to-field move. Variable indexing.

Calculation Specifications

Operation	Factor 2	Result Field Name	Result Field Leng
28 29 30 31 32	33 34 35 36 37 38 39 40 41 42	43 44 45 46 47 48	49 50
MOVEA	ARRX,N	FIELD	

N=3

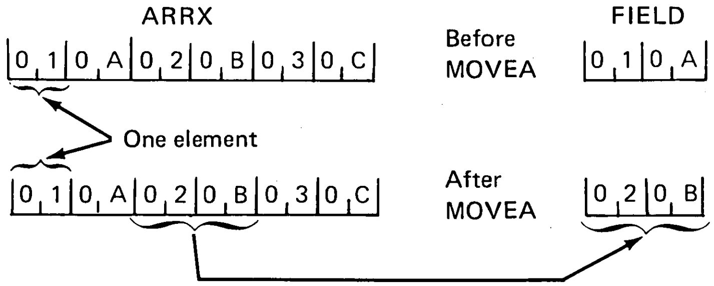


Figure 10-23 (Part 3 of 3). MOVEA Operation

MOVEL (MOVE LEFT)

Indicators				Result	Resulting Indicators		
7-8	9-17	Factor 1	Factor 2	Field	54-55	56-57	58-59
Opt	Opt	Blank	Required	Required	Blank	Blank	Blank

The MOVEL operation transfers characters from factor 2 to the leftmost positions in the result field. Moving begins with the leftmost character in factor 2.

When a numeric field is moved into an alphameric field, both digit and zone portions of the rightmost character are transferred if that character is to be moved.

A summary of the rules for MOVEL operations for three conditions based on field lengths is as follows:

1. Factor 2 is the same length as the result field:
   a. If factor 2 and the result field are numeric, the sign is moved with the rightmost position.
   b. If factor 2 is numeric and the result field is alphameric, the sign is moved with the rightmost position.
   c. If factor 2 is alphameric and the result field is numeric, a minus zone is moved into the rightmost position of the result field if the zone from the rightmost position of factor 2 is a D (minus zone). However, if the zone from the rightmost position of factor 2 is not a D, a positive zone is moved into the rightmost position of the result field. Digit portions are converted to their corresponding numeric characters.
   d. If factor 2 and the result field are alphameric, all characters are moved.

2. Factor 2 is longer than the result field:
   a. If factor 2 and the result field are numeric, the sign from the rightmost position of factor 2 is moved into the rightmost position of the result field.
   b. If factor 2 is numeric and the result field is alphameric, the result field contains only numeric characters.
   c. If factor 2 is alphameric and the result field is numeric, a minus zone is moved into the rightmost position of the result field if the zone from the rightmost position of factor 2 is a D (minus zone). However, if the zone from the rightmost position of factor 2 is not a D, a positive zone is moved into the rightmost position of the result field. Other result field positions contain only numeric characters.
   d. If factor 2 and the result field are alphameric, only the number of characters needed to fill the result field are moved.

3. Factor 2 is shorter than the result field:
   a. If factor 2 is either numeric or alphameric and the result field is numeric, the digit portion of factor 2 replaces the contents of the leftmost positions of the result field. The sign in the rightmost position of the result field is not changed.
   b. If factor 2 is either numeric or alphameric and the result field is alphameric, the characters in factor 2 replace the equivalent number of leftmost positions in the result field. No change is made in the zone of the rightmost position of the result field.

The MOVEL operation is summarized in Figure 10-24.

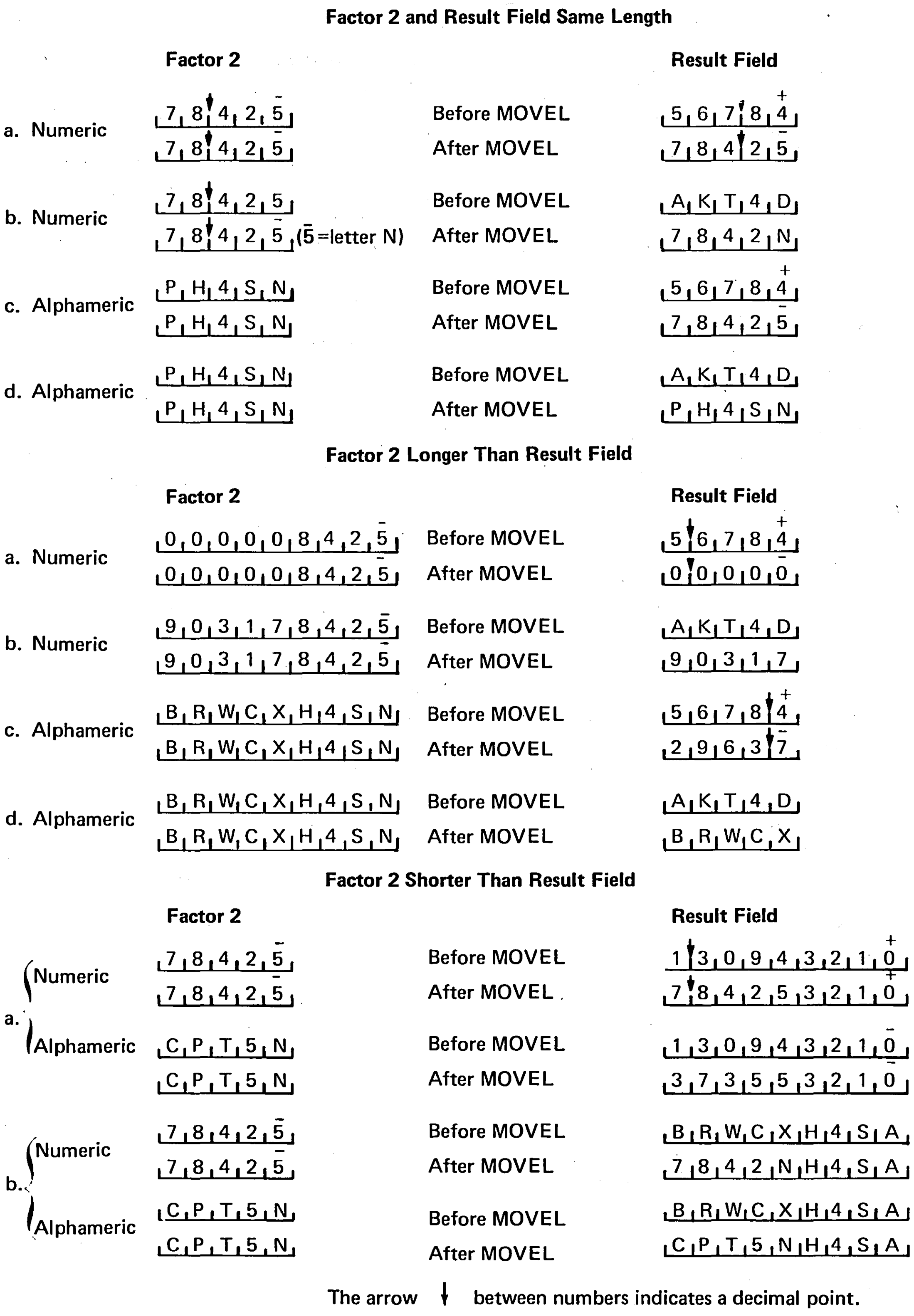


Figure 10-24. MOVEL Operations

MULT (MULTIPLY)

Indicators		Factor 1	Factor 2	Result Field	Resulting Indicators		
7-8	9-17				54-55	56-57	58-59
Opt	Opt	Optional	Required	Required	Optional	Optional	Optional

Factor 1 is multiplied by factor 2, and the product is placed in the result field. Factor 1 and factor 2 are not changed. If factor 1 is not present, the result field is multiplied by factor 2, and the product is placed in the result field.

Be sure that the result field is large enough to hold the product. To determine the minimum length of the result field, use this rule: the length of the result field equals the length of factor 1 plus the length of factor 2.

MVR (MOVE REMAINDER)

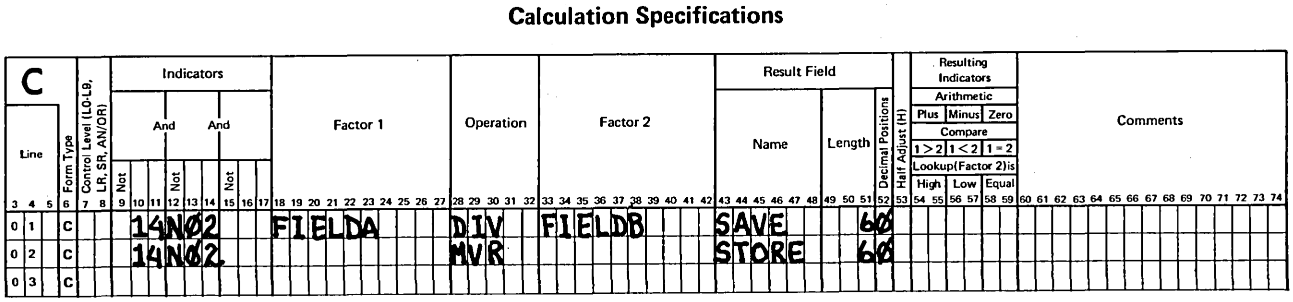
Indicators		Factor 1	Factor 2	Result Field	Resulting Indicators		
7-8	9-17				54-55	56-57	58-59
Opt	Opt	Blank	Blank	Required	Optional	Optional	Optional

The MVR operation moves the remainder from the previous divide operation to a separate field named as the result field. Factor 1 and factor 2 must not be used. This operation must immediately follow the divide operation.

The maximum length of the remainder (including decimal positions) is 15. The number of significant decimal positions is the greater of:

- The number of decimal positions in factor 1 of the previous divide operation.
- The sum of the decimal positions in factor 2 and the result field of the previous divide operation.

The maximum whole number positions in the remainder is equal to the whole number positions in factor 2 of the previous divide operation. Figure 10-25 shows the specification for a move remainder operation.



Calculation Specifications

Line (3-5)	Form Type (6)	Control Level (LO-L9, LR, SR, AN/OR) (7-8)	Indicators (9-17)	Factor 1 (18-27)	Operation (28-32)	Factor 2 (33-42)	Result Field Name (43-48)	Length (49-51)	Decimal Positions (52)	Half Adjust (H) (53)	Resulting Indicators (54-59)	Comments (60-74)
01	C		14NØ2	FIELDA	DIV	FIELDB	SAVE	6	Ø			
02	C		14NØ2		MVR		STORE	6	Ø			
03	C											

Figure 10-25. Move Remainder Operation

NEXT (NEXT)

Indicators				Result	Resulting Indicators		
7-8	9-17	Factor 1	Factor 2	Field	54-55	56-57	58-59
Opt	Opt	Required	Required	Blank	Blank	Optional	Blank

The NEXT operation code forces the next input to the program to come from the device specified in factor 1. If NEXT is specified more than once between input operations, only the last operation is executed. The NEXT operation code can be used only for a WORKSTN file.

Note: For WORKSTN files, a device can be either a display station or an SSP-ICF session.

To use this operation, enter NEXT in columns 28 through 32. In factor 1, enter the name of a 2-character field that contains the device identification or a 2-character alphameric literal that is the device identification. In factor 2, enter the name of the WORKSTN file for which the operation is requested.

An indicator can be specified in columns 56 and 57. This indicator is set on if an exception/error occurs on the NEXT operation. If the INFSR subroutine is specified and columns 56 and 57 do not contain an indicator, the subroutine automatically receives control when an exception/error occurs. (For more information on the INFSR subroutine, see *WORKSTN Exception/Error Handling* in Chapter 13.) If the INFSR subroutine is not specified and columns 56 and 57 do not contain an indicator, the program halts when an exception/error occurs.

For more information on the NEXT operation code, see Chapter 13, *WORKSTN File Considerations and Sample Programs.*

POST (POST)

Indicators				Result	Resulting Indicators		
7-8	9-17	Factor 1	Factor 2	Field	54-55	56-57	58-59
Opt	Opt	Required	Blank	Required	Blank	Optional	Blank

The POST operation allows the programmer to retrieve status information for a specified display station that is attached to a WORKSTN file. The status information (for the POST operation, the display screen size–960 or 1920 characters) is placed in the INFDS data structure that was specified in the result field. The program must contain the INFDS data structure for the WORKSTN file to use POST.

Factor 1 must contain a variable or an alphameric literal that identifies the display station whose status is being requested. The result field contains the name of the INFDS data structure in which this information is to be posted. Columns 56 and 57 can specify an indicator that is set on if an error occurs on the POST operation. An error occurs if the specified work station ID is not attached to the file for which the INFDS data structure is specified.

If columns 56 and 57 do not specify an indicator but the program contains the INFSR subroutine, the subroutine automatically receives control when an error occurs. If the INFSR subroutine is not present and columns 56 and 57 do not contain an indicator, the program halts when an exception/error occurs. The display station must be attached to the WORKSTN file. If it is not attached, the device will be *not found* and an error will occur on POST. (For more information on the INFSR subroutine, see *WORKSTN Exception/Error Handling* in Chapter 13.)

Columns 33 through 42, 49 through 55, and 58 and 59 must be blank for a POST operation.

READ (READ A RECORD)

Indicators				Result	Resulting Indicators		
7-8	9-17	Factor 1	Factor 2	Field	54-55	56-57	58-59
Opt	Opt	Blank	Required	Blank	Blank	Optional	Optional

The READ operation calls for immediate input from a demand file during the calculation phase of the program cycle. This operation differs from the FORCE operation because FORCE calls for certain input on the next program cycle, not the present one.

The operation code READ must appear in columns 28 through 32. Factor 2 contains the name of the file from which a record should be read immediately. An indicator can be used in columns 58 and 59. This indicator turns on when an end-of-file condition is reached for the demand file or for each READ operation after an end-of-file condition is reached. If columns 58 and 59 are blank, a halt occurs on an end-of-file condition and on subsequent READ operations after the end-of-file condition is reached. Indicators can be specified in columns 7 through 17.

An indicator can be specified in columns 56 and 57 if the READ operation is issued to a WORKSTN file. This indicator is set on if an exception condition occurs (that is, the operator pressed one of the function control keys: Roll Up, Roll Down, Clear, Print, Record Backspace, or Help) or if an input error occurs. If columns 56 and 57 do not contain an indicator and either of these conditions occurs, the program halts unless the INFSR subroutine is specified. If the INFSR subroutine is specified, the subroutine automatically receives control and an exception/error occurs. (For more information on the INFSR subroutine, see *WORKSTN Exception/Error Handling* in Chapter 13.)

The following columns must remain blank for a READ operation: columns 18 through 27 (factor 1), columns 43 through 48 (result field), columns 49 through 51 (field length), column 52 (decimal positions), column 53 (half adjust), and columns 54 and 55 (resulting indicators).

The following files can appear as factor 2 in a READ operation:

- Sequential disk files processed consecutively and specified as input or update files.
- Indexed disk files processed sequentially by key and specified as input or update files.
- Indexed disk files processed sequentially by limits and specified as input or update files.
- Direct files processed consecutively as input or update files.
- WORKSTN files.
- SPECIAL files.

REL (RELEASE)

Indicators				Result	Resulting Indicators		
7-8	9-17	Factor 1	Factor 2	Field	54-55	56-57	58-59
Opt	Opt	Required	Required	Blank	Blank	Optional	Blank

The REL operation releases the device specified in factor 1 from the program. Either a requesting or nonrequesting device can be released with the REL operation code. The specified device is released when the REL operation is encountered during the calculations unless the device is the requestor of a single requestor program. If the device specified in factor 1 is the requestor of a single requestor program, the device is released at end of job, not when the operation code is encountered in the calculations. (If the device is a display station, it is no longer available to the program, but it is available for system log messages.)

If an exception/error occurs on the attempt to release the device, the indicator specified in columns 56 and 57 is set on. If no indicator is specified, the program halts unless the INFSR subroutine is specified in the program. If the INFSR subroutine is specified, the INFSR subroutine automatically receives control when an exception/error occurs and no indicator is specified in columns 56 and 57.

When all devices are released from a primary WORKSTN file, the file goes to end of file and RPG II sets on the LR indicator. If the program containing the primary file is an NEP, the system operator must enter the STOP SYSTEM command before the WORKSTN file will go to end of file.

When all device are released from a demand WORKSTN file and the program is not an NEP, the first READ operation after the last REL operation causes the READ end-of-file indicator to be set on (columns 58 and 59). The programmer can then set on the LR indicator unless the LR indicator was specified as the end-of-file indicator. If the program containing the demand WORKSTN file is an NEP, the end-of-file indicator is set on when the system operator enters the STOP SYSTEM command. The programmer can then set on the LR indicator unless the LR indicator was specified as the end-of-file indicator.

If RESTORE-NO is specified on the WORKSTN OCL statement, a display format from the program may appear on the screen after the display station has been released. If RESTORE-YES is specified on the WORKSTN OCL statement, the COMMAND display appears on the screen immediately when the display station is released.

For more information on the ACQ and REL operation codes, see Chapter 13, *WORKSTN File Considerations and Sample Programs.*

RLABL (RPG II LABEL)

Indicators				Result	Resulting Indicators		
7-8	9-17	Factor 1	Factor 2	Field	54-55	56-57	58-59
Blank	Blank	Blank	Blank	Required	Blank	Blank	Blank

The RLABL operation allows the subroutine specified in an EXIT operation to reference a field, data structure, table, array, or indicator defined in the RPG II program. RLABL operations must be specified immediately after the EXIT operation that refers to the subroutine using the field, data structure, table, array, or indicator in the RLABL specification (see Figure 10-26). All external subroutines become part of the root segment and are not to be put into overlays.

The rules for use of RLABL on the calculation specifications are as follows:

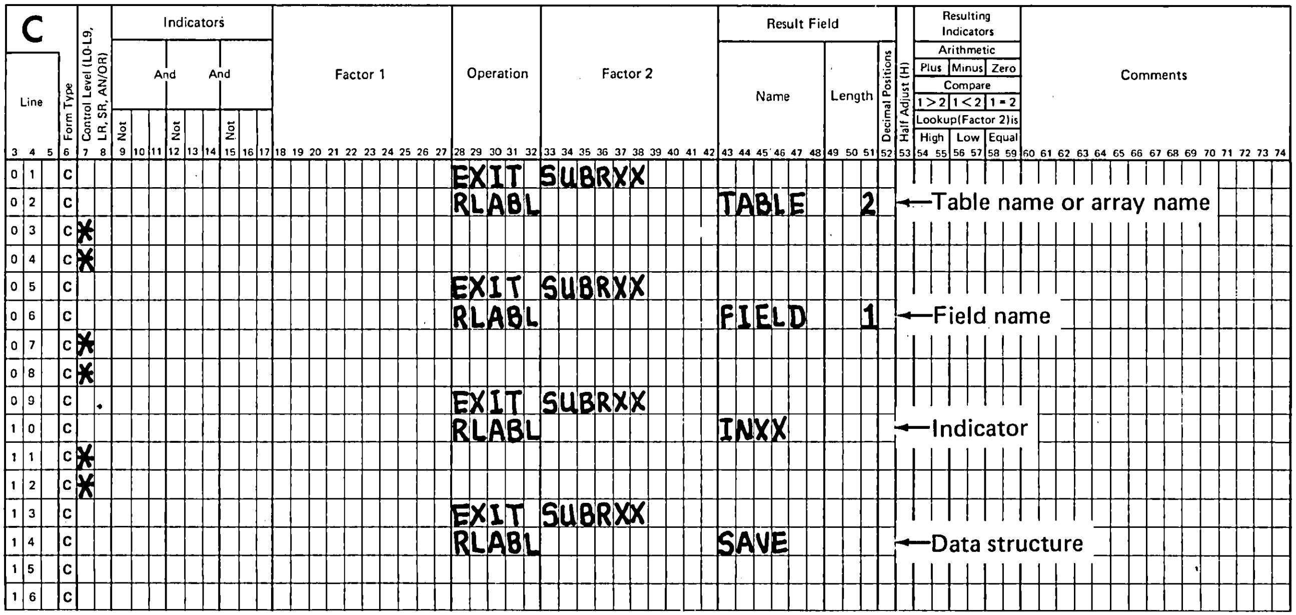
Columns	**Entry**
Operation (28-32)	RLABL
Result field 43-48	Field, data structure, table, or array name, or indicator (INxx, where xx is the indicator)
Field length (49-51)	Length of field (optional)
Decimal positions (52)	Decimal indication (optional)

Only RLABL operations specifying a field, table, or array name can have entries for field length (columns 49 through 51) and decimal positions (column 52).

The following columns must be blank for an RLABL operation: columns 7 and 8 (control level), columns 9 through 17 (indicators), columns 18 through 27 (factor 1), columns 33 through 42 (factor 2), column 53 (half adjust), and columns 54 through 59 (resulting indicators).

A name defined by a TAG, BEGSR, or ENDSR specification cannot be used in an RLABL specification.

Calculation Specifications



Line	Form Type	Control Level (L0-L9, LR, SR, AN/OR)	Indicators	Factor 1	Operation	Factor 2	Result Field Name	Length	Decimal Positions	Half Adjust (H)	Resulting Indicators	Comments
01	C				EXIT	SUBRXX						
02	C				RLABL		TABLE	2				←Table name or array name
03	C	*										
04	C	*										
05	C				EXIT	SUBRXX						
06	C				RLABL		FIELD	1				←Field name
07	C	*										
08	C	*										
09	C				EXIT	SUBRXX						
10	C				RLABL		INXX					←Indicator
11	C	*										
12	C	*										
13	C				EXIT	SUBRXX						
14	C				RLABL		SAVE					←Data structure
15	C											
16	C											

Figure 10-26. RPG II Coding for RLABL Field Entries

Referencing an Indicator

When an indicator is specified in an RLABL operation, use the form INxx as the result field, where xx is the indicator to be transferred to the subroutine. For example, if the MR indicator is to be transferred to a subroutine, specify INMR as the result field for the RLABL operation.

When an indicator is specified in the RLABL operation, the RPG II compiler generates the following parameters and passes them to the assembler subroutine:

```
B     SUBRxx
DC    XL1 '00'
DC    XL1 'Mask for the indicator'
DC    XL1 'Displacement to the indicator from
      XR1'
```

Referencing a Field

When a field name is specified in the RLABL operation, the RPG II compiler generates the following parameters and passes them to the assembler subroutine:

```
B     SUBRxx
DC    IL1 'Field length-1'
DC    AL2 (right address of field)
```

Referencing a Data Structure

When a data structure is specified in the RLABL operation, the RPG II compiler generates the following parameters and passes them to the assembler subroutine:

```
B     SUBRxx
DC    XL3 'FFFFFF'
DC    IL2 'Data structure length-1'
DC    AL2 (leftmost address of data structure)
```

Referencing a Table or Array

The subroutine can refer to a table or array defined in the RPG II program by using the control field created for that table or array. This control field is called the DTT (define the table), and one is created for each table or array built by the RPG II program. The control field is in the following format:

Bytes	**Meaning**
0-1	Address of rightmost byte of the first entry
2-3	Address of rightmost byte of the last entry
4-5	Initialized to the address of the rightmost byte of first entry; used at object time for address of right byte of the last looked-up entry
6-7	Length of an entry
8-13	Array name (arrays only)

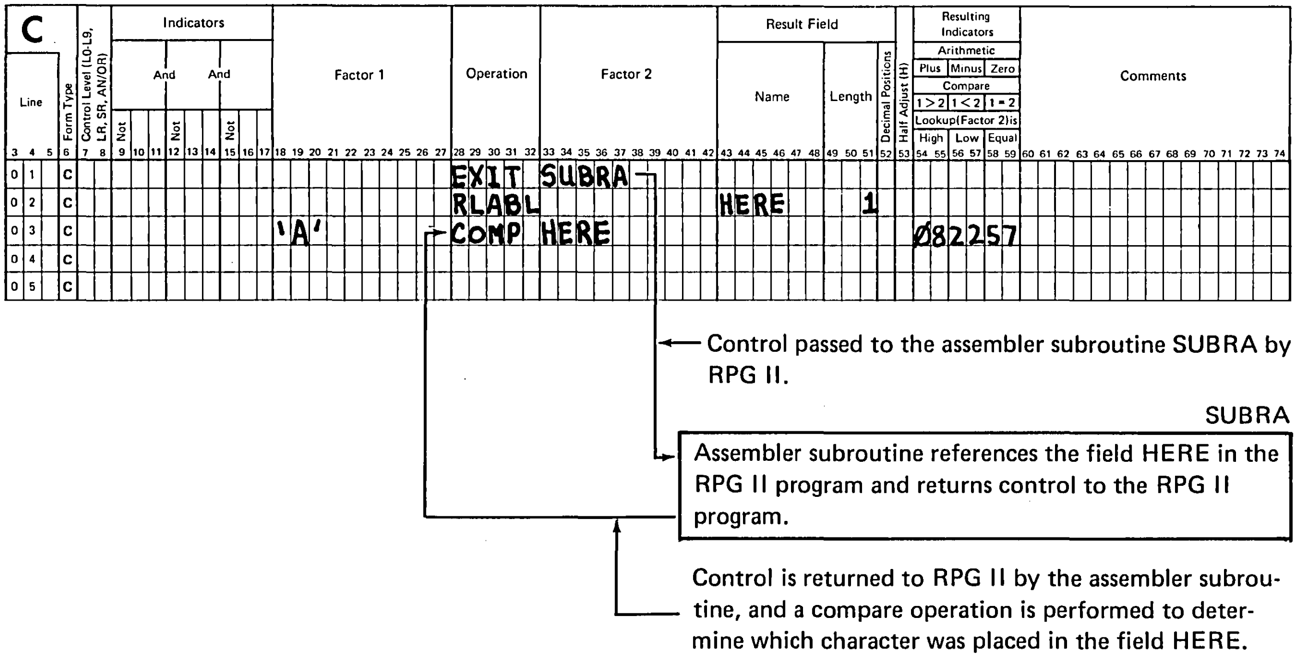
The subroutine can obtain the data retrieved from the preceding LOKUP operation by using the address in bytes 4 and 5. To access the table or array itself, the address in bytes 0 and 1 must be used. Data the subroutine uses is left unpacked.

When a table or array is specified in the RLABL operation, the RPG II compiler generates the following parameters and passes them to the assembler subroutine:

B	SUBRxx
DC	IL1 'Entry length-1'
DC	AL2 (leftmost address of the DTT)

See Figures 10-27 and 10-28 for examples of RPG II linkage specifications.

Calculation Specifications



Line	Form Type	Control Level (L0-L9, LR, SR, AN/OR)	Indicators	Factor 1	Operation	Factor 2	Result Field Name	Length	Decimal Positions	Half Adjust (H)	Resulting Indicators High	Low	Equal	Comments
01	C				EXIT	SUBRA								
02	C				RLABL		HERE	1						
03	C			'A'	COMP	HERE					Ø8	22	57	
04	C													
05	C													

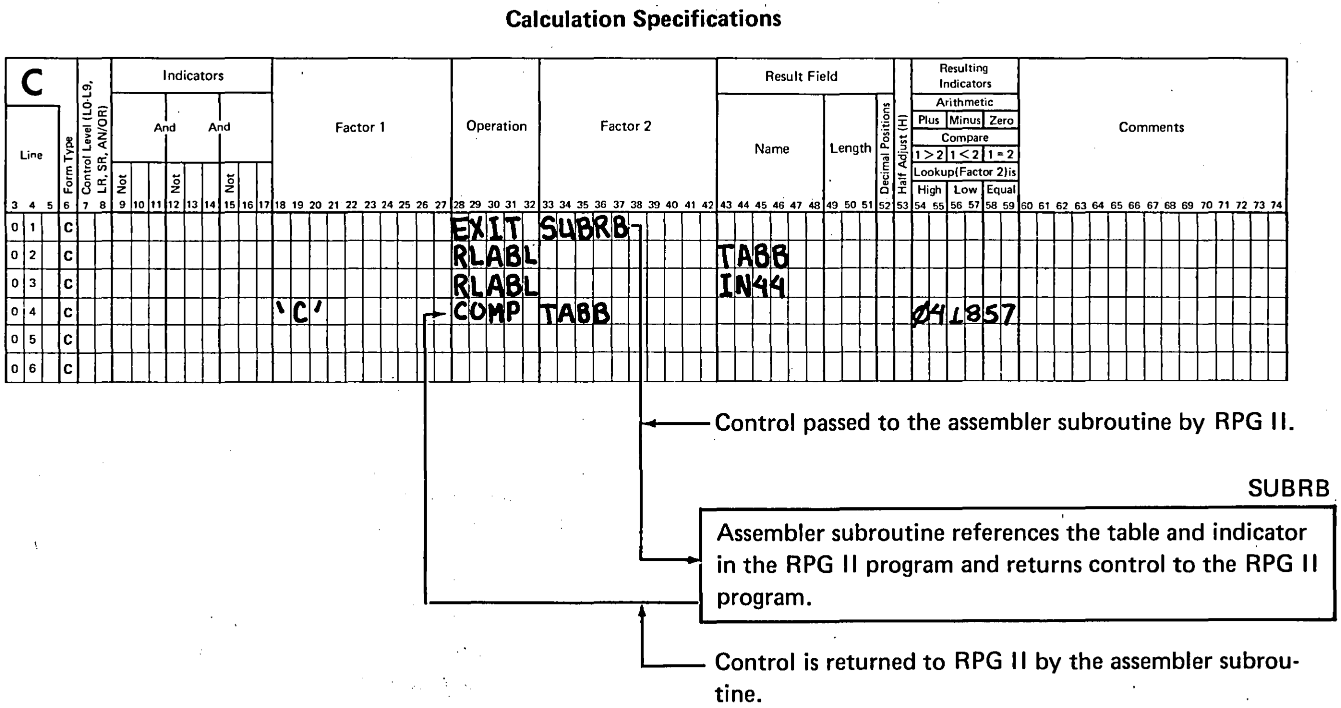
Control passed to the assembler subroutine SUBRA by RPG II.

SUBRA

Assembler subroutine references the field HERE in the RPG II program and returns control to the RPG II program.

Control is returned to RPG II by the assembler subroutine, and a compare operation is performed to determine which character was placed in the field HERE.

Figure 10-27. RPG II Linkage to an Assembler Language Subroutine



Calculation Specifications

Line	Form Type	Control Level	Indicators	Factor 1	Operation	Factor 2	Result Field Name	Length	Decimal Positions	Half Adjust (H)	Resulting Indicators High / Plus / 1>2	Low / Minus / 1<2	Equal / Zero / 1=2	Comments
01	C				EXIT	SUBRB								
02	C				RLABL		TABB							
03	C				RLABL		IN44							
04	C			'C'	COMP	TABB					Ø4	18	57	
05	C													
06	C													

Control passed to the assembler subroutine by RPG II.

SUBRB

Assembler subroutine references the table and indicator in the RPG II program and returns control to the RPG II program.

Control is returned to RPG II by the assembler subroutine.

The subroutine refers to both RLABL entries. It first tests the indicator. If the indicator is off, control is returned to the RPG II program. If the indicator is on, a character C is moved into the last looked-up entry in the table, TABB. When control is returned to the RPG II program, a compare operation is performed to see whether or not the subroutine placed a C in TABB.

Figure 10-28. RPG II Linkage to an Assembler Language Subroutine

Considerations for the Assembler Programmer

To write an assembler subroutine that is linked to an RPG II program, the assembler programmer must be aware of the following:

- The name of the subroutine must be the same as the name specified in factor 2 of the RPG II EXIT operation.

- Upon entry to the assembler language subroutine, the address recall register (ARR) contains a pointer to the parameters that represent the RPG II fields to be referenced by the assembler subroutine. The return point to the RPG II program is the first byte after the parameters.

- If the subroutine makes use of registers 1 and 2, the contents of these registers must be stored upon entry to, and restored before exit from, the subroutine.

Note: The user-written subroutines should be placed in #LIBRARY or #RPGLIB (the library the RPG compiler resides in), not in the same library as the RPG II source program.

Message Retrieve Subroutine (SUBR23)

The message retrieve subroutine (SUBR23) allows you to retrieve messages from a user message member. After the message has been retrieved, it can be modified and written to an output file.

Linkage to SUBR23 is by the EXIT operation code, and input parameters are passed to SUBR23 by RLABL operation codes. To use SUBR23, specify EXIT in columns 28 to 31 and SUBR23 in columns 33 to 38. Four RLABL operation codes must be specified after the EXIT operation with the following result field entries:

Result Field	Description
MIC number	Name of a four-digit numeric field that contains the MIC (message identification code) of the text to be retrieved.
Text area	Name of the alphameric field or data structure into which the message text is read. The maximum length of a level-1 message is 75 characters and of a level-2 message is 225 characters.
Level	Name of a one-digit numeric field that designates the user message member level. A value of 1 in this field indicates a message level of 1; a value of 2 indicates a message level of 2.
Rcode	Name of a one-digit numeric field that contains the return codes. The return code and their meanings are as follows:

Return Code	Meaning
0	Message was successfully retrieved with no truncation.
1	Message was successfully retrieved; however, it was truncated because the length of the text area was less than the message length.
2	Message was not found.
3	The field indicating the message was invalid.
4	An invalid MIC value was diagnosed.
5	Message member was not found or message text length exceeds the level 1 maximum.

The text area, which is specified by the second RLABL operation, is blanked before each attempt to retrieve a message; therefore, a blank text area is returned to the user program when the return code value is 2 or greater. A total of 225 positions in the text area is blanked unless the text area is less than 225 characters in length.

SET (SET)

Indicators				Result	Resulting Indicators		
7-8	9-17	Factor 1	Factor 2	Field	54-55	56-57	58-59
Opt	Opt	Optional	Optional	Blank	Optional	Optional	Optional

The SET operation can be used only with input files assigned to the device KEYBORD, which is specified in columns 40 through 46 of the file description specifications. All SET operations are directed to the display station that loaded the program or to the display station assigned to the program by the WORKSTN OCL statement.

The SET operation allows any or a combination of the following:

- Command keys to be pressed
- The field, literal, or table or array element specified in factor 1 to be displayed on the display screen
- User messages (from USER1 message member) 0001 to 0099 to be displayed when numbers 01 to 99 respectively are specified in the nn portion of the SETnn and KEYnn operation codes
- The buffer for a CONSOLE file to be blanked if ERASE is specified in the result field of the SET operation

Allowing Command Keys to be Pressed

The SET operation allows you to specify command keys that the operator is allowed to press at this point in the program. When the operator presses a command key, the corresponding command key indicator turns on. These command key indicators can be used to condition subsequent calculation or output operations. Command key indicators remain on until they are used again in a SET operation or until they are turned off by the SETOF operation.

When the program is at a particular specification line, you can give the operator the option of pressing one to three command keys. For each command key to be pressed, the operator first presses the Cmd key and then presses the digit key corresponding to the command key indicator (KA through KN, or KP through KY). After all command key responses have been entered, the operator presses an entry function key.

A total of 24 command keys are designated for the top row of the keyboard. In the lowercase position, key 1 corresponds to command key indicator KA, key 2 to KB ... -(minus) to KK and = (equal) to KL. In the uppercase position, key | corresponds to command key indicator KM, @ to KN ... and + to KY.

If command keys were erroneously pressed and an entry function key has not been pressed, the operator can reset all the command keys by pressing the Cmd key followed by the character backspace (Clear) key while holding down the Shift key. The operator can then rekey all the correct keys. If any invalid command keys and an entry function key were pressed, an error message is issued.

If no command keys are to be pressed, the operator responds to the SET operation by pressing only an entry function key; thus causing the indicators to be turned off. This is called a *null response.* Using this null response in your programs is not recommended because of the possibility of an accidental null response. For example, if the operator neglects to press the Cmd key before pressing the appropriate command key, a null response occurs.

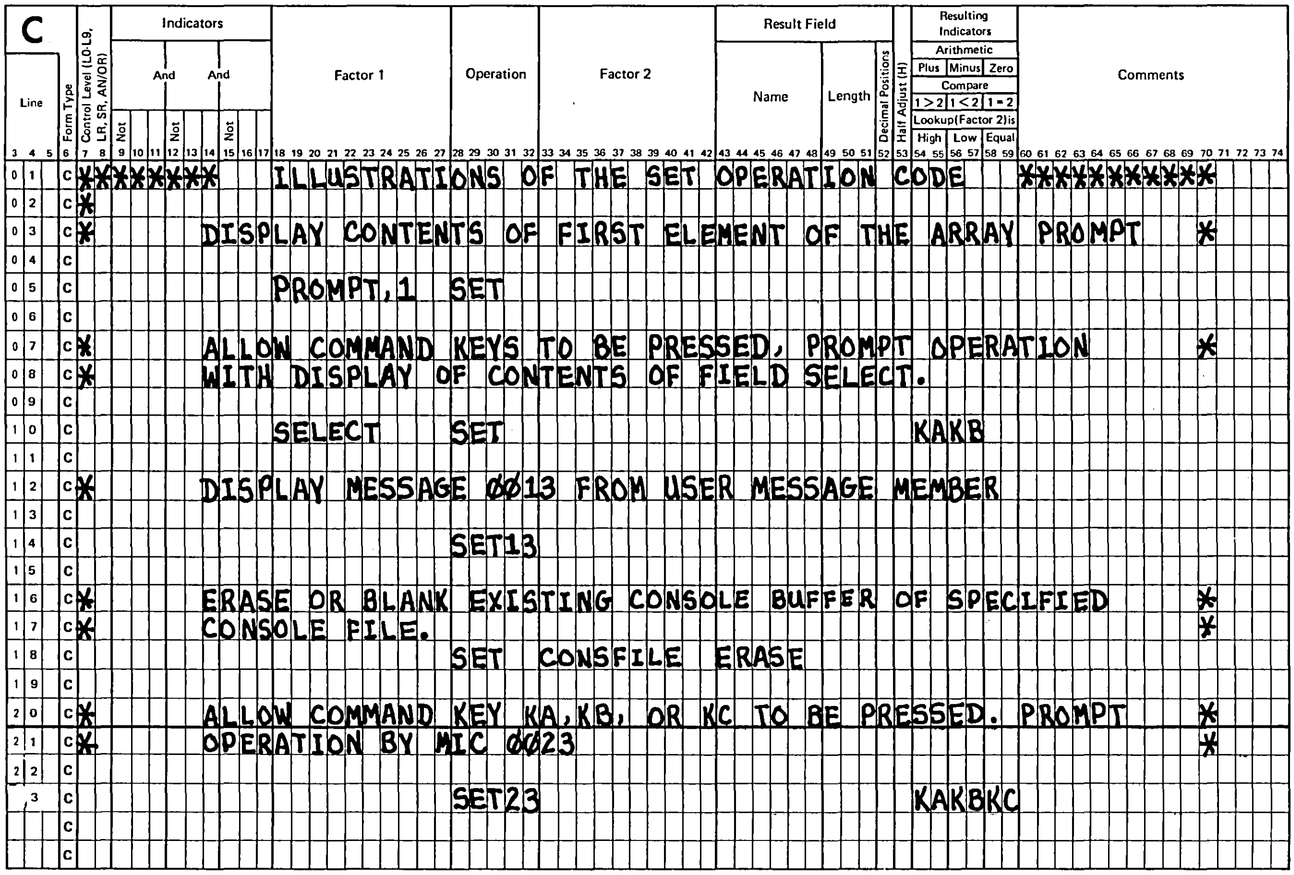
Specifications for SET Operations

The specifications required for a SET operation vary depending upon which function, or combination of functions, is to be performed. Figure 10-29 shows a summary of these specifications, and Figure 10-30 shows the possible combination of these SET functions.

Columns 7-8: Enter any valid conditioning indicator. However, leave these columns blank if the SET operation is not a part of a subroutine or if it is to be performed only at detail time.

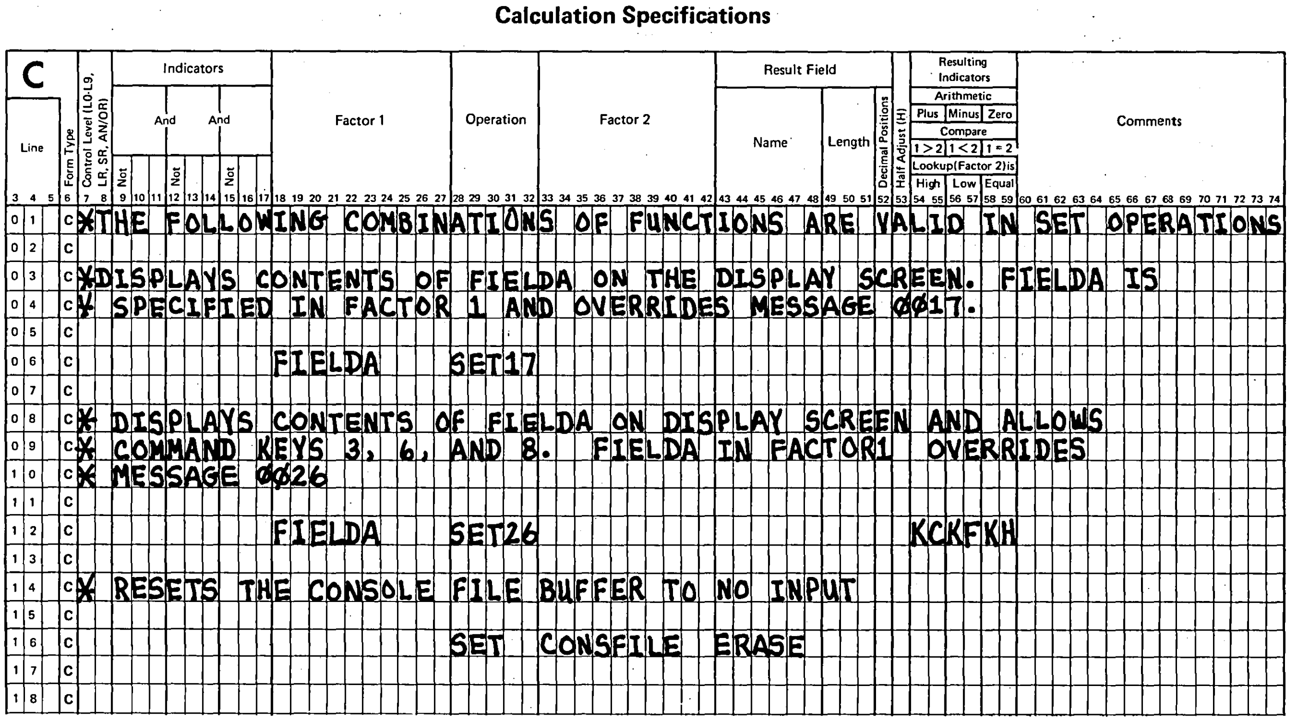
Columns 9-17: Enter any valid conditioning indicators for any SET operation. However, leave these columns blank if the SET operation is to be performed on every program cycle.

Calculation Specifications



```
  01 C********   ILLUSTRATIONS OF THE SET OPERATION CODE   ***********
  02 C*
  03 C*      DISPLAY CONTENTS OF FIRST ELEMENT OF THE ARRAY PROMPT   *
  04 C
  05 C           PROMPT,1  SET
  06 C
  07 C*      ALLOW COMMAND KEYS TO BE PRESSED, PROMPT OPERATION      *
  08 C*      WITH DISPLAY OF CONTENTS OF FIELD SELECT.
  09 C
  10 C           SELECT    SET                       KAKB
  11 C
  12 C*      DISPLAY MESSAGE ØØ13 FROM USER MESSAGE MEMBER
  13 C
  14 C                     SET13
  15 C
  16 C*      ERASE OR BLANK EXISTING CONSOLE BUFFER OF SPECIFIED     *
  17 C*      CONSOLE FILE.                                           *
  18 C                     SET  CONSFILE  ERASE
  19 C
  20 C*      ALLOW COMMAND KEY KA,KB, OR KC TO BE PRESSED. PROMPT    *
  21 C*      OPERATION BY MIC ØØ23                                   *
  22 C
  23 C                     SET23                     KAKBKC
     C
     C
```

Figure 10-29. Summary of Calculation Specifications for SET Operations



Calculation Specifications

Line	Form Type	Control Level (L0-L9, LR, SR, AN/OR)	Indicators	Factor 1	Operation	Factor 2	Result Field Name	Length	Decimal Positions	Half Adjust (H)	Resulting Indicators (Plus/Minus/Zero; High/Low/Equal)	Comments
01	C	*THE FOLLOWING COMBINATIONS OF FUNCTIONS ARE VALID IN SET OPERATIONS										
02	C											
03	C	*DISPLAYS CONTENTS OF FIELDA ON THE DISPLAY SCREEN. FIELDA IS										
04	C	* SPECIFIED IN FACTOR 1 AND OVERRIDES MESSAGE ØØ17.										
05	C											
06	C			FIELDA	SET17							
07	C											
08	C	* DISPLAYS CONTENTS OF FIELDA ON DISPLAY SCREEN AND ALLOWS										
09	C	* COMMAND KEYS 3, 6, AND 8. FIELDA IN FACTOR1 OVERRIDES										
10	C	* MESSAGE ØØ26										
11	C											
12	C			FIELDA	SET26						KCKFKH	
13	C											
14	C	* RESETS THE CONSOLE FILE BUFFER TO NO INPUT										
15	C											
16	C				SET	CONSFILE	ERASE					
17	C											
18	C											

Figure 10-30. Possible Combinations of Functions Performed by SET Operation

Columns 18-27: Enter the constant, literal, field name, or table or array element to be displayed on the display screen.

Columns 28-30: Enter the operation code SET.

Columns 31-32: Enter the message identification code (MIC) corresponding to the message in the user message member file that is to be displayed on the display screen. This message prompts the operator to perform a SET operation. Valid entries are 01 to 99. An entry is required when command keys are specified in columns 54 through 59 and columns 18 through 27 are blank. If no user message member is specified prior to execution with a MEMBER OCL statement or there is no message for the specified MIC, the system prompt 'nn-MESSAGE INDICATOR' is displayed, where nn is the contents of columns 31 and 32.

Columns 33-42: When the ERASE function is specified in columns 43 through 48 for a CONSOLE file, enter the CONSOLE filename in columns 33 through 42. For all other SET operations, leave these columns blank.

Columns 43-48: Enter ERASE in these columns to clear the CONSOLE file specified in columns 33 through 42. For all other SET operations, leave these columns blank.

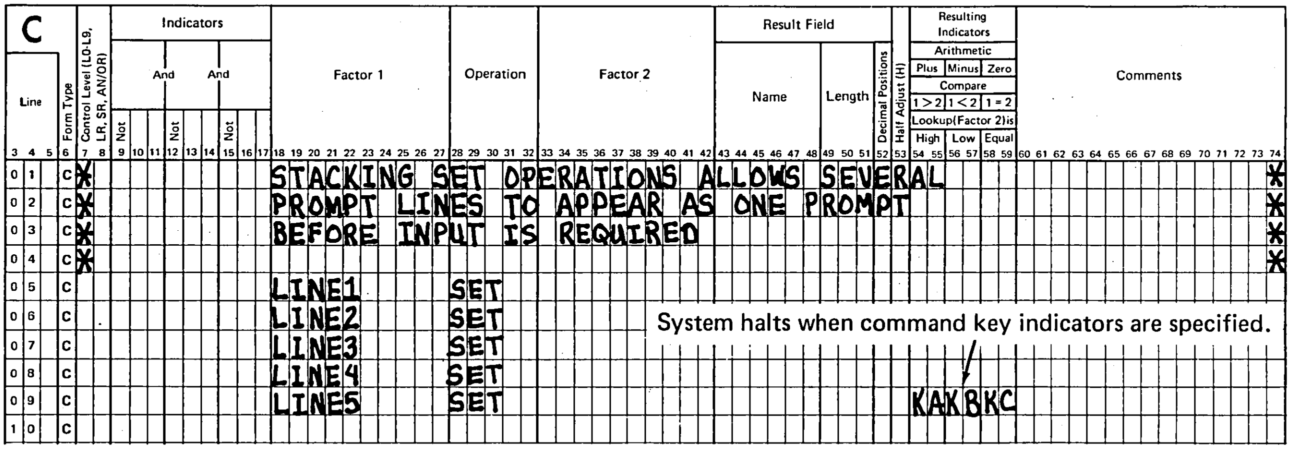
Columns 49-53: Leave these columns blank.

Columns 54-59: Enter the command keys (KA through KN, KP through KY) that the operator is allowed to press when the program is at this specification line. One to three command keys can be specified. If only one or two command keys are specified, they can be entered in any of the three sets of columns. When the operator presses a command key specified in these columns, that command key indicator turns on and remains on until it is used again in a SET operation or until it is turned off by the SETOF operation. A halt occurs if the operator presses a command key other than those specified in columns 54 through 59 of a SET operation.

Either factor 1 or message indicators in columns 31 and 32 must be specified on a SET operation. If both factor 1 and message indicators are present, the message indicators are ignored.

If you stack SET operations with a factor 1 (or MIC) and no command key entries (see Figure 10-31), you can display several lines on the display screen before the system halts for input. You can display up to six lines if the record length is 40 or less. You can display a full screen if the record length is more than 40. The system does not halt until a command key function is encountered or a KEY operation is specified.

Calculation Specifications



```
01  C*          STACKING SET OPERATIONS ALLOWS SEVERAL                 *
02  C*          PROMPT LINES TO APPEAR AS ONE PROMPT                   *
03  C*          BEFORE INPUT IS REQUIRED                               *
04  C*                                                                 *
05  C           LINE1     SET
06  C           LINE2     SET
07  C           LINE3     SET
08  C           LINE4     SET
09  C           LINE5     SET                         KAKBKC
10  C
```

System halts when command key indicators are specified.

C
Line
Form Type
Control Level (L0-L9, LR, SR, AN/OR)
Indicators
And
And
Not
Not
Not
Factor 1
Operation
Factor 2
Result Field
Name
Length
Decimal Positions
Half Adjust (H)
Resulting Indicators
Arithmetic
Plus Minus Zero
Compare
1>2 1<2 1=2
Lookup(Factor 2)is
High Low Equal
Comments

Figure 10-31. Using SET Operation to Display Multiple-Line Prompt

Using SET Operations in Subroutines

Sometimes it is necessary to write a program that has the KEY and SET operations performed at several different points in the program. Instead of writing these KEY operations and related SET operations every time they are needed, you can write them just once in a subroutine. Then, call the subroutine each time it is needed (see *Subroutine Operations* in this chapter for information on specifying and using subroutines).

User Message Member

The System/34 system support program product lets you create your own message members, which are called user message members. These message members can contain prompts or informational messages to be displayed during your RPG II program.

For information on creating message members, see *$MGBLD Utility Program* in the *System Support Reference Manual.* The messages contained in user message members are displayed when you specify the halt indicators (H1 through H9) or the message indicator option of the SET and KEY operation codes. The messages displayed must be formatted so MIC numbers 0001 through 0109 are assigned to the specific function as follows:

MIC	Function
0001-0099	Message to be displayed as specified by the nn portion of the SET and KEY operation codes (SETnn, KEYnn, where nn = 01 to 99).
0100	Message to be displayed at the end of an RPG II cycle when the system is finished processing outstanding halt indicators.
0101-0109	Message to be displayed at the end of an RPG II cycle in which the system has encountered H1 through H9 halt indicators (0101 through 0109 correspond to H1 through H9 respectively).

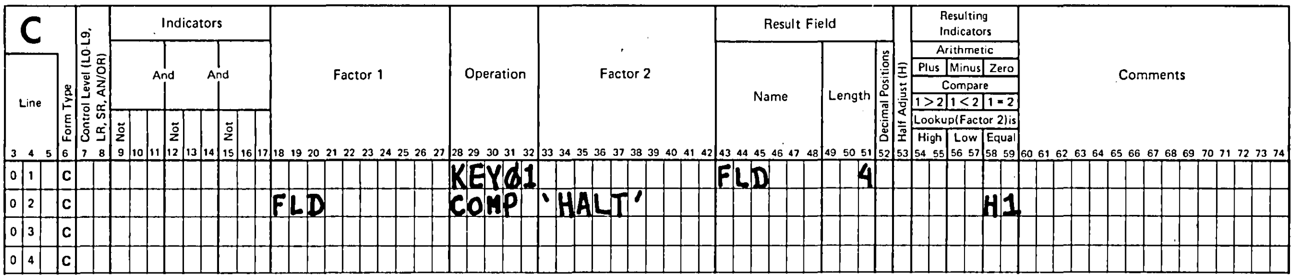
For a message contained in a user message member to be displayed, the message text must exist in an object message member. The message member must be specified in the MEMBER USER1 OCL statement, and the SETnn or KEYnn operation or an H1 to H9 indicator must be used in the program. (See *MEMBER Statement* in the *System Support Reference Manual.*)

Note: The specified user message member remains active until the system processes another MEMBER statement or the display station session is ended (the display station operator signs off). If one of the user message members is active (either USER1 or USER2) and an execution time error is encountered, you will receive a user message rather than the appropriate system message (unless you have copied the system messages into your user message member). For more information on how long a user message member remains active, see *MEMBER Statement* in the *System Support Reference Manual.*

The first level message corresponding to the H1 to H9 halt indicators can be described in more detail with an associated second level message member with the same MIC number. The text of a second level message can be up to 225 characters long. Second level messages can only be specified with the H1 to H9 halt indicators, and a MEMBER USER2 OCL statement must be included at execution time. After the halt indicators (H1-H9) turn on, all calculations and detail output operations are performed for the record before processing stops, and a message is issued. If the halt indicators (H1-H9) are turned on during LR time, the program does not halt processing and continues to completion.

Figure 10-32 shows the coding and OCL statements necessary to issue a message. The source member (MESG1 for this example) must be loaded into a library via the source entry utility or the $MAINT utility. The object member message MESG1 must be created prior to execution. For information on loading the message source member, see *Sign-On Procedures* in the *SEU Reference Manual*; for information on creating a message load member, see the *System Support Reference Manual.*

Calculation Specifications



Line (3-5)	Form Type (6)	Control Level (L0-L9, LR, SR, AN/OR) (7-8)	Indicators (9-17)	Factor 1 (18-27)	Operation (28-32)	Factor 2 (33-42)	Result Field Name (43-48)	Result Field Length (49-51)	Decimal Positions (52)	Half Adjust (H) (53)	Resulting Indicators: Plus / 1>2 / High (54-55)	Minus / 1<2 / Low (56-57)	Zero / 1=2 / Equal (58-59)	Comments (60-74)
01	C				KEYØ1		FLD	4						
02	C			FLD	COMP	'HALT'							H1	
03	C													
04	C													

Messages are received from the system message member because no overriding MEMBER statement was specified:

// LOAD USER
// RUN

01-MESSAGE INDICATOR is the text issued as the prompt for the KEY operation. If HALT is keyed, the indicator H1 turns on and the RPG II operator message 0101 RPG II INDICATOR H1 IS ON is displayed. If the 0 option is selected, message 0100 ALL HALT INDICATORS PRE—VIOUSLY DISPLAYED is issued.

Messages are retrieved from the user-created message member MESG1 specified in the MEMBER statement:

// LOAD USER
// MEMBER USER1-MESG1
// RUN

The prompt KEY HALT TO TERMINATE PROGRAM is the text issued for the KEY01 operation. This is the contents of the object member loaded into the user message member specified by the MEMBER statement. Subsequently, the USER 0101 message text that is issued when the literal HALT is keyed is HALT HAS BEEN ENTERED WITH KEY OP.

The second message issued, USER 0100, has not been defined in the user message member; thus it cannot be retrieved, and the message MESSAGE NOT RETRIEVED (SEE MSG MANUAL) is issued.

Figure 10-32. Issuing a Message

Special Combinations of the SET and KEY Operations

Normally, the operator must press an entry function key after each KEY operation or after command keys specified in a SET operation are pressed. However, it is possible to combine these operations so that the operator can press command keys (specified in columns 54 through 59 of a SET operation), key a field (specified in a KEY operation), and only press an entry function key once. This is only possible if:

- The SET operation immediately precedes the KEY operation.
- The SET and KEY operations are conditioned by the same indicators (columns 7 through 17). Indicators for both operations must be specified in the same order.
- The SET and KEY operations contain the same message indicators (see Figure 10-33). If factor 1 is used to display messages, then columns 31 and 32 can be blank to satisfy this requirement.
- Factor 1 for the SET and KEY operations can be the same, different, or missing from one operation. If factor 1 is specified for both the SET and KEY operations, the contents of each factor 1 are displayed.

If the data field is numeric, the operator must first press the specified command key, key the field, and then press the Field Exit, Field +, or Field - key. The Enter/Rec Adv key is not allowed as an entry function key for a numeric field. For an alphameric field, the operator must perform the same sequence of steps if the Field Exit or Field + key is pressed. However, if the Enter/Rec Adv key is used, the operator can press the command key and then key the field, or key the field and then press the command key before pressing the Enter/Rec Adv key.

SETLL (SET LOWER LIMIT)

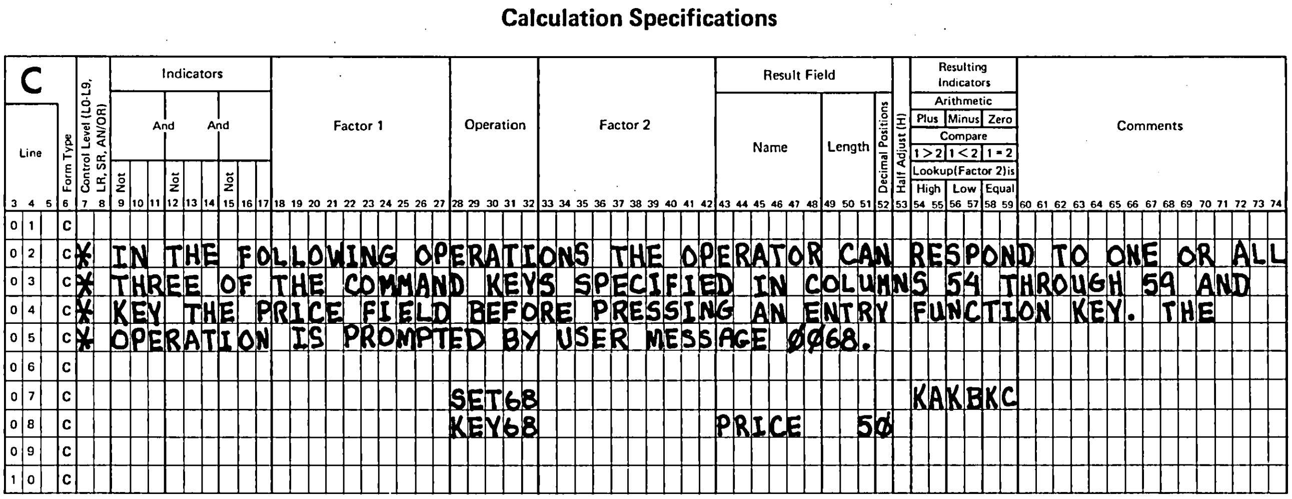
Indicators				Result	Resulting Indicators		
7-8	9-17	Factor 1	Factor 2	Field	54-55	56-57	58-59
Opt	Opt	Required	Required	Blank	Blank	Blank	Blank

The SETLL operation allows the lower limits for an indexed demand file being processed sequentially within limits to be set during calculations.

Factor 1 must contain a field name or literal representing the value of the lower limit being set. The length of the field or literal must be equal to the length of the key specified on the filename in factor 2.

Factor 2 must contain the name of the file for which the lower limit is to be set. If a read is performed to the file prior to a SETLL operation, the record with the lowest key in the file is fetched. Figure 10-34 shows an example of SETLL coding.

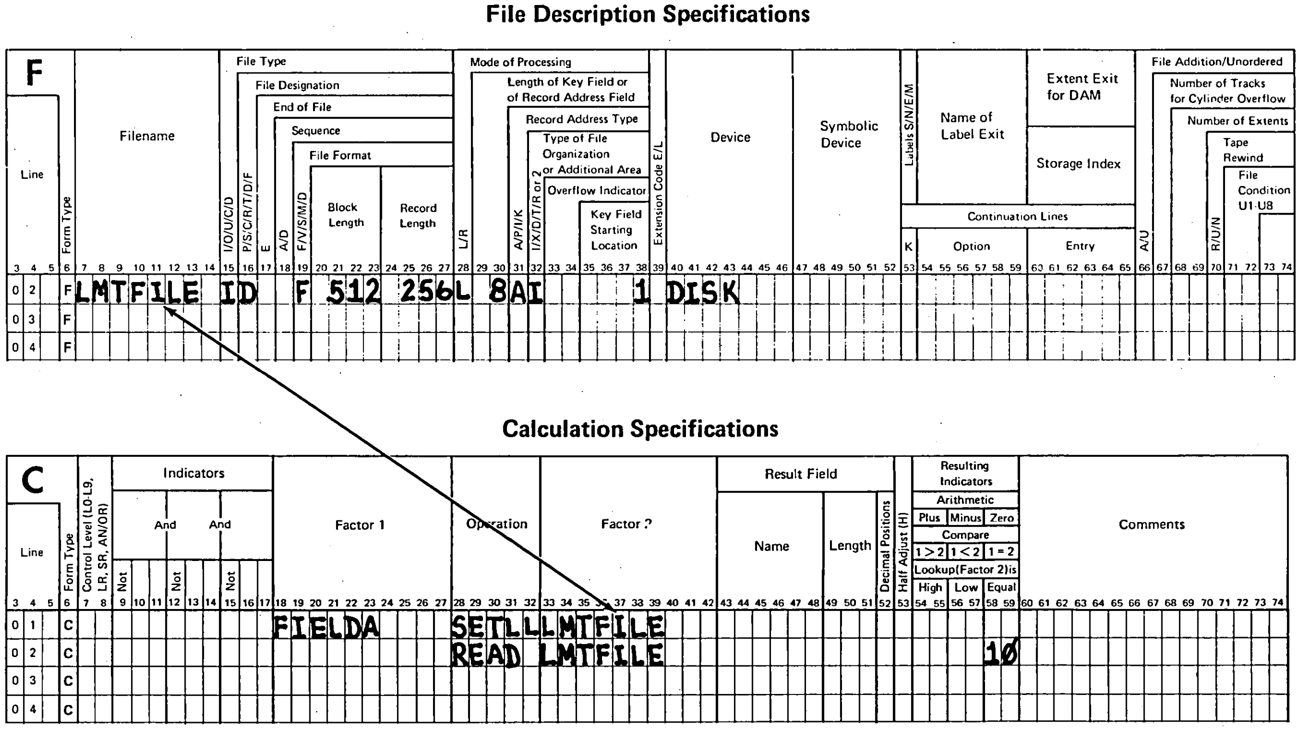
Note: When a lower limit is specified by SETLL, the end-of-file indicator specified for the READ operation to the file being processed is not set off by the RPG II cycle.



Calculation Specifications

```
01 C
02 C* IN THE FOLLOWING OPERATIONS THE OPERATOR CAN RESPOND TO ONE OR ALL
03 C* THREE OF THE COMMAND KEYS SPECIFIED IN COLUMNS 54 THROUGH 59 AND
04 C* KEY THE PRICE FIELD BEFORE PRESSING AN ENTRY FUNCTION KEY. THE
05 C* OPERATION IS PROMPTED BY USER MESSAGE ØØ68.
06 C
07 C                     SET68                     KAKBKC
08 C                     KEY68          PRICE  5Ø
09 C
10 C
```

Figure 10-33. Special Combination of SET and KEY Operations



File Description Specifications

```
02 FLMTFILE ID  F 512 256L 8AI     1 DISK
03 F
04 F
```

Calculation Specifications

```
01 C           FIELDA    SETLLLMTFILE
02 C                     READ LMTFILE                  1Ø
03 C
04 C
```

FIELDA is defined on input specifications as an eight-position alphameric field.

Figure 10-34. SETLL Operation Coding

SETOF (SET OFF)

Indicators				Result	Resulting Indicators		
7-8	9-17	Factor 1	Factor 2	Field	54-55	56-57	58-59
Opt	Opt	Blank	Blank	Blank	1 required		

The SETOF operation turns off any indicators specified in columns 54 through 59. At least one resulting indicator must be specified in columns 54 through 59.

SETON (SET ON)

Indicators				Result	Resulting Indicators		
7-8	9-17	Factor 1	Factor 2	Field	54-55	56-57	58-59
Opt	Opt	Blank	Blank	Blank	1 required		

The SETON operation turns on any indicators specified in columns 54 through 59. At least one resulting indicator must be specified in columns 54 through 59.

SHTDN (SHUT DOWN)

Indicators				Result	Resulting Indicators		
7-8	9-17	Factor 1	Factor 2	Field	54-55	56-57	58-59
Opt	Opt	Blank	Blank	Blank	Required	Blank	Blank

The SHTDN operation sets on the resulting indicator specified in columns 54 and 55 if the system operator has requested shutdown. The indicator can then be used to condition termination of the program in an orderly manner, such as printing some partial totals and going to normal end of job.

Columns 28 through 32 must contain SHTDN, and columns 54 and 55 must contain one of the following valid indicators: 01 through 99, L1 through L9, U1 through U8, H1 through H9, or LR.

SORTA (SORT AN ARRAY)

Indicators				Result	Resulting Indicators		
7-8	9-17	Factor 1	Factor 2	Field	54-55	56-57	58-59
Opt	Opt	Blank	Required	Blank	Blank	Blank	Blank

The SORTA operation allows you to sequence the elements of an array during execution of a program. You can ensure that the elements of the array are in the proper sequence for LOKUP operation by performing a SORTA operation.

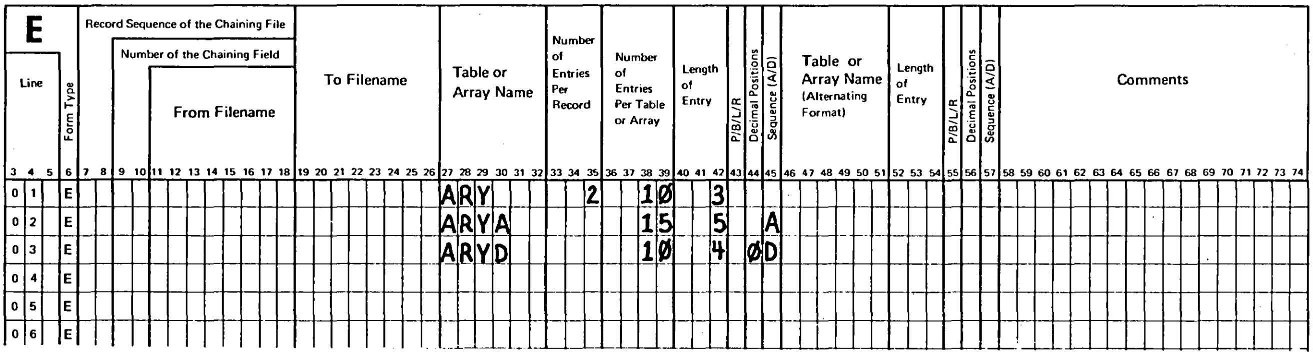
In Figure 10-35, the array ARY is sorted into ascending order because no entry is specified for sequence (columns 45) in the extension specifications. ARYA is sorted into ascending order because column 45 of the extension specifications contains A; ARYD is sorted into descending order because column 45 contains D.

The array specified in factor 2 is sorted into the sequence specified in the extension specifications for the array. If no sequence is specified, the array is sorted into ascending sequence. The standard EBCDIC collating sequence is used for the SORTA operation. If an alternate collating sequence has been defined, it is not used.

For examples of the SORTA operation, see Figure 10-35.

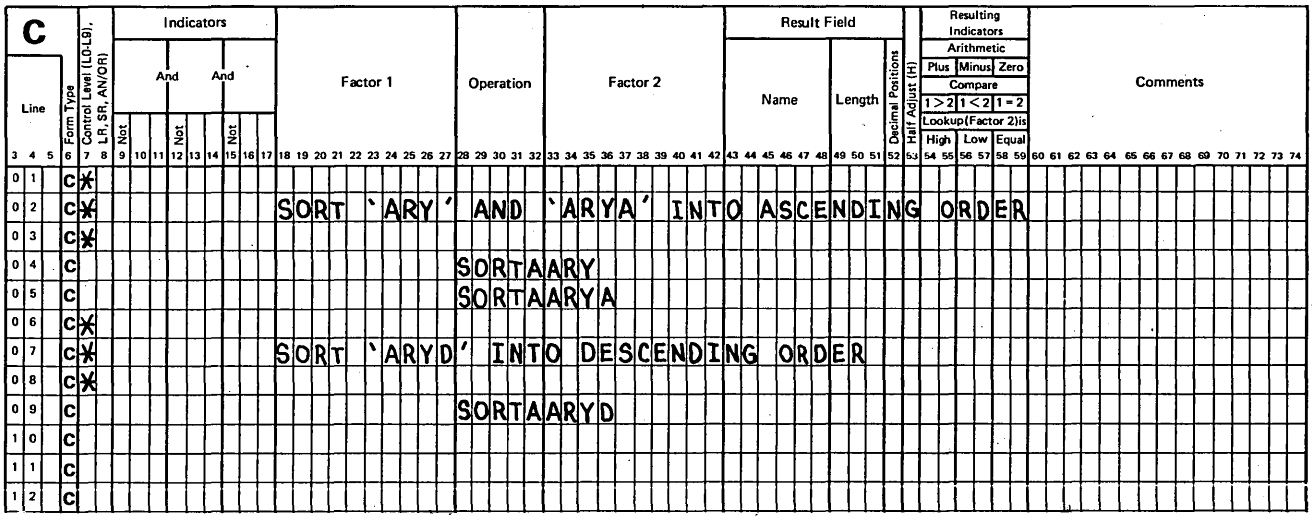
Note: Columns 18 through 27 (factor 1) and 43 through 59 (result field, half adjust, and resulting indicators) must be blank if an SORTA operation is specified.

Extension Specifications



```
01 E                    ARY     2  1Ø  3
02 E                    ARYA      15  5  A
03 E                    ARYD      1Ø  4 ØD
04 E
05 E
06 E
```

Calculation Specifications



```
01 C*
02 C*          SORT 'ARY' AND 'ARYA' INTO ASCENDING ORDER
03 C*
04 C                     SORTAARY
05 C                     SORTAARYA
06 C*
07 C*          SORT 'ARYD' INTO DESCENDING ORDER
08 C*
09 C                     SORTAARYD
10 C
11 C
12 C
```

Figure 10-35. Examples of the SORTA Operation

SQRT (SQUARE ROOT)

Indicators				Result	Resulting Indicators		
7-8	9-17	Factor 1	Factor 2	Field	54-55	56-57	58-59
Opt	Opt	Blank	Required	Required	Blank	Blank	Blank

The SQRT operation derives the square root of the field named in factor 2. The square root of factor 2 is placed in the result field. Factor 1 is not used.

An entire array can be used in a SQRT operation if factor 2 and the result field contain array names.

The number of decimal places in the result field can be either less than or greater than the number of decimal places in factor 2. However, the result field should not have less than half the number of decimal places in factor 2. The result of a SQRT operation is always half-adjusted.

If the value of the factor 2 field is negative, the job halts. The operator can continue processing by responding to the halt. When processing is continued, the result field is set to zero.

SUB (SUBTRACT)

Indicators				Result	Resulting Indicators		
7-8	9-17	Factor 1	Factor 2	Field	54-55	56-57	58-59
Opt	Opt	Optional	Required	Required	Optional	Optional	Optional

Factor 2 is subtracted from factor 1. The difference is placed in the result field. Factor 1 and factor 2 are not changed by the operation. Subtracting a field from itself is a method of setting the result field to zeros. If factor 1 is not present, factor 2 is subtracted from the result field, and the difference is placed in the result field.

TAG (TAG)

Indicators				Result	Resulting Indicators		
7-8	9-17	Factor 1	Factor 2	Field	54-55	56-57	58-59
Opt	Blank	Required	Blank	Blank	Blank	Blank	Blank

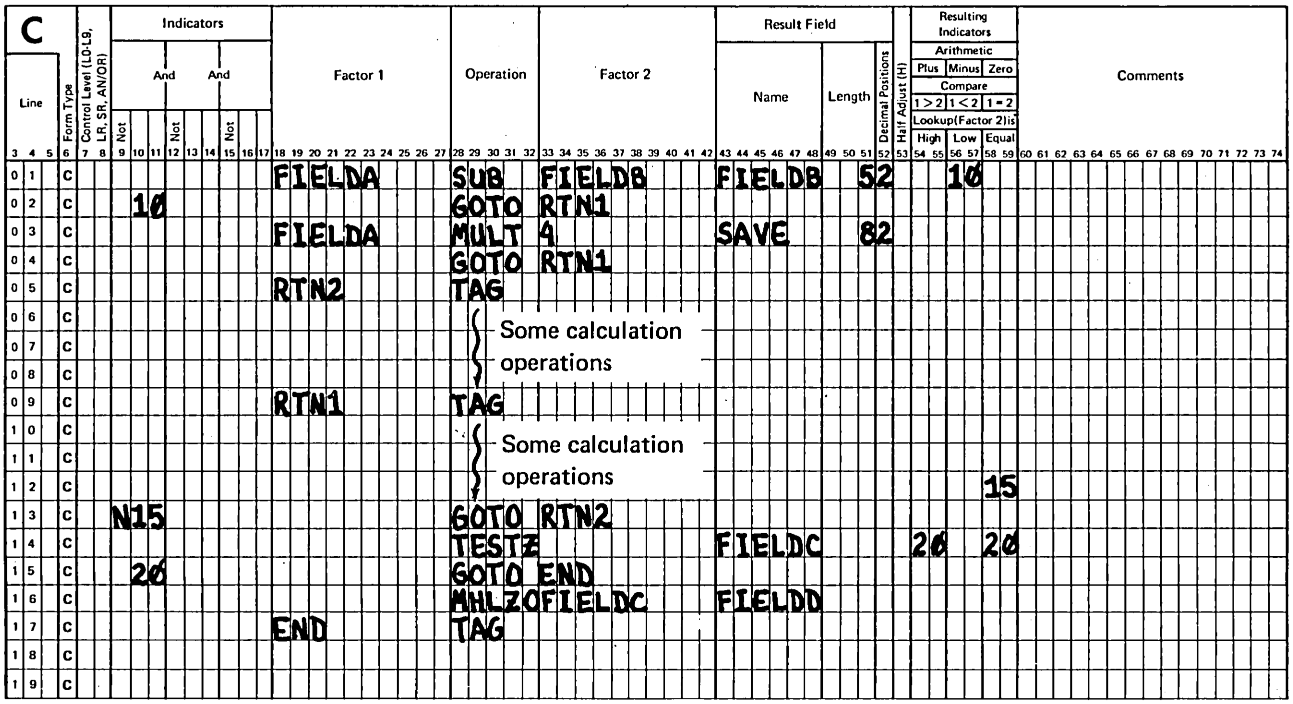
The TAG operation names the operation to which the program branches in the GOTO operation. If the TAG appears within a subroutine, the associated GOTO must appear within the same subroutine.

Factor 1 contains the label that must begin in column 18. The same label cannot be used for more than one TAG operation (or elsewhere as a subroutine name or ENDSR label).

Factor 2 and the result field are not used. No indicators can be entered in columns 9 through 17 for a TAG operation. Control level indicators must be used, however, if branching is to occur only when the information in a control field has changed.

See Figures 10-36 and 10-37 for examples of the TAG operation.

Calculation Specifications



Line	Form Type	Control Level	Indicators	Factor 1	Operation	Factor 2	Result Field Name	Length	Decimal Positions	Half Adjust (H)	Plus / 1>2 / High	Minus / 1<2 / Low	Zero / 1=2 / Equal	Comments
01	C			FIELDA	SUB	FIELDB	FIELDB	5	2			10		
02	C		10		GOTO	RTN1								
03	C			FIELDA	MULT	4	SAVE	8	2					
04	C				GOTO	RTN1								
05	C			RTN2	TAG									
06	C													
07	C					Some calculation								
08	C					operations								
09	C			RTN1	TAG									
10	C													
11	C					Some calculation								
12	C					operations							15	
13	C		N15		GOTO	RTN2								
14	C				TESTZ		FIELDC				20		20	
15	C		20		GOTO	END								
16	C				MHLZO	FIELDC	FIELDD							
17	C			END	TAG									
18	C													
19	C													

1. If the result of the subtraction in line 01 is minus (indicator 10 is on), a branch is taken to RTN1 (routine 1) named by the TAG operation code in line 09. Notice that both the GOTO (line 02) and TAG (line 09) are not conditioned by control level indicators.

2. If the branch is not taken in line 02, the multiplication in line 03 is performed. Then the branch to RTN1 (line 09) must be taken because this branch is not conditioned by indicators.

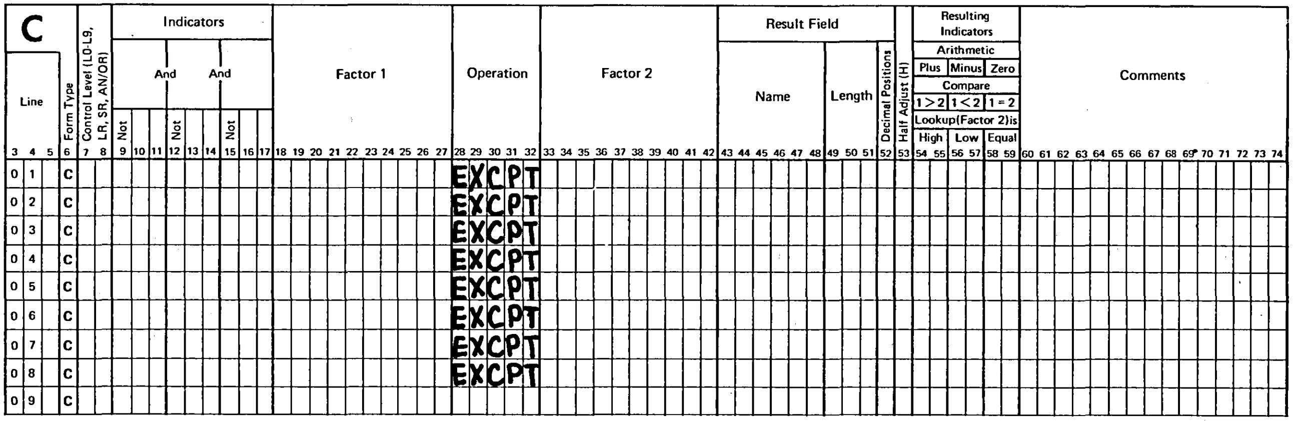
3. Operations in lines 10 through 12 are then done. If the operation in line 12 does not turn indicator 15 on, a branch is taken backwards to RTN2 (line 05).

4. Operations then go in the order specified again from lines 06 through 12. Nothing is done in line 09 because TAG gives only a name. These same operations are performed again and again until indicator 15 does turn on.

5. When indicator 15 is on, the branch to RTN2 is not taken. The TESTZ operation is then performed. If this operation causes indicator 20 to turn on, a branch is taken to line 17 (GOTO END). If indicator 20 is not on, the operation in line 16 is done.

Figure 10-36. Using GOTO and TAG (Skipping Operations)

Calculation Specifications

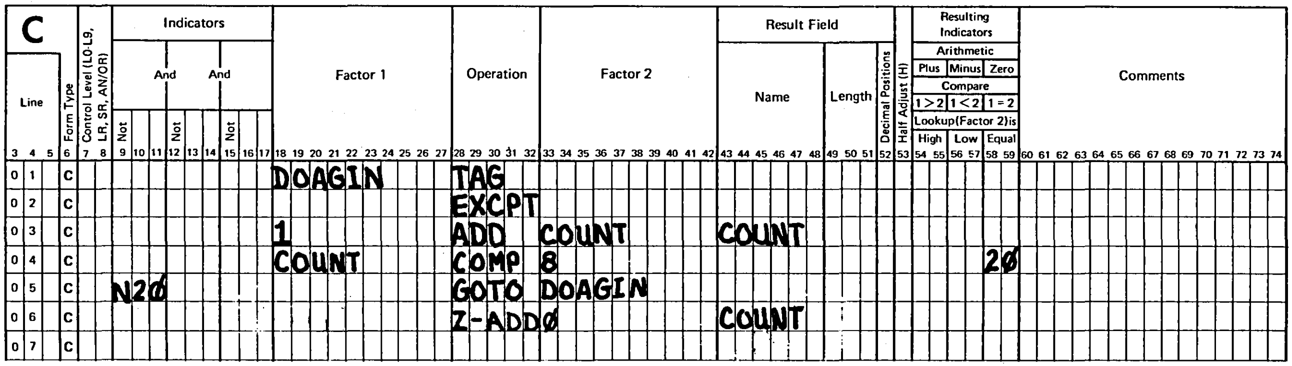


```
01 C           EXCPT
02 C           EXCPT
03 C           EXCPT
04 C           EXCPT
05 C           EXCPT
06 C           EXCPT
07 C           EXCPT
08 C           EXCPT
09 C
```

Assume you want to make eight mailing labels for every customer you have. The customer's name and address are found on an input record. Because you want to write eight labels for each record, you can use exception lines and the operation EXCPT instead of coding eight identical output line specifications. (See *Exception* (EXCPT) in this chapter for further information.)

However, by using branching, you can code it all in five lines as shown below. An EXCPT line is printed out. One is added to COUNT to keep track of how many times the line is printed. Then COUNT is compared to 8. If COUNT does not equal 8, a branch is taken back to the beginning (GOTO DOAGIN). If COUNT equals 8, the branch is not taken. Instead, the COUNT field is set to zero for the next cycle.

Calculation Specifications



```
01 C           DOAGIN    TAG
02 C                     EXCPT
03 C           1         ADD  COUNT     COUNT
04 C           COUNT     COMP 8                          2Ø
05 C  N2Ø                GOTO DOAGIN
06 C                     Z-ADDØ         COUNT
07 C
```

Figure 10-37. Using GOTO and TAG to Eliminate Duplicate Coding

TESTB (TEST BIT)

Indicators				Result	Resulting Indicators		
7-8	9-17	Factor 1	Factor 2	Field	54-55	56-57	58-59
Opt	Opt	Blank	Required	Required	1 required		

The TESTB operation compares the bits identified in factor 2 with the corresponding bits in the field named as the result field. Resulting indicators in columns 54 through 59 reflect the status of the result field bits. Factor 2 is always a source of bits for the result field. The result field is the field in which corresponding bits are compared with the bits specified in factor 2.

Factor 2 can contain:

- *Bit numbers 0-7:* From 1 to 8 bits can be tested per operation. The bits to be tested are identified by the numbers 0 through 7 (0 is the leftmost bit). The bit numbers must be enclosed in apostrophes and the entry must begin in column 33. For example, to test bits 0, 2, and 5, enter '025' in factor 2.

- *Field name:* The name of a one-position alphameric field, table element, or array element can be specified in factor 2. In this case, the bits that are on in the field, table element, or array element are tested in the result field; bits that are off are not tested.

See Figure 10-38 for a summary of TESTB operations.

Indicators assigned in columns 54 through 59 reflect the status of the result field bits. At least one indicator must be assigned, and as many as three can be assigned for one operation. Two indicators can be the same for a TESTB operation, but not three. For TESTB operations, the resulting indicators turn on as follows:

Columns 54-55: An indicator in these columns turns on if each bit specified in factor 2 or each bit that is on in the factor 2 field is off in the result field.

Columns 56-57: An indicator in these columns turns on if the bits specified in factor 2 or the bits that are on in the factor 2 field are of mixed status (some on, some off) in the result field.

Columns 58-59: An indicator in these columns turns on if each bit specified in factor 2 or each bit that is on in the factor 2 field is on in the result field.

Note: If the field in factor 2 has no bits on, then this indicator turns on.

The operation code TESTB must appear in columns 28 through 32. Conditioning indicators can be used in columns 7 through 17. At least one resulting indicator should be assigned in columns 54 through 59. As many as three resulting indicators can be assigned, but not more than two can be the same. Factor 1, decimal positions, and the half-adjust columns must be blank.

Calculation Specifications

Line	Form Type	Control Level (L0-L9, LR, SR, AN/OR)	Indicators (Not, And, Not, And, Not)	Factor 1	Operation	Factor 2	Result Field Name	Length	Decimal Positions	Half Adjust (H)	Resulting Indicators Arithmetic: Plus Compare: 1>2 Lookup(Factor 2)is: High	Minus 1<2 Low	Zero 1=2 Equal	Comments
3 4 5	6	7 8	9 10 11 12 13 14 15 16 17	18 19 20 21 22 23 24 25 26 27	28 29 30 31 32	33 34 35 36 37 38 39 40 41 42	43 44 45 46 47 48	49 50 51	52	53	54 55	56 57	58 59	60 61 62 63 64 65 66 67 68 69 70 71 72 73 74
01	C													
02	C													
03	C													
04	C													
05	C													
06	C													
07	C													
08	C													

The following TESTB operation compares bits 0 and 7 with corresponding bits in the field named BITS. If bits 0 and 7 are off in the field named BITS, indicator 20 turns on. If bits 0 and 7 are of mixed status (one on, one off) in the field named BITS, indicator 21 turns on. If bits 0 and 7 are on in the field named BITS, indicator 22 turns on.

Line	Form Type	Control Level	Indicators	Factor 1	Operation	Factor 2	Result Field Name	Length	Decimal Positions	Half Adjust	High	Low	Equal	Comments
09	C													
10	C				TESTB	'Ø7'	BITS				2Ø	21	22	
11	C													
12	C													
13	C													
14	C													
15	C													
16	C													
17	C													
18	C													
19	C													
20	C													

The following operation compares the bits that are on in the field named ALPHA with corresponding bits in the field named BITS. If the bits that are on in the field named ALPHA are off in the field named BITS, indicator 20 turns on. If the bits that are on in the field named ALPHA are of mixed status (some on, some off) in the field named BITS, indicator 21 turns on. If the bits that are on in the field named ALPHA are on in the field named BITS, indicator 22 turns on.

Line	Form Type	Control Level	Indicators	Factor 1	Operation	Factor 2	Result Field Name	Length	Decimal Positions	Half Adjust	High	Low	Equal	Comments
21	C													
22	C				TESTB	ALPHA	BITS				2Ø	21	22	
23	C													
24	C													
25	C													
26	C													
27	C													
28	C													
29	C													
30	C													
31	C													
32	C													
33	C													
34	C													

The following operations use a one-position array element either as a source of bits or as a result field, or both. In the first operation, the bits that are on in the field named ALPHA are compared to corresponding bits in the array element ARR,NX. For example, assume that bits 1 and 4 are on in the field named ALPHA. If bits 1 and 4 are off in array element ARR,NX, indicator 20 turns on. If bits 1 and 4 are of mixed status (one on, one off) in array element ARR,NX, indicator 21 turns on. If bits 1 and 4 are on in array element ARR,NX, indicator 22 turns on.

Line	Form Type	Control Level	Indicators	Factor 1	Operation	Factor 2	Result Field Name	Length	Decimal Positions	Half Adjust	High	Low	Equal	Comments
35	C													
36	C													
37	C				TESTB	ALPHA	ARR,NX				2Ø	21	22	
38	C													
39	C				TESTB	'24'	ARE,12				2Ø	21	22	
40	C													
41	C				TESTB	ARE,12	ARR,NX				2Ø	21	22	
42	C													
43	C													

Figure 10-38. Summary of TESTB Operations

TESTZ (TEST ZONE)

Indicators				Result	Resulting Indicators		
7-8	9-17	Factor 1	Factor 2	Field	54-55	56-57	58-59
Opt	Opt	Blank	Blank	Required	1 required		

The TESTZ operation tests the zone of the leftmost character in the result field. The result field must be alphameric because this operation can be done only on alphameric characters. Resulting indicators turn on according to the results of the test. The characters &, A through I, and any other character with the same zone as the character A turn the plus indicator on. The characters - (minus), J through R, and any other character with the same zone as the character J turn the minus indicator on. Characters with any other zone turn the zero indicator on. Factor 1 and factor 2 are not used in this operation.

TIME (TIME OF DAY)

Indicators				Result	Resulting Indicators		
7-8	9-17	Factor 1	Factor 2	Field	54-55	56-57	58-59
Opt	Opt	Blank	Blank	Required	Blank	Blank	Blank

The TIME operation accesses the system time of day and, if specified, the system date.

The system date accessed is not the same field as UDATE and may or may not contain the same information. See the *System Support Reference Manual* for a complete description of the system date and the DATE OCL statement.

Columns 28 through 32 must contain the operation code TIME, and columns 43 through 48 (the result field) must specify the name of a numeric field with zero decimal positions into which the time of day or the time of day and the system date are written.

To access the time of day only, specify the result field as a 6-digit numeric field. To access both the time of day and the system date, specify the result field as a 12-digit numeric field. The time of day is always placed in the first six positions of the result field in the format hhmmss, where hh is hours, mm is minutes, and ss is seconds. If the system date is included, it is placed in positions 7 through 12 of the result field. The date format depends on the system date format and can be mmddyy, ddmmyy, or yymmdd.

XFOOT (SUMMING THE ELEMENTS OF AN ARRAY)

Indicators				Result	Resulting Indicators		
7-8	9-17	Factor 1	Factor 2	Field	54-55	56-57	58-59
Opt	Opt	Blank	Required	Required	Optional	Optional	Optional

The XFOOT operation can be used only on numeric arrays. XFOOT adds the elements of the array together and places the sum into the field specified as the result field. Factor 1 is not used. Factor 2 contains the name of the array.

Z-ADD (ZERO AND ADD)

Indicators				Result	Resulting Indicators		
7-8	9-17	Factor 1	Factor 2	Field	54-55	56-57	58-59
Opt	Opt	Blank	Required	Required	Optional	Optional	Optional

Factor 2 is added to a field of zeros. The sum is placed in the result field. Factor 1 is not used.

Z-SUB (ZERO AND SUBTRACT)

Indicators				Result	Resulting Indicators		
7-8	9-17	Factor 1	Factor 2	Field	54-55	56-57	58-59
Opt	Opt	Blank	Required	Required	Optional	Optional	Optional

Factor 2 is subtracted from a field of zeros. The difference, which is actually the negative of factor 2, is placed in the result field. This operation can be used to change the sign of a field. Factor 1 is not used.

Chapter 11. Multifile Processing

In an RPG II program, the processing of more than one input file, with or without match fields, is termed multifile processing. Selection of records from more than one file based on the contents of match fields is known as multifile processing by matching records.

Multifile processing can be used with input files that are designated as primary or secondary files but not as chained or demand files. One of the files used in multifile processing must be a primary file. If a WORKSTN file is designated as the primary file, no secondary files can be specified; therefore, multifile processing by matching records cannot be used.

NO MATCH FIELDS

When no match fields are used in multifile processing, records are selected from one file at a time. When the records from one file are all processed, the records from the next file are selected. The files are selected in this order:

1. Primary file, if specified

2. Secondary files in the order in which they are described in the file description specifications

MATCH FIELDS

When match fields are used in multifile processing, the program selects the records for processing according to the contents of the match fields. The program reads one record from every input file and compares the match fields in the records. If the records are in ascending order, the program selects the record with the lowest match field. If the records are in descending order, the program selects the record with the highest match field.

When a record is selected from a file, the program then reads the next record from that file. At the beginning of the next program cycle, the new record is compared with the other records in the read area that are waiting for selection, and one record is selected for processing.

Records without match fields can also be included in the files. Such records are selected for processing before records with match fields. If two or more of the records being compared have no match fields, selection of those records is determined by the priority of the files from which the records came. The priority of the files is:

1. Primary file, if specified

2. Secondary files in the order in which they are described in the file description specifications

When the primary file record matches one or more of the secondary records, the MR (matching record) indicator turns on. This indicator can be used to condition calculation or output operations for the record that is selected. When one of the matching records must be selected, the selection is determined by the priority of the files from which the records came.

Figure 11-1 shows when the MR indicator is turned on during the RPG II cycle. For more information on the RPG II cycle, see Chapter 17, *Detailed RPG II Program Logic.*

One file can be defined with match fields while all other input files have no match fields. The files without the match fields are then processed completely according to the previously mentioned priority of files. The file with the match fields is processed last, and sequence checking occurs for that file.

Assigning Match Field Values (M1-M9)

When assigning match field values (M1 through M9) to fields on the input specifications in columns 61 and 62, consider the following:

- Sequence checking is automatically done for all record types with match field specifications. The contents of the fields to which M1 through M9 are assigned are checked for correct sequence. An error in sequence stops the program, and the record that caused the halt is not processed. When the system is restarted, the next record from the same file is read. Thus, all match fields must be in the same order, either all ascending or all descending. See *Column 18 (Sequence)* in Chapter 3, *File Description Specifications.*

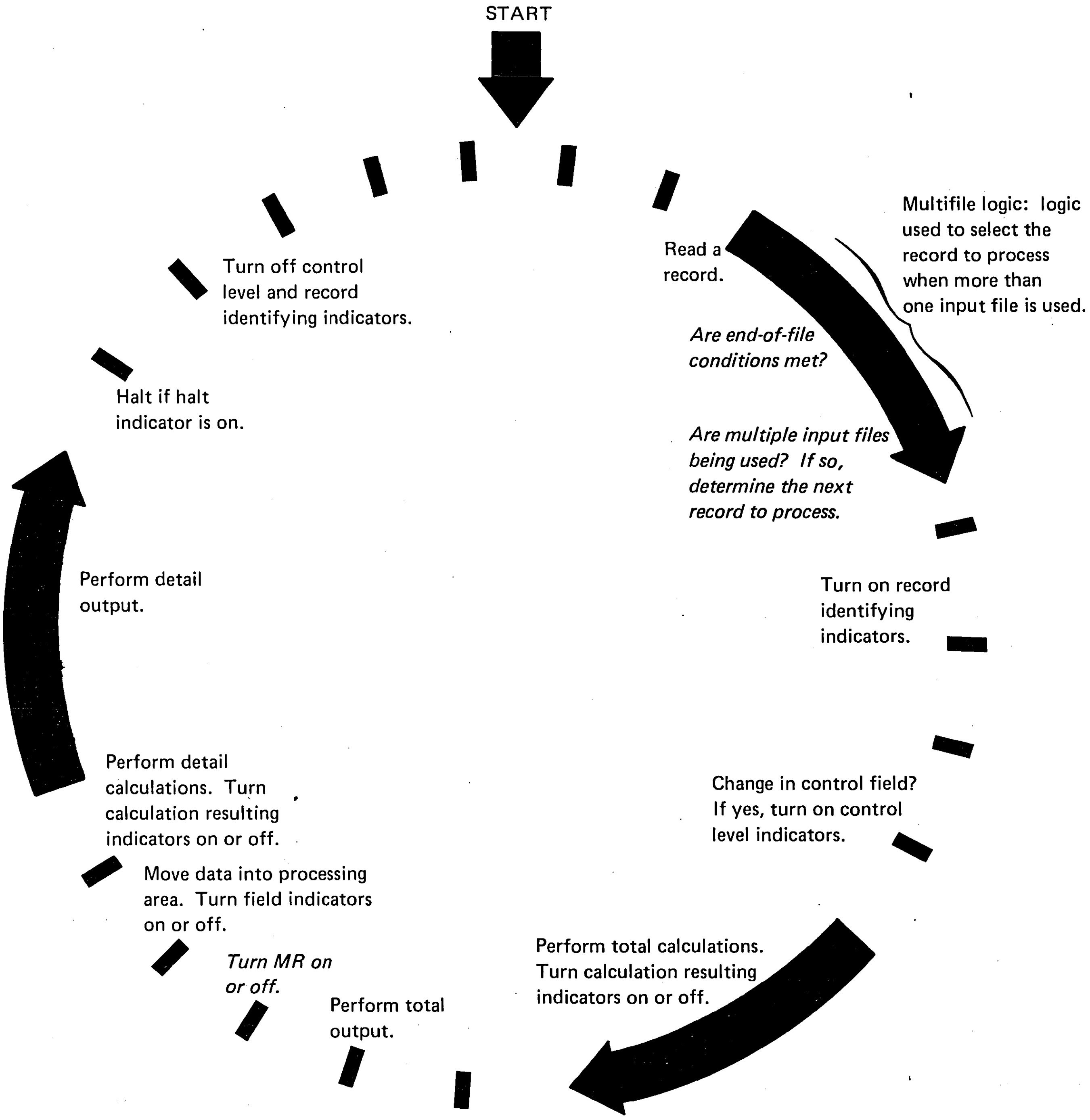

START
Read a record.
Multifile logic: logic used to select the record to process when more than one input file is used.
Are end-of-file conditions met?
Are multiple input files being used? If so, determine the next record to process.
Turn on record identifying indicators.
Change in control field? If yes, turn on control level indicators.
Perform total calculations. Turn calculation resulting indicators on or off.
Perform total output.
Turn MR on or off.
Move data into processing area. Turn field indicators on or off.
Perform detail calculations. Turn calculation resulting indicators on or off.
Perform detail output.
Halt if halt indicator is on.
Turn off control level and record identifying indicators.

Figure 11-1. Simplified Matching Record Logic

- Not all files used in the job must have match fields. Not all record types within one file must have match fields either. However, at least one record type from two files must have match fields if files are ever to be matched.
- The same number of match fields must be specified for all record types that are used in matching. The same matching record values must also be used for all types (see Figure 11-2).
- All match fields given the same matching record value (M1 through M9) must be the same length and type (alphameric or numeric). If the match field contains packed data, the zoned decimal length, which is (2 x the packed length) -1, is considered as the length of the match field.
- Record positions of different match fields can overlap, but the total length of all fields must not exceed 144 characters.
- If more than one match field is specified for a record type, all the fields are combined and treated as one continuous match field (see Figure 11-2). The fields are combined according to descending sequence (M9 to M1) of matching record values.
- Match fields cannot be split, that is, the same matching field value cannot be used twice for one type of record.
- Match fields can be either alphameric or numeric. However, all match fields given the same matching record value (M1 through M9) are considered numeric if any one of the match fields is described as numeric.
- When numeric fields having decimal positions are matched, they are treated as if they had no decimal position. For instance, 3.46 is considered equal to 346.
- Only the digit portions of numeric match fields are compared. Even though a field is negative, it is considered to be positive because the sign of the numeric field is ignored. Thus, a -5 will match with a +5.
- Whenever more than one matching record value is used, all match fields must match before the MR indicator turns on. For example, if match fields M1, M2, and M3 are specified, all three fields from one record must match all three fields from the other record. A match on only the M1 and M2 fields will not turn on the MR indicator (see Figure 11-2).
- Field names are ignored in matching record operations. Therefore, fields from different record types that are assigned the same match level can have the same name.
- If an alternate collating sequence is defined for the program, alphameric fields are matched according to the alternate sequence specified.
- A field specified as binary (B in column 43) cannot be assigned a match field value. However, a field specified as packed (P in column 43) can be assigned a match field value.

Note: Additional rules applying to matching records when used with entries in the field record relation columns are discussed under *Columns 63-64*, in Chapter 7, *Input Specifications.*

Figure 11-3 is an example of how match fields are specified.

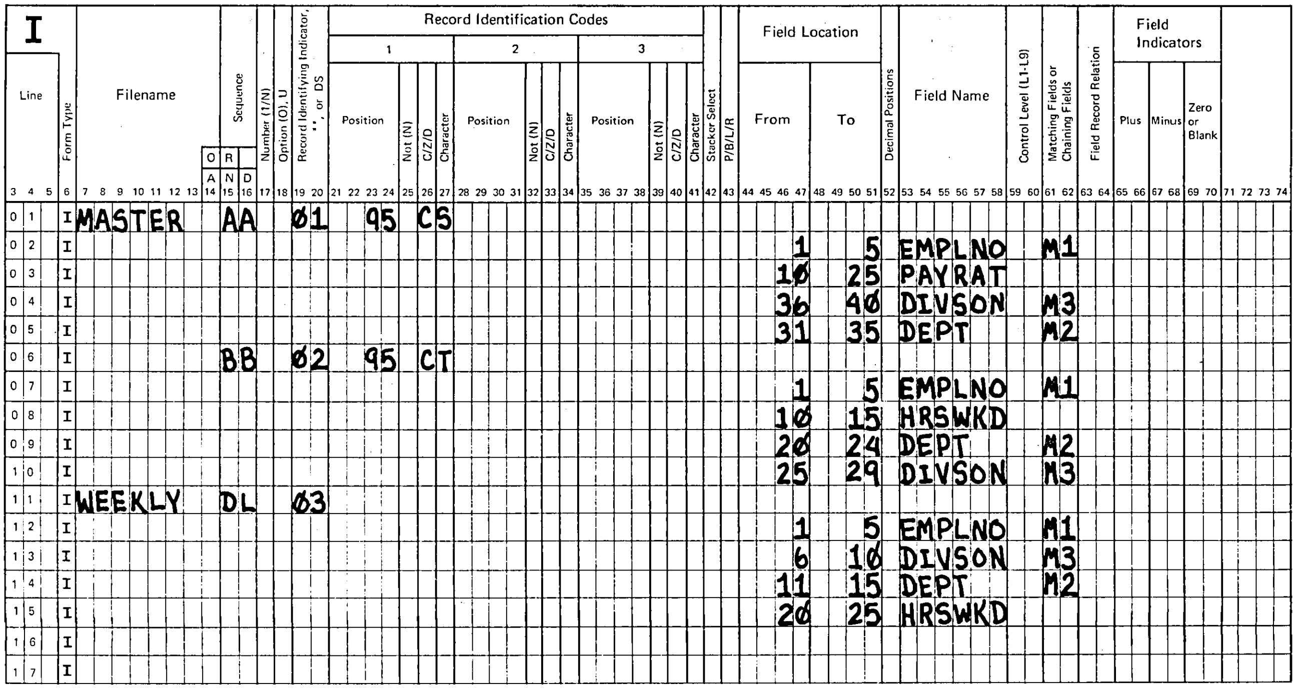
Processing Matching Records

Matching records for two or more files are processed in the following manner:

- Whenever a record from the primary file matches a record from the secondary file, the primary file is processed first. Then the matching secondary file record is processed. The record identifying indicator that identifies the record type just selected is on at the time the record is processed. This indicator is often used to control the type of processing that takes place.
- Whenever records from ascending files do not match, the record having the lowest match field content is processed first. Whenever records from descending files do not match, the record having the highest match field content is processed first.
- A record type that has no match field specification is processed immediately after the record it follows. The MR indicator is off. If this record type is first in the file, it is processed first even if it is not in the primary file.
- The matching of records makes it possible to enter data from primary records into their matching secondary records because the primary record is processed before the matching secondary record. However, the transfer of data from secondary records to matching primary records can be done only when look-ahead fields are specified (see *Look-Ahead Fields under Columns 19-20* in Chapter 7, *Input Specifications).*

Figures 11-4 and 11-5 show how records from three disk files are selected for processing.

Input Specifications



Line	Form Type	Filename	Sequence	Record Identifying Indicator	Position	C/Z/D	Character	From	To	Field Name	Matching Fields or Chaining Fields
01	I	MASTER	AA	Ø1	95	C	S				
02	I							1	5	EMPLNO	M1
03	I							1Ø	25	PAYRAT	
04	I							36	4Ø	DIVSON	M3
05	I							31	35	DEPT	M2
06	I		BB	Ø2	95	C	T				
07	I							1	5	EMPLNO	M1
08	I							1Ø	15	HRSWKD	
09	I							2Ø	24	DEPT	M2
10	I							25	29	DIVSON	M3
11	I	WEEKLY	DL	Ø3							
12	I							1	5	EMPLNO	M1
13	I							6	1Ø	DIVSON	M3
14	I							11	15	DEPT	M2
15	I							2Ø	25	HRSWKD	
16	I										
17	I										

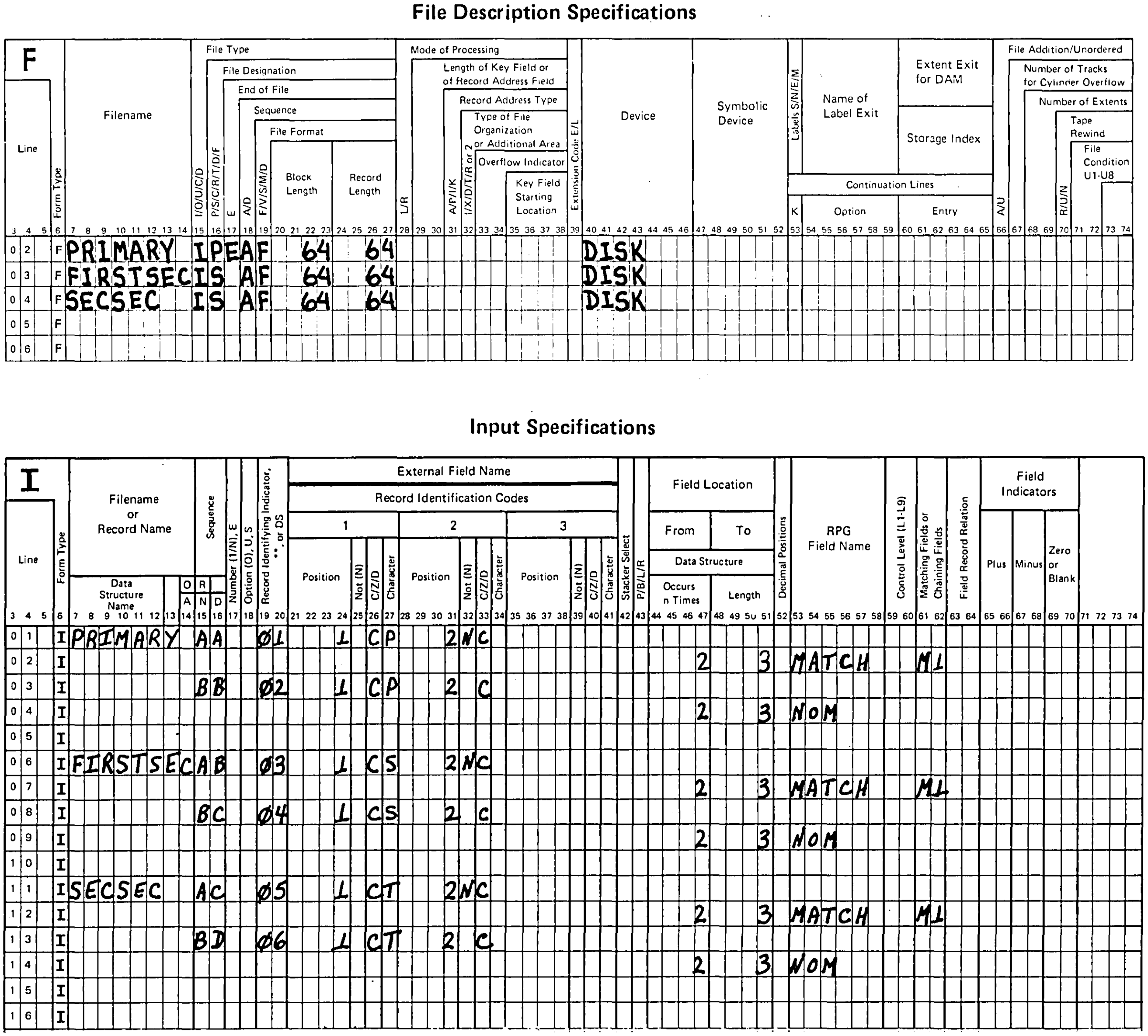
Figure 11-2. Match Fields

Three record types are used in matching records. All record types have three match fields specified, and all use the same values (M1, M2, M3) to indicate which fields must match. The MR indicator turns on only if all three match fields in either of the record types from the MASTER file are the same as all three fields from the record in the WEEKLY file.

The three match fields in each record type are combined and treated as one match field organized as follows:

DIVSON	DEPT	EMPLNO
M3	M2	M1

The order in which the fields are specified by the input specifications does not affect the organization of the match fields in the computer.



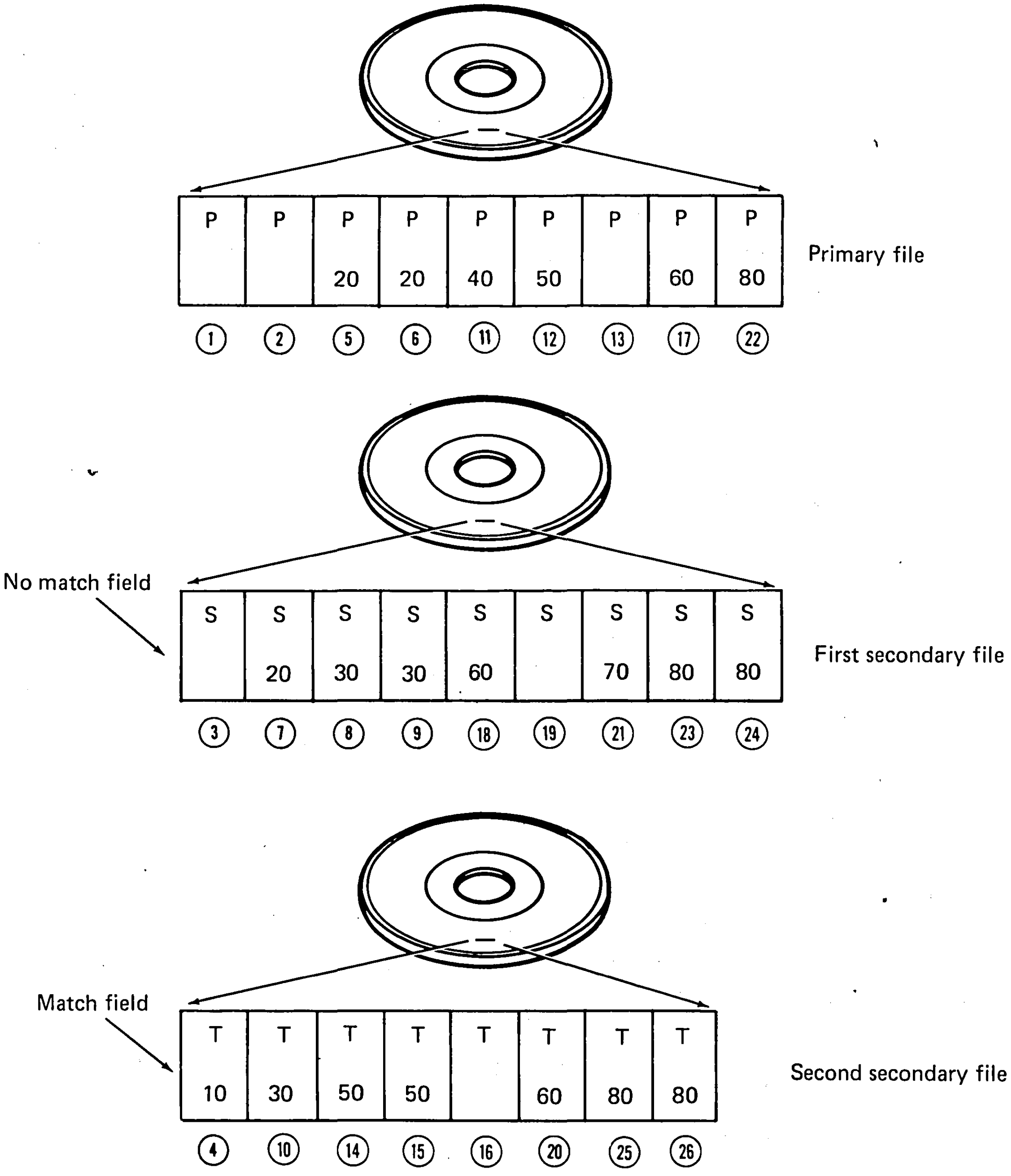
File Description Specifications

Line	Form Type	Filename	File Type	File Designation	End of File	Sequence	File Format	Block Length	Record Length	Device
02	F	PRIMARY	I	P	E	A	F	64	64	DISK
03	F	FIRSTSEC	I	S		A	F	64	64	DISK
04	F	SECSEC	I	S		A	F	64	64	DISK
05	F									
06	F									

Input Specifications

Line	Form Type	Filename or Record Name	Sequence	Record Identifying Indicator	1 Position	1 Not (N)	1 C/Z/D	1 Character	2 Position	2 Not (N)	2 C/Z/D	2 Character	Field Location From	Field Location To	RPG Field Name	Matching Fields or Chaining Fields
01	I	PRIMARY	AA	Ø1	1		C	P	2	N	C					
02	I												2	3	MATCH	M1
03	I		BB	Ø2	1		C	P	2		C					
04	I												2	3	NOM	
05	I															
06	I	FIRSTSEC	AB	Ø3	1		C	S	2	N	C					
07	I												2	3	MATCH	M1
08	I		BC	Ø4	1		C	S	2		C					
09	I												2	3	NOM	
10	I															
11	I	SECSEC	AC	Ø5	1		C	T	2	N	C					
12	I												2	3	MATCH	M1
13	I		BD	Ø6	1		C	T	2		C					
14	I												2	3	NOM	
15	I															
16	I															

Figure 11-3. Match Field Specifications for Three Disk Files

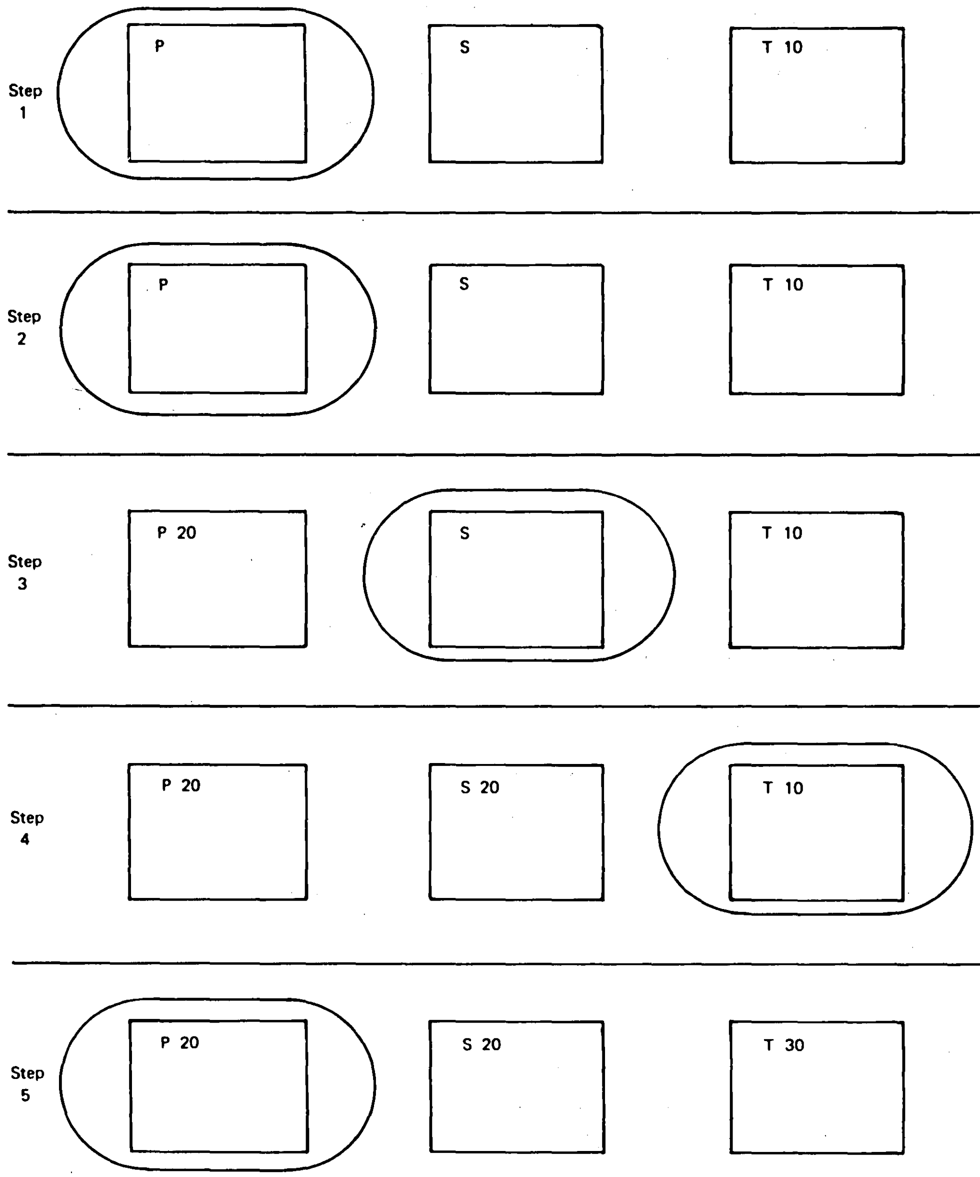


The records from the three disk files are selected in the order indicated by the circled numbers.

Figure 11-4 (Part 1 of 2). Normal Record Selection from Three Disk Files

Cycle	File Processed	Indicators On	Reason for Record Selection
1	PRIMARY	02	No match field specified
2	PRIMARY	02	No match field specified
3	FIRST SEC	04	No match field specified
4	SEC SEC	05	Second secondary low No primary match
5	PRIMARY	01, MR	Primary matches first secondary
6	PRIMARY	01, MR	Primary matches first secondary
7	FIRST SEC	03, MR	First secondary matches primary
8	FIRST SEC	03	First secondary low No primary match
9	FIRST SEC	03	First secondary low No primary match
10	SEC SEC	05	Second secondary low No primary match
11	PRIMARY	01	Primary low No secondary match
12	PRIMARY	01, MR	Primary matches second secondary
13	PRIMARY	02	No match field specified
14	SEC SEC	05, MR	Second secondary matches primary
15	SEC SEC	05, MR	Second secondary matches primary
16	SEC SEC	06	No match field specified
17	PRIMARY	01, MR	Primary matches both secondary files
18	FIRST SEC	03, MR	First secondary matches primary
19	FIRST SEC	04	No match field specified
20	SEC SEC	05, MR	Second secondary matches primary
21	FIRST SEC	03	First secondary low No primary match
22	PRIMARY	01, MR	Primary matches both secondary files
23	FIRST SEC	03, MR	First secondary matches primary
24	FIRST SEC	03, MR	First secondary matches primary
25	SEC SEC	05, MR	Second secondary matches primary
26	SEC SEC	05, MR	Second secondary matches primary

Figure 11-4 (Part 2 of 2). Normal Record Selection from Three Disk Files



The first record from each file is read. The P and S records have no match field, so they are processed before the T record that has a match field. Because the P record comes from the primary file, it is selected for processing first.

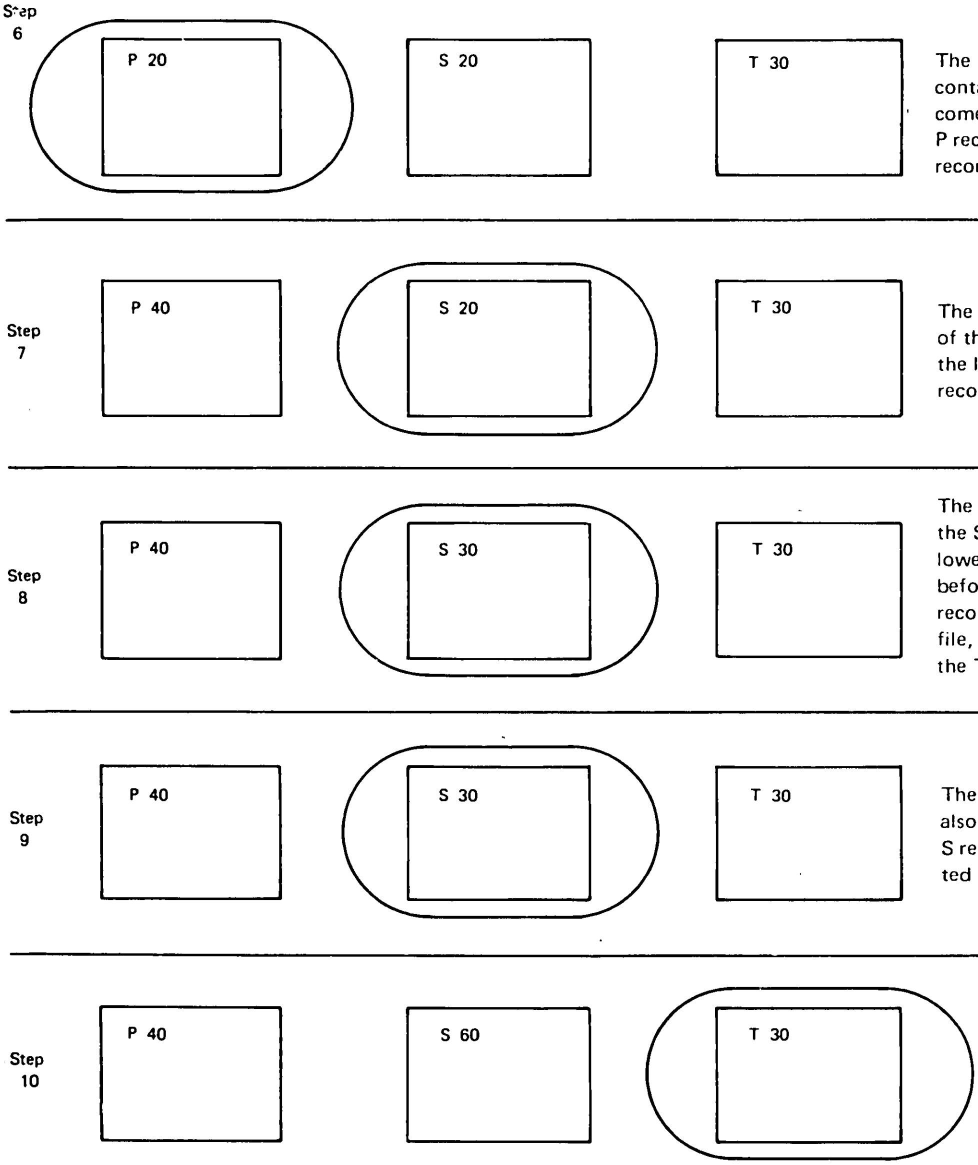
The next P record is read. It contains no match field and comes from the primary file, so the new P record is also selected for processing before the S record.

The next P record read has a match field. The S record has no match field, so it is selected for processing.

The next S record is read. All three records have match fields. Because the value in the match field of the T record is lower than the value in the other two, the T record is selected for processing.

The next T record is read. The matching P and S records both have the low match field value, so they are processed before the T record. Because the matching P record comes from the primary file, it is selected for processing first.

Figure 11-5 (Part 1 of 2). Normal Record Selection from Three Disk Files



The next P record is read. Because it contains the same match field and comes from the primary file, the new P record is selected instead of the S record.

The next P record is read. The value of the match field in the S record is the lowest of the three, so the S record is selected for processing.

The next S record is read. Because the S and T records match and have the lowest match field, they are selected before the P record. Because the S record comes from the first secondary file, it is selected for processing before the T record.

The next S record is read. Because it also has the same match field as the S record just selected, it too is selected before the T record.

The next S record is read. The T record contains the lowest match field value, and is selected for processing.

Figure 11-5 (Part 2 of 2). Normal Record Selection from Three Disk Files

Chapter 12. CONSOLE File Considerations

Using the CONSOLE device in an RPG II program enables the operator to enter input to an executing RPG II program from a display station. Display screen prompts are automatically generated from the input specifications to prompt the operator for data entry. As the operator enters data in response to the prompt, the data is displayed on the display screen. Internally the data is placed in a buffer, which is independent of the execution of the RPG II program. At input time of the RPG II cycle, the program retrieves the data from the buffer. Data for a CONSOLE file can be entered anytime during the RPG II cycle. The operator indicates a normal end of input (end-of-file condition) to the CONSOLE file by pressing command key 12, that is, the Cmd and = (equal) keys.

RESTRICTIONS FOR USING A CONSOLE FILE

The following restrictions apply to the use of a CONSOLE file:

- Only one CONSOLE file is allowed per program.
- Only one display station is associated with the program; therefore, prompting for input to the program is directed to the display station that loaded the program or that is assigned to the program by the WORKSTN OCL statement.
- If a program contains a WORKSTN file, it cannot contain a CONSOLE, KEYBORD, or CRT file.
- The maximum number of record types that can be used for a CONSOLE file is 10.
- The maximum record length is 1,518 characters.
- The maximum alphameric field length is 66 characters.
- The maximum numeric field length is 15 characters.
- The maximum number of input fields that can be displayed on the screen is 80.

WHEN USED WITH KEYBORD/CRT DEVICE

Both device names CONSOLE and KEYBORD refer to a display station, that is, a keyboard and a display screen. When the device KEYBORD is specified in the same program as the device CONSOLE and the operator is entering data for the CONSOLE file, the following occurs when the program encounters a KEY or SET operation for the KEYBORD file:

- The operator is required to complete data entry for the current record for the CONSOLE file.
- The prompt for the SET or KEY operation, or the output to the CRT file, is then displayed on the display screen. Normal RPG II processing continues after the SET or KEY operation is completed. The display screen and keyboard are reenabled for input to the CONSOLE file when the RPG II program begins processing the saved record during the next input cycle.

AUTOMATIC GENERATION OF DISPLAY SCREEN FORMATS

When a program includes a CONSOLE device and the RPG command is used to compile the program, the RPG II format generator (RPGR) uses the input specifications to generate source input to the screen format generator (the $SFGR utility program of the system support program product). The screen format generator compiles this source input and generates a display screen format load member for the program. The source input to $SFGR is not saved when the RPG command is used. To save the source input, specify the NOGEN parameter on the RPG command statement. The RPGR procedure can be called separately to generate the source input for the $SFGR utility program, which is then called to generate a display screen format load member for the CONSOLE file.

To execute an RPG program containing a CONSOLE device from a display station with a 960-character display screen, specify the GEN960 parameter on the RPG or AUTO command statement. GEN960 indicates that the screen formats for the CONSOLE file will be generated for a 12-line, 960-character screen.

See Chapter 18, *Compiling and Executing RPG II Programs*, for a description of the RPG and RPGR command statements. For a complete description of the $SFGR utility program, see the *System Support Reference Manual.*

RPG II forms the name of this source member and load member by adding FM to the RPG II program name given in columns 75 through 80 of the control specifications at the time of compilation. Therefore, the format name cannot be changed after compilation. For example, if the name of the program is PRNAME, the name of the display screen format source member and load member for that program is PRNAMEFM.

If multiple formats are generated for the same program, the record identifying indicator is appended to the program name to distinguish each distinct format in the program. For example, if the program PRNAME contains the three record types with the indicators 01, 02, and 03 specifying the record types, the display format names generated are:

PRNAMEFM

PRNAME01

PRNAME02

PRNAME03

PRNAMEFM is the name by which the set of display formats is referenced in the library, and PRNAME01, PRNAME02, and PRNAME03 are the names of the actual display screen format members of the set.

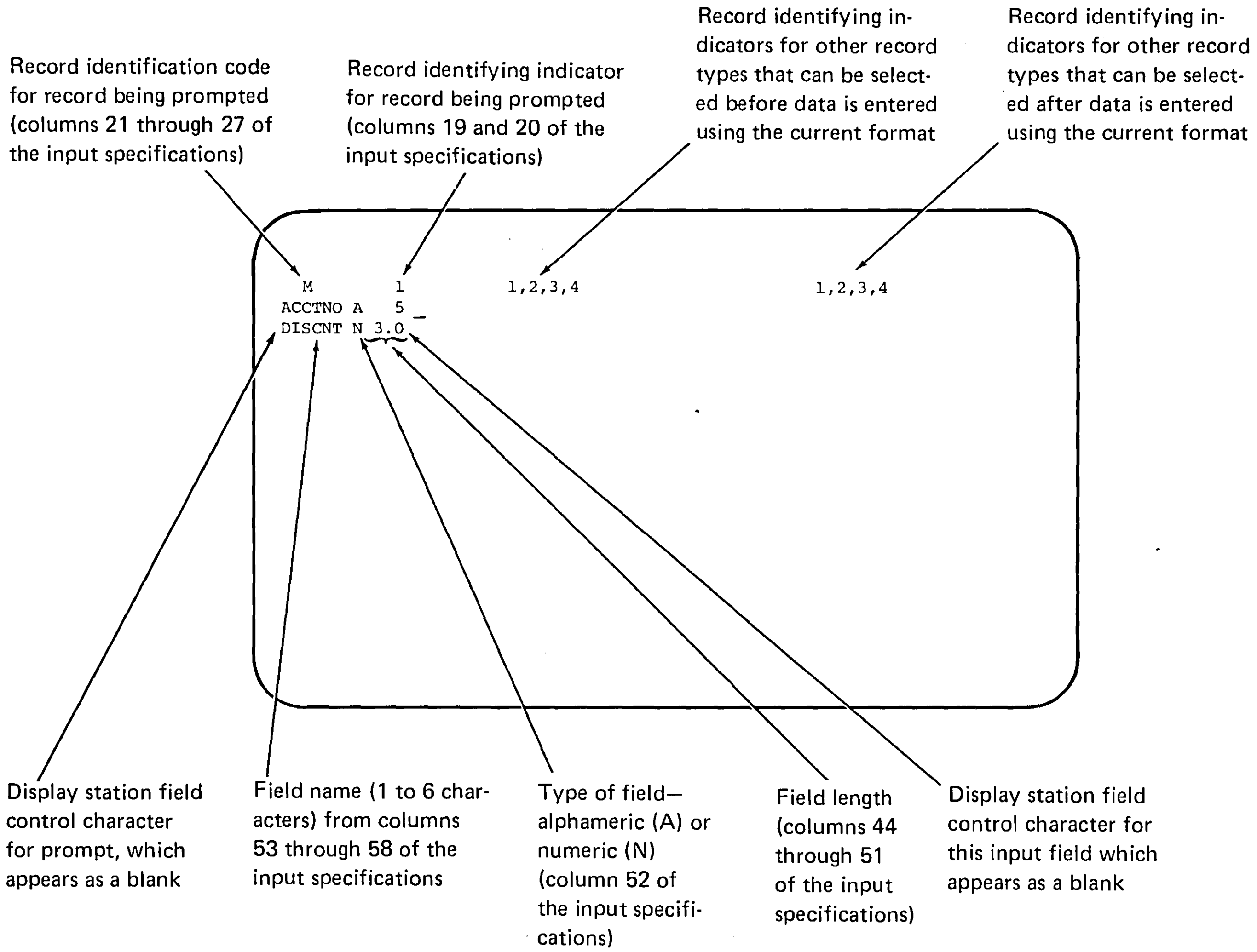
If OR lines are used on the input specifications to identify the same record, only one format is associated with the record.

When CONSOLE is used as a record address file to provide keys for a limits file, FM is added to the program name to form the name of the display screen format source and load members.

Display Format

The top line on the display screen contains control information that is used by the operator to identify the current record and to specify the next record type to be prompted (see Figure 12-1). The remaining 11 or 23 lines are used for the formatted record. For each field defined, 14 characters are reserved to contain the field name and its attributes. Therefore, the record length is limited to 1,518 characters. The format generated for a record depends on the size and number of fields in the record. The possible formats are:

- *One column*: The one-column format is generated whenever the number of fields prompted for is less than 12 or 24.
- *Two columns*: The two-column format is generated whenever the number of fields is equal to or greater than 12 or 24 and less than or equal to 22 or 46 (see Figure 12-2). If any field length in the record exceeds 26 characters, the screen format is altered to handle these fields.
- *Three columns*: The three-column format is generated whenever the number of fields is greater than 22 or 46 and less than or equal to 33 or 69 (see Figure 12-2). If any field length in the record exceeds 12 characters, the format is altered to allow these fields.
- *Four columns*: The four-column format occurs whenever the number of fields is greater than 33 or 69 and less than or equal to 44 or 80 (see Figure 12-2). If any field length in the record exceeds 6 characters, the format is altered to allow these fields. If the format is altered so that the four-column format is invalid, an error message is issued. If an error message is issued, you must reduce the number of fields in the record or change the order of the fields. Remember that all fields for the record must fit on one display screen format.

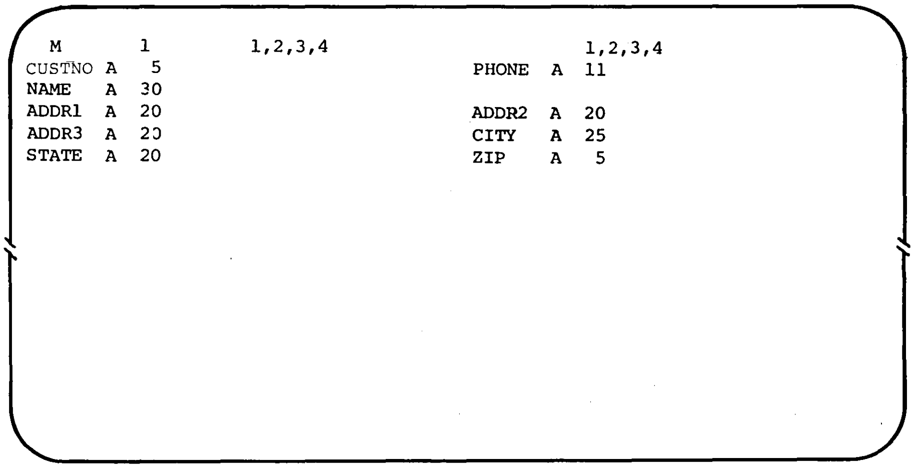


```
M        1          1,2,3,4                       1,2,3,4
ACCTNO A    5 _
DISCNT N 3.0
```

Record identification code for record being prompted (columns 21 through 27 of the input specifications)
Record identifying indicator for record being prompted (columns 19 and 20 of the input specifications)
Record identifying indicators for other record types that can be selected before data is entered using the current format
Record identifying indicators for other record types that can be selected after data is entered using the current format
Display station field control character for prompt, which appears as a blank
Field name (1 to 6 characters) from columns 53 through 58 of the input specifications
Type of field—alphameric (A) or numeric (N) (column 52 of the input specifications)
Field length (columns 44 through 51 of the input specifications)
Display station field control character for this input field which appears as a blank

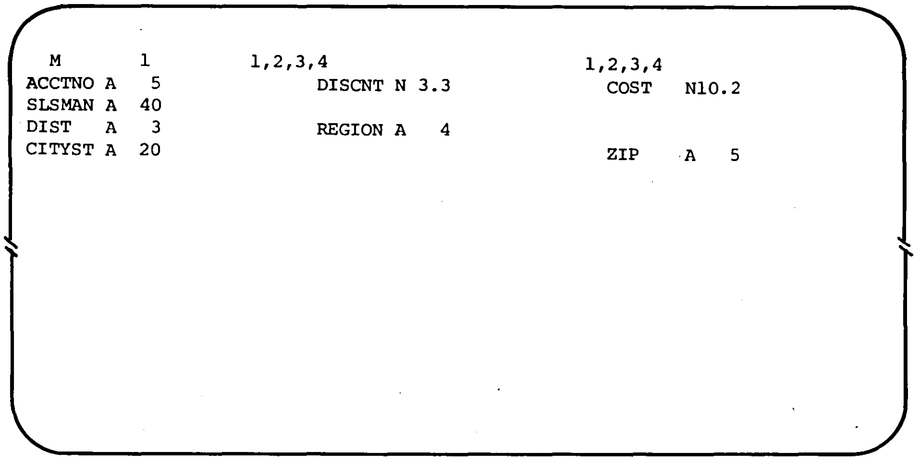
Figure 12-1. Display Generated for the CONSOLE File

Two-Column Format



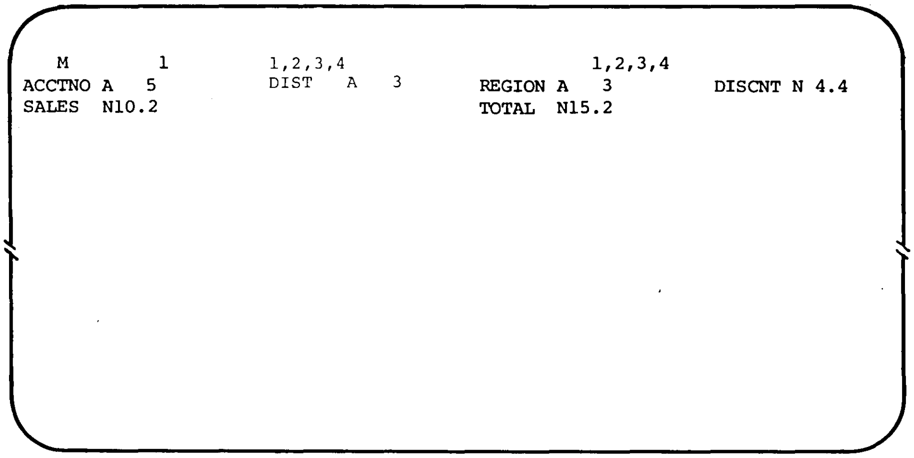
```
  M       1         1,2,3,4                      1,2,3,4
CUSTNO A   5                           PHONE  A  11
NAME   A  30
ADDR1  A  20                           ADDR2  A  20
ADDR3  A  20                           CITY   A  25
STATE  A  20                           ZIP    A   5
```

Three-Column Format



```
  M       1         1,2,3,4                      1,2,3,4
ACCTNO A   5              DISCNT N 3.3             COST   N10.2
SLSMAN A  40
DIST   A   3              REGION A   4
CITYST A  20                                       ZIP    A   5
```

Four-Column Format



```
   M        1         1,2,3,4                       1,2,3,4
ACCTNO A   5          DIST   A   3      REGION A   3          DISCNT N 4.4
SALES  N10.2                            TOTAL  N15.2
```

Figure 12-2. Display Screen Formats Generated by the $SFGR Utility Program

Prompt Format

The prompts, which are generated from the field names on the input specifications, are 14 positions long and have the following format:

Position	Explanation
1	Display station field control character for the prompt, which appears as a blank.
2-7	Field name.
8	Blank.
9	N for a numeric field or A for an alphameric field.
10-13	Length of field. For an alphameric field, positions 10 and 11 are blank and positions 12 and 13 contain the length. For a numeric field, positions 10 and 11 contain the length of the field, position 12 contains a decimal point, and position 13 indicates the number of decimal positions in the field.
14	Display station field control character for the input field, which appears as a blank.

Altering the Display Screen Format

Once the RPG II format generator has generated the source input for the screen format generator and the input has been cataloged in the library, you can alter this source input, if desired, by using SEU (see the *SEU Reference Manual*) and the FORMAT procedure (see the *System Support Reference Manual*). See *RPGR Command Statement* in Chapter 18 for an explanation of how to save the source input.

SPECIFICATIONS FOR A CONSOLE FILE

To use the CONSOLE file in a program, you must enter certain codes on the file description specifications and the input specifications.

File Description Specifications

Entries for the columns not included in the following list are described in Chapter 3, *File Description Specifications*.

Column 15: Enter an I in this column to specify this file as an input file.

Column 16: Enter a P (primary), S (secondary), D (demand), or R (record address) as the file designation.

Column 17: Leave this column blank if the program can end whether or not all records from the file have been processed, or enter an E if all records from the file must be processed before the program can end. If this column is blank for all files in the program, all records from every file must be processed before the program can end. For a CONSOLE file, the operator indicates a normal end of file by pressing command key 12, that is, the Cmd and = (equal) keys.

Columns 20-23: The entry for block length must either equal the record length entered in columns 24 through 27 or be blank.

Columns 24-27: The record length is defined as the highest to-field location specified on the input specifications. This record length cannot be less than 2 or greater than 1,518. If the CONSOLE file is designated as a record address file, determine the record length by multiplying the length of the record address field by 2. This record length cannot be less than 2 or greater than 58.

Column 28: Leave this column blank.

Columns 29-30: These columns must be blank if column 16 contains a P, S, or D. If column 16 contains an R, enter the length (1 to 29) of the record key.

Column 31: Use this column only for record address files. Leave the column blank if the keys in the record address file are the same as the keys in the index file. Enter an A for an indexed file with zoned decimal keys.

Columns 32-38: Leave these columns blank.

Column 39: Leave this column blank if column 16 contains a P, S, or D. If column 16 contains an R for record address file, this column must contain an E.

Columns 40-46: Enter CONSOLE as the device name. The file can be an input data file or a record address file. There can be only one CONSOLE file in a program.

Input Specifications

File and Record Identification Specifications

Entries for the columns not included in the following list are described in Chapter 7, *Input Specifications.*

Columns 14-16: These columns *must not* contain the characters AND; however, columns 14 and 15 can contain the characters OR. These OR lines can be used to indicate a relationship between record identifying indicators or record types. If columns 14 and 15 contain OR, the same number of record identification codes must be described on this specification line as are described on the preceding line.

Columns 15-18: RPG II uses the entries from columns 15 and 16 (sequence), column 17 (number), and column 18 (option) to determine which record types are allowable alternatives to the default record type (those record types that can be selected before data is entered for the record type currently displayed) and which record types are valid after data is entered for the record type currently displayed. RPG II inserts the valid *before* and *after* record types in the top line of the display format generated for a CONSOLE file (see Figure 12-1).

Columns 19-20: Enter a record identifying indicator (01 to 10) to define the command key the operator enters to select this record type. The same indicator cannot be used to define more than one record type within the input specifications for one program.

Columns 21-34: Position 1 or positions 1 and 2 of each record in the CONSOLE file must contain a record identification code to identify which record was keyed. The 1- or 2-character code specified in these columns is automatically inserted into each new record when it is prompted.

The rules for coding columns 21 through 34 are:

- *Columns 21-23:* Leave these columns blank.
- *Column 24*: Enter the number 1.
- *Column 25*: Leave this column blank.
- *Column 26*: Enter a C in this column.
- *Column 27*: Enter the alphabetic character, special character, or digit that is present in position 1 of the input record.
- *Columns 28-34*: If a 1-character record identification code is used, leave these columns blank. If a 2-character record identification code is used, code these columns the same as columns 21 through 27, except that column 31 must contain the number 2 to indicate record position 2.

Columns 35-74: Leave these columns blank.

Field Specifications

Entries for the columns not included in the following list are described in Chapter 7, *Input Specifications.*

Columns 14-16: AND lines are not allowed for CONSOLE file records.

Columns 53-58: Enter a descriptive field name (1 to 6 alphameric characters) to be used as a prompt for this data. To enter data into a whole array for a CONSOLE file, define the whole array as a subfield within a field of the CONSOLE file record or define each element of the array with an index and place this entry in columns 53 through 58. The index must be an integer value.

Columns 59-60: If this is a primary or secondary file, a control level indicator (L1 through L9) can be entered to indicate that a control break occurs on a change in the field's contents.

Columns 61-62: If this is a primary or secondary file, a match field value (M1 to M9) can be entered to indicate a match field. Otherwise, leave these columns blank.

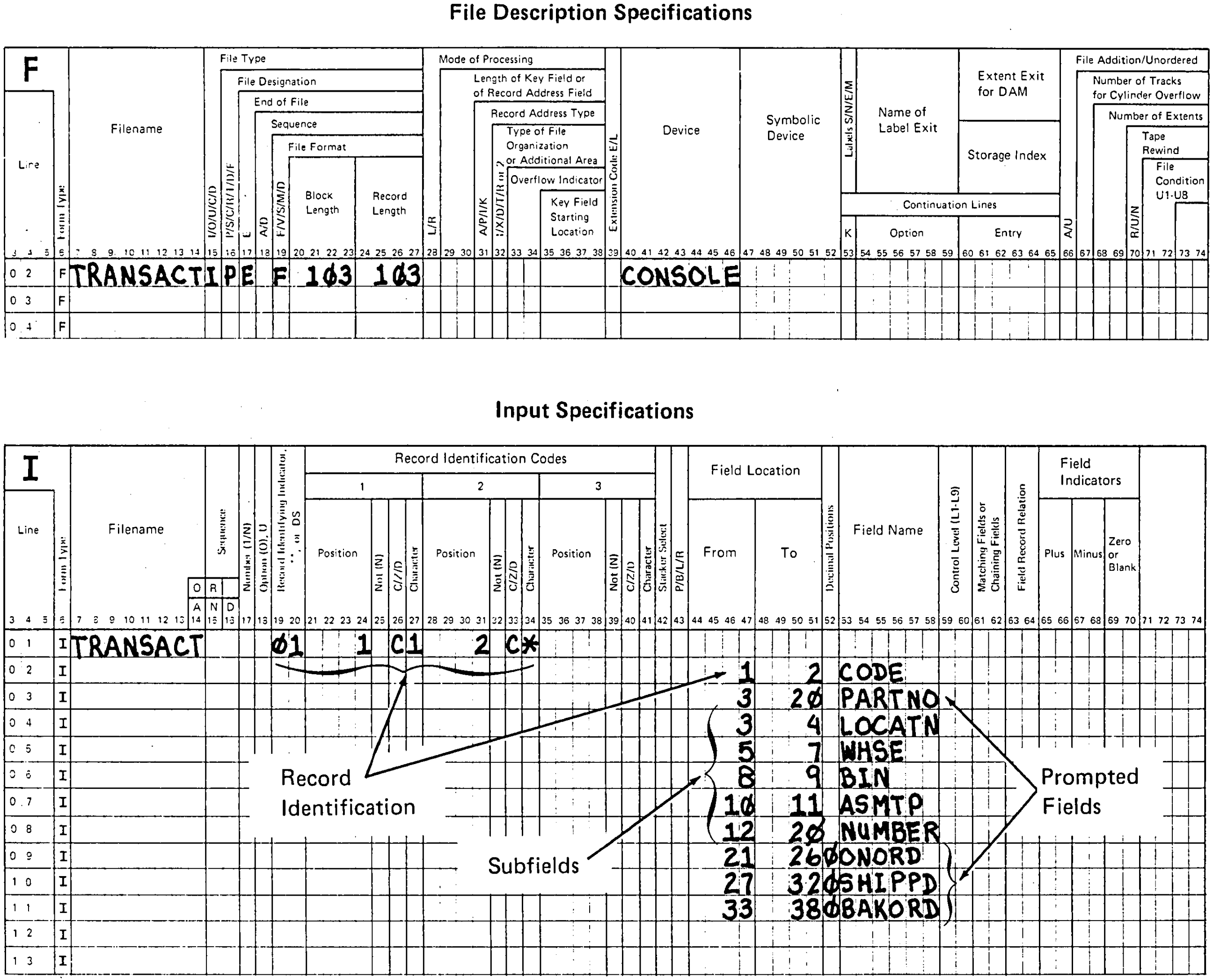
When describing the fields in a CONSOLE file record, remember the following considerations:

- The maximum alphameric field length is 66 characters.
- The maximum numeric field length is 15 digits.
- The maximum record length is 1,518 characters.
- Subfields can be specified within the fields of a CONSOLE file record. The from and to field locations for subfields must not overlap the from and to field locations for another field. Subfields are not prompted for, but are assigned values from the prompted field and can be used in calculation and output specifications. In Figure 12-3, the part number 01ROC43CP843987831 is entered in response to the prompt field PARTNO. LOCATN, WHSE, BIN, ASMTP, and NUMBER are subfields within the PARTNO field. The values for the subfields are extracted from the PARTNO field.

Erasing the CONSOLE File Buffer

To erase, or blank, the entire CONSOLE file buffer, use the ERASE function with the SET operation code in the calculation specifications. You must enter the operation code SET in columns 28 through 30, the filename of the CONSOLE file in columns 33 through 42, and ERASE in columns 43 through 47. ERASE indicates to the system that the buffer is to be set to blanks just before the program reads a record at the beginning of the next RPG II cycle.

Because the buffer is not erased until the beginning of the next RPG II cycle, processing of the current record continues after the ERASE operation is encountered. If the ERASE operation is executed because of invalid input data, you should insert code in your program to avoid further calculations and to return to the start of the RPG II cycle. A correct form of the record containing invalid input data and any records that were entered after that record can then be reentered.



File Description Specifications

Line	Form Type	Filename	File Type	File Designation	File Format	Block Length	Record Length	Device
0 2	F	TRANSACTI	P	E	F	1Ø3	1Ø3	CONSOLE
0 3	F							
0 4	F							

Input Specifications

Line	Form Type	Filename	Record Identifying Indicator	Position	C/Z/D	Character	Position	C/Z/D	Character	From	To	Decimal Positions	Field Name
0 1	I	TRANSACT	Ø1	1	C	1	2	C	*				
0 2	I									1	2		CODE
0 3	I									3	2Ø		PARTNO
0 4	I									3	4		LOCATN
0 5	I									5	7		WHSE
0 6	I									8	9		BIN
0 7	I									1Ø	11		ASMTP
0 8	I									12	2Ø		NUMBER
0 9	I									21	26	Ø	ONORD
1 0	I									27	32	Ø	SHIPPD
1 1	I									33	38	Ø	BAKORD
1 2	I												
1 3	I												

Prompted Field: PARTNO

01ROC43CP843987831

Subfields:

LOCATN:	01
WHSE:	ROC
BIN:	43
ASMTP:	CP
NUMBER:	843987831

Figure 12-3. Specifying Subfields for a CONSOLE File

Chapter 13. WORKSTN File Considerations and Sample Programs

The WORKSTN file allows concurrent entry of data from one or more devices attached to an executing RPG II program. The devices that can be attached to a WORKSTN file are display stations and SSP-ICF sessions. The number of devices that can be attached to a WORKSTN file is dependent on the value associated with the NUM continuation line option (for more information, see *Continuation Line Options* under *File Description Specifications* in this chapter).

WORKSTN can be specified as the input/output device if you use the display screen format specifications to define the display screen format. These specifications must then be compiled by the $SFGR utility program of the system support program product. Input and output fields, as well as the indicators that control the attributes of these fields, are defined on the display screen format specifications. The RPG II input and output specifications must relate input and output fields to the fields in the order defined on the display screen format specifications. The indicators defined on the display screen format specifications are controlled by the RPG II program. For a complete description of the $SFGR utility program, see the *System Support Reference Manual.*

A WORKSTN file must be specified when the Interactive Communications Feature (SSP-ICF) is used. The WORKSTN file requires no special coding when it is used for SSP-ICF. The file description entries are the same as those for a WORKSTN file that allows an RPG program to communicate with a display station. In this chapter, a device can be either a display station or an SSP-ICF session. For further information on SSP-ICF, see the *Interactive Communications Feature Reference Manual.*

Note: WORKSTN file processing is the same whether display stations or SSP-ICF sessions are attached to the file. Therefore, you should have a thorough understanding of WORKSTN file processing before attempting to use the Interactive Communications Feature.

PROGRAM ATTRIBUTES

When a WORKSTN file is specified, an RPG II program can be coded to support one device or multiple devices. The supported devices can be either acquired devices or requestor devices. A requestor is defined as the device that calls, or requests, the RPG II program. A program that allows only one requestor is called an SRT (single requestor terminal) program, and a program that allows multiple requestors is called an MRT (multiple requestor terminal) program.

The SRT and MRT attributes, which are assigned to a program on the COMPILE OCL statement, are external to the RPG II program. To change an SRT program to an MRT program, you must recompile the program and specify a value for the MRTMAX parameter. MRTMAX specifies the number of requestors that can be attached to the program. You must also add the NUM keyword on a continuation line for the WORKSTN file description specifications. NUM specifies the maximum number of devices that can be attached to the WORKSTN file at any one time. NUM must be equal to the MRTMAX value plus the maximum number of nonrequestor devices to be attached to the file at any one time.

For a more complete description of the SRT and MRT program attributes, see the *System Support Reference Manual.*

SRT (Single Requestor Terminal) Program

An SRT program is specified if the COMPILE OCL statement does not include an MRTMAX parameter when the program is compiled. Although an SRT program allows only one requestor per program, more than one device can execute the program concurrently. In this situation, the system support program product loads and initiates a *separate* copy of the program for each requesting device.

An SRT program can have multiple devices attached to it during execution. In this situation, the requestor calls the program, and other devices are acquired for the program by use of the WORKSTN OCL statement or the ACQ operation code. A display station can be acquired if it is not attached to an executing program and if it is not in command mode. An SSP-ICF session can be acquired if it is not already attached as a requestor. A device acquired by the WORKSTN OCL statement or by the ACQ operation code can be released from the program by the REL operation code or by an R entry in column 16 of the output specifications.

The NUM keyword is required on a file description specifications continuation line for an SRT program that contains any acquired devices. The NUM keyword for an SRT must specify a value greater than or equal to the number of nonrequestor devices plus one (for the single requestor).

When using the SRT program attribute, keep the following in mind:

- If the program is called by more than one requestor, each requestor has a separate copy of the program executing concurrently.
- The requestor provides the external indicators (U1 through U8).
- The requestor provides the display station local data area for the data structure defined by a U in column 18 of the input specifications.
- Program error messages go to the requestor.
- The program, including any acquired devices, is suspended via inquiry.

MRT (Multiple Requestor Terminal) Program

An MRT program is specified by the MRTMAX parameter on the COMPILE OCL statement. The value specified on the MRTMAX parameter limits the number of requesting devices that the program can process concurrently. An MRTMAX value of one is valid and means that only one device can request the program and that multiple copies of that program are not initiated when more than one requestor uses the same procedure to call the MRT program.

Note: Ordinarily different MRT procedures should not call the same MRT program because more than one copy of the MRT program could be in storage at the same time.

Each requestor of an MRT program is attached to the same program during execution. The first requestor of an MRT program causes the program to be loaded and initiated. Each succeeding requestor of an MRT program attaches to the executing program at the beginning of an input cycle or when a READ operation code is issued to the WORKSTN file. If the program is handling the maximum number of requestors (specified by the MRTMAX parameter), the next requestor of the program is queued by the system support program product. When the program releases one of its requestors, the program can process the queued request.

When using the MRT program attribute, keep the following in mind:

- If the program is called by more than one requestor, the first requestor:
  - Causes the program to be initiated
  - Provides external indicators (U1 through U8)
  - Provides the display station local data area for the data structure defined by a U in column 18 of the input specifications

- Each succeeding requestor after the first becomes attached to the program at input time for primary files or upon execution of a READ operation code to the WORKSTN file.

- Program error messages go to the system console operator.

- Requestors may leave the program without suspending the program or other devices.

- The display station local data area and external indicators can be made available to the program for each requestor if you use SUBR20 and SUBR21.

- When inquiry is used in an MRT program, the processing of information from the display station requesting the interrupt is suspended. However, the program continues to process information from other program-requesting devices.

ATTACHING A DEVICE TO A PROGRAM

A display station or SSP-ICF session can be attached to an RPG II program in one of two ways:

- As the requestor.

- Using the ACQ operation code in the calculation operations. If the device is not available or is already attached to the program, the indicator specified in columns 56 and 57 is set on.

A display station can be attached to an RPG II program in two additional ways:

- The procedure that is called to execute the RPG II program includes a WORKSTN OCL statement with the parameter REQD-YES. The display station is attached to the program by the system support program product and must be available for the program to be initiated.

- The procedure that calls the RPG II program contains a WORKSTN OCL statement with the parameter REQD-NO. RPG II attempts to acquire the display station during startup processing (see Figure 13-2). If the acquire attempt fails, the display station is deleted from the file.

If a WORKSTN program is initiated from the input job queue, a device must be attached either by an OCL statement or by the ACQ operation code. If the device is not attached by one of these methods, unexpected results can occur.

Note: A display station is available if it is not attached to any executing program and if it is not in command mode. An SSP-ICF session is available if it is not attached to any executing program.

WORKSTN FILE INPUT PROCESSING

Figure 13-1 shows the RPG II program logic cycle and the steps that are affected by WORKSTN file input processing. All steps in the cycle except steps 1, 3, 11 through 13, and 15 are the same as those for the regular RPG II program cycle.

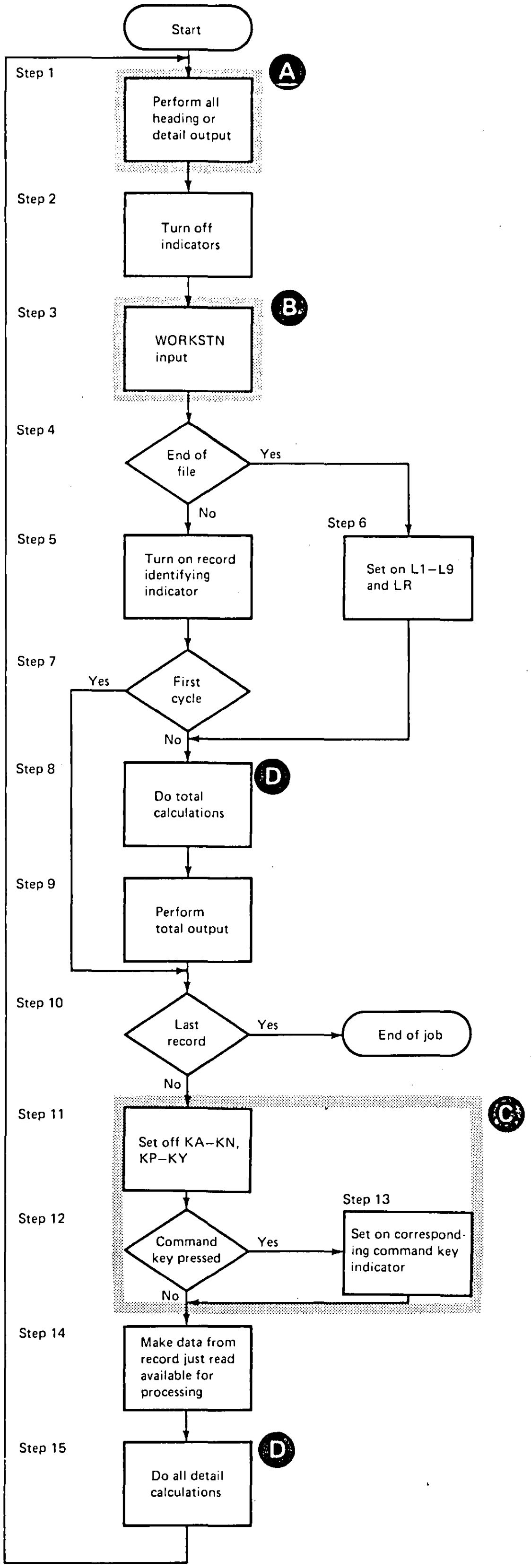


Figure 13-1. RPG II Program Logic Cycle With a WORKSTN File

A 1P output to a WORKSTN file is not allowed.

B WORKSTN input processing can include:

- Saving the common IND/SAVDS area in the IND/SAVDS area for the device from which the last input record was read (if specified).
- Getting the display station record. If the display station is new to the file, the record may be blank. Only the last input-capable display screen format can be read into the program.
- Restoring the IND/SAVDS area of the device from which the last input record was read (if specified).
- Inserting value into ID field (if specified).

For a detailed explanation of WORKSTN input processing, see Figure 13-2 for an expansion of step 3.

Data keyed at a display station is returned (input) to the RPG II program for processing when the operator presses a command key or the Enter/Rec Adv key. The operator can also cause the data to return to the program by pressing the Field Exit, Field +, or Field –key if the last input field in the format is defined as an auto record advance field (column 36 of the D specification).

C All command key indicators are turned off; then the appropriate one, if any, is turned on.

Note: If an exception/error occurs on the read, the command key indicators are not reset.

D If the READ operation code is used, it combines steps 3, 5, 11, 12, 13, and 14. If the EXCPT operation code is specified, it uses the ID field to direct output to the display station whose ID is contained in the field.

Step 3 of Figure 13-1, which is the WORKSTN input processing step, is expanded and shown in Figure 13-2. The following explanations refer to the steps shown in Figure 13-2.

Step 3-1. RPG determines whether the IND and/or SAVDS continuation line option is specified on the file description specifications for this file. If neither option is specified, RPG goes to step 3-3.

Step 3-2. If the IND and/or SAVDS option is specified, RPG moves the common IND/SAVDS area to the IND/SAVDS area for the device from which the last input record was read.

Step 3-3. RPG determines whether this is the first cycle for the first requestor of the program. If it is, RPG goes to step 3-10. All requestors of the program except the first enter the program at step 3-10.

Step 3-4. If the device is not a requestor, RPG determines whether all devices in the internal device table have been started. If all are started, the program goes to step 3-10.

A device is started when it has been acquired and a successful input or output operation has been executed. If a device is acquired via the ACQ operation code, the device is not considered to be started unless output is sent to the device in the same cycle.

Step 3-5. If not all devices have been started, RPG locates a device that has not been started.

Step 3-6. If the device located is a display station, RPG determines whether it has been acquired.

Step 3-7. If the display station has not been acquired, RPG calls work station data management to acquire the display station.

Step 3-8. RPG determines whether the acquire was successful. If it was not successful, RPG goes back to step 3-5.

Step 3-9. If the device is acquired, RPG creates a first-time blank record to satisfy the first read to the device. RPG then goes to step 3-11.

Step 3-10. RPG reads in the record from the device. Remember that all requestors of the program except the first enter the program at this point.

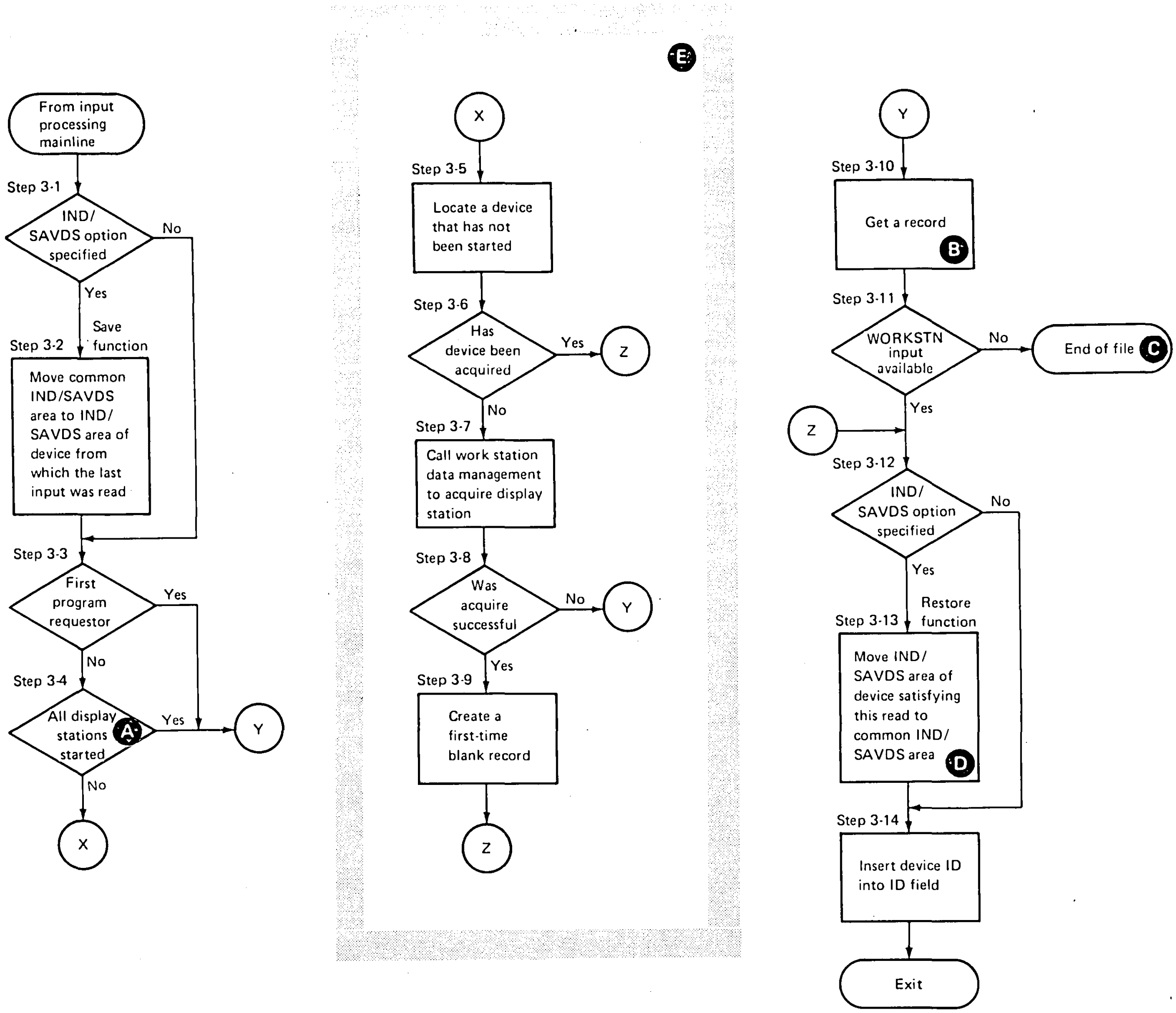
Step 3-11. RPG determines whether WORKSTN input is available. If it is not, end of file has been reached.

Step 3-12. RPG determines whether the IND and/or SAVDS continuation line option is specified on the file description specifications for this file. If an option is not specified, RPGgoes to step 3-14.

Step 3-13. If the IND and/or SAVDS option is specified, RPG moves the IND/SAVDS area for the device that satisfied the read to the common IND/SAVDS area.

Step 3-14. RPG inserts the device ID of the device that satisfied the read into the ID field if the ID continuation line option is specified on the file description specifications.

After the WORKSTN input file processing, RPG goes to step 4 as shown in Figure 13-1.


E
From input processing mainline
Step 3-1
IND/ SAVDS option specified
No
Yes
Save function
Step 3-2
Move common IND/SAVDS area to IND/ SAVDS area of device from which the last input was read
Step 3-3
First program requestor
Yes
No
Step 3-4
All display stations started
A
Yes
Y
No
X
X
Step 3-5
Locate a device that has not been started
Step 3-6
Has device been acquired
Yes
Z
No
Step 3-7
Call work station data management to acquire display station
Step 3-8
Was acquire successful
No
Y
Yes
Step 3-9
Create a first-time blank record
Z
Y
Step 3-10
Get a record
B
Step 3-11
WORKSTN input available
No
End of file
C
Yes
Z
Step 3-12
IND/ SAVDS option specified
No
Yes
Restore function
Step 3-13
Move IND/ SAVDS area of device satisfying this read to common IND/ SAVDS area
D
Step 3-14
Insert device ID into ID field
Exit

A One RPG cycle is used to start each nonrequestor, which is a device that has been acquired by the OCL statement // WORKSTN or by the RPG operation code ACQ. RPG checks each nonrequestor to see if an input or output operation to the device has previously been specified. If it has not, RPG physically acquires the device if necessary and creates a blank record to satisfy the first read.

B All requestors of the program except the first enter the program at this step.

Figure 13-2. WORKSTN File Input Processing

C End of file occurs for a WORKSTN file at the time of the read if:

- All devices have been released.
- Input is not allowed from any of the attached display stations because:
  - A new format has not been displayed at the display station since data was last keyed.
  - Suppress input-yes is specified in column 35 of the S specification for the format currently displayed and no input-capable formats are concurrently displayed at the display station.
- The program is an NEP, if all devices have been released, and if the operator has entered a STOP SYSTEM command.

D If this is the first input for this device, the indicators specified in the IND field are off and the SAVDS field is blank.

E Steps 3-5 through 3-9 occur only for nonrequestors.

End-of-File Considerations

The ways in which a primary or a demand WORKSTN file reaches end of file and causes the program to go to end of job (the LR indicator is on) are described in the following discussion. (For a description of the end-of-job steps that occur when a read under format is performed, see *Read Under Format* in this chapter.)

Primary WORKSTN File

For a primary file in a program that does not have an NEP attribute, end of file causes RPG II to set on the LR indicator in one of two ways.

- The LR indicator is turned on when all devices have been released. Devices can be released from the program in one of the following ways:
  - An R entry in column 16 of the output specifications. The device is released when the output specification is encountered during the output operations. If a format name is specified in the same specifications that contain an R in column 16, the format is displayed or the interactive communications operation is performed and then the device is released.
  - The REL operation code. The device specified in factor 1 is released from the program when the REL operation is encountered during the calculation specifications unless the device is the requestor of a single requestor program. If the device specified in factor 1 is the requestor of a single requestor program, the device is released at end of job, not when the operation is encountered during calculations. However, no subsequent operations to the device are allowed.

Note: If the device is a display station, it is still available for system log messages after it has been released.

- The LR indicator is turned on when input is not allowed from any attached device. If the device is a display station, input is not allowed when:
  - A new format has not been displayed since data was last keyed.
  - Suppress input-yes is specified in column 35 of the S specification for the last format displayed and no input-capable formats are concurrently displayed at the display station.

For information on when input is not allowed from an SSP-ICF session, see the *Interactive Communications Feature Reference Manual.*

If the programmer sets on the LR indicator in this type of program, the program goes to end of job.

If the program has an MRT-NEP attribute and all devices have been released or no input-capable formats are outstanding, end of file does not occur and RPG II does not set on the LR indicator until the system operator enters the STOP SYSTEM command. However, if the programmer sets on the LR indicator, the program goes to end of job (that is, the system operator does not have to enter the STOP SYSTEM command).

Note: An MRT program should not set on the LR indicator until end-of-file is reached for the WORKSTN file. If the LR indicator is set before end-of-file is reached, undesirable results occur for requestors that are in the sign on process.

Demand WORKSTN File

For a demand WORKSTN file in a program that does not have an NEP attribute, the programmer must set on the LR indicator. When all devices are released from a demand WORKSTN file or no input-capable formats are outstanding, the first READ operation after the last REL operation causes the READ end-of-file indicator (columns 58 and 59) to be set on. The programmer can then set on the LR indicator or the LR indicator can be specified as the READ end-of-file indicator.

If the program has an NEP attribute and all devices have been released or no input-capable formats are outstanding, the end-of-file indicator on the READ operation is not set on until the system operator enters a STOP SYSTEM command. However, if the programmer sets on the LR indicator based on some condition other than the end-of-file indicator on the READ operation, the program goes to end of job (that is, the system operator does not have to enter the STOP SYSTEM command).

WRITING A PROGRAM WITH A WORKSTN FILE

Creating a Display Screen Format

The first step in writing an RPG II program that contains the WORKSTN device is to define the display screen formats to be used. You can use the display screen layout sheet (GX21-9271) as shown in Figure 13-3 to lay out all the fields as they will appear on the display screen. You can then use the completed layout sheet as a guide for filling out the two parts (S specifications and D specifications) of the display screen format specifications (GX21-9253) as shown in Figure 13-4.

You can also use the Screen Design Aid (SDA) to interactively design the display screen formats. SDA allows you to design the appearance and specify the attributes of a new display screen format before the format is generated. SDA generates the display screen format specifications for you so you do not need to fill out the display screen format specification sheets. You can also specify an option of SDA that generates RPG II specifications. The specifications generated are the control, file description, input, and output specifications for the WORKSTN file:

- The control specifications have columns 52 and 53 (number of formats) completed.
- The file description specifications have the entries for a WORKSTN file completed.
- The input and output specifications include the input and output fields that correspond to the input and output fields specified for the display screen formats.

For a complete description of SDA, see the *SDA Reference Manual.*

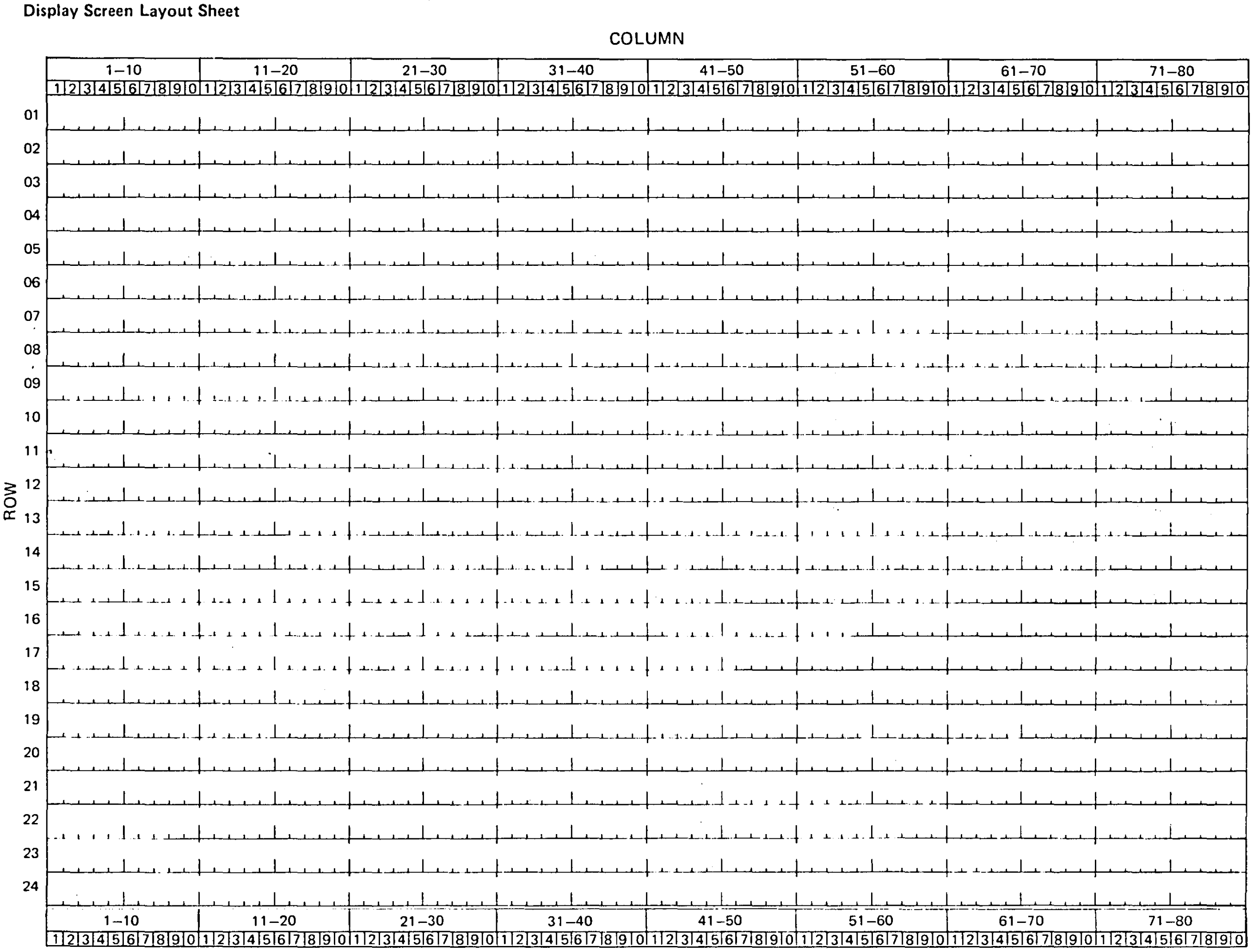
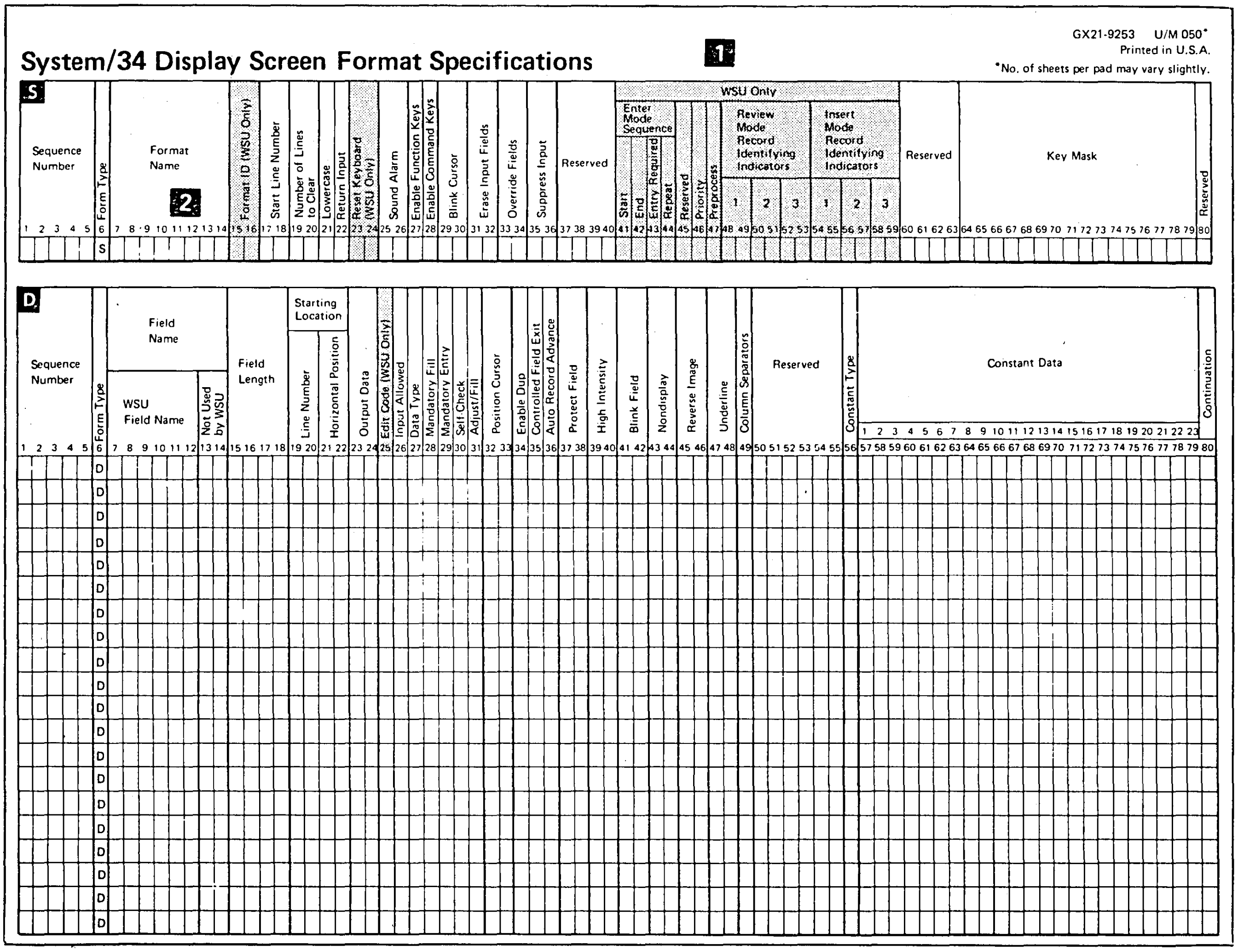


Figure 13-3. Display Screen Layout Sheet



GX21-9253 U/M 050*
Printed in U.S.A.
*No. of sheets per pad may vary slightly.

System/34 Display Screen Format Specifications 1

S

Sequence Number	Form Type	Format Name 2	Format ID (WSU Only)	Start Line Number	Number of Lines to Clear	Lowercase	Return Input	Reset Keyboard (WSU Only)	Sound Alarm	Enable Function Keys	Enable Command Keys	Blink Cursor	Erase Input Fields	Override Fields	Suppress Input	Reserved	WSU Only: Enter Mode Sequence Start	End	Entry Required	Repeat	Reserved	Priority	Preprocess	Review Mode Record Identifying Indicators 1	2	3	Insert Mode Record Identifying Indicators 1	2	3	Reserved	Key Mask	Reserved
1 2 3 4 5	6	7 8 9 10 11 12 13 14	15 16	17 18	19 20	21	22	23 24	25 26	27	28	29 30	31 32	33 34	35 36	37 38 39 40	41	42	43	44	45	46	47	48 49	50 51	52 53	54 55	56 57	58 59	60 61 62 63	64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79	80
	S																															

D

Sequence Number	Form Type	Field Name: WSU Field Name	Not Used by WSU	Field Length	Starting Location: Line Number	Horizontal Position	Output Data	Edit Code (WSU Only)	Input Allowed	Data Type	Mandatory Fill	Mandatory Entry	Self-Check	Adjust/Fill	Position Cursor	Enable Dup	Controlled Field Exit	Auto Record Advance	Protect Field	High Intensity	Blink Field	Nondisplay	Reverse Image	Underline	Column Separators	Reserved	Constant Type	Constant Data	Continuation
																												1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23	
1 2 3 4 5	6	7 8 9 10 11 12	13 14	15 16 17 18	19 20	21 22	23 24	25	26	27	28	29	30	31	32 33	34	35	36	37 38	39 40	41 42	43 44	45 46	47 48	49	50 51 52 53 54 55	56	57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79	80
	D																												
	D																												
	D																												
	D																												
	D																												
	D																												
	D																												
	D																												
	D																												
	D																												
	D																												
	D																												
	D																												
	D																												
	D																												
	D																												
	D																												
	D																												
	D																												
	D																												

Notes:

1 The shaded columns are not used by $SFGR to generate the screen formats to be used by an RPG II program.

2 The display screen format name used in columns 7 through 14 of the S specifications must be entered on the output specifications beginning in column 45. The format name in the output specifications must be enclosed in apostrophes.

Figure 13-4. Display Screen Format Specifications

Each field in the record must be described on the display screen format specifications. You can specify the following three types of fields:

- An *output field* contains data that cannot be changed by the operator. For a normal output operation, the data in the field is supplied either as part of the field definition when the format is generated or is supplied by the RPG II program (and must be described on the output specifications) when the format is displayed. Whether the output data is supplied by the field definition or by the RPG II program depends on the status of the indicator specified in columns 23 and 24 (output data) of the D specifications. Output data is supplied as follows:
  - If the indicator specified in columns 23 and 24 (output data) is on, the output data comes from the RPG II program.
  - If the indicator specified in columns 23 and 24 (output data) is off, the output data comes from the constant specified in columns 57 through 79 of the D specification. This constant is often blanks.
  - For an explanation of the relationship between the indicator in columns 23 and 24 (output data) and the indicator in columns 33 and 34 (override fields) of the S specification, see *Overriding Fields in a Format* in this chapter.
- An *input field* is a field on the display reserved for keyboard entry. Data for the field is entered by the display station operator. This field type is described in RPG II on the input specifications.
- An *output/input field* is both displayed (output) on the screen and read (input) back into the program. This field type is described in RPG II on both the input and output specifications.

All three types of fields can be used in the same display screen format.

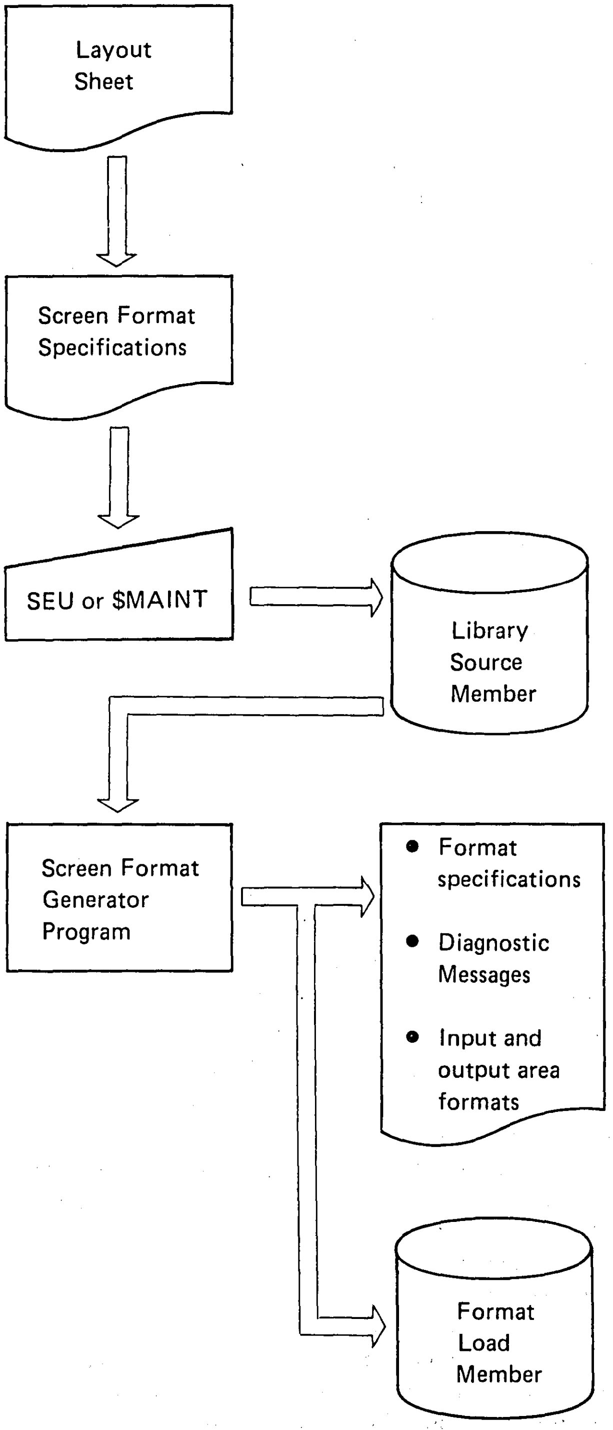
In addition to specifying the type of field, you can specify the type of data that can be contained in the field. The types of data that can be specified are:

- Alphabetic only
- Numeric only. If special characters are entered in an N-type field, the data read by the RPG program may not be as expected. The program uses only the digit portion of characters entered in an N-type field. The zone portion is forced to hex F, except for the sign position (see *Zoned Decimal Format* in Chapter 7).
- Alphameric
- Signed numeric (the last position of the field is reserved for a sign). For signed numeric data, the length of the field as specified on the display screen format specifications must be one more than the length specified on the RPG II specifications.

You can also specify other attributes of the field, such as display intensity, whether the field is a blinking field, and whether data must be entered into the field by the operator.

On the S specification, you can indicate the number of lines to be cleared before the format is displayed. You can also specify a variable starting line number for the format. For an example of these two functions, see the sample program shown in Figure 13-14. For a complete description of the entries that can be made on the display screen format specifications, see the $SFGR utility program in the *System Support Reference Manual.* For a summary of the possible entries, see Appendix A in this manual.

When you have completed the display screen format specifications, use the $MAINT utility program or SEU to enter the specifications into a source member. Then run the screen format generator program ($SFGR utility program) to process the source specifications. For the display screen format load member name, the compiler uses the name specified as the value for the FMTS continuation line option. If the FMTS continuation line option is not specified, the compiler uses the characters specified in columns 75 through 80 of the control specifications (the program name) and adds the characters FM to the end of the program name. The format name cannot be changed after compilation. Figure 13-5 shows the steps for creating a display screen format.



1. Arrange all fields on the layout sheet, just as they will appear on the display screen.

2. Use the completed layout sheet as a guide for filling out the two parts of the display screen format specifications.

3. Use SEU or $MAINT to transfer the specifications to a library source member.

4. Run the screen format generator utility program ($SFGR) by entering the required control statements or by running the FORMAT procedure. The screen format generator program produces the following output:
   - List of source specifications
   - Diagnostic messages
   - List of information about the input and output records for the format
   - The display screen format (or formats) in the specified load member

Figure 13-5. Steps for Creating a Display Screen Format

Restrictions for Using a WORKSTN File

The following restrictions apply to the use of a WORKSTN file in an RPG II program:

- Only one WORKSTN file is allowed per program.

- A program cannot contain a KEYBORD, CRT, or CONSOLE file if it contains a WORKSTN file.

- The WORKSTN file must be specified as a combined file (capable of both input and output).

- If the WORKSTN file is specified as a primary file, no secondary files are allowed in the program.

- Control level indicators, match field values, and look-ahead fields are not allowed.

- The first page indicator (1P) is not allowed.

- Packed and binary data should not be sent to a display station as output data. Such data can contain IBM 5251 Display Station control characters. The presence of 5251 control characters can cause undesirable results. Packed and binary data can be sent to an SSP-ICF session.

- For an SRT program, output can precede input from the requestor in one of three ways:
  - No ID is specified.
  - The ID field contains the ID of the requestor.
  - The ID field contains blanks.

- For an MRT program, the first input/output operation issued to a requestor of the program must be an input operation.

- For an acquired device, output can precede input if the ID of the device is first placed in the ID field.

Command Keys

All 24 command keys can be used for a WORKSTN file. When RPG makes the data read from a display station available for processing, all command key indicators are set off. If a command key was pressed when the data was read into the program, the corresponding command key indicator is set on. You can then use the command key indicator to condition calculation and output operations.

For example, you can specify that the operator press command key 2 (rather than the Enter/Rec Adv key) when the last item for an invoice has been keyed at the display station. You can then use command key indicator KB in the program to condition calculation operations and output operations, such as presenting the next screen to the operator.

You can use the display screen format S specifications to selectively enable or disable certain command keys for your program. For a description of how to selectively enable or disable command keys, see column 28 of the S specifications summary in Appendix A or see the *System Support Reference Manual.* For a discussion of how to determine whether a command key was pressed, see *Specifications for the INFDS Data Structure* in this chapter.

To document the use of the command keys for the operator, you can use the template assignment form on the IBM 5251 Display Station Keyboard Template Assignment Sheet and Display Screen Layout Sheet.

Control Specifications

Columns 52 and 53 of the control specifications must specify the number of formats contained in the display screen format load member. This number must include all the formats in the display screen format load member, not just the number of formats used by the program. If no entry is made in columns 52 and 53, the program assumes an entry of 32. The entry in columns 52 and 53 can be greater than the actual number of formats in the load member but not greater than 32.

File Description Specifications

The following file description specifications must be made for a WORKSTN file. If multiple display stations are attached to the WORKSTN file or if you want to specify the file information data structure (INFDS) or the WORKSTN exception/error processing subroutine (INFSR), continuation lines can also be specified. The continuation lines are described after the file description specifications.

Column	Entry
6	F (the form type).
7-14	A valid filename.
15	Must contain a C for combined file.
16	Must contain a P for primary file or a D for demand file.
17-18	Must be blank.
19	Must contain an F or be blank for fixed-length record.
20-23	Must be blank. The block length equals the record length.
24-27	The length of the longest record, which is equal to the highest end position specified on the input or output specifications.
28-39	Must be blank.
40-46	Device name WORKSTN.
47-70	Must be blank.
71-72	File conditioning indicators (U1 through U8), if needed.
73-74	Must be blank.
75-80	Can contain the program identification.

For an example of the file description specifications, see the sample programs at the end of this chapter.

Continuation Line Options

Several options can be specified for a WORKSTN file on the file description specifications continuation lines. These options can be specified if more than one device is attached to a WORKSTN file or if you want to specify the file information data structure (INFDS) or the WORKSTN exception/error processing subroutine (INFSR). To specify a continuation line, enter a K in column 53, the option keyword in columns 54 through 59, and the value in columns 60 through 65 (columns 60 through 67 can be used if the FMTS option is specified). For examples of specifying continuation line options, see Figure 13-6.

The following keywords and values can be specified in columns 54 through 59 and 60 through 65 respectively:

Keyword Value

NUM Maximum number of devices that can be attached to this file at the same time. The number specified must be right-justified in columns 60 through 65. If not specified, 1 is assumed. If specified, NUM must be greater than or equal to the number of requestors specified by the MRTMAX parameter on the COMPILE OCL statement plus the number of acquired devices (those specified on the WORKSTN OCL statement or in the ACQ operation). The number specified on the MRTMAX parameter is reserved for requestors. The difference between the MRTMAX value and the NUM value is the maximum number of devices (nonrequestors) that can be attached to the program at one time via OCL statements or the ACQ operation code. For example, if the MRTMAX value is 5 and the NUM value is 6, only one nonrequestor device can be attached to the program, even if only one requestor is presently signed on.

Note: If NUM is greater than 1, use caution when updating a file (see *Update Files* in Chapter 3).

Keyword Value

SAVDS Name of a data structure that is to be saved and restored for each device attached to this file. The data structure cannot be a display station local data area, a compile-time array, or a preexecution-time array. If SAVDS is not specified, no data area swapping is done.

This data structure can contain information that is unique to each display station. For example, it can contain a field that is used to accumulate the number of records read or to store a field that is not used until later cycles, such as credit limit.

IND Number of indicators, beginning with 01, that are to be saved and restored by the display station. Indicators 01 through the number specified here are saved and restored for each display station. If IND is not specified or if NUM equals 1, no indicator swapping is done. The entry must be right-justified in columns 60 through 65.

For example, error indicators or indicators used for security clearance may be unique by display station.

Note: For SAVDS and IND, only one copy of the data structure and indicators is available at a time. The indicators and data structure that are available are those associated with the device from which the last input was read.

The data structure and indicators that are available change each time an input operation (either a primary file input or a demand file read) is executed. On an input operation, the present copy of the data structure and indicators in the program is written to a save area for the device from which the previous input was read. The data structure and indicators for the device now being read from are then written from the save area associated with the device to the program SAVDS and IND areas. For further information, see Figure 13-2 in this chapter.

Keyword **Value**

SLN Name of a two-digit numeric field whose value determines the first line on the display screen where the display screen format is to begin if a variable starting line number was specified (V in column 17 of the display screen format S specifications). If SLN is not specified, all formats having a variable starting line number will begin on line 01.

FMTS *NONE. Indicates that there are only SSP-ICF formats present in this program.

Display screen format load member name. The compiler uses the name entered here as the display screen format load member name. The format load member name entered here can be from 1 to 8 characters in length, left-justified in columns 60 through 67. If a name is not entered, the compiler assumes the display screen format load member name is the program name (from columns 75 through 80 of the control specifications) with FM added to the end of the name.

ID Name of a 2-character self-defining alphameric field (that is, the field does not have to be specified as an input or result field) that contains the ID of the device that supplied the record currently being processed in this file. The ID field is updated whenever a record is read from the WORKSTN file. Therefore, it always contains the identification of the device from which the last record was read (unless your program moves a different identification into the ID field). You can direct output to different devices in a multiple WORKSTN file by changing the value in this field to the symbolic ID of another device in the file.

Display station IDs are in the form AX, where A is an alphabetic character (A-Z, #, @, or $) and X is any character. If a WORKSTN OCL statement exists for the display station, the ID is the same as the value of the SYMID parameter.

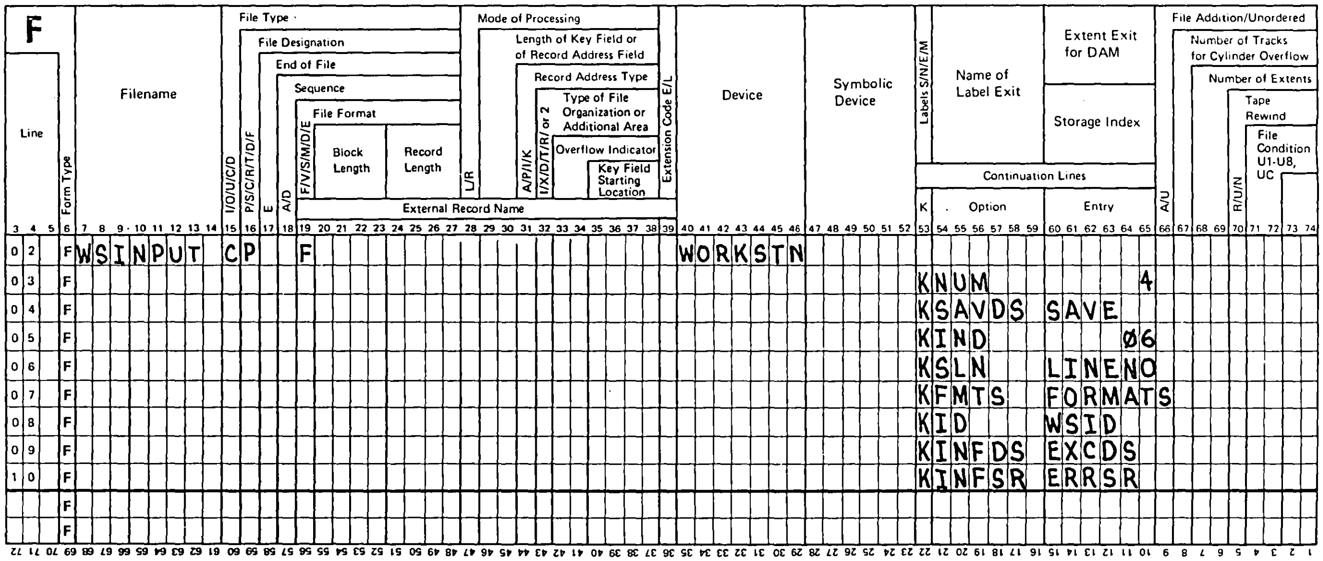
Keyword **Value**

SSP-ICF session IDs can be in two formats. They are either NN where N is numeric (0-9), or NA where N is numeric and A is alphabetic (A-Z, #, @, or $). If the format is NA, a SESSION OCL statement must be specified with a SYMID parameter whose value is also in an NA format.

INFSR Name of the user-written calculation subroutine that may receive control when an exception/error occurs. This subroutine, referred to as the WORKSTN exception/error processing subroutine, is written by the RPG II programmer to handle the WORKSTN exception conditions or input/output errors in the manner best suited for the application program. The following WORKSTN operations transfer control to the INFSR subroutine: ACQ, REL, NEXT, POST, input (READ operation or primary input), or output (EXCPT or normal cycle output). If INFSR is not specified, error recovery is handled by RPG II. See *WORKSTN Exception/Error Handling* in this chapter for more information on INFSR.

INFDS Name of the data structure that contains the unique identification of the exception/error and the WORKSTN operation that caused the exception/error condition. This data structure is referred to as the WORKSTN file information data structure. The INFDS data structure also contains the display screen size (*SIZE), either 960 or 1920 characters, and information about ideographic support. If the INFDS data structure is not specified, this information is not available to the RPG II program. See *WORKSTN Exception/Error Handling* in this chapter for more information on INFDS.

File Description Specifications



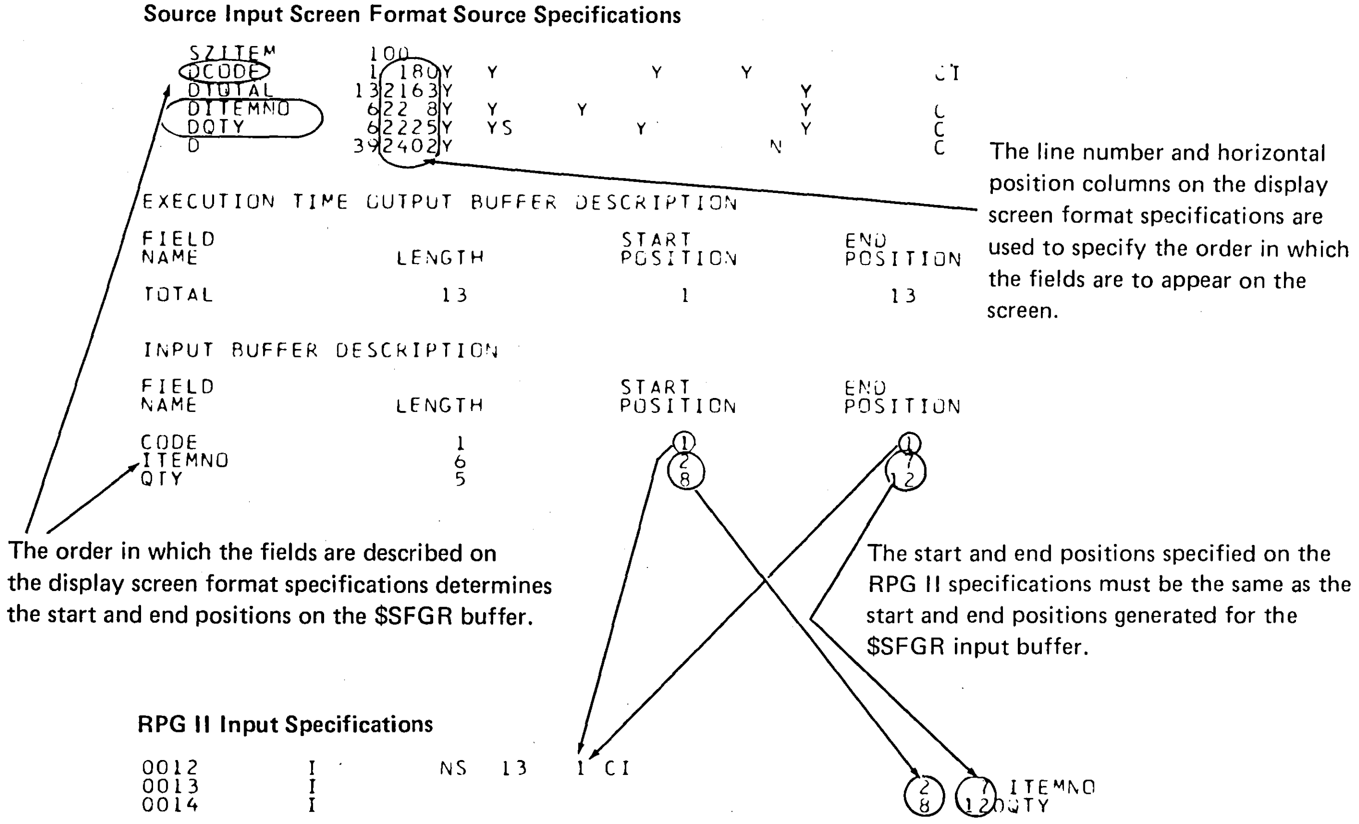
Line	Form Type	Filename	File Type	File Designation	File Format	Device	K	Option	Entry
02	F	WSINPUT	C	P	F	WORKSTN			
03	F						K	NUM	4
04	F						K	SAVDS	SAVE
05	F						K	IND	Ø6
06	F						K	SLN	LINENO
07	F						K	FMTS	FORMATS
08	F						K	ID	WSID
09	F						K	INFDS	EXCDS
10	F						K	INFSR	ERRSR
	F								
	F								

Figure 13-6. WORKSTN Continuation Line Options

Input Specifications

Entries for the input specifications are described in Chapter 7. However, the following considerations apply when the input file is a WORKSTN file:

- Control level indicators (columns 59 and 60), match field values (columns 61 and 62), and look-ahead fields (columns 19 and 20) are not allowed.
- The field location entries in columns 44 through 51 refer to the location of the field in the input record, and in the case of a display station, not to the location of the field as it is displayed on the screen. The input fields are placed in the input record in the order in which they are described on the display screen format specifications. However, the line number and horizontal position columns on the display screen format specifications can be used to change the order in which the fields appear on the screen. See Figure 13-7 for an example of the relationship between the display screen format specifications and the RPG II input specifications.
- Each record including the blank record at the first read to a device should be identified on the input specifications. For display screens, specify a nondisplay, protected output/input field on each display screen format for the record code.
- The first input record read from a device is blank unless PDATA-YES is specified in the procedure that executes the program; unless the program is a requestor, other than the first, of an MRT program (see the $MAINT utility program in the *System Support Reference Manual)*; or unless a read under format is performed (see *Read Under Format* in this chapter).



Source Input Screen Format Source Specifications

```
  SZITEM        100
  DCODE        1   180Y   Y          Y      Y                CI
  DTOTAL      132163Y                              Y
  DITEMNO      622  8Y   Y       Y                 Y          C
  DQTY         622225Y   YS         Y              Y          C
  D           392402Y                         N               C

EXECUTION TIME OUTPUT BUFFER DESCRIPTION

FIELD                                  START                END
NAME                 LENGTH            POSITION             POSITION

TOTAL                  13                 1                   13

INPUT BUFFER DESCRIPTION

FIELD                                  START                END
NAME                 LENGTH            POSITION             POSITION

CODE                    1                 1                    1
ITEMNO                  6                 2                    7
QTY                     5                 8                   12
```

The line number and horizontal position columns on the display screen format specifications are used to specify the order in which the fields are to appear on the screen.

The order in which the fields are described on the display screen format specifications determines the start and end positions on the $SFGR buffer.

The start and end positions specified on the RPG II specifications must be the same as the start and end positions generated for the $SFGR input buffer.

RPG II Input Specifications

```
0012        I          NS   13      1 CI
0013        I                                          2    7 ITEMNO
0014        I                                          8   12 QTY
```

Note: For the complete program in which this format is used, see *WORKSTN Sample Programs* later in this chapter.

Figure 13-7. Relationship Between the RPG II Input Specifications and the $SFGR Input Buffer

Calculation Specifications

Three operation codes (see Figure 13-8) are unique to a WORKSTN file:

- ACQ acquires the device specified in factor 1 for the program. Factor 2 must contain the filename for the WORKSTN file. If the device is available, ACQ attaches it to the program. If it is not available or is already attached to the program, the indicator specified in columns 56 and 57 is set on. However, if no indicator is specified in columns 56 and 57 but the program contains the INFSR (WORKSTN exception/error processing) subroutine, the INFSR subroutine automatically receives control when an exception/error occurs on the ACQ operation. If no indicator is specified and the program does not contain the INFSR subroutine, the programs halts when an exception/error condition occurs. The operator can then continue the job or retry the ACQ operation. No input or output operation occurs when the ACQ operation is executed.

- REL releases the device specified in factor 1 from the program. Factor 2 must contain the filename for the WORKSTN file. Either a requesting or a nonrequesting device can be released with the REL operation code. The specified device is released when the REL operation is encountered during calculations unless the device is the requestor of a single requestor program. If the device specified in factor 1 is the requestor of a single requestor program, the device is released at end of job, not when the operation code is encountered in the calculations.

  Note: If the device is a display station, the display station is no longer available to the program. The display station is available for system log messages.

  An indicator can be specified in columns 56 and 57 and is set on if an exception/error occurs on the REL operation. If columns 56 and 57 do not contain an indicator and an exception/error occurs, the program halts unless the INFSR subroutine is specified. If the INFSR subroutine is specified, the subroutine automatically receives control when an exception/error occurs. (For more information on the INFSR subroutine, see *WORKSTN Exception/Error Handling* in this chapter.)

- NEXT forces the next input to the program to come from the device specified in factor 1. Factor 2 must contain the filename for the WORKSTN file. If NEXT is specified more than once between input operations, only the last operation is executed. An indicator can be specified in columns 56 and 57. This indicator is set on if an exception/error occurs on the NEXT operation. If columns 56 and 57 do not contain an indicator and an exception/error occurs, the program halts unless the INFSR subroutine is specified. If the INFSR subroutine is specified, the subroutine automatically receives control when an exception/error occurs. (For more information on the INFSR subroutine, see *WORKSTN Exception/Error Handling* in this chapter.)

The READ operation code can also be used for a WORKSTN file. For an SRT program, output can precede input from the requestor in one of three ways:

- No ID is specified.

- The ID field contains the ID of the requestor.

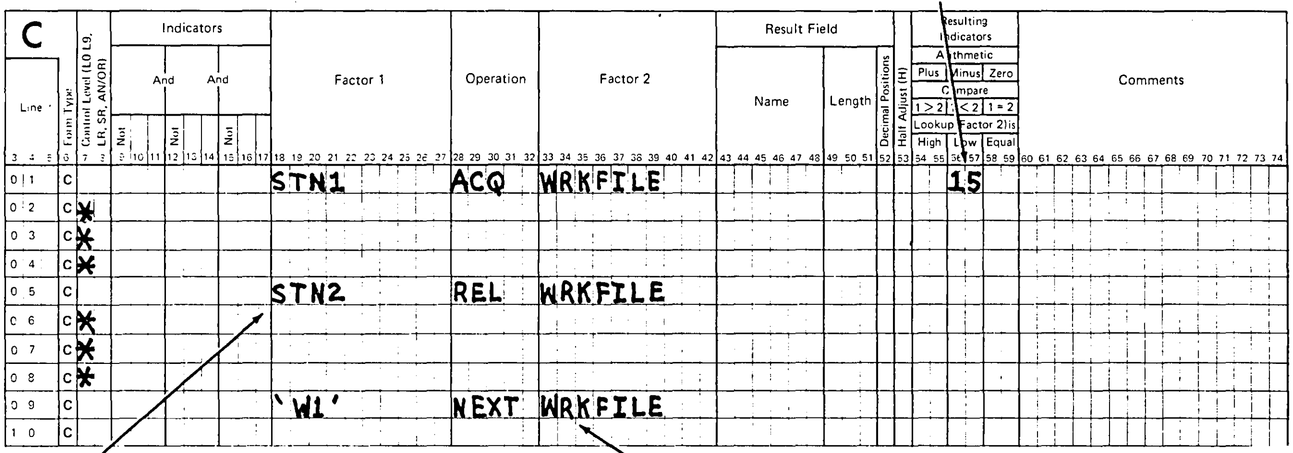
- The ID field contains blanks.

For an MRT program, the first input/output operation issued to a requestor of the program must be an input operation.

For an acquired device, output can precede input if the ID of the device is first placed in the ID field.

Indicator 15 turns on if the device identified in the field STN1 cannot be acquired.

Calculation Specifications



C		Control Level (L0-L9, LR, SR, AN/OR)	Indicators			Factor 1	Operation	Factor 2	Result Field		Decimal Positions	Half Adjust (H)	Resulting Indicators			Comments
				And	And				Name	Length			Arithmetic			
													Plus	Minus	Zero	
													Compare			
													1 > 2	1 < 2	1 = 2	
													Lookup (Factor 2) is			
Line	Form Type		Not	Not	Not								High	Low	Equal	
3 4 5	6	7 8	9 10 11	12 13 14	15 16 17	18 19 20 21 22 23 24 25 26 27	28 29 30 31 32	33 34 35 36 37 38 39 40 41 42	43 44 45 46 47 48	49 50 51	52	53	54 55	56 57	58 59	60 61 62 63 64 65 66 67 68 69 70 71 72 73 74
0 1	C					STN1	ACQ	WRKFILE						15		
0 2	C	*														
0 3	C	*														
0 4	C	*														
0 5	C					STN2	REL	WRKFILE								
0 6	C	*														
0 7	C	*														
0 8	C	*														
0 9	C					'W1'	NEXT	WRKFILE								
1 0	C															

The device identified by the field STN2 is released from the program.

NEXT forces the next input to come from device 1 (identified by the alphameric literal W1).

Figure 13-8. WORKSTN Operation Codes

Output Specifications

Entries for the output specifications are described in Chapter 9. However, the following considerations apply when the output file is a WORKSTN file:

- The first page (1P) indicator is not allowed.
- The end position entry in columns 40 through 43 refers to the end position of the field in the output record. For display stations it does not refer to the end position of the field as it appears on the screen. Use the output from the $SFGR utility program as a guide when coding the output specifications for display stations (see Figure 13-9).
- The output fields start in position 1.
- For display stations, the fields must be described on the output specifications in the same order as they are described on the display screen format specifications.
- A device can be released after the output is performed if you place an R in column 16. If OR lines are specified, column 16 must contain an R for each line. The device is released when that output specification is encountered during the output operations. If a format name is specified in the same specification line that contains an R in column 16, the format is displayed or the interactive communications operation is performed and then the device is released.
- The display screen format name from columns 7 through 14 of the display screen format specifications for this program or the predefined SSP-ICF format name must be specified as a constant in columns 45 through 54. The line containing the format name cannot be conditioned by any indicators. You must also enter Kn in columns 42 and 43, where n is the length of the format name. For example, if the format name is FORM1, enter K5 in columns 42 and 43 and 'FORM1' in columns 45 through 51.
- One format name is required for each output record for the WORKSTN file. The specification of more than one format name per record is not allowed.

Note: When the RPG II Z edit code is used for an output field in a WORKSTN file and the value of the field is zero, RPG II sends a blank field to work station data management, which places a zero in the rightmost position of a signed numeric field.

WORKSTN EXCEPTION/ERROR HANDLING

The WORKSTN exception/error processing subroutine (INFSR) and error indicators in columns 56 and 57 of the WORKSTN operation codes (REL, ACQ, NEXT, POST, and READ) allow the programmer to control the program logic if exception/error conditions occur during WORKSTN file processing. The WORKSTN file information data structure (INFDS) contains status information that the programmer can check to determine what type of exception/error occurred. Using the information in the INFDS, the programmer can then determine which exception/error conditions he wants to handle in the INFSR subroutine and which exception/error conditions he wants RPG II to handle.

If neither the INFSR subroutine nor error indicators are specified, an exception/error is handled by the RPG II error handling routine, which causes a program to halt and the operator must choose the appropriate option.

Exception conditions refer to the input of an enabled function control key (Print, Roll Up, Roll Down, Clear, Help, or Record Backspace) to the program. The display screen format S specifications must be used to enable these function control keys for an RPG program. To enable the function control keys, the program must also contain the INFDS data structure, which will contain an indication of the exception/error condition, and either the INFSR subroutine or an error indicator specified in columns 56 and 57 of a READ operation. If the program does not contain the INFDS data structure, the program cannot determine whether one of the function control keys was pressed. No automatic programmed function is associated with the function control keys. The programmer can test the INFDS to determine whether a function control key was pressed and then use that indication as a signal from the operator to perform certain routines in the program. For information on how the function control keys are enabled, see *Enabling/Disabling Function Control Keys* in this chapter.

Error conditions refer to input/output errors that occur during an implicit input/output operation (such as a primary file read, EXCPT output, or normal cycle output operations) or during an explicit input/output operation (such as ACQ, REL, NEXT, or READ).

Source Input Screen Format Source Specifications

```
SSHOWITEM 100
D         8 5 3Y                   Y       ITEM NO.
D        22 514Y                   Y       DESCRIPTION
D         9 541Y                   Y       PRICE
D         6 553Y                   Y       ONHAND
D         6 562Y                   Y       SOLD
DITEMNO   6 6 3Y
DONHAND   6 653Y
DPENDNG   6 662Y
DPRICE    9 641Y
DDESC    22 614Y
```

The line number and horizontal position columns on the display screen format specifications can be used to change the order in which the fields appear on the screen.

```
EXECUTION TIME OUTPUT BUFFER DESCRIPTION

FIELD                              START         END
NAME              LENGTH           POSITION      POSITION

ITEMNO               6                 1             6
ONHAND               6                 7            12
PENDNG               6                13            18
PRICE                9                19            27
DESC                22                28            49
```

The end positions generated for the $SFGR output buffer and the end positions specified on the RPG II output specifications must be the same for each field.

RPG II Output Specifications

```
0017      O       D           03N99
0018      O                                  K8 'SHOWITEM'
0019      O                         ITEMNO     6
0020      O                         ONHANDL   12
0021      O                         PENDNGL   18
0022      O                         PRICE 2   27
0023      O                         DESC      49
```

The fields on the display screen format specifications must be described in the same order as specified by the end positions in the output specifications.

Note: For the complete program in which this format is used, see *WORKSTN Sample Programs* later in this chapter.

Figure 13-9. Relationship Between the RPG II Output Specifications and the $SFGR Output Buffer

The INFDS data structure, if specified, contains an identification of the exception/error that occurred and an identification of the WORKSTN operation for which the exception/error occurred. The INFDS also contains status information on normal conditions (not exceptions or errors) such as whether a command key was pressed, whether end of file has occurred, or whether the size of the display screen is 960 or 1920 characters. The INFDS data structure must be specified if function control keys are enabled for the program or if the POST operation code is to be executed; otherwise the indication that a function control key was pressed and the results of the POST operation are not available to the program. The information in the INFDS data structure is updated for each WORKSTN operation; however, *SIZE is only updated when the POST operation is executed. If an exception/error condition occurs, the programmer can use the INFDS information to determine the type of exception/error that occurred and use that information to control the program logic or to control the return point from the INFSR subroutine, if specified.

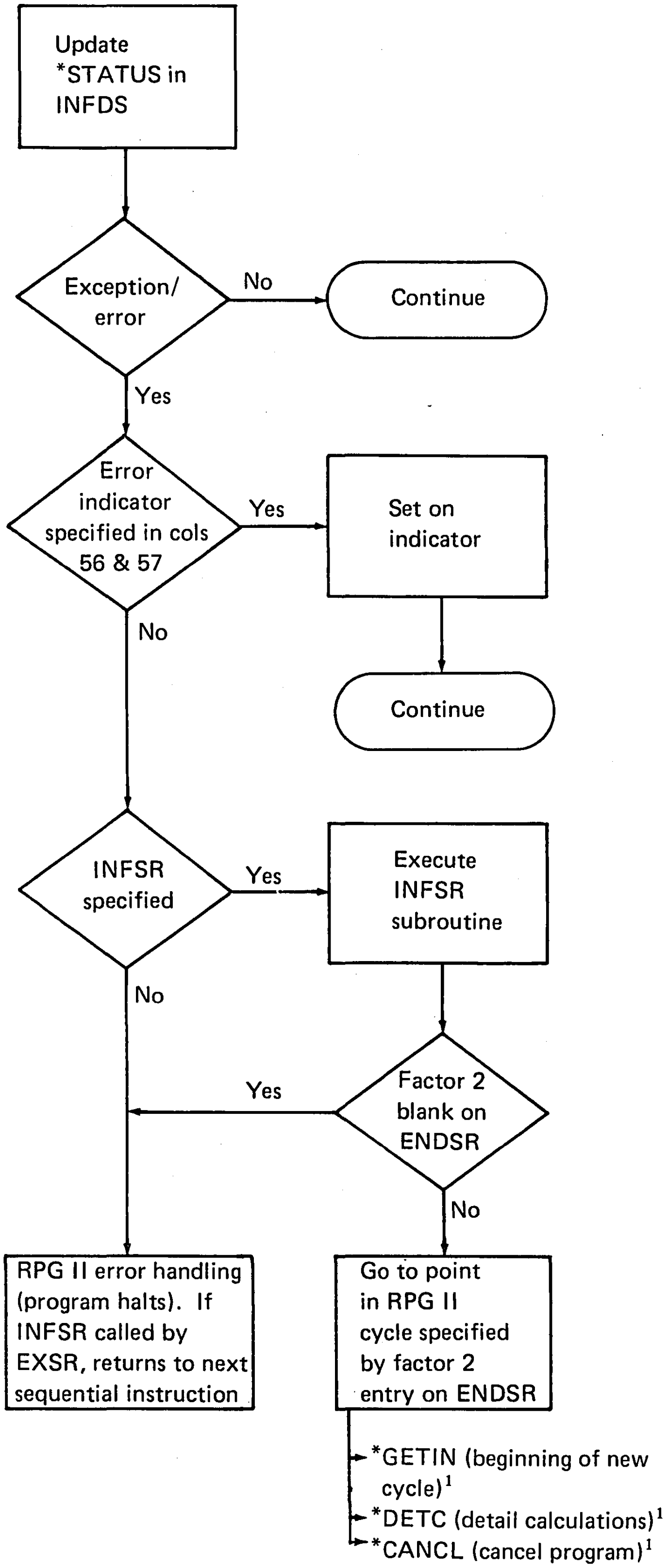
When an exception/error condition occurs for a WORKSTN operation, information in the INFDS is updated and control automatically passes to the INFSR subroutine, if specified, under the following conditions:

- If an exception/error occurs on a primary file read, on EXCPT output, or on a normal cycle output operation
- If an exception/error occurs on an explicit input/output operation (ACQ, REL, NEXT, or READ) that does not have an indicator specified in columns 56 and 57

In addition, the INFSR subroutine can be called directly from detail or total calculations by the EXSR operation.

The indicator specified in columns 56 and 57 for a READ, ACQ, REL, POST, or NEXT operation is set on if an exception/error condition occurs on that operation. Control is then passed to the next executable statement in the program. If an error indicator is specified for one of these operations, the INFSR subroutine must be called by the EXSR operation if the subroutine is to be executed. Control does not automatically pass to the INFSR subroutine if the exception/error occurs on a READ, ACQ, REL, POST, or NEXT operation that has an indicator specified in columns 56 and 57.

The relationship between these exception/error handling techniques is shown in Figure 13-10. These exception/error handling techniques are optional and can be used individually or in any combination. However, if function control keys are enabled for the program, the INFDS data structure and either the INFSR subroutine or an error indicator on the READ operation must be specified. The programmer can choose the techniques that best suit his application program.


Update *STATUS in INFDS
Exception/ error
No
Continue
Yes
Error indicator specified in cols 56 & 57
Yes
Set on indicator
No
Continue
INFSR specified
Yes
Execute INFSR subroutine
No
Yes
Factor 2 blank on ENDSR
No
RPG II error handling (program halts). If INFSR called by EXSR, returns to next sequential instruction
Go to point in RPG II cycle specified by factor 2 entry on ENDSR
*GETIN (beginning of new cycle)[1]
*DETC (detail calculations)[1]
*CANCL (cancel program)[1]

[1] For the exact point in the cycle that is specified by these keywords, see Chapter 17.

Figure 13-10. WORKSTN Exception/Error Handling

Specifications for the INFDS Data Structure

The INFDS data structure can be used to pass the WORKSTN file exception/error information to the RPG program. The INFDS data structure contains the identification of the type of exception/error that occurred and an indication of the WORKSTN operation that was executing when the error occurred. The INFDS data structure also contains status information on normal conditions such as whether end of file has occurred, whether the display screen size is 960 or 1920 characters (*SIZE), and whether your system supports ideographic processing. The INFDS data structure must be specified if function control keys are to be enabled for the program. If the INFDS is not specified, this information is not available to the RPG II program.

The name of the data structure to be used as the WORKSTN file information data structure must be specified on a continuation line for the WORKSTN file on the file description specifications along with the INFDS keyword. The entries for the continuation lines are:

Column	Entry
6	F
7-52	Must be blank
53	K
54-59	INFDS
60-65	Name of the data structure to be used as the INFDS
66-74	Must be blank
75-80	Program identification (optional)

The rules for defining the INFDS data structure on the input specifications are the same as for any other data structure. However, the name of the data structure must be previously defined on the file description specifications continuation line with the INFDS keyword. The location of the subfields containing the status information in the data structure is defined by special keywords on the input specifications. The keywords must be placed left-justified in columns 44 through 50. These keywords identify the location of self-defining subfields within the data structure. The keywords are not labels, however, and cannot be used to reference the subfields. A name must be assigned in columns 53 through 58 in order to reference the subfields (see Figure 13-11). The valid keywords are *STATUS, *OPCODE, *SIZE, *RECORD, *MODE, *INP, and *OUT.

In addition to these keyword-defined subfields, there is an alphameric subfield in the INFDS that contains return codes (see Appendix C). This subfield is filled in for all WORKSTN files. The subfield is located in positions 23 through 26 and must be defined on the input specifications. The subfield is referenced by the name specified in columns 53 through 58. (For information on return codes resulting from the use of SSP-ICF, see the *Interactive Communications Feature Reference Manual.*)

**STATUS Keyword*

The *STATUS keyword identifies a self-defining five-digit numeric subfield with zero decimal positions within the INFDS data structure. This subfield contains a five-digit code that identifies the exception/error condition. The codes are as follows:

Exception/Error Conditions (Function Control Keys)	**Code**
Print	01121
Roll Up	01122
Roll Down	01123
Clear	01124
Help	01125
Record Backspace	01126
(Error Status Codes)	
Input rejected, buffer too small	01201
Permanent I/O error occurred	01251
Invalid device, or maximum number of display stations already attached	01261
Device busy	01271
Display station released by operator	01275
Input rejected, keyboard disabled, device not available or not found	01281
Attempt to acquire a device already owned	01285
Other input errors	01299
Change direction received with no data	01311
Request for change direction received	01321
Time interval expiration	01331

Note: If an exception/error condition occurs, RPG II bypasses the move field logic, no fields are changed, no record identifying indicators are turned on, and the command key indicators are not reset.

In addition, the programmer also has access to the following successful status codes that are placed in *STATUS after any input/output operation:

Condition	**Code**
No exception (with a display station, either the Enter/Rec Adv or Auto Record Advance key was pressed)	00000
Any of the 24 command keys	00002
End of file (input rejected, no display stations ready)	00011

Any code in *STATUS greater than 99 is considered to be an exception/error condition, and the error indicator, if specified, is set on. If no error indicator is specified on an ACQ, REL, NEXT, POST, or READ operation or if the operation is a primary file read, normal output, or EXCPT output, control is automatically passed to the INFSR subroutine.

For information on return codes resulting from the use of the Interactive Communications Feature, see the *Interactive Communications Feature Reference Manual.* Appendix C of this manual contains a list of return codes issued by WORKSTN and SSP-ICF data management.

**OPCODE Keyword*

The *OPCODE keyword identifies a self-defining, 5-character alphameric subfield within the INFDS data structure. This subfield contains a value that identifies which WORKSTN operation was executing when the exception/error occurred. The value inserted in the *OPCODE subfield is READ, ACQ, REL, NEXT, POST, or WRITE (for output operations). A value is inserted in the *OPCODE subfield when a nonzero value is placed in *STATUS.

**RECORD Keyword*

The *RECORD keyword identifies a self-defining, 8-character alphameric subfield within the INFDS data structure. This subfield contains the format name if *OPCODE contains WRITE. If *OPCODE does not contain WRITE, *RECORD is blank.

**SIZE Keyword*

The *SIZE keyword identifies a self-defining, four-digit numeric subfield within the INFDS data structure that contains the identification of the character size of the display screen (1920 or 960). This subfield is reset each time the POST operation code is executed. For the 1920-character display, the digits 1920 are stored in the *SIZE subfield; for the 960-character display, the digits 0960 are stored in the *SIZE subfield.

**MODE Keyword*

The *MODE keyword identifies a two-digit numeric field that indicates if ideographic support was requested when the user signed on.

Value	Explanation
10	Ideographic support was requested.
00	Ideographic support was not requested.

**INP Keyword*

The *INP keyword identifies a two-digit numeric field that indicates if this display station is capable of producing ideographic characters for input to a program.

Value	Explanation
10	The keyboard is capable of ideographic data entry.
00	The keyboard is not capable of ideographic data entry, or is not input-capable.

**OUT Keyword*

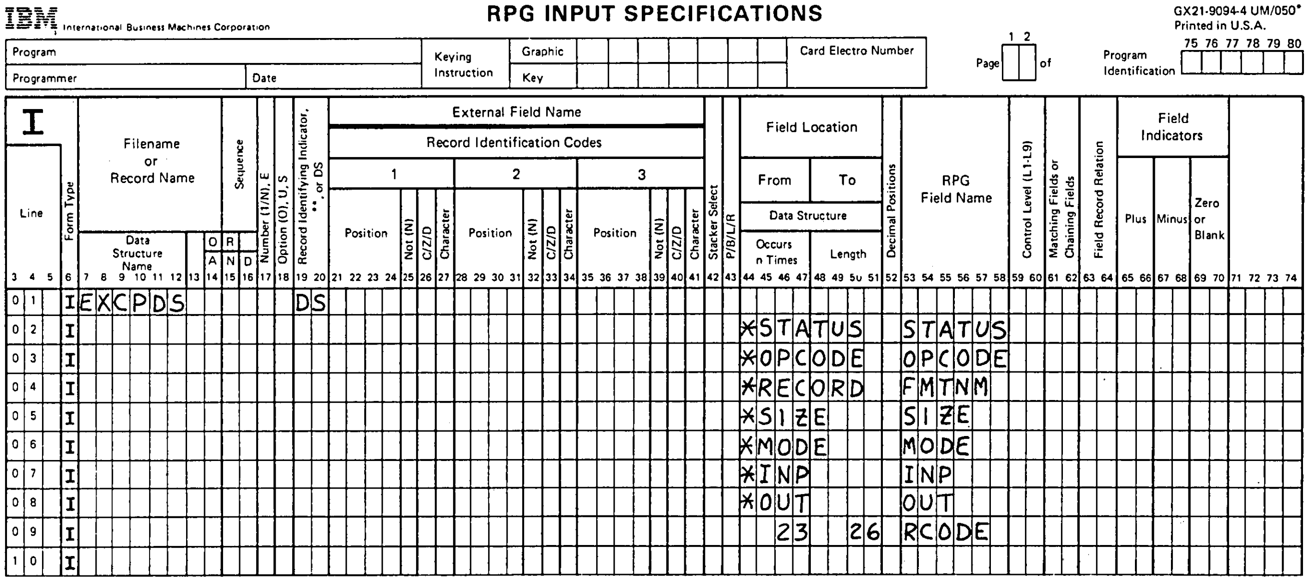
The *OUT keyword identifies a two-digit numeric field that indicates if this display station's display screen is capable of displaying ideographic characters.

Value	Explanation
10	The display screen can display. ideographic characters.
00	The display screen cannot display ideographic characters, or is not output-capable.

Positions 23-26

Positions 23 through 26 of the data structure defined as the INFDS contain the 4-character data management return code for WORKSTN files. This subfield is filled in for all WORKSTN files. The subfield must be defined on the input specifications. The subfield is referenced by the name specified in columns 53 through 58 of the input specifications. For information on RPG return codes for display stations, see Appendix C. For information on SSP-ICF return codes, see the *Interactive Communications Feature Reference Manual.*

Note: This subfield is not updated for an *STATUS value of 01261 because no call is made to data management. If the *STATUS value is 01281, this subfield will not be updated unless the error occurs on a read or ACQ operation.



IBM International Business Machines Corporation

RPG INPUT SPECIFICATIONS

GX21-9094-4 UM/050*
Printed in U.S.A.

Program | Keying Instruction: Graphic, Key | Card Electro Number | Page □□ of | Program Identification □□□□□□
Programmer | Date

Line	Form Type	Filename or Record Name / Data Structure Name (7-12)	Record Identifying Indicator, or DS (19-20)	Field Location From (44-47)	Field Location To (48-51)	RPG Field Name (53-58)
01	I	EXCPDS	DS			
02	I			*STATUS		STATUS
03	I			*OPCODE		OPCODE
04	I			*RECORD		FMTNM
05	I			*SIZE		SIZE
06	I			*MODE		MODE
07	I			*INP		INP
08	I			*OUT		OUT
09	I			23	26	RCODE
10	I					

The keywords and predefined from and to locations in columns 44 through 51 define the location and size of the subfields in the INFDS data structure, which contain the status information. Field names must be assigned in columns 53 through 58 so the subfields can be referenced in the program.

Figure 13-11. Subfield Keywords for the INFDS Data Structure

Specifications for the INFSR Subroutine

The INFSR subroutine can perform any function normally allowed in calculations, including exits to other calculation subroutines and input/output operations. The INFSR subroutine returns control to the point specified by the optional factor 2 entry for the ENDSR operation.

Specify the INFSR keyword and the name of the exception/error processing subroutine on a continuation line for the WORKSTN file on the file description specifications. Valid entries for the continuation line are:

Column	Entry
6	F
7-52	Must be blank
53	K
54-59	INFSR
60-65	Name of the calculation subroutine that is to be executed if a WORKSTN exception/error occurs on a READ, ACQ, REL, POST, or NEXT operation for which no indicator is specified in columns 56 and 57 or on implicit input/output operations.
66-74	Must be blank
75-80	Program identification (optional)

The rules for coding the INFSR subroutine are the same as for any calculation subroutine. However, the name specified in factor 1 must be previously defined on the file description specifications continuation line with the INFSR keyword. The INFSR subroutine returns control to the point specified by an optional entry in factor 2 of the ENDSR operation. Valid entries for factor 2 are:

Factor 2	Description
Blank (no entry)	• If the INFSR subroutine was called explicitly by the EXSR operation, control returns to the operation following the EXSR statement. • If the subroutine was called indirectly (that is, control was automatically passed to the subroutine), the subroutine is executed and control is passed to the RPG II error handling routine, which in most cases causes the program to halt, and the operator must choose the appropriate option.
Literal	The literal must be one of the following keywords. (The keyword must be enclosed in apostrophes.) '*GETIN'–Control returns to the beginning of a new cycle. '*DETC'–Control returns to the beginning of detail calculations. '*CANCL'–Files are closed and program is canceled. '–A literal value of blanks is the same as no entry.

Factor 2	Description
Literal (continued)	If an exception/error occurs on an operation that attempts to read data from a file and the exception/error handling subroutine receives control, the programmer must ensure that an output operation is issued to the WORKSTN file before another read is issued. Two consecutive read operations cannot be issued to the same WORKSTN file. For example, if the WORKSTN file is a primary file and the exception/error subroutine ENDSR statement specified a return point of *GETIN, an output operation must be issued to the file before the ENDSR is executed. The *GETIN routine will attempt to read from a WORKSTN primary file.
Array element or field name	The array element or field name specifies a 6-character alphameric field that contains one of the reserved keywords, *GETIN, *DETC, or *CANCL or that contains blanks that define the return point from the subroutine. The reserved keywords must be left-justified and padded with blanks in the field specified. By specifying the return point in a field, the programmer can use the subroutine to process all types of exceptions and errors that occur on the WORKSTN file.

If a field name or array element is specified in factor 2, the field or array element is set to blanks upon each exit from the subroutine. Therefore, the programmer can control the return point of the INFSR within the program by placing the return point in the field that best fits the particular exception/error that occurred. If no value is placed in factor 2, the subroutine is executed and control passes to the RPG II error handling routine if the subroutine was called indirectly. If the INFSR subroutine was explicitly called by the EXSR operation and factor 2 is blank, control returns to the calculation immediately following the EXSR operation.

SPECIAL DISPLAY FORMAT CONSIDERATIONS

Overriding Fields in a Format

An override operation allows you to override fields in a format when you redisplay the same format. To perform an override operation, you must specify an indicator in columns 33 and 34 of the S specification. An override operation is performed if the indicator is on when the format is displayed (see Figure 13-12). A normal output operation is performed if the indicator is off when the format is displayed.

During an override operation (the indicator in columns 33 and 34 is on), the following occurs:

- Any field that has an indicator specified in columns 23 and 24 of the D specification and that indicator is off is unchanged. If data was keyed into the field, that data is unchanged. Any field that had Y, N, or blank specified in columns 23 and 24 is also unchanged.

- Any field that has an indicator specified in columns 23 and 24 of the D specification and that indicator is on is displayed with data from the RPG II program. Any data that was keyed into the field by the operator is lost. Output information is displayed from the *same locations in the output record area* as for a normal display.

- For all fields, the use of indicator-controlled attributes such as highlight or reverse image is determined by the state of that indicator. All field attributes that are not controlled by indicators are unchanged.

For example, you may want to override fields in a display if the operator keys incorrect data into a field. To do this, specify an indicator in columns 33 and 34 of the S specification, which allows the format to be overridden. If the operator keys incorrect data into a field, you can then set on the indicator in columns 33 and 34 and redisplay the format. If the indicator specified in columns 23 and 24 of the D specification is off for the field, the incorrect data is unchanged. If the indicator is on, data from the RPG II program is displayed. You can also use indicators for field attributes such as highlight and reverse image and set these indicators on when the override indicator is set on.

Read Under Format

A read under format allows one program in a procedure to display a format and the next program in the procedure to read it. The first program displays the format using a normal output operation and then goes to end of job or releases the display station. While the second program is initiating, the operator keys information into the displayed format. When the operator presses the Enter/Rec Adv key, the input information from the display is sent to the second program.

The following steps occur in a read under format:

1. With a normal output operation, the first program displays a format at the display station.

2. The first program goes to end of job (if the program is an SRT program) or releases the requesting display station (if the program is an MRT). The display station is released when column 16 of the output specifications contains an R, when the REL operation code is used in the calculations, or when the program goes to end of job.

3. The second program is initiated. (Data should not be passed to the second program from an INCLUDE OCL statement.)

4. The second program performs a normal read operation (either by a READ operation code or by a primary file input operation).

Processing the Duplicate Character Value (Hex 1C)

If you specify enable dup (column 34 of the D specification) for a field in a display screen format, the operator can press the Dup key to indicate to the program that the contents of the field are to be duplicated from the field in the previous record. When the Dup key is pressed, the field, from the position of the cursor to the end of the field, is filled with the duplicate character value (hex 1C), which is displayed as the character $\overline{*}$. The Dup key does not duplicate any characters; therefore, you must process the duplicate character values in your program.

If you want the operator to either duplicate the entire field or key the entire field, you need to test only one character in the field to determine whether the Dup key was pressed. For example, you can test the last character in an alphameric field for the duplicate character value by using the TESTB operation code. If the last character in the field is not a duplicate character value, move the contents of the test field to the processing field (see Figure 13-13).

You can also write your program to allow the operator to change the first part of a field and duplicate the latter part of the field. For example, if the operator changes the first 4 characters in a 10-character field and then presses the Dup key, positions 5 through 10 of the field will contain the duplicate character value (hex 1C). In your program, you then have to test each character in the field to determine where the first duplicate character occurs, and replace the appropriate positions with the data to be duplicated.

		Indicator in Columns 33 and 34 of the S specification	
		OFF	ON
Indicator in Columns 23 and 24 of the D specification	OFF	Output data comes from D specification (columns 57 through 79).	No change occurs to data on the screen.
	ON	Output data comes from the RPG II program.	Output data comes from the RPG II program.
		Normal Output Operation	Override Operation

Figure 13-12. Effect of Indicators on Output Data During an Override Operation

Enabling/Disabling Command Keys

The display screen format S specification allows you to specify in column 28 (enable command keys) that certain command keys be enabled or disabled for a format. (Normally all command keys are enabled for an RPG II WORKSTN program.) If the operator presses a disabled command key, an error message is displayed. The operator can press the Error Reset key and then press the correct command key.

For a description of the entries required to enable or disable command keys, see the summary of the display screen format S specification in Appendix A or see the *System Support Reference Manual.*

The INFDS data structure can be used to determine whether an enabled command key was pressed. For more information, see *WORKSTN Exception/Error Handling* in this chapter.

Enabling/Disabling Function Control Keys

The display screen format S specification allows you to specify in column 27 (enable function keys) that six function control keys (Print, Roll Up, Roll Down, Clear, Help, and Record Backspace) be enabled or disabled for your program. For these function control keys to be enabled for an RPG II program, the program must contain the INFDS data structure *and* either the INFSR subroutine or a READ operation with an indicator specified in columns 56 and 57. If the INFDS data structure is not specified in the program, the indication that a function control key was pressed is not available to the RPG program.

Function control keys not supported by the program can be masked off (disabled) and, therefore, are not returned to the program. If the operator presses a disabled function control key, an error message is displayed. The operator can then press the Error Reset key, followed by the correct function control key.

For a description of the entries required to enable or disable function control keys, see the summary of the display screen format S specification in Appendix A or see the *System Support Reference Manual.*

IBM-WRITTEN SUBROUTINES SUBR20 AND SUBR21

Setting and Restoring External Indicators (SUBR20)

The IBM-written *subroutine SUBR20 allows you to set and restore the external indicators for each requesting display station when multiple display stations are requestors in a WORKSTN file. To call the subroutine, you must make the following four entries on the calculation specifications:

			Result Field			
Factor 1	Operation	Factor 2	Name	Length	Decimal Positions	Half Adjust (H)
18 19 20 21 22 23 24 25 26 27	28 29 30 31 32	33 34 35 36 37 38 39 40 41 42	43 44 45 46 47 48	49 50 51	52	53
	EXIT	SUBR2Ø				
	RLABL		OP	1		
	RLABL		TNAME	2		
	RLABL		RCODE	1		

OP is a 1-character field that contains an I to indicate that the external indicators are to be input to the program for this display station, or an O to indicate that the external indicators are to be output for this display station. To enter the appropriate code in the OP field, you can use a MOVE operation before calling the subroutine.

TNAME is a 2-character field that contains the work station ID of the display station. Normally the field you specify for TNAME is the same field you specified on the file description specifications continuation line as the ID field.

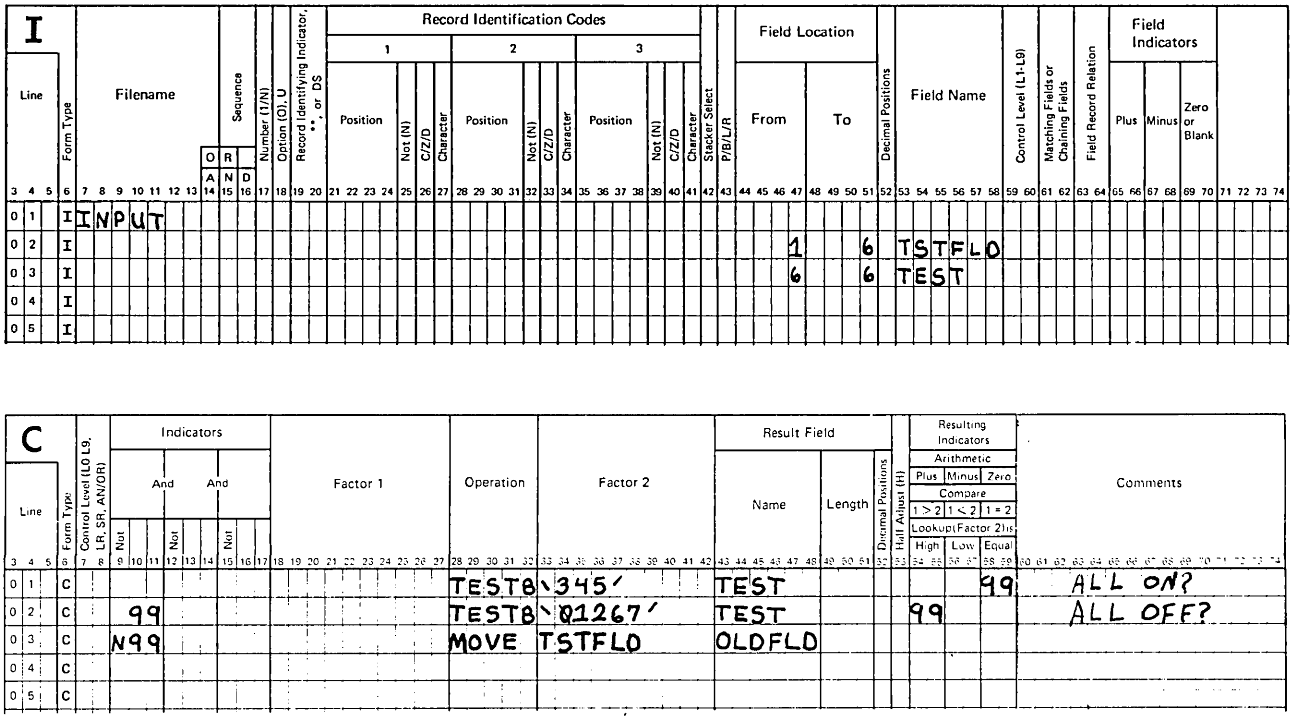
RCODE is a 1-character field that contains the following return code:

0 = successful

1 = unsuccessful (the display station is not attached to the program)

2 = unsuccessful (the display station is not a requestor)

The external indicators for the requestor of an SRT program are automatically available to the program without the use of SUBR20 and are written out at end of job. The external indicators for the first requestor of an MRT program are available without the use of SUBR20, but they are not automatically written out at end of job.



Line	Form Type	Filename	Field Location From	Field Location To	Field Name
01	I	INPUT			
02	I		1	6	TSTFLD
03	I		6	6	TEST
04	I				
05	I				

Line	Form Type	Indicators	Factor 1	Operation	Factor 2	Result Field Name	Length	Plus / 1>2 / High	Minus / 1<2 / Low	Zero / 1=2 / Equal	Comments
01	C			TESTB	'345'	TEST				99	ALL ON?
02	C	99		TESTB	'01267'	TEST		99			ALL OFF?
03	C	N99		MOVE	TSTFLD	OLDFLD					
04	C										
05	C										

Figure 13-13. Testing for a Duplicate Character Value (Hex 1C)

Reading and Writing the Display Station Local Data Area (SUBR21)

The IBM-written subroutine SUBR21 allows you to read and write the display station local data area for each display station when multiple display stations are requestors in a WORKSTN file. (For a complete description of the display station local data area, see the *System Planning Guide.*) To call the subroutine, you must make the following five entries on the calculation specifications:

Factor 1	Operation	Factor 2	Result Field			
			Name	Length	Decimal Positions	Half Adjust (H)
18 19 20 21 22 23 24 25 26 27	28 29 30 31 32	33 34 35 36 37 38 39 40 41 42	43 44 45 46 47 48	49 50 51	52	53
	EXIT	SUBR21				
	RLABL		OP	1		
	RLABL		TNAME	2		
	RLABL		RCODE	1		
	RLABL		AREA	256		

OP is a 1-character field that contains an I to indicate that the display station local data area is to be input to the program for this display station, or an O to indicate that the display station local data area is to be output for this display station.

TNAME is a 2-character field that contains the work station ID for the display station.

RCODE is a 1-character field that contains the following return code:

0 = successful

1 = unsuccessful (the display station is not attached to the program)

2 = unsuccessful (the display station is not a requestor)

AREA is a field or data structure into which or from which the display station local data area is read or written. AREA can be up to 256 characters long. Position 1 of the display station local data area is always placed in position 1 of this field. If AREA is to be used to pass parameters to OCL, the special characters ? and / (slash) should not be used.

The display station local data area for the requestor of an SRT program or the *first* requestor of an MRT program can be referenced in RPG II if you define a data structure with a U in column 18 of the input specifications (see Chapter 7).

WORKSTN SAMPLE PROGRAMS

Sample Program ITMINQ

The sample program ITMINQ displays records from the chained disk file INV. The display prompts the operator to enter an item number that is used to chain to the record in the disk file. If the record is found, the item number, description, price, onhand quantity, and quantity sold are displayed on line 6. If the item is not found, the message ITEM NOT FOUND is displayed on line 24. After the item record or the error message has been displayed, the display prompts the operator for the next item number. The operator enters a / (slash) to end the job. See Figure 13-14 for the specifications for ITMINQ, and see Figure 13-15 for the compiler listing for the program.

Display Screen Format

Figure 13-14 (Part 1) shows the display screen layout sheet that is used to format the display. The individual display screen format specifications ZITEM, SHOWITEM, and ZERROR are contained in the format load member ITMINQFM. (To form the format load member name, add FM to the end of the RPG II program name.) The display screen format names are specified as constants on the output specifications (see Part 4 of Figure 13-14). The end positions for these constants are specified as Kn, where n is the length of the format name.

The fields on the D specifications for the display screen formats must be specified in the same order as they appear in the RPG II output record. However, the line number and horizontal position columns can be used to change the order in which the fields appear on the screen.

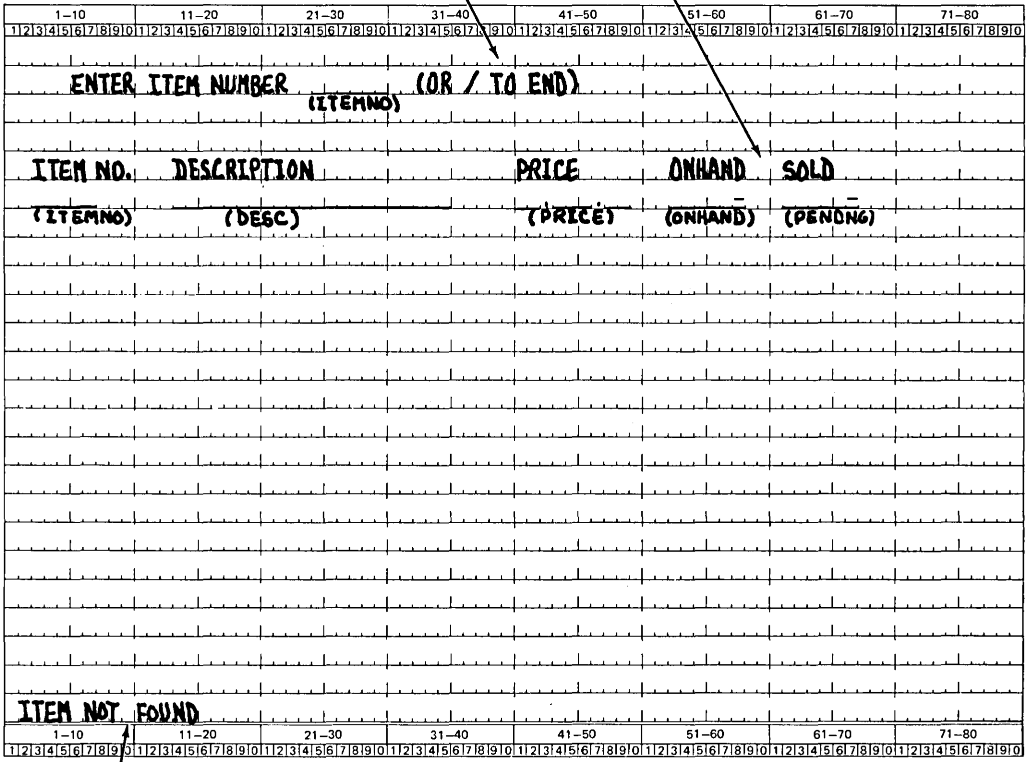
Indicator 02 is set on after the execution of the first READ from the display screen. This initial READ is satisfied by reading a blank record from the display screen. The ZITEM format is written to the screen on the next cycle because indicator 02 is on. The operator can then enter an item number in response to the ZITEM format.

Indicator 03 is set on when the item number is read from the ZITEM format (position 1 does not contain a /). Indicator 03 causes both the ZITEM and SHOWITEM formats to be written to the display screen. The ZITEM format is not input-capable, as indicator 03 is on (indicator 03 is specified in columns 35 and 36, suppress input, of the ZITEM format S specification). The ZITEM format is made input-capable when the SHOWITEM format is displayed (the SHOWITEM format has an N specified in columns 35 and 36 of its S specification, which enables the ZITEM format for input). The ZERROR format enables the ZITEM format for input in the same way.

Note: Only the last format displayed on the display screen is input-capable. However, through the use of the suppress input facility, more than one format on the same display screen can be made input-capable. For more information on suppress input, see *$SFGR—Screen Format Generator Utility Program* in the *System Support Reference Manual.*

This displayed line is described by display screen format ZITEM.

Lines 05 and 06 are described by display screen format SHOWITEM.



The error message is described by display screen format ZERROR, and the constant is described in the RPG II output specifications.

The individual formats ZITEM, SHOWITEM, and ZERROR are contained in the format load member ITMINQFM.

Indicator 03 suppresses input from this format until another format is displayed during the same RPG II cycle with suppress input-no specified.

GX21-9253 U/M 050*
Printed in U.S.A.

System/34 Display Screen Format Specifications

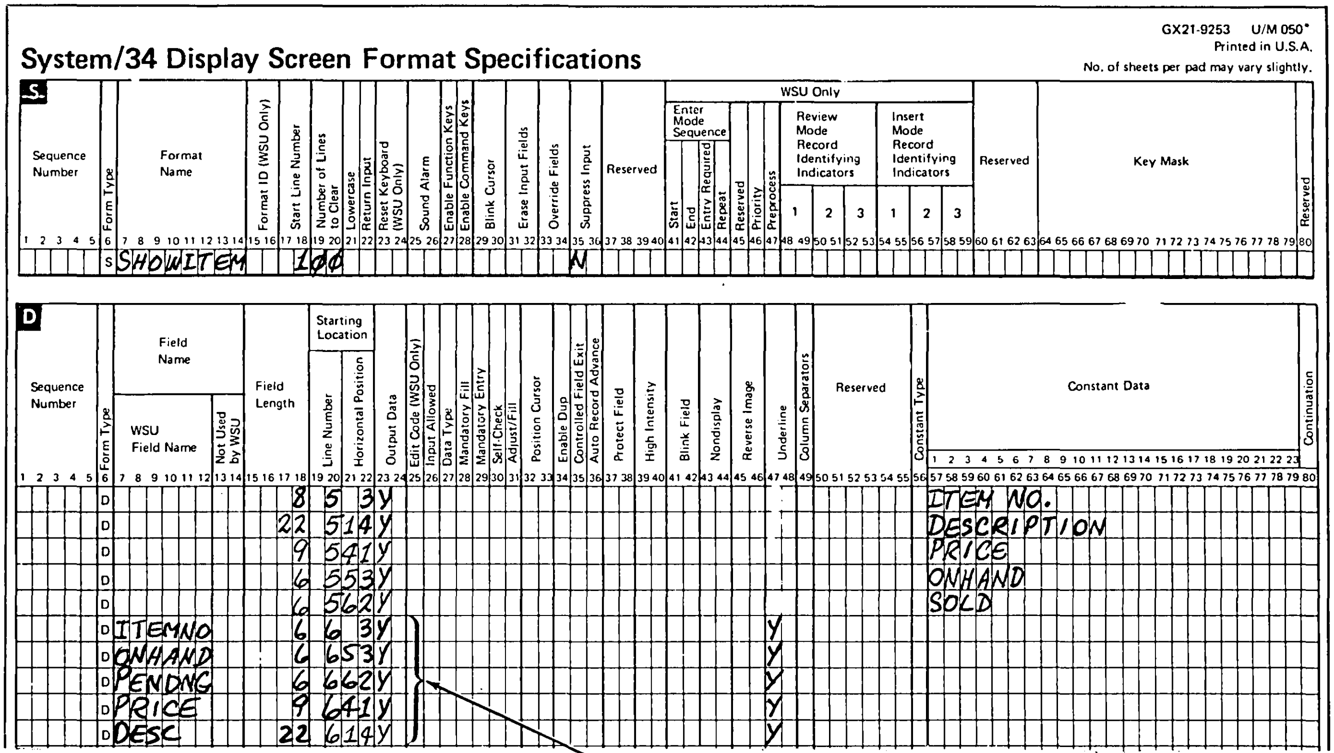
S

Sequence Number	Form Type	Format Name	Format ID (WSU Only)	Start Line Number	Number of Lines to Clear	Lowercase	Return Input	Reset Keyboard (WSU Only)	Sound Alarm	Enable Function Keys	Enable Command Keys	Blink Cursor	Erase Input Fields	Override Fields	Suppress Input	Reserved	Enter Mode Sequence Start	Enter Mode Sequence End	Entry Required	Repeat	Reserved	Priority	Preprocess	Review Mode Record Identifying Indicators 1	2	3	Insert Mode Record Identifying Indicators 1	2	3	Reserved	Key Mask	Reserved
	S	ZITEM		Ø1	24										Ø3																	

D

Sequence Number	Form Type	WSU Field Name	Not Used by WSU	Field Length	Line Number	Horizontal Position	Output Data	Edit Code (WSU Only)	Input Allowed	Data Type	Mandatory Fill	Mandatory Entry	Self-Check	Adjust/Fill	Position Cursor	Enable Dup	Controlled Field Exit	Auto Record Advance	Protect Field	High Intensity	Blink Field	Nondisplay	Reverse Image	Underline	Column Separators	Reserved	Constant Type	Constant Data	Continuation
	D			17	2	6	Y																				C	ENTER ITEM NUMBER	
	D	ITEMNO		6	2	25			Y						Y			Y						Y					
	D			13	2	33	Y																				C	(OR / TO END)	

Figure 13-14 (Part 1 of 4).*Sample Program ITMINQ



GX21-9253 U/M 050*
Printed in U.S.A.
No. of sheets per pad may vary slightly.

System/34 Display Screen Format Specifications

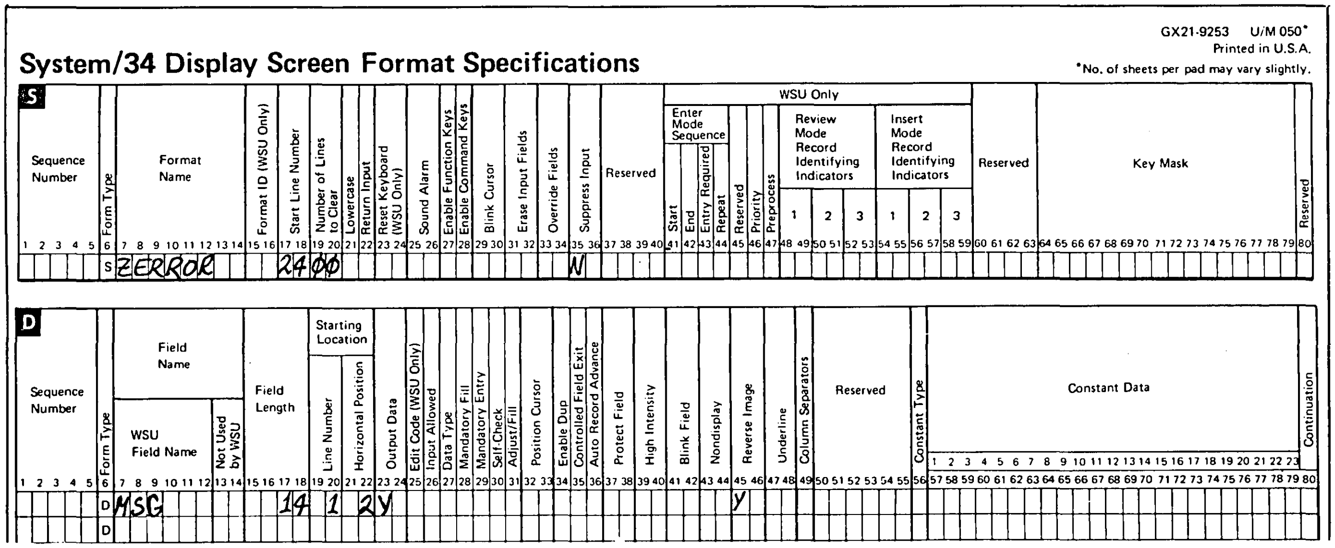
S

Sequence Number (1-5)	Form Type (6)	Format Name (7-14)	Format ID (WSU Only) (15-16)	Start Line Number (17-18)	Number of Lines to Clear (19-20)	Lowercase (21)	Return Input (22)	Reset Keyboard (WSU Only) (23-24)	Sound Alarm (25-26)	Enable Function Keys (27)	Enable Command Keys (28)	Blink Cursor (29-30)	Erase Input Fields (31-32)	Override Fields (33-34)	Suppress Input (35-36)	Reserved (37-40)	WSU Only: Enter Mode Sequence Start (41)	End (42)	Entry Required (43)	Repeat (44)	Reserved (45)	Priority (46)	Preprocess (47)	Review Mode Record Identifying Indicators 1 (48-49)	2 (50-51)	3 (52-53)	Insert Mode Record Identifying Indicators 1 (54-55)	2 (56-57)	3 (58-59)	Reserved (60-63)	Key Mask (64-79)	Reserved (80)
	S	SHOWITEM		1	ØØ										N																	

D

Sequence Number (1-5)	Form Type (6)	Field Name: WSU Field Name (7-12)	Not Used by WSU (13-14)	Field Length (15-18)	Starting Location: Line Number (19-20)	Horizontal Position (21-22)	Output Data (23-24)	Edit Code (WSU Only) (25)	Input Allowed (26)	Data Type (27)	Mandatory Fill (28)	Mandatory Entry (29)	Self-Check (30)	Adjust/Fill (31)	Position Cursor (32-33)	Enable Dup (34)	Controlled Field Exit (35)	Auto Record Advance (36)	Protect Field (37-38)	High Intensity (39-40)	Blink Field (41-42)	Nondisplay (43-44)	Reverse Image (45-46)	Underline (47-48)	Column Separators (49)	Reserved (50-55)	Constant Type (56)	Constant Data (57-79)	Continuation (80)
	D			8	5	3	Y																					ITEM NO.	
	D			22	5	14	Y																					DESCRIPTION	
	D			9	5	41	Y																					PRICE	
	D			6	5	53	Y																					ONHAND	
	D			6	5	62	Y																					SOLD	
	D	ITEMNO		6	6	3	Y																	Y					
	D	ONHAND		6	6	53	Y																	Y					
	D	PENDNG		6	6	62	Y																	Y					
	D	PRICE		9	6	41	Y																	Y					
	D	DESC		22	6	14	Y																	Y					

These fields are described in the same order as they appear in the RPG II input and output records. However, the line number and horizontal position columns have been used to change the order in which the fields appear on the display screen.



GX21-9253 U/M 050*
Printed in U.S.A.
*No. of sheets per pad may vary slightly.

System/34 Display Screen Format Specifications

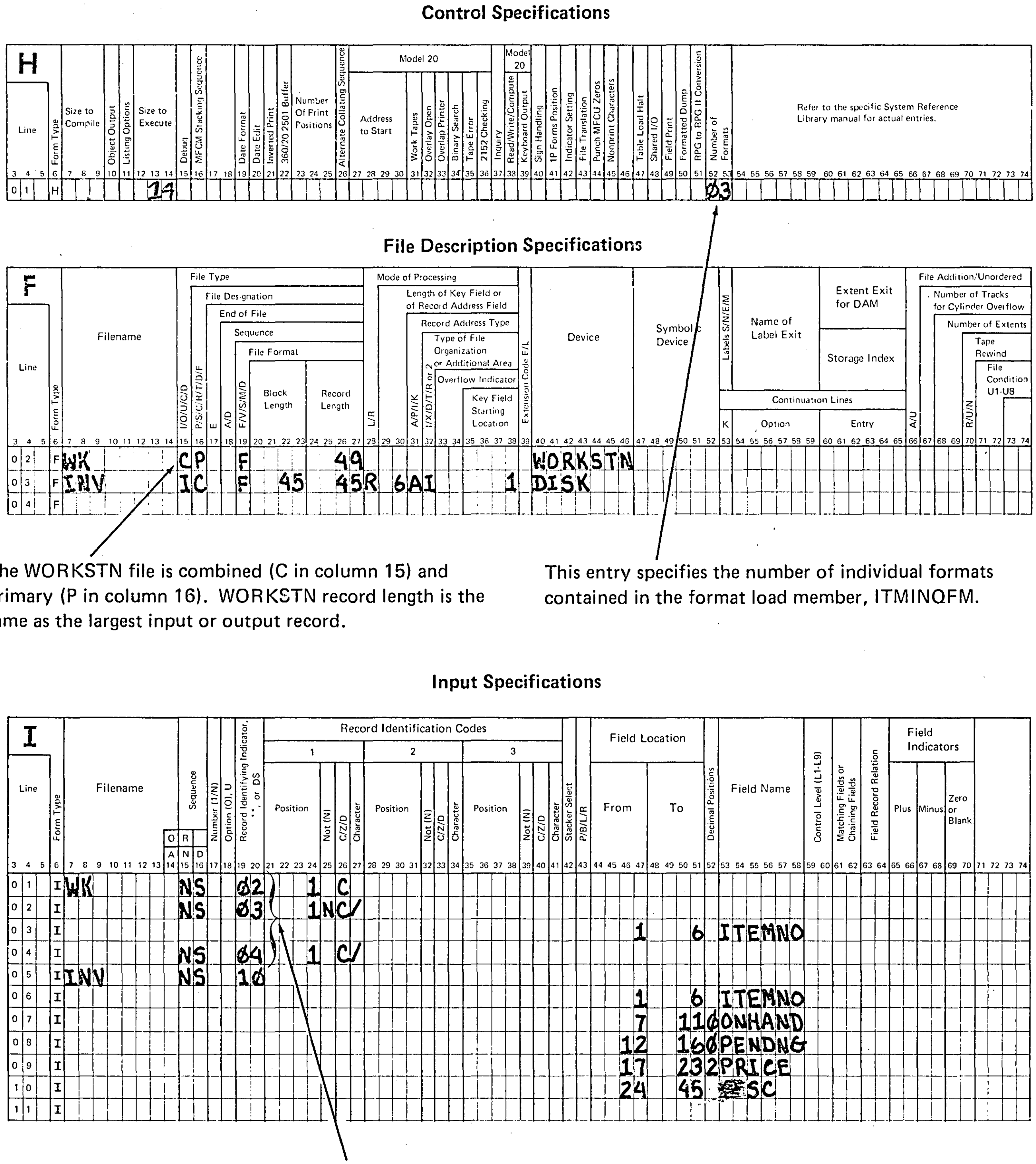
S

Sequence Number (1-5)	Form Type (6)	Format Name (7-14)	Format ID (WSU Only) (15-16)	Start Line Number (17-18)	Number of Lines to Clear (19-20)	Lowercase (21)	Return Input (22)	Reset Keyboard (WSU Only) (23-24)	Sound Alarm (25-26)	Enable Function Keys (27)	Enable Command Keys (28)	Blink Cursor (29-30)	Erase Input Fields (31-32)	Override Fields (33-34)	Suppress Input (35-36)	Reserved (37-40)	WSU Only: Enter Mode Sequence Start (41)	End (42)	Entry Required (43)	Repeat (44)	Reserved (45)	Priority (46)	Preprocess (47)	Review Mode Record Identifying Indicators 1 (48-49)	2 (50-51)	3 (52-53)	Insert Mode Record Identifying Indicators 1 (54-55)	2 (56-57)	3 (58-59)	Reserved (60-63)	Key Mask (64-79)	Reserved (80)
	S	ZERROR		24	ØØ										N																	

D

Sequence Number (1-5)	Form Type (6)	Field Name: WSU Field Name (7-12)	Not Used by WSU (13-14)	Field Length (15-18)	Starting Location: Line Number (19-20)	Horizontal Position (21-22)	Output Data (23-24)	Edit Code (WSU Only) (25)	Input Allowed (26)	Data Type (27)	Mandatory Fill (28)	Mandatory Entry (29)	Self-Check (30)	Adjust/Fill (31)	Position Cursor (32-33)	Enable Dup (34)	Controlled Field Exit (35)	Auto Record Advance (36)	Protect Field (37-38)	High Intensity (39-40)	Blink Field (41-42)	Nondisplay (43-44)	Reverse Image (45-46)	Underline (47-48)	Column Separators (49)	Reserved (50-55)	Constant Type (56)	Constant Data (57-79)	Continuation (80)
	D	MSG		14	1	2	Y																Y						
	D																												

Figure 13-14 (Part 2 of 4). Sample Program ITMINQ



Control Specifications

```
  01 H      14                                     Ø3
```

File Description Specifications

```
  02 FWK      CP  F      49            WORKSTN
  03 FINV     IC  F  45  45R 6AI     1 DISK
  04 F
```

The WORKSTN file is combined (C in column 15) and primary (P in column 16). WORKSTN record length is the same as the largest input or output record.

This entry specifies the number of individual formats contained in the format load member, ITMINQFM.

Input Specifications

```
  01 IWK      NS  Ø2   1 C
  02 I        NS  Ø3   1NC/
  03 I                                        1   6 ITEMNO
  04 I        NS  Ø4   1 C/
  05 IINV     NS  1Ø
  06 I                                        1   6 ITEMNO
  07 I                                        7  11ØONHAND
  08 I                                       12  16ØPENDNG
  09 I                                       17  232PRICE
  10 I                                       24  45 DESC
  11 I
```

Indicator 02 denotes a blank record. Most WORKSTN files will start with a blank record for each display station. Indicator 03 is used for the item number record, and indicator 04 is used for end of job.

Figure 13-14 (Part 3 of 4). Sample Program ITMINQ

Calculation Specifications

Line	Form Type	Control Level	Indicators (9-17)	Factor 1	Operation	Factor 2	Result Field Name	Length	Decimal Positions	Half Adjust	Resulting Indicators High (54-55)	Low (56-57)	Equal (58-59)	Comments
01	C				SETOF						99			
02	C		Ø3	ITEMNO	CHAIN	INV					99			

Indicator 99 is set on if the item is not found, and is used to condition the error message ITEM NOT FOUND (see the output specifications, line 11).

Output Specifications

Line	Form Type	Filename	Type (14-16)	Space	Skip	Output Indicators (23-31)	Field Name	Edit Codes (38)	End Position in Output Record	Constant or Edit Word
01	O	WK	D			Ø2				
02	O		OR			Ø3N99				
03	O								K5	'ZITEM'
04	O		D			Ø3N99				
05	O								K8	'SHOWITEM'
06	O						ITEMNO		6	
07	O						ONHAND	L	12	
08	O						PENDNG	L	18	
09	O						PRICE	2	27	
10	O						DESC		49	
11	O		D			99				
12	O								K6	'ZERROR'
13	O								14	'ITEM NOT FOUND'
14	O		DR			Ø4				
15	O									

Commas	Zero Balances to Print	No Sign	CR	-	
Yes	Yes	1	A	J	X = Remove Plus Sign
Yes	No	2	B	K	Y = Date Field Edit
No	Yes	3	C	L	Z = Zero Suppress
No	No	4	D	M	

An R in column 16 releases the display station from the program. No format name is necessary on the release. If one is given, the format is written to the display station, then the display station is released from the program.

The PRICE field is edited by RPG II (edit code 2 in column 38). ONHAND and PENDNG are edited by the RPG II L edit code. Because these fields contain punctuation, they cannot be used as numeric input fields.

Figure 13-14 (Part 4 of 4). Sample Program ITMINQ

```
SOURCE INPUT SCREEN FORMAT SOURCE SPECIFICATIONS

   SZITEM    0124             03
   D         17 2 6Y                                CENTER ITEM NUMBER
   DITEMNO    6 225   Y     Y   Y          Y
   D         13 233Y                                C OR / TO END

INPUT BUFFER DESCRIPTION

FIELD                          START          END
NAME             LENGTH        POSITION       POSITION

ITEMNO               6             1              6



SOURCE INPUT SCREEN FORMAT SOURCE SPECIFICATIONS

   SSHOWITEM  100             N                    Y         ITEM NO.
   D           8 5 3Y                              Y         DESCRIPTION
   D          22 514Y                              Y         PRICE
   D           9 541Y                              Y         ONHAND
   D           6 553Y                              Y         SOLD
   D           6 562Y
   DITEMNO     6 6 3Y
   DONHAND     6 653Y
   DPENDNG     6 662Y
   DPRICE      9 641Y
   DDESC      22 614Y

EXECUTION TIME OUTPUT BUFFER DESCRIPTION

FIELD                          START          END
NAME             LENGTH        POSITION       POSITION

ITEMNO               6             1              6
ONHAND               6             7             12
PENDNG               6            13             18
PRICE                9            19             27
DESC                22            28             49



SOURCE INPUT SCREEN FORMAT SOURCE SPECIFICATIONS

   SZERROR   2400             N          Y
   DMSG      14 1 2Y

EXECUTION TIME OUTPUT BUFFER DESCRIPTION

FIELD                          START          END
NAME             LENGTH        POSITION       POSITION

MSG                 14             1             14
```

```
 ITMINQFM   SCREEN FORMAT LOAD MEMBER

FORMAT ZITEM    REQUIRES   256 BYTES OF STORAGE
FORMAT SHOWITEM REQUIRES   256 BYTES OF STORAGE
FORMAT ZERROR   REQUIRES   256 BYTES OF STORAGE
```

Figure 13-15 (Part 1 of 2). Compiler Listing for Sample Program ITMINQ

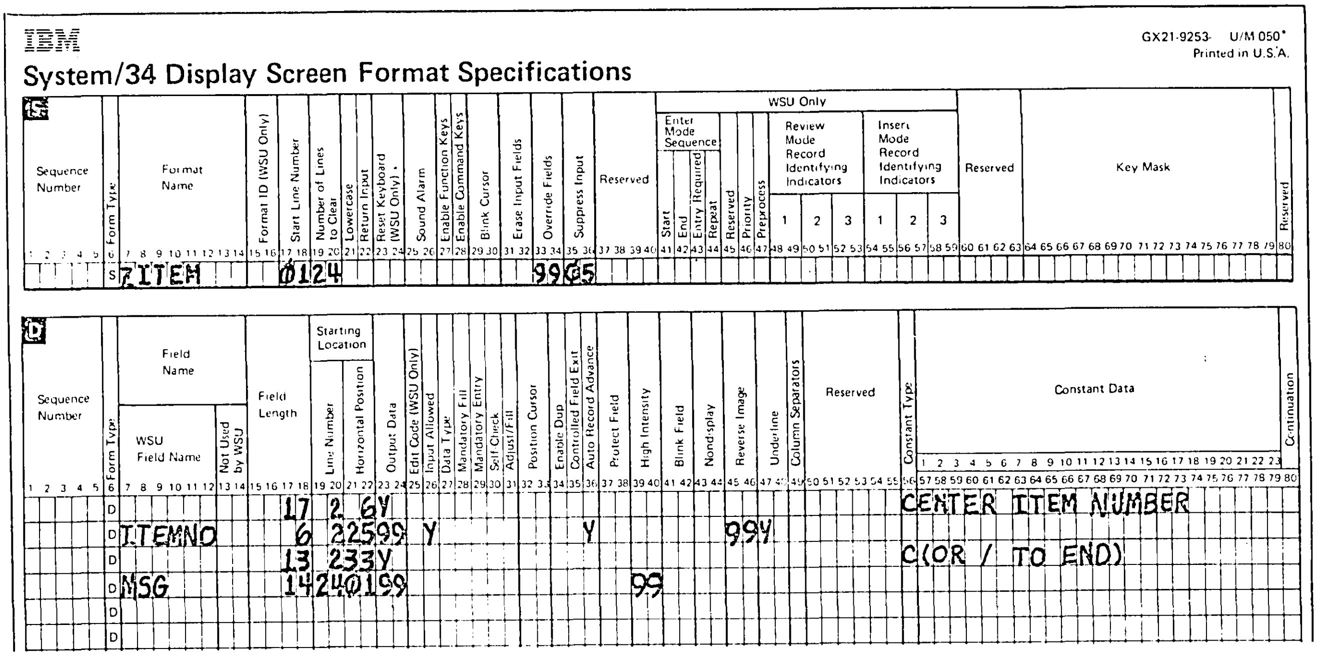
```
           IBM SYSTEM/34 RPGII COMPILER


            H      14                                     03                       ITMINQ


 0001       FWK      CP  F      49           WORKSTN
 0002       FINV     IC  F  45  45R 6AI     1 DISK
 0003       IWK      NS  02    1 C
 0004       I        NS  03    1NC/
 0005       I                                       1   6 ITEMNO
 0006       I        NS  04    1 C/
 0007       IINV     NS  10
 0008       I                                       1   6 ITEMNO
 0009       I                                       7  1100NHAND
 0010       I                                      12  160PENDNG
 0011       I                                      17  232PRICE
 0012       I                                      24  45 DESC
 0013       C                     SETOF                     99
 0014       C   03      ITEMNO    CHAININV                  99
 0015       OWK      D         02
 0016       O       OR         03N99
 0017       O                                   K5 'ZITEM'
 0018       O        D         03N99
 0019       O                                   K8 'SHOWITEM'
 0020       O                         ITEMNO     6
 0021       O                         ONHANDL   12
 0022       O                         PENDNGL   18
 0023       O                         PRICE 2   27
 0024       O                         DESC      49
 0025       O        D         99
 0026       O                                   K6 'ZERROR'
 0027       O                                   14 'ITEM NOT FOUND'
 0028       O        DR        04

INDICATORS USED
        02 03 04 10 99

RPG-0305 INDICATORS UNREFERENCED
        10

  FIELD NAMES USED
 STMT#  NAME   DEC  LNG   DISP
  0005 ITEMNO       0006   0105
  0009 ONHAND  0    0005   0120
  0010 PENDNG  0    0005   0125
  0011 PRICE   2    0007   012C
  0012 DESC         0022   011B

 ERROR NUMBER  STATEMENT NUMBER
   RPG-0097           0004
   RPG-0097           0007

  ERROR SEVERITY                                       TEXT
RPG-0097   W    NO FIELD DESCRIBED FOR THIS OR PREVIOUS RECORD OR DATA STRUCTURE.  IF DATA STRUCTURE, LENGTH DEFAULTS TO ONE.
RPG-0305   W    INDICATOR ASSIGNED BUT NOT USED TO CONDITION OPERATIONS.


               MAIN STORAGE USAGE OF RPG II CODE

    START  NAME IF  CODE    NAME    TITLE
    ADDR   OVERLAY  LENGTH
    0000            0636    RGROOT ROOT
    0636            00A8    @PGTS  WORK STATION SCAN SUBROUTINE
    0752            008D    RGMAIN INPUT MAINLINE
    06DE            006C    RGSUBS INPUT CONTROL ROUTINE
    07DF            0062    RGSUBS RECORD IDENTIFICATION
    0841            0026    RGSUBS CONTROL FIELDS
    074A            0008    RGSUBS INPUT HOOK
    0867            06EE    @PGTI  WORK STATION INPUT PROCESSING
    0F55            023D    @PGTD  WORK STATION RETURN CODE & *STATUS UPDATE
    119A            0037    RGMAIN DETAIL CALCULATIONS
    1192            0008    RGSUBS INPUT HOOK
    1230            0016    @PGDL  DATA MANAGEMENT CALL
    11D1            005F    RGSUBS CHAIN CODE
    12C4            00B1    RGMAIN DETAIL OUTPUT
    1246            0051    RGSUBS OUTPUT CONTROL ROUTINE
    12A3            0021    RGSUBS CONSTANTS
    1297            000C    RGSUBS OUTPUT HOOK
    1375            02D9    @PGTO  WORK STATION OUTPUT PROCESSING
    164E            001D    RGMAIN LR & OVERFLOW PROCESSING
    166B            0022    RGMAIN INPUT FIELDS
    168D            00A9    RGMAIN OPEN MAINLINE
    1736            0030    RGMAIN CLOSE MAINLINE

                   05992    ITMINQ MAIN STORAGE REQUIRED TO EXECUTE.
                #LIBRARY    SOURCE MEMBER INPUT LIBRARY.
                #LIBRARY    LOAD MEMBER OUTPUT LIBRARY.
                    0027    LIBRARY SECTORS REQUIRED FOR OBJECT PROGRAM.
```

Figure 13-15 (Part 2 of 2). Compiler Listing for Sample Program ITMINQ

Sample Program INQPUT

The sample program INQPUT, shown in Figure 13-16, is similar to the ITMINQ program. However, the override field function of the display screen format specifications is used in INQPUT to display the error message in the ZITEM format instead of the separate ZERROR format.

The first display (ZITEM) prompts the operator to enter an item number. After the operator enters the item number, the program retrieves the inventory information and displays it with the SHOWITEM format. If the operator enters an item number not in the inventory file, the program turns on indicator 99 and redisplays the first format. Indicator 99 is specified in columns 33 and 34 (override fields) of the S specification for the ZITEM format and is used in columns 23 and 24 of the D specification to condition the message field (MSG) and the ITEMNO field. The MSG field is displayed only when indicator 99 is on, and the ITEMNO field is displayed as a reverse image field when indicator 99 is on.



IBM

GX21-9253- U/M 050*
Printed in U.S.A.

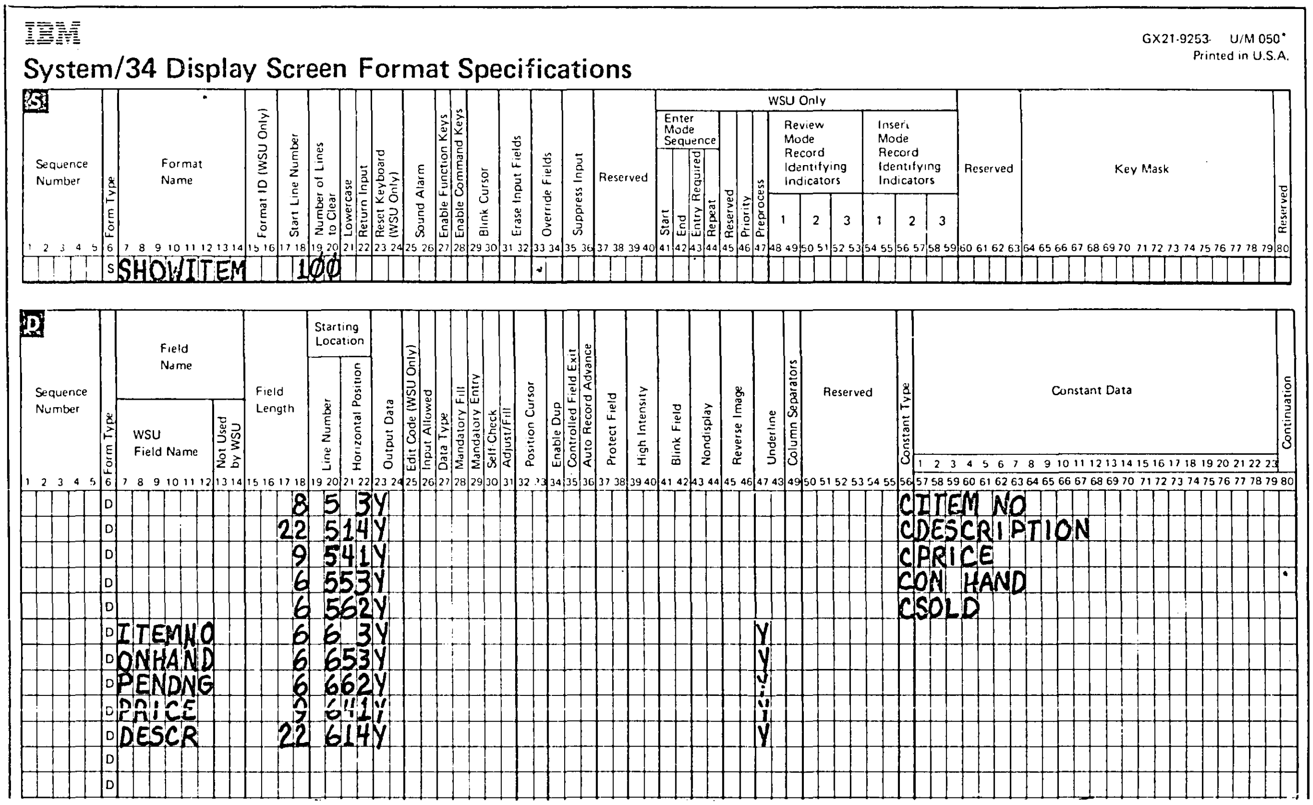
System/34 Display Screen Format Specifications

Form Type	Format Name	Start Line Number	Number of Lines to Clear	Override Fields	Suppress Input
S	ZITEM	Ø1	24	99	Ø5

Form Type	Field Name	Field Length	Line Number	Horizontal Position	Output Data	Input Allowed	Auto Record Advance	High Intensity	Reverse Image	Underline	Constant Type	Constant Data
D		17	2	6	Y						C	ENTER ITEM NUMBER
D	ITEMNO	6	22	5	99	Y	Y		99	Y		
D		13	23	3	Y						C	(OR / TO END)
D	MSG	14	24	Ø1	99			99				
D												
D												

Indicator 99 is specified in columns 33 and 34 (override fields) of the S specification for the ZITEM format. This allows the program to override (modify) fields in the display without retransmitting the entire display. When the program sets on indicator 99 (when an item number does not exist in the inventory file), the message field (MSG) for line 24 is displayed and the ITEMNO field, containing the invalid item number, is redisplayed as a reverse image field.

If indicator 05 is on, input is not allowed to the ZITEM display. If an invalid item number was entered, indicator 05 is not on. If a valid item number was entered, indicator 05 is on, which suppresses input to the ZITEM format. However, input is not suppressed on the SHOWITEM format; therefore, input is allowed to the ZITEM format when the SHOWITEM format is displayed. When multiple formats are used, a suppress input-no specification on the last format displayed allows the last format that contains input fields to accept input.



IBM

GX21-9253- U/M 050*
Printed in U.S.A.

System/34 Display Screen Format Specifications

Form Type	Format Name	Start Line Number	Number of Lines to Clear
S	SHOWITEM	1Ø	Ø

Form Type	Field Name	Field Length	Line Number	Horizontal Position	Output Data	Underline	Constant Type	Constant Data
D		8	5	3	Y		C	ITEM NO
D		22	5	14	Y		C	DESCRIPTION
D		9	5	41	Y		C	PRICE
D		6	5	53	Y		C	ON HAND
D		6	5	62	Y		C	SOLD
D	ITEMNO	6	6	3	Y	Y		
D	ONHAND	6	6	53	Y	Y		
D	PENDNG	6	6	62	Y	Y		
D	PRICE	9	6	41	Y	Y		
D	DESCR	22	6	14	Y	Y		
D								
D								

Figure 13-16 (Part 1 of 3). Sample Program INQPUT with Override Fields

Control Specifications

Line	Form Type	Size to Compile	Object Output	Listing Options	Size to Execute	Debug	MFCM Stacking Sequence	Date Format	Date Edit	Inverted Print	360/20 2501 Buffer	Number Of Print Positions	Alternate Collating Sequence	Model 20: Address to Start	Work Tapes	Overlay Open	Overlay Printer	Binary Search	Tape Error	2152 Checking	Inquiry	Model 20: Read/Write/Compute	Keyboard Output	Sign Handling	1P Forms Position	Indicator Setting	File Translation	Punch MFCU Zeros	Nonprint Characters	Table Load Halt	Shared I/O	Field Print	Formatted Dump	RPG to RPG II Conversion	Number of Formats	Refer to the specific System Reference Library manual for actual entries.
01	H																																		2	

File Description Specifications

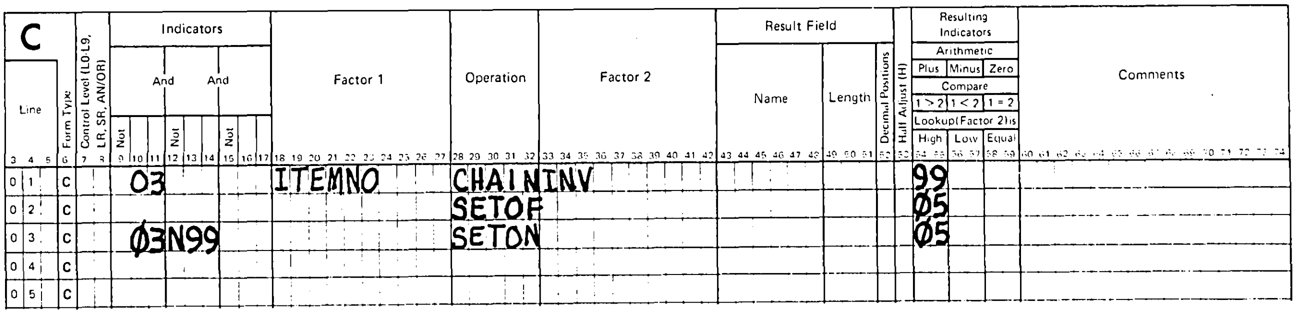
Line	Form Type	Filename	File Type (I/O/U/C/D)	File Designation (P/S/C/R/T/D/F)	End of File (E)	Sequence (A/D)	File Format (F/V/S/M/D)	Block Length	Record Length	Mode of Processing (L/R)	Length of Key Field or of Record Address Field	Record Address Type (A/P/I/K)	Type of File Organization or Additional Area (I/X/D/T/R or 2)	Overflow Indicator	Key Field Starting Location	Extension Code E/L	Device	Symbolic Device	Labels S/N/E/M	Name of Label Exit	Extent Exit for DAM / Storage Index	Continuation Lines: K	Option	Entry	A/U	Number of Tracks for Cylinder Overflow	Number of Extents	Tape Rewind (R/U/N)	File Condition U1-U8
02	F	WK	C	P			F		49								WORKSTN												
03	F																					K	NUM	4					
04	F	INV	I	C				45	45	R	6	A	I		1		DISK												
05	F																												
06	F																												

Input Specifications

Line	Form Type	Filename	Sequence	Number (1/N)	Option (O), U	Record Identifying Indicator, **, or DS	1: Position	1: Not (N)	1: C/Z/D	1: Character	2: Position	2: Not (N)	2: C/Z/D	2: Character	3: Position	3: Not (N)	3: C/Z/D	3: Character	Stacker Select	P/B/L/R	Field Location From	Field Location To	Decimal Positions	Field Name	Control Level (L1-L9)	Matching Fields or Chaining Fields	Field Record Relation	Field Indicators Plus	Minus	Zero or Blank
01	I	WK	NS			Ø2	1		C																					
02	I		NS			Ø3	1	N	C	/																				
03	I																				1	6		ITEMNO						
04	I		NS			Ø4	1		C	/																				
05	I	INV	NS			1Ø																								
06	I																				1	6		ITEMNO						
07	I																				7	11	Ø	ONHAND						
08	I																				12	16	Ø	PENDNG						
09	I																				17	23	2	PRICE						
10	I																				24	45		DESC						
11	I																													
12	I																													

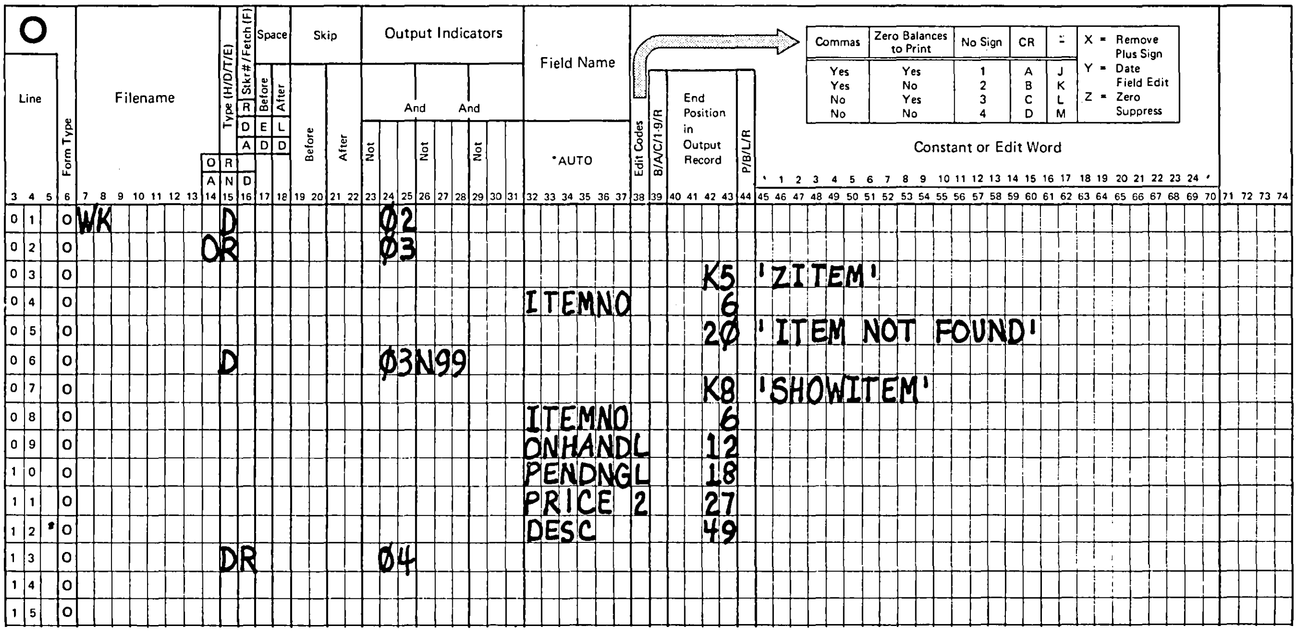
Record identifying indicator 02 is set on when the first record (a blank record) is read from the display station. Record identifying indicator 03 is set on when the operator enters an item number. Record identifying indicator 04 is set on when the operator requests end of job.

Figure 13-16 (Part 2 of 3). Sample Program INQPUT with Override Fiel.'s



Line	Form Type	Control Level	Indicators	Factor 1	Operation	Factor 2	Result Field Name	Length	Decimal Positions	Half Adjust (H)	Resulting Indicators	Comments
01	C		03	ITEMNO	CHAIN	INV					99	
02	C				SETOF						Ø5	
03	C		Ø3N99		SETON						Ø5	
04	C											
05	C											

The CHAIN operation uses the ITEMNO field to retrieve the record from the INV file. Indicator 99 is set on if a record is not found in the INV file; indicator 05 is set off. If indicator 99 is not on, indicator 05 is set on. Indicator 05, if on, suppresses input to the ZITEM format.



Commas	Zero Balances to Print	No Sign	CR	-
Yes	Yes	1	A	J
Yes	No	2	B	K
No	Yes	3	C	L
No	No	4	D	M

X = Remove Plus Sign
Y = Date Field Edit
Z = Zero Suppress

Line	Form Type	Filename	AND/OR	Type (H/D/T/E)	Stkr#/Fetch (F)	Space Before	Space After	Skip Before	Skip After	Output Indicators	Field Name	Edit Codes	B/A/C/1-9/R	End Position in Output Record	P/B/L/R	Constant or Edit Word
01	O	WK		D						Ø2						
02	O		OR							Ø3						
03	O													K5		'ZITEM'
04	O										ITEMNO			6		
05	O													2Ø		'ITEM NOT FOUND'
06	O			D						Ø3N99						
07	O													K8		'SHOWITEM'
08	O										ITEMNO			6		
09	O										ONHAND	L		12		
10	O										PENDNG	L		18		
11	O										PRICE	2		27		
12	O										DESC			49		
13	O			D	R					Ø4						
14	O															
15	O															

The ZITEM format is written to the display if indicator 02 or indicator 03 is on. The SHOWITEM format is written to the display if indicator 03 is on and indicator 99 is not on. If indicator 04 is on, the display station is released from the program by the R in column 16.

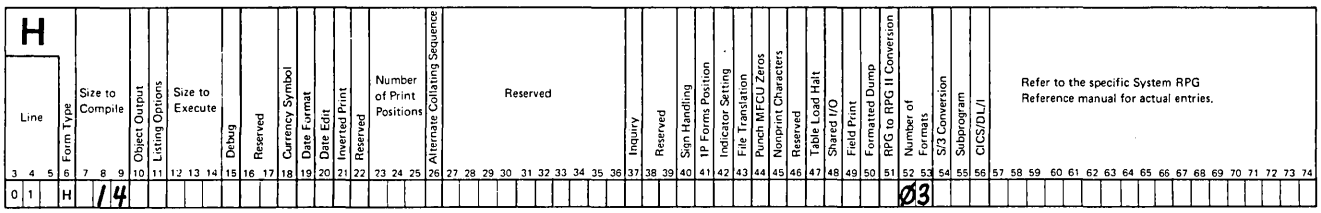
Figure 13-16 (Part 3 of 3). Sample Program INQPUT with Override Fields

Sample Program INQRY2

The sample program INQRY2, shown in Figure 13-17, is very similar to sample program ITMINQ. The only difference is that the WORKSTN file, WK, is processed as a demand file, rather than as a primary file. The display prompts the operator to enter an item number that is used to chain to the record in the disk file. Since WK is a demand file, the item number must be explicitly read into the program via the READ operation code. If the record is found, the item number, description, price, onhand quantity and quantity sold are displayed on line 6.

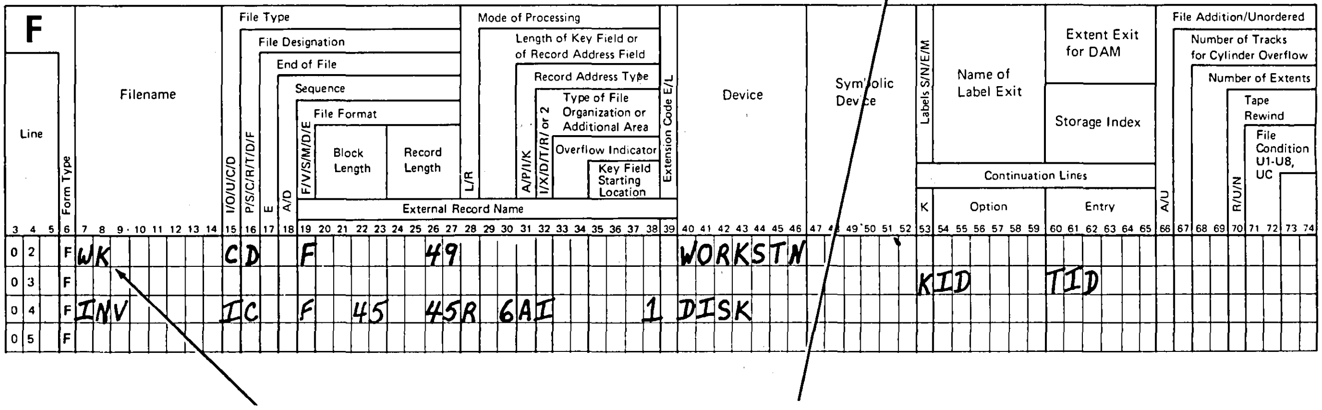
If the item number does not match an item in the disk file, the message ITEM NOT FOUND is displayed on line 24. After the item record or the error message is displayed, the display prompts the operator for a new item number. The operator ends the job by entering a / (slash). 'Figure 13-14 (parts 1 and 2) show the display screen format specifications for sample program ITMINQ. The same display screen format specifications are used for sample program INQRY2. For a description of these display screens, see *Display Screen Format* under *Sample Program ITMINQ* in this chapter.

Control Specifications



```
01 H 14                                          Ø3
```

File Description Specifications



```
02 FWK      CD  F      49            WORKSTN
03 F                                              KID    TID
04 FINV     IC  F  45  45R 6AI     1 DISK
05 F
```

The WORKSTN file is combined (C in column 15) and demand (D in column 16). WORKSTN record length is the same as the largest input or output record.

This entry specifies the number of individual formats in the format load member, INQRY2FM.

Figure 13-17 (Part 1 of 3). Sample Program INQRY2

Input Specifications

Line	Form Type	Filename or Record Name	Sequence	Record Identifying Indicator	Record Identification Codes 1: Position	Not (N)	C/Z/D	Character	Field Location: From	To	Decimal Positions	RPG Field Name
01	I	WK	NS	Ø2	1		C					
02	I		NS	Ø3	1	N	C	/				
03	I								1	6		ITEMNO
04	I		NS	Ø4	1		C	/				
05	I	INV	NS	1Ø								
06	I								1	6		ITEMNO
07	I								7	11	Ø	ONHAND
08	I								12	16	Ø	PENDNG
09	I								17	23	2	PRICE
10	I								24	45		DESC
11	I											

Indicator 02 is set on when a blank record is read from the display station. A blank record is constructed to satisfy the first read to a device. Indicator 03 is set on when an item number is read from the display station, and indicator 04 is set on when a slash is read (end-of-job).

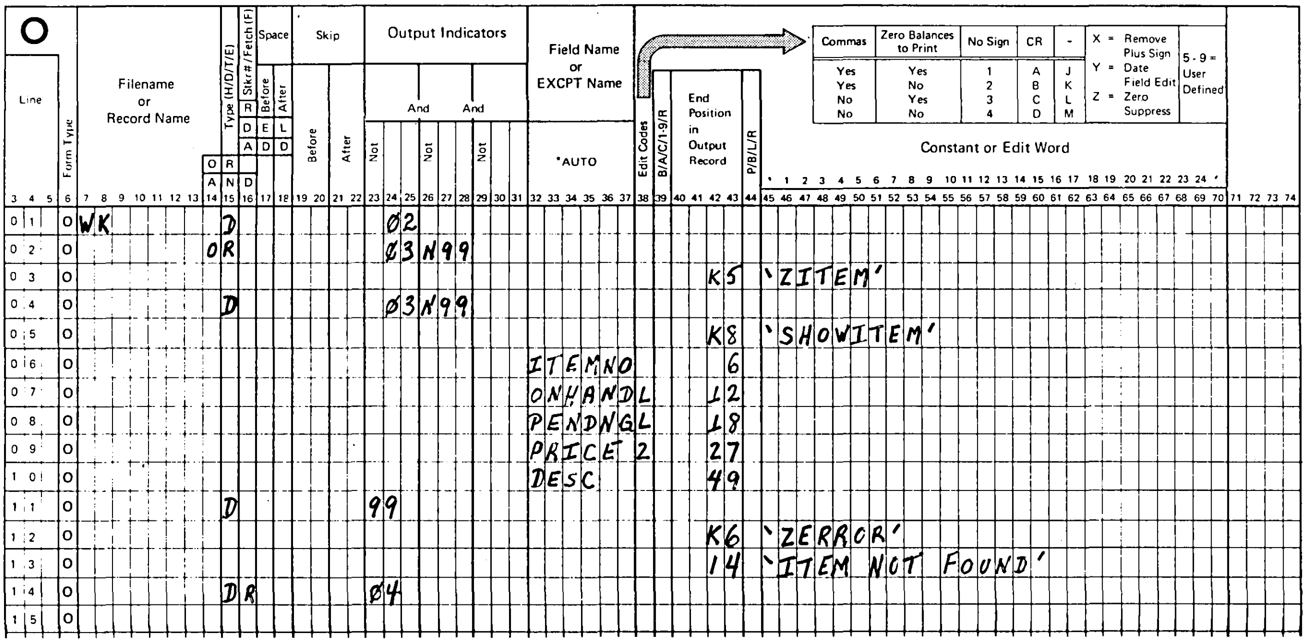
Calculation Specifications

Line	Form Type	Indicators	Factor 1	Operation	Factor 2	Resulting Indicators: Plus / 1>2 / High	Minus / 1<2 / Low	Zero / 1=2 / Equal
01	C			SETOF		99		
02	C			READ	WK			LR
03	C	Ø3	ITEMND	CHAIN	INV	99		
04	C							

A WORKSTN demand file requires a READ operation to read data from a display station into the program.

Indicator 99 is set on if the item is not found, and is used to condition the error message ITEM NOT FOUND (see the output specifications, lines 11 through 13).

Figure 13-17 (Part 2 of 3). Sample Program INQRY2



Line	Form Type	Filename or Record Name	Type (H/D/T/E)	Stkr #/Fetch (F) / Release	Output Indicators	Field Name or EXCPT Name	Edit Codes	End Position in Output Record	Constant or Edit Word
01	O	WK	D		Ø2				
02	O	OR			Ø3 N99				
03	O							K5	'ZITEM'
04	O		D		Ø3 N99				
05	O							K8	'SHOWITEM'
06	O					ITEMNO		6	
07	O					ONHAND	L	12	
08	O					PENDNG	L	18	
09	O					PRICE	2	27	
10	O					DESC		49	
11	O		D		99				
12	O							K6	'ZERROR'
13	O							14	'ITEM NOT FOUND'
14	O		D	R	Ø4				
15	O								

Commas	Zero Balances to Print	No Sign	CR	-
Yes	Yes	1	A	J
Yes	No	2	B	K
No	Yes	3	C	L
No	No	4	D	M

X = Remove Plus Sign
Y = Date Field Edit
Z = Zero Suppress
5 - 9 = User Defined

An R in column 16 releases the display station from the program. No format name is necessary on the release. If one is given, the format is written to the display station as output, then the display station is released from the program.

The PRICE field is edited by the RPG II edit code 2. ONHAND and PENDING are edited by the RPG II edit code L.

Figure 13-17 (Part 3 of 3). Sample Program INQRY2

Saving Fields (SAVDS) or Indicators (IND) for a WORKSTN File

When a WORKSTN file allows multiple devices (continuation line keyword NUM is greater than 1), certain fields and indicators can be placed in SAVDS and IND for each device. This allows you to save the state of an indicator or the contents of a field that is unique to this display station. SAVDS and IND allow you to save your place in the program while another requestor is executing the program. For more information on what is in the SAVDS and IND areas and when this information is updated, see *Continuation Line Options* in this chapter.

For example, in the sample WORKSTN file program ORD (see Chapter 19), the array IQPD and its index should be placed in SAVDS, but the customer address should not. Indicator 01 (first-time indicator for each display station) should be placed in IND, but the other indicators should not. All other indicators in the sample program are reset before the next input record is processed: Indicators 10 through 13, 91, and 92 are record identifying indicators; indicators 96 through 99 are reset by the CMDKEY subroutine; indicators KA, KB, and KC are reset on input from a display station; indicator 90 is reset by calculation statement 0058 or 0098; and indicator 15 is never on at input time.

Note: Indicators may need to be reset in the program; they are not always reset by RPG II in time to be useful to the programmer.

The following types of fields and indicators do not need to be placed in SAVDS and IND:

- Work fields that are used during one cycle (between input operations for the WORKSTN file), but can then be destroyed. For example, in the sample WORKSTN file program ORD (see Chapter 19), the field IQPD00 that is moved to IQPD is a work field. The fields such as AMOUNT that are recalculated each time are also work fields. All fields in the data structure CRECD are output to the display screen on one cycle, and read in and written to disk on the next cycle; therefore, they do not have to be saved.

- Job fields that are used by all display stations but are not destroyed. An example of a job field would be a field such as TOTAL1 that contains the sum of the total fields at the end of each order, or a field such as COUNT that contains a count of the remaining records in the file TRANS. An example of a job level indicator would be one that is set on if the number of records remaining in the TRANS file is less than 100. If that indicator is on, no new orders can be started.

Using MICs with a WORKSTN File

When a MIC (message identification code) is to be displayed for a WORKSTN file, the length of the message must be entered in the field length column of the display screen format specifications and the constant type column must contain an M. The name of a 6-character field or a 6-character constant must then be specified on the RPG II output specifications. The contents of the field or the constant must be in the form xxxxyy, where xxxx is the MIC number and yy is the 2-character message member identifier. For a complete description of the message member identifier, see Columns 57-79 (Constant Data) under *Field Definition Specifications* in Chapter 4 of the *System Support Reference Manual.*

DEBUGGING WORKSTN PROGRAMS

Because the logic for WORKSTN file processing is supplied by both the RPG II program and the display screen format specifications, it may be more difficult to isolate coding errors for the WORKSTN file than for other files. The following techniques may help you debug a WORKSTN program:

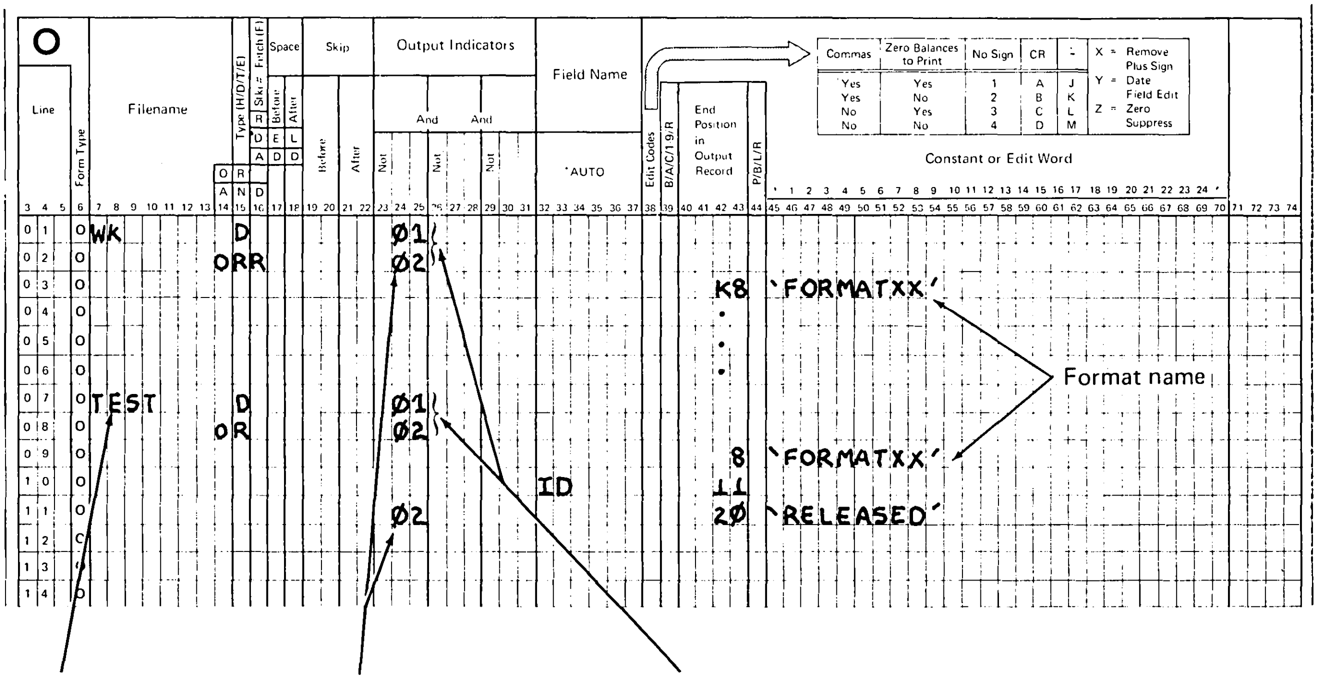
- Always compare the $SFGR listing to the RPG II input and output specifications. The from/to and end positions used on the RPG II specifications must match the from/to and end positions listed for the $SFGR input and output buffers.
- If the wrong format is displayed, check the status of the indicators to be certain the status is as you expected. If the status of the indicators is incorrect, the wrong format may be displayed or a correct format may be followed by an additional format that overlays and thereby hides the correct format. The specification of erase input (columns 31 and 32) or override fields (columns 33 and 34) may also cause a partial format to be displayed that overlays the correct format.
- Always include a record type for blank records. Blank records can occur in one of two ways: (1) if the record is the first input record for a display station (in most programs the first input record for a display station is blank); (2) if N (no) is specified in column 22 (return input) of the display screen format S specification and no data keys were pressed.
- If the program goes to end of job prematurely, check whether all display stations have been released or whether Y (yes) was specified in column 35 (suppress input) of the S specification. Either situation can result in no display stations being allowed to enter input, which causes end of file on the WORKSTN file. If the program is an NEP and either of the preceding conditions is true and if the operator enters a STOP SYSTEM command, the WORKSTN file goes to end of file.
- If the COMMAND display unexpectedly follows a program display, the program may have gone to end of job *before* any data was entered for the display (see the RESTORE parameter of the WORKSTN OCL statement in the *System Support Reference Manual*). If RESTORE-NO is specified, a display from the program may be on the screen after the program has gone to end of job so it appears as if the program is still executing. If RESTORE-YES is specified, the COMMAND display appears on the screen immediately when the program goes to end of job.
- Avoid using multiple formats on the same section of the screen until the program logic is debugged.
- During the debugging operations, display a constant on the screen for every format. This should help you analyze the screen contents.
- Use the DEBUG operation code in selected locations to trace the program flow. Suggested locations and the resulting debug information are as follows:

Location	**Debug Information**
As first calculation	Shows the contents of the specified input record and the indicator status for a primary file
After any READ operation	Shows the contents of the specified input record and the indicator status for a demand file
Before every EXCPT operation	Shows the status of the indicators that control which records (formats) are to be produced as output
As last detail calculation	Shows the indicator status before heading and detail output
After an ACQ operation	Shows the work station ID and the indicator off if the operation was successful
After a REL operation	Shows the display stations that are released from the program
After every TAG operation	Shows the program flow
As first statement in each subroutine	Shows the program flow
Conditioned by LR	Shows when the program ends

- After each WORKSTN output record, define a record with the same conditioning indicators and write that record to the DEBUG file (see Figure 13-18). The record should contain:
  - The format name
  - The work station ID, if used in the program
  - The release status if the display station is released in the output specifications
  - SLN (starting line number), if used in the program
  - Data fields as needed

If the following types of error messages occur, check the probable causes listed:

- Error messages involving program checks to the WORKSTN device are probably caused by (1) invalid use of erase input fields (columns 31 and 32 of the S specification) or (2) clearing all or a portion of the screen containing the input fields.
- Error messages involving invalid WORKSTN IDs are probably caused by an earlier release of the display station in either calculation or output operations.


O
Line
Form Type
Filename
Type (H/D/T/E)
Space
Skip
Output Indicators
Field Name
*AUTO
Edit Codes
B/A/C/1-9/R
End Position in Output Record
P/B/L/R
Constant or Edit Word
Commas
Zero Balances to Print
No Sign
CR
X = Remove Plus Sign
Y = Date Field Edit
Z = Zero Suppress
Format name

Line	Form Type	Filename	AND/OR	Type	Release	Output Indicators	Field Name	End Position	Constant or Edit Word
01	O	WK		D		Ø1			
02	O		OR		R	Ø2			
03	O							K8	'FORMATXX'
04	O							•	
05	O							•	
06	O							•	
07	O	TEST		D		Ø1			
08	O		OR			Ø2			
09	O							8	'FORMATXX'
10	O						ID	11	
11	O					Ø2		2Ø	'RELEASED'
12	O								
13	O								
14	O								

Use the same file that is used for DEBUG operations.

Indicator 02 shows release status of display station.

Use the same conditioning indicators for both files.

Figure 13-18. Writing the WORKSTN Output Record to the DEBUG File

Chapter 14. Tables and Arrays

Tables and arrays are systematic arrangements of data items having like characteristics; that is, the same field length, data type (alphameric or numeric), and number of decimal positions. A table generally contains constant data that is used for calculation or printing with variable transaction data. Arrays are generally used for variable data and totals that are used independently of the variable transaction data. Both tables and arrays are described on the extension specifications (see Chapter 4).

Important differences exist, however, in defining and processing tables and arrays. Tables and arrays are defined in terms of when they are loaded for use by the program. Tables can be loaded at compile time or preexecution time, and arrays can be loaded at compile time, preexecution time, or execution time. During the processing of a program, tables can be searched one item at a time for a specific item of data with a unique identifier. Arrays can also be searched for a uniquely identified data item. Unlike tables, however, array items can also be referenced by their relative position to other items. To do this, you must provide an index to a specific item in the array. In addition, an entire array can be processed sequentially when you specify the array name only once in certain calculation operations.

The following terms are used to describe tables and arrays:

- *Compile-time tables and arrays* are loaded with the source program and become a permanent part of the object program. The initial content of a compile-time table or array can be changed only when the source program is recompiled with the revised table or array.
- *Preexecution-time tables and arrays* are loaded with the object program before actual execution of the RPG II program begins; that is, before any input files are read, calculations are performed, or output functions are performed.
- *Execution-time arrays* are loaded or created by input or calculation specifications. The arrays are loaded after actual execution of the RPG II program has begun; that is, they are read in as input data or created during calculations in the program. An execution-time array is also described on the extension specifications. Tables cannot be specified for execution time.
- *Related tables and arrays* are two tables or two arrays that are used together. The items in the tables or arrays are called corresponding items; each item in the second table or array gives additional information about its corresponding item in the first table or array. Tables can be related or arrays can be related; however, a table cannot be related to an array or vice versa.

  Figure 14-1 shows related tables A (TABA) and B (TABB). An item in table A gives a part number, and the corresponding item in table B gives the part cost. Although all items within one table or array must have the same characteristics, corresponding items of related tables or arrays can have different characteristics. Related tables and arrays do not have to have the same number of entries. However, unpredictable results can occur if the LOKUP operation is used for related tables or arrays with an unequal number of entries (see *LOKUP with Two Tables* in Chapter 10, *Operation Codes*).
- *Short tables and arrays* are those in which not all of the entries contain data. The unused entries in numeric tables and arrays are filled with zeros; the unused entries in alphameric tables and arrays are filled with blanks. Usually short tables or arrays are created if only a few table or array items are available when the table or array is built, but more items are to be included. Short tables and arrays must have at least one entry.
- *Full tables and arrays* are those in which all possible entries contain data.
- *Entry* is one element in a single table or array or corresponding items in related tables or arrays.

TABA (part number)	TABB (unit code)
345126	373
38A473	498
39K143	1297
40B125	93
41C023	3998
42D893	87
43K832	349
44H111	679
45P673	898
46C732	47587

Related tables TABA and TABB can be described as two separate table files or as one table file in alternating format.

Records for TABA and TABB when described as two separate table files

TABA entry	TABA entry	TABA entry	TABA entry	TABA entry	TABA entry	TABA entry	TABA entry	TABA entry	TABA entry	—— blank ——
1 2 3 4 5 6	7 8 9 10 11 12	13 14 15 16 17 18	19 20 21 22 23 24	25 26 27 28 29 30	31 32 33 34 35 36	37 38 39 40 41 42	43 44 45 46 47 48	49 50 51 52 53 54	55 56 57 58 59 60	61 62 63 64 65 66 67 68 69 70

This record contains TABA entries in positions 1 through 60.

TABB entry	TABB entry	TABB entry	TABB entry	TABB entry	TABB entry	TABB entry	TABB entry	TABB entry	TABB entry	—— blank ——
1 2 3 4 5	6 7 8 9 10	11 12 13 14 15	16 17 18 19 20	21 22 23 24 25	26 27 28 29 30	31 32 33 34 35	36 37 38 39 40	41 42 43 44 45	46 47 48 49 50	51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70

This record contains TABB entries in positions 1 through 50.

Records for TABA and TABB when described as one table file in alternating format

TABA entry	TABB entry	TABA entry	TABB entry	TABA entry	TABB entry	TABA entry	TABB entry	TABA entry	TABB entry	TABA entry	TABB entry
1 2 3 4 5 6	7 8 9 10 11	12 13 14 15 16 17	18 19 20 21 22	23 24 25 26 27 28	29 30 31 32 33	34 35 36 37 38 39	40 41 42 43 44	45 46 47 48 49 50	51 52 53 54 55	56 57 58 59 60 61	62 63 64 65 66

TABA entry	TABB entry	TABA entry	TABB entry	TABA entry	TABB entry	TABA entry	TABB entry	—— blank ——
67 68 69 70 71 72	73 74 75 76 77	78 79 80 81 82 83	84 85 86 87 88	89 90 91 92 93 94	95 96 97 98 99	100 101 102 103 104 105	106 107 108 109 110	111 112 113 114 115 116 117 118 119 120

This record contains TABA and TABB entries in alternating format in positions 1 through 110.

Figure 14-1. Related Tables

RULES FOR CREATING TABLE OR ARRAY INPUT RECORDS

Table and array input records must be formatted according to certain rules:

- The first table or array entry for each input record must begin in position 1.
- An entire record need not be filled with entries. If it is not, blanks or comments can be included after the entries (see Figure 14-2).
- Each input record except the last must contain the same number of entries. A record can contain just one entry or as many entries as the record can hold.
- Each entry must be contained entirely on one input record. An entry cannot be split between two records; thus, the length of a single entry is limited to the maximum record length for the input device. If related tables or arrays are used and are described in alternating format, corresponding items must be on the same input record; together they cannot exceed the maximum record length for the device.
- Related tables or arrays can be described separately or in alternating format. Alternating format means that, together, the corresponding items are considered one table or array entry.
- The number of table and/or array names or data structures used in a program cannot exceed 75. The number of compile-time tables and/or arrays cannot exceed 70.

DEFINING TABLES AND ARRAYS

All tables and arrays must be described on the extension specifications sheet. One line describes each set of table or array input records. See Chapter 4, *Extension Specifications*, for a complete description of the columns on the extension specifications sheet.

If only one table or array is described, columns 11 through 45 are used. If alternating tables or arrays are described on one set of input records, the second table or array is described in columns 46 through 57 of the same line as the first table or array. If preexecution-time tables and arrays are being described, entries in columns 11 through 18 and 27 through 45 are required. Columns 19 through 26 are used if the table or array is to be written at the end of the job. Columns 11 through 18 are not used for compile-time tables and arrays or execution-time arrays.

Tables and arrays can be specified in any sequence. Compile-time and preexecution-time tables and arrays can be mixed. However, the sequence in which tables and arrays are specified on the extension specifications determines the order in which they must be loaded at the start of the job.

```
1234512345 Comments can be
1234512345              anywhere out here
1234512345                                  or here
12345        or here (that is, after the last entry position for the longest record).

1234512345
1234512345
1234512345
12345  If comments begin here, RPG cannot tell if you intend them as comments or if you
       provided too much data for the table/array. Therefore, it issues a warning message.
```

Each of the two tables/arrays contains seven entries, each entry 5 positions long, with two entries per record. The last record contains only one entry. The remaining 5 positions in the last record should be left blank, because using these positions for comments causes warning message RPG-0333, TABLE/ARRAY FULL OR NO TABLES/ARRAYS FOR FOLLOWING DATA. Therefore, comments should begin after the last entry position for the longest record; that is, (the number of entries per record x the number of positions per entry) + 1.

Figure 14-2. Table Input Record With Comments

Figure 14-3 shows the extension specifications required for the three types of arrays:

- Line 1 specifies a compile-time array, ARC, which has a total of eight elements (three elements per record). Each element is 12 characters in length, with four decimal positions.

- Line 3 specifies a preexecution-time array, ARE, to be read from file DISKIN. ARE has 250 alphameric elements (12 elements per record). Each element is 5 characters long. The elements are arranged in ascending sequence.

- Line 5 specifies an execution-time array, ARI, to be read from input records. ARI has 10 numeric elements, each 10 characters long with zero decimal positions.

Any of these specifications can also include entries in columns 19 through 26 that define the name of a file to which the array is written at end of job and in columns 46 through 57 that define an alternating array.

LOADING TABLES AND ARRAYS

Tables and arrays can be loaded at compilation time or preexecution time. Only arrays can be loaded at execution time.

Compile-Time Tables and Arrays

Tables and arrays loaded at compilation time are compiled along with the RPG II source program and become a part of that program. Rules for loading tables and arrays at compile time are as follows:

- Compile-time tables must be entered into the system following the RPG II source program. See *How RPG II Works* in Chapter 1 for a description of entering the source program.

- A record with **ƀ in positions 1 through 3 must precede the first table or array input record.

- Tables and arrays must be loaded in the same order as they are described on the extension specifications.

- A compile-time array must have entries in columns 33 through 35 of the extension specifications and must not have entries in columns 11 through 18 of the extension specifications.

- Compile-time tables and arrays must not be in packed or binary format.

- For compile-time arrays, the maximum length of an alphameric entry is 96 because the maximum length of a record in the source program is 96 characters.

Figure 14-4 shows the arrangement on disk of the RPG II source program when compile-time tables are loaded.

Preexecution-Time Tables and Arrays

Preexecution-time tables and arrays are not part of the source program. They are loaded by the RPG II object program similar to other data files.

Preexecution-time tables or arrays are loaded from the disk. The table or array file must have been created earlier. OCL statements are used just prior to program execution to identify the table or array file of the disk. If two or more tables or arrays are to be loaded, they must be loaded from different disk files.

No error is indicated when a sequenced table or array is not completely filled at preexecution time, even though the unfilled entries are initialized to blank or zero and, therefore, create a logical sequence error.

Execution-Time Arrays

To load an array from information in input records, describe that information in the input specifications. The specifications made depend on whether the array information is contained in one or more than one record. Any type of array (compile-time, preexecution-time, or execution-time) can be referenced in the input specifications. Execution-time arrays are not sequence checked; however, the array sequence (A or D in column 45) must be specified if a high or low LOKUP operation is used.

Array Information In One Record

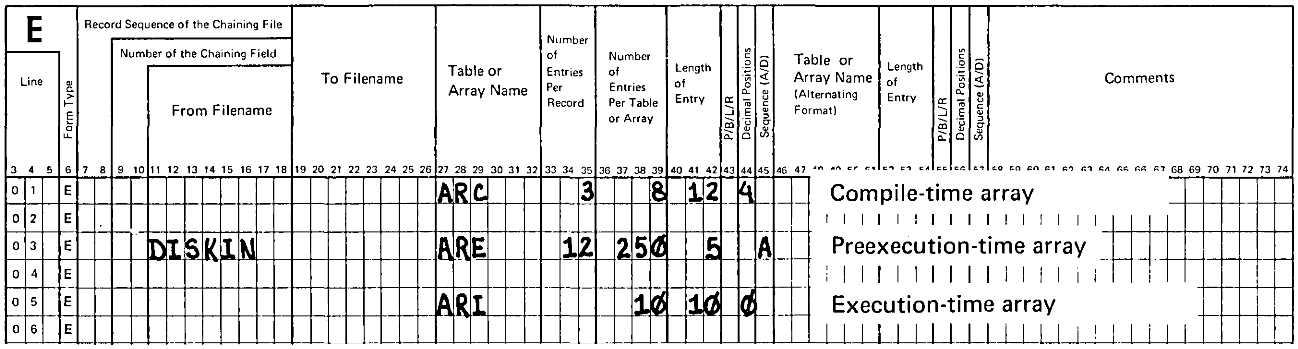
If the array information is contained in one record, the information can occupy consecutive positions in the record or it can be scattered throughout the record.

If the array elements are consecutive on the input record, the array can be loaded with a single input specification. Figure 14-5 shows the specifications for loading an array, INPARR, of six elements (12 characters each) from a single record from the file ARRFILE.

If the array elements are scattered throughout the record, they can be defined and loaded one at a time, with one element described on a specification line. Figure 14-6 shows the specifications for loading an array, ARRX, of six elements with 12 characters each, from a single record from file ARRFILE; a blank separates each of the elements from the others.

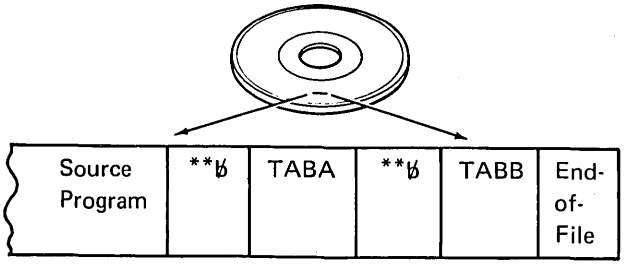
An array can be located in a data structure only if the whole array is contained in the data structure and all elements are consecutive. See *Data Structure* under *Columns 19-20* in Chapter 7.

Extension Specifications



Line (3-5)	Form Type (6)	Record Sequence of the Chaining File (7-8)	Number of the Chaining Field (9-10)	From Filename (11-18)	To Filename (19-26)	Table or Array Name (27-32)	Number of Entries Per Record (33-35)	Number of Entries Per Table or Array (36-39)	Length of Entry (40-42)	P/B/L/R (43)	Decimal Positions (44)	Sequence (A/D) (45)	Table or Array Name (Alternating Format) (46-51)	Length of Entry (52-54)	P/B/L/R (55)	Decimal Positions (56)	Sequence (A/D) (57)	Comments (58-74)
01	E					ARC	3	8	12		4							Compile-time array
02	E																	
03	E			DISKIN		ARE	12	25Ø	5			A						Preexecution-time array
04	E																	
05	E					ARI		1Ø	1Ø		Ø							Execution-time array
06	E																	

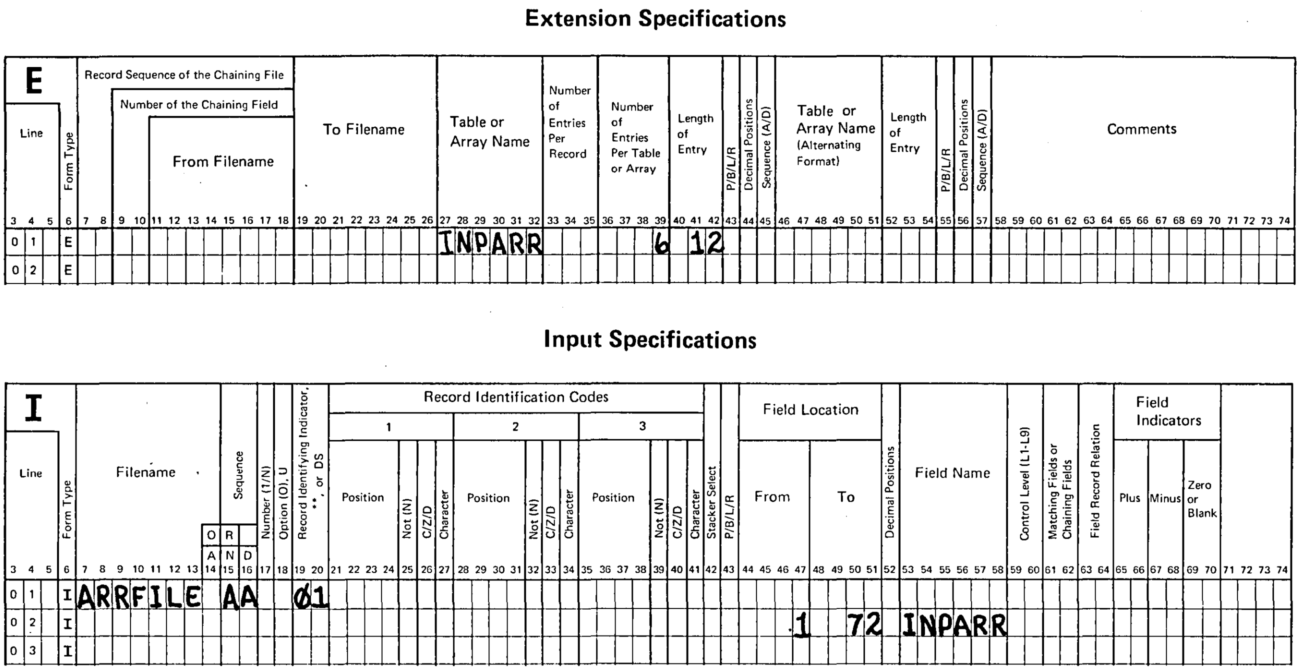
Figure 14-3. Extension Specifications for Three Types of Arrays



These tables are located in the library with the source program.

b̸ = blank

Figure 14-4. Arrangement of the RPG II Source Program and Nonalternating Compile-Time Table Data



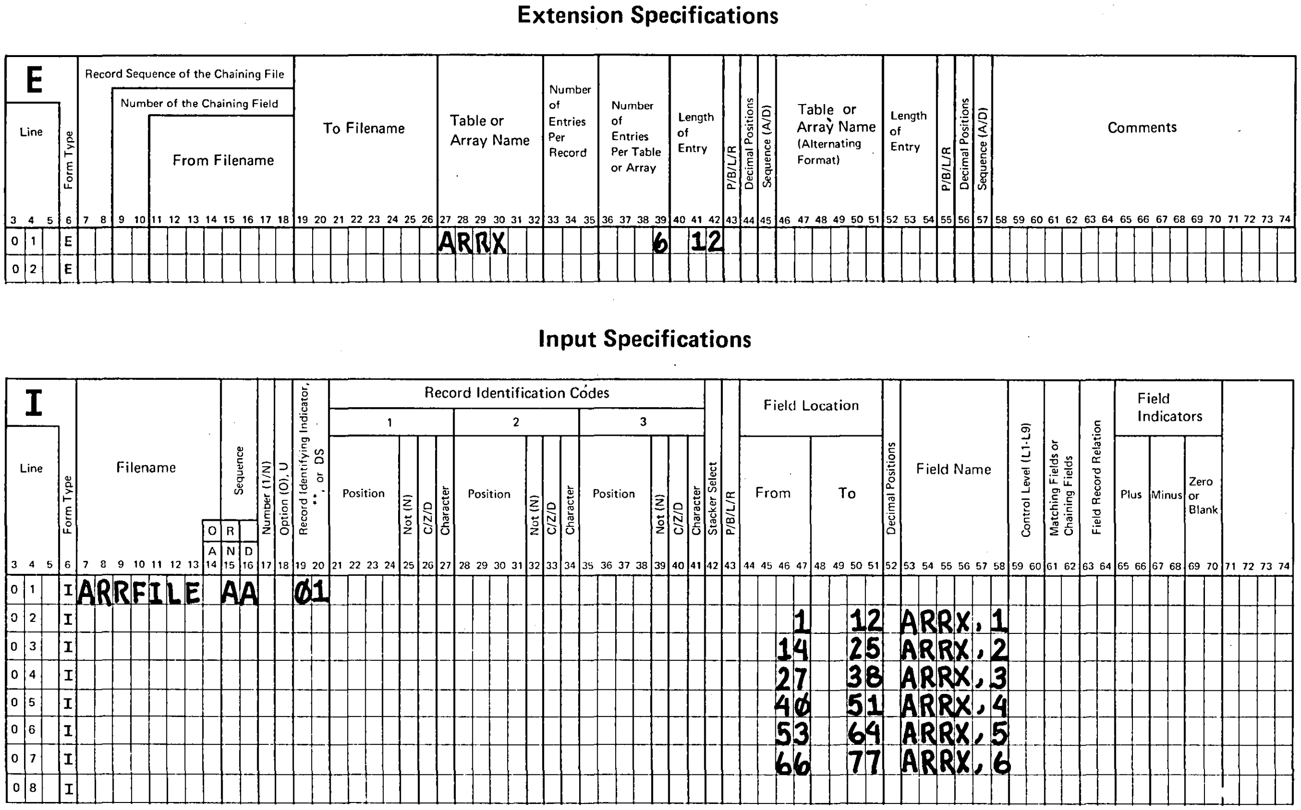
Extension Specifications

Line	Form Type	Record Sequence of the Chaining File	Number of the Chaining Field	From Filename	To Filename	Table or Array Name	Number of Entries Per Record	Number of Entries Per Table or Array	Length of Entry	P/B/L/R	Decimal Positions	Sequence (A/D)	Table or Array Name (Alternating Format)	Length of Entry	P/B/L/R	Decimal Positions	Sequence (A/D)	Comments
01	E					INPARR		6	12									
02	E																	

Input Specifications

							Record Identification Codes														Field Location							Field Indicators		
							1				2				3															
Line	Form Type	Filename	Sequence	Number (1/N)	Option (O), U	Record Identifying Indicator, **, or DS	Position	Not (N)	C/Z/D	Character	Position	Not (N)	C/Z/D	Character	Position	Not (N)	C/Z/D	Character	Stacker Select	P/B/L/R	From	To	Decimal Positions	Field Name	Control Level (L1-L9)	Matching Fields or Chaining Fields	Field Record Relation	Plus	Minus	Zero or Blank
01	I	ARRFILE	AA			Ø1																								
02	I																				1	72		INPARR						
03	I																													

Figure 14-5. Defining an Execution-Time Array With Consecutive Elements



Extension Specifications

Line	Form Type	Record Sequence of the Chaining File	Number of the Chaining Field	From Filename	To Filename	Table or Array Name	Number of Entries Per Record	Number of Entries Per Table or Array	Length of Entry	P/B/L/R	Decimal Positions	Sequence (A/D)	Table or Array Name (Alternating Format)	Length of Entry	P/B/L/R	Decimal Positions	Sequence (A/D)	Comments
01	E					ARRX		6	12									
02	E																	

Input Specifications

							Record Identification Codes														Field Location							Field Indicators		
							1				2				3															
Line	Form Type	Filename	Sequence	Number (1/N)	Option (O), U	Record Identifying Indicator, **, or DS	Position	Not (N)	C/Z/D	Character	Position	Not (N)	C/Z/D	Character	Position	Not (N)	C/Z/D	Character	Stacker Select	P/B/L/R	From	To	Decimal Positions	Field Name	Control Level (L1-L9)	Matching Fields or Chaining Fields	Field Record Relation	Plus	Minus	Zero or Blank
01	I	ARRFILE	AA			Ø1																								
02	I																				1	12		ARRX,1						
03	I																				14	25		ARRX,2						
04	I																				27	38		ARRX,3						
05	I																				4Ø	51		ARRX,4						
06	I																				53	64		ARRX,5						
07	I																				66	77		ARRX,6						
08	I																													

Figure 14-6. Defining an Execution-Time Array With Scattered Elements

The following input specifications are required for loading an array from a single input record:

Column	Entry
6	I
7-42	Blank
43	P (packed), B (binary), or blank (zoned decimal).
44-47 and 48-51	Field location of either an entire array (consecutive elements) or individual field locations of single elements of the array.
52	This column must be left blank.
53-58	The name of the array or the name of a single element (array name with index). This array name must be the same name as that used on the extension specifications.
59-62	Blank
63-64	Field record relation indicator. See *Columns 63-64 (Field Record Relation)* in Chapter 7, *Input Specifications*, for information on this entry.
65-74	Blank

Array Information in More Than One Record

If the array information is contained in two or more records, many methods can be used to load the array. The method used is primarily based on the size of the array and whether the array elements are consecutive in the input records. Figure 14-7 shows the array that results when array information is loaded from more than one input record. Each record identified by a 1 or 3 in column 1 contains 6 items of array information. Records identified by a 2 in column 1 do not contain array information, although they appear in the same input file. Examples of loading and storing array information are found in *Examples of Using Arrays* in this chapter. The RPG II program processes one record at a time; therefore, the entire array cannot be processed until all of the records containing the array information are read and the information is moved into the array fields. It may, therefore, be necessary to suppress calculation and output operations until the entire array is read into the system.

SEARCHING TABLES AND ARRAYS

The LOKUP operation can be used to search tables and arrays. See *Lookup Operation* in Chapter 10, *Operation Codes*, for a description of how to use LOKUP.

REFERENCING ARRAYS

Arrays can be used in input, output, or calculation specifications (see *Examples of Using Arrays*). The elements in an array can be referenced individually, or the array can be referenced as a whole. Individual elements are referenced by an array name plus an index. The array name alone references the entire array.

Array Name and Index

The array name is specified beginning in column 27 or column 46 of the extension specifications and must be a valid RPG II name. The array name cannot exceed 6 characters.

If the entire array is to be referenced, use the array name alone. However, if individual elements of the array are to be referenced, the array name requires an index. The index can be a numeric field with zero decimal positions or an unsigned literal (no plus or minus sign). The index must not be zero, negative, or greater than the number of elements in the array. The array name and index must be separated by a comma (for example, AR,IND). The array name with comma and index can never be less than 3 characters long. The total length of an array name with comma and index usually cannot exceed 6 characters. However, if the array name plus index is specified only in factor 1 or factor 2 of the calculation specifications, the array name plus comma and index can be up to 10 characters long.

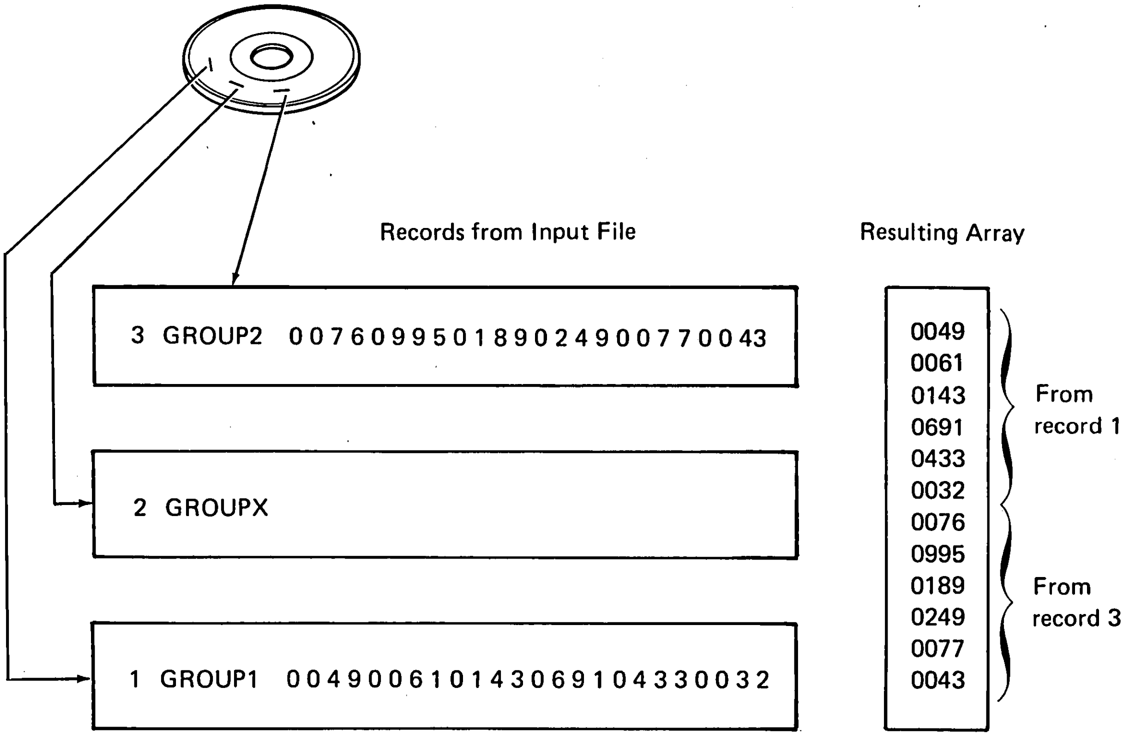


Figure 14-7. Loading an Array from Input Records

Some examples of array names with and without an index are:

Valid	
ARAYOL	
B	
AR,1	First element of array AR.
X,YY2	YY2 is a field name.
Invalid	
BALANCE	Array name has more than 6 characters.
6TOTAL	First character is not alphabetic.
TOTAL-	Name contains special character.
CR TOT	Name contains a blank.
A1,A1	Array is used as index.
BAL,XX1	Name including comma has more than 6 characters. This name is valid for factor 1 and factor 2 of the calculation specifications only.
AR,+1	Array has invalid signed index.

Referencing an Array in Calculations

An entire array or individual elements in an array can be referenced in calculation specifications. Process individual elements like normal fields. Remember, if an array element is to be used as a result field, the array name with the comma and index cannot exceed 6 characters.

To reference an entire array, use only the array name, which can be used as factor 1, factor 2, or the result field. The following operations can be used with an array name: ADD, Z-ADD, SUB, Z-SUB, MULT, DIV, SQRT, MOVE, MOVEL, MOVEA, MLLZO, MLHZO, MHLZO, MHHZO, DEBUG, XFOOT, SORTA, and LOKUP.

Several other operations can be used with an array element only, not the array name alone. These operations are COMP, TESTZ, BITON, BITOF, TESTB, KEY, SET, and MVR.

When specified with an array name, certain operations are repeated for each element in the array. These are ADD, Z-ADD, SUB, Z-SUB, MULT, DIV, SQRT, MOVE, MOVEL, MLLZO, MLHZO, MHLZO, and MHHZO. The following rules apply when these operations are specified with an array name:

- When factors 1 and 2 and the result field are arrays with the same number of elements, the operation uses the first element from every array, then the second element from every array until all elements in the arrays are processed. If the arrays do not have the same number of entries, the operation ends when the last element of the array with the fewest elements has been processed.

- When one of the factors is a field, constant, or figurative constant and the other factor and the result field are arrays, the operation is performed once for every element in the shorter array. The same field, constant, or figurative constant is used in all of the operations.

- The result field must always be an array.

- Resulting indicators (columns 54 through 59) cannot be used because of the number of operations being performed.

- If an operation code uses factor 2 only (such as Z-ADD, Z-SUB, or SQRT) and the result field is an array, the operation is performed once for every element in the array. The same field, constant, or figurative constant is used in all of the operations.

If an array is used as a result field, and an element of that array is used as a factor, the value of that element becomes altered as a result of the calculation. After this occurs, all calculations will use the new value of the element. For example, consider two numeric arrays having the following element values:

ARR1,1 = 2	ARR2,1 = 2
ARR1,2 = 4	ARR2,2 = 8
ARR1,3 = 6	ARR2,3 = 1

If every element of ARR1 is added to element ARR2,2 and the result is placed in ARR2 (the RPG command statement–ARR1 ADD ARR2,2 ARR2 is executed), the elements of the result array ARR2 are:

ARR2,1 = 10(2+8)	
ARR2,2 = 12(4+8)	The value of ARR2,2 is now not 8. The new value will be used for the remaining calculations.
ARR2,3 = 18(6+12)	

MODIFYING CONTENTS OF TABLES AND ARRAYS

Tables and arrays can be temporarily changed during the execution of a job when the table or array name is used as a result field in an arithmetic or move operation. (Arrays are also changed temporarily when the SORTA operation is executed). The appropriate entry in the table or array is modified for the duration of the job. The next time the job is executed, however, the table or array contains the original entries. Temporary changes can be made permanent if you change the table input records.

Figure 14-8 shows the specifications for modifying the contents of related tables TABFIL and TABLIT.

Adding Entries to Short Tables or Arrays

Entries can be added to short tables and arrays before or during execution of the job. The simplest way to add entries to a table or array is to write additional entries on the input records before program execution. However, entries that are created by calculation operations or read from an input record can also be added during execution of a program.

Figure 14-9 shows how entries are added to related numeric tables with the LOKUP and MOVE operations. Such entries are only temporary unless they are written in table input records. If these entries are to become a permanent part of the short table, they must be written in records and included with the other table file records.

TABLE AND ARRAY OUTPUT

Entire tables and arrays can be written to an output file under control of RPG II at end of job when the LR indicator is on. To indicate that an entire table or array is to be written, specify the name of the output file to be used in columns 19 through 26 of the extension specifications.

If an entire array is to be written on an output record (via output specifications), describe the array along with any normal fields for the record:

- Columns 32 through 37 of the output specifications must contain the same array name used on the extension specifications.

- Columns 40 through 43 of the output specifications must contain the record position where the last element of the array is to end.

If an output record is to contain only certain elements from a table or array, describe the elements in the same way as normal fields, using either an array name with an index or a table name as the field name.

Editing Entire Arrays

When editing is specified for an entire array, all elements of the array are edited. If different editing is required for various elements, reference them individually.

When an edit code is specified for an entire array (column 38), two blanks are automatically inserted between elements in the array, that is, to the left of every element in the array except the first. When an edit word is specified instead, the blanks are not inserted. The edit word must contain all the blanks to be inserted.

Calculation Specifications

C			Indicators						Result Field				Resulting Indicators			
Line	Form Type	Control Level (L0-L9, LR, SR, AN/OR)	Not	And Not	And Not	Factor 1	Operation	Factor 2	Name	Length	Decimal Positions	Half Adjust (H)	Arithmetic: Plus / Compare: 1 > 2 / Lookup(Factor 2)is: High	Minus / 1 < 2 / Low	Zero / 1 = 2 / Equal	Comments
3-5	6	7-8	9-11	12-14	15-17	18-27	28-32	33-42	43-48	49-51	52	53	54-55	56-57	58-59	60-74
01	C															
02	C					25	LOKUP	TABFIL	TABLIT						1Ø	
03	C		1Ø				MOVE	5ØØ	TABLIT							
04	C		1Ø				MOVE	3Ø	TABFIL							
05	C															

The item in TABFIL that contains 25 is to be changed to 30. The corresponding item in TABLIT is to be changed to 500. The search word is the constant 25. When a match is found in the table TABFIL, the item from TABFIL and its corresponding item in TABLIT are placed in their respective storage areas. The number 500 is then moved into the storage area for TABLIT; the number 30 is moved into the storage area for TABFIL. The contents of the appropriate original table entry are now modified to agree with the new entry in the special storage areas.

Figure 14-8. Changing Table Data With MOVE Operations

Calculation Specifications

C			Indicators						Result Field				Resulting Indicators			
Line	Form Type	Control Level (L0-L9, LR, SR, AN/OR)	Not	And Not	And Not	Factor 1	Operation	Factor 2	Name	Length	Decimal Positions	Half Adjust (H)	Arithmetic: Plus / Compare: 1 > 2 / Lookup(Factor 2)is: High	Minus / 1 < 2 / Low	Zero / 1 = 2 / Equal	Comments
3-5	6	7-8	9-11	12-14	15-17	18-27	28-32	33-42	43-48	49-51	52	53	54-55	56-57	58-59	60-74
01	C															
02	C		Ø1			ØØØ	LOKUP	TABA	TABB						35	
03	C		35	Ø1			MOVE	NEWA	TABA							
04	C		35	Ø1			MOVE	NEWB	TABB							
05	C															

The LOKUP operation is conditioned by indicator 01. Indicator 01 is on when a record containing information in the fields NEWA and NEWB is read. These fields are to be added to the short tables TABA and TABB respectively. To get the entry in the correct place in the table, a search is made to find the first empty entry. Unfilled entries in short numeric tables are filled with zeros. Thus, the search word used is 000. When the first 000 entry is found, indicator 35 turns on, and NEWA and NEWB are moved into the storage areas of the related tables TABA and TABB. They, in turn, become part of the tables.

Figure 14-9. Adding Entries to Short Tables

EXAMPLE OF USING TABLES

A payroll job requires two related tables: TABNUM is the search table containing employee numbers, and TABRAT is the related table containing employee salary rates (see Figure 14-10). After an employee's rate is found, rate is multiplied by the number of hours worked. The result is the amount earned.

The table input records with eight entries in each record are organized in alternating format. Each table has 500 entries. Therefore, 63 records are required. The first 62 contain characters in positions 1 through 72 (5-character element in TABA plus a 4-character element in TABB, times eight entries per record). The last record has only four entries and contains characters in positions 1 through 36. Figure 14-10 shows the RPG II specifications needed for the job. The following paragraphs explain the entries made.

File Description Specifications

The input records are contained in the input file TIMECARD, which is designated as a primary file (P in column 16). When this file reaches end of file, processing ends (E in column 17). This file is read in from the disk.

The related tables are contained in the input table file. This file is designated as a table file by the T in column 16. The file is read in from disk prior to execution time. An E is required in column 39 to show that additional information about the file is specified in the extension specifications.

Extension Specifications

The extension specifications complete the definition of the file RATETABL. The table searched is TABNUM (columns 27 through 32), which has eight entries in each record (columns 33 through 35) and 500 entries in the table (columns 36 through 39). Each table entry is 5 characters long (columns 40 through 42) with zero decimal positions (column 44). The table is organized in ascending sequence (column 45).

The alternating table is TABRAT (columns 46 through 51). Each entry is 4 characters long (columns 52 through 54) with two decimal positions (column 56).

Input Specifications

The input file TIMECARD is assigned a sequence of AA (columns 15 and 16). Record identifying indicator 01 turns on whenever an input record is present for processing. No record identification codes are specified in columns 21 through 41 because there is only one record type. Lines 02 and 03 describe the locations of the two input fields used by the program. The employee number (EMPNUM) is in positions 1 through 5 of the input record. The number of hours worked by the employee (HRSWKD) is in positions 42 through 44 of the input record.

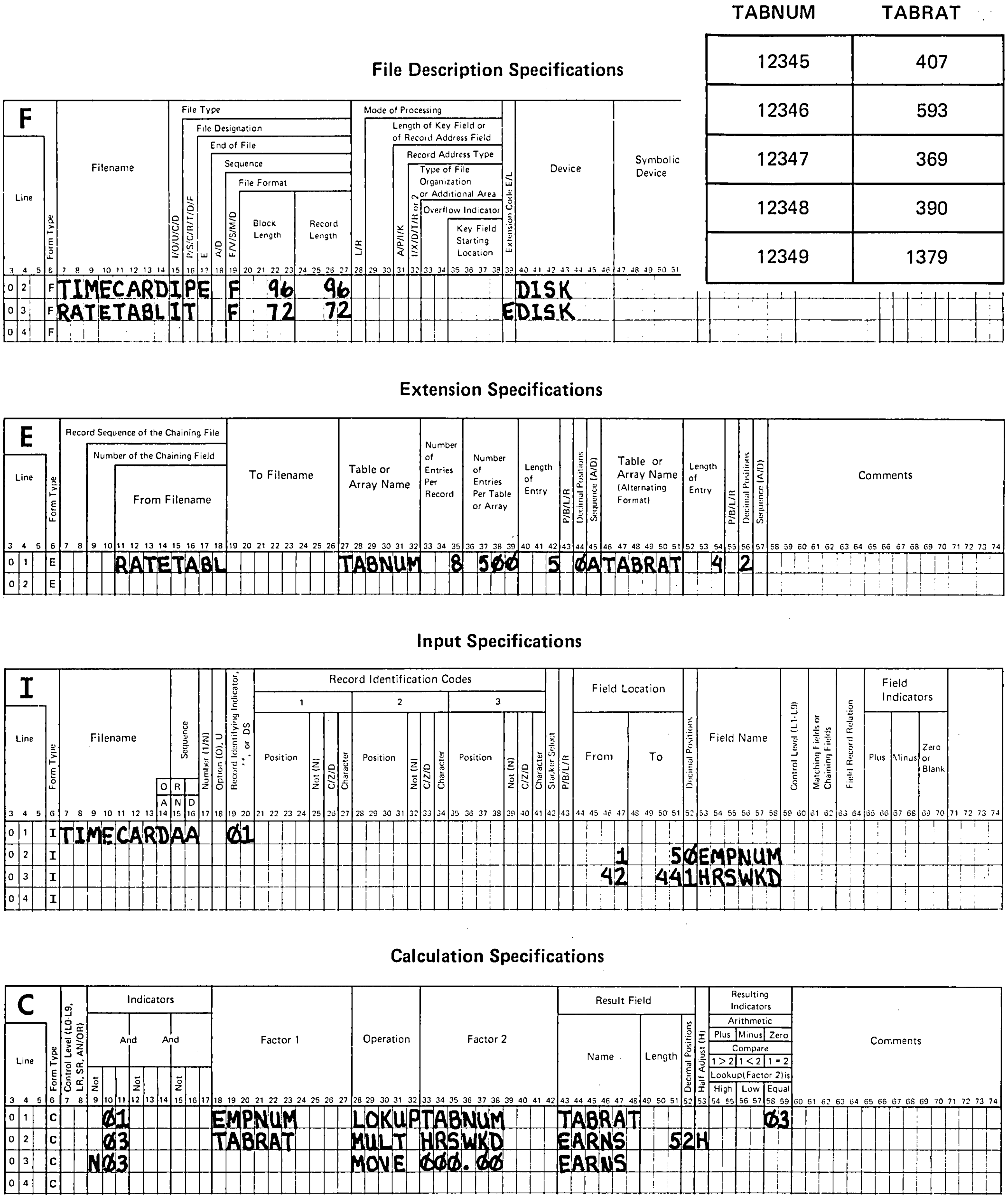
Calculation Specifications

On line 01, factor 1 specifies the search word EMPNUM (employee number). The LOKUP operation code is specified in columns 28 through 32. Factor 2 contains the name of the table to be searched, TABNUM. The result field contains the name of the related table, TABRAT.

The LOKUP operation causes the employee number (EMPNUM) to be used as the search word for the data contained in TABNUM. Indicator 03 turns on when an entry that is equal to the search word is found in the search table.

The operation in line 02 is performed when indicator 03 is on. The rate for the employee, taken from the related table TABRAT, is multiplied by the number of hours worked (HRSWKD). The result is stored in the field EARNS, which is 5 characters long with two decimal positions. The result is half-adjusted.

When an equal entry is not found in TABNUM (indicator 03 is not on), the operation in line 03 is performed. The literal 000.00 is then moved to the field EARNS, specifying that the employee does not have an entry in the table.



TABNUM	TABRAT
12345	407
12346	593
12347	369
12348	390
12349	1379

File Description Specifications

Line	Form Type	Filename	File Type	File Designation	End of File	File Format	Block Length	Record Length	Extension Code E/L	Device
02	F	TIMECARD	I	P	E	F	96	96		DISK
03	F	RATETABL	I	T		F	72	72	E	DISK
04	F									

Extension Specifications

Line	Form Type	From Filename	Table or Array Name	Number of Entries Per Record	Number of Entries Per Table or Array	Length of Entry	Decimal Positions	Sequence (A/D)	Table or Array Name (Alternating Format)	Length of Entry	Decimal Positions
01	E	RATETABL	TABNUM	8	5ØØ	5	Ø	A	TABRAT	4	2
02	E										

Input Specifications

Line	Form Type	Filename	Sequence	Record Identifying Indicator	Field Location From	Field Location To	Decimal Positions	Field Name
01	I	TIMECARD	AA	Ø1				
02	I				1	5	Ø	EMPNUM
03	I				42	44	1	HRSWKD
04	I							

Calculation Specifications

Line	Form Type	Indicators	Factor 1	Operation	Factor 2	Result Field Name	Length	Decimal Positions	Half Adjust (H)	Resulting Indicators Equal
01	C	Ø1	EMPNUM	LOKUP	TABNUM	TABRAT				Ø3
02	C	Ø3	TABRAT	MULT	HRSWKD	EARNS	5	2	H	
03	C	NØ3		MOVE	ØØØ.ØØ	EARNS				
04	C									

Figure 14-10. Related Tables Used in Payroll Job

EXAMPLES OF USING ARRAYS

Building an Array Using Field Indexes

Figure 14-11 illustrates a method of loading an array using fields in input records as indexes. The figure shows a sample 12-element array, with each element 5 characters long. The array could be defined with any number of elements (to a maximum of 99) without additional input specifications. To do this you would assign different values to fields X1 through X10 on each input record type 03 and to fields X1 and X2 on each input record type 04. Succeeding type 03 records can then load 10 additional elements into array AR, up to the maximum defined in the array; each type 04 record can load two additional elements.

Blanks and other fields can appear on the input records because the array elements and their index are identified by the from and to entries.

This method requires a minimum of coding and no calculations to set up the array. Extra work, however, is required to set up the indexing scheme for the input records.

Building an Array Using Fixed Indexes

In Figure 14-12 eighteen 5-character elements of array AR1 are loaded with only two specification lines. On succeeding input specifications, other elements of AR1 are loaded one after another until the array is full. Each additional element is coded on a separate line. Each new record requires a separate means of identification. For example, if another 03 record followed the first, the fields on the second record overlay the fields read in from the first record. This method works well for small arrays.

Calculating Totals with Arrays

The specifications in Figure 14-13 tabulate three levels of totals. As they are read from input records, the fields FIELDA, FIELDB, FIELDC, and FIELDD are added to the first level totals L1A, L1B, L1C, and L1D. These first level totals are added at the time of an L1 control break to totals L2A, L2B, L2C, and L2D. Similarly, at an L2 control break the second level totals are added to third level totals L3A, L3B, L3C, L3D. In addition, as control breaks occur, L1, L2, and L3 total output is performed; and total fields are set to zeros after they are written on the output device.

Figure 14-14 shows the same tabulations being performed on arrays. Note the reduction in coding required to specify the functions. For example, line 5 of the calculation specifications performs the same function as lines 5 through 8 of the calculation specifications shown in Figure 14-13. Similarly, the output specifications are reduced from 15 lines to 6. The method using arrays results in only two positions between array elements.

Using Arrays to Format Field Output

Figure 14-15 illustrates the use of three arrays to format field output. The arrays are defined as follows:

Array Name	Number of Elements	Element Length
ARA	4	5
ARB	5	10
ARC	6	4

Array ARA is contained in the input records with record identifying indicator 01, ARB in the records with record identifying indicator 02, and ARC in both types of records. Array ARC and the element of array ARA are to be included together in an output record as are arrays ARC and an element (identified by field X1) of array ARB. Every element in array ARC is edited according to the edit word '0ƀ.ƀƀ&CR' (where ƀ represents a blank).

The contents of the arrays in the first two input records are:

Record	Array	Array Contents
1	ARA	12345678901234567890
	ARC	012345678901234567898 76N (note that N equals minus 5)
2	ARB	JOHNƀDOEƀƀJOEƀSMITHƀLEEƀMARXƀƀJIMƀKNOTSƀTIMƀTYLERƀ
	ARC	(the same as in record 1)

Extension Specifications

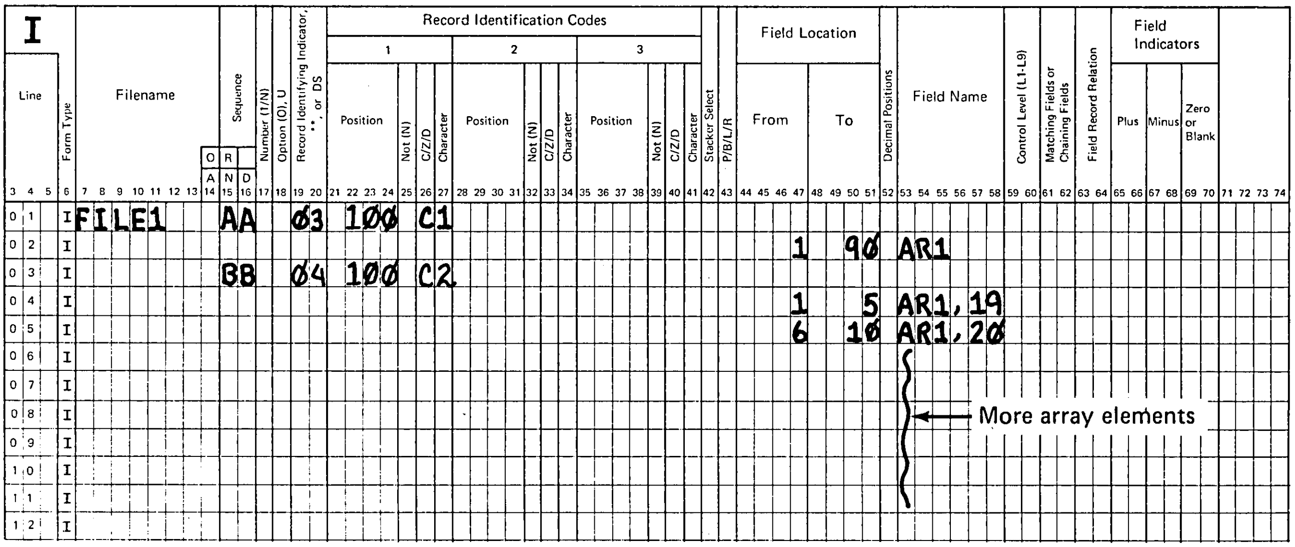
Line (3-5)	Form Type (6)	Record Sequence of the Chaining File (7-8)	Number of the Chaining Field (9-10)	From Filename (11-18)	To Filename (19-26)	Table or Array Name (27-32)	Number of Entries Per Record (33-35)	Number of Entries Per Table or Array (36-39)	Length of Entry (40-42)	P/B/L/R (43)	Decimal Positions (44)	Sequence (A/D) (45)	Table or Array Name (Alternating Format) (46-51)	Length of Entry (52-54)	P/B/L/R (55)	Decimal Positions (56)	Sequence (A/D) (57)	Comments (58-74)
0 1	E					AR		12	5									
0 2	E																	

Input Specifications

Line (3-5)	Form Type (6)	Filename (7-14)	Sequence (15-16)	Number (1/N) (17)	Option (O), U (18)	Record Identifying Indicator, **, or DS (19-20)	Record Identification Codes 1 Position (21-24)	Not (N) (25)	C/Z/D (26)	Character (27)	2 Position (28-31)	Not (N) (32)	C/Z/D (33)	Character (34)	3 Position (35-38)	Not (N) (39)	C/Z/D (40)	Character (41)	Stacker Select (42)	P/B/L/R (43)	Field Location From (44-47)	To (48-51)	Decimal Positions (52)	Field Name (53-58)	Control Level (L1-L9) (59-60)	Matching Fields or Chaining Fields (61-62)	Field Record Relation (63-64)	Field Indicators Plus (65-66)	Minus (67-68)	Zero or Blank (69-70)	(71-74)
0 1	I	FILE1	AA			Ø3	8Ø		C	1																					
0 2	I																				2	3	Ø	X1							
0 3	I																				4	8		AR,X1							
0 4	I																				9	10	Ø	X2							
0 5	I																				11	15		AR,X2							
0 6	I																				16	17	Ø	X3							
0 7	I																				18	22		AR,X3							
0 8	I																														
0 9	I																														
1 0	I																														
1 1	I																				54	55	Ø	X1Ø							
1 2	I																				56	6Ø		AR,X1Ø							
1 3	I		BB			Ø4	8Ø		C	2																					
1 4	I																				2	3	Ø	X1							
1 5	I																				4	8		AR,X1							
1 6	I																				9	10	Ø	X2							
1 7	I																				11	15		AR,X2							
1 8	I																														

Figure 14-11. Building an Array Using Input Fields as Indexes

Input Specifications



Line	Form Type	Filename	Sequence	Number (1/N)	Option (O), U	Record Identifying Indicator, **, or DS	Record Identification Codes 1 Position	Not (N)	C/Z/D	Character	Record Identification Codes 2 Position	Not (N)	C/Z/D	Character	Record Identification Codes 3 Position	Not (N)	C/Z/D	Character	Stacker Select	P/B/L/R	Field Location From	Field Location To	Decimal Positions	Field Name	Control Level (L1-L9)	Matching Fields or Chaining Fields	Field Record Relation	Field Indicators Plus	Minus	Zero or Blank	
01	I	FILE1	AA			Ø3	1ØØ		C	1																					
02	I																				1	9Ø		AR1							
03	I		BB			Ø4	1ØØ		C	2																					
04	I																				1	5		AR1,19							
05	I																				6	1Ø		AR1,2Ø							
06	I																														
07	I																														
08	I																														
09	I																														
10	I																														
11	I																														
12	I																														

More array elements

Figure 14-12. Building an Array Using Fixed Indexes

Calculation Specifications

Line	Form Type	Control Level (L0-L9, LR, SR, AN/OR)	Indicators	Factor 1	Operation	Factor 2	Result Field Name	Length	Decimal Positions	Half Adjust (H)	Resulting Indicators	Comments
01	C			FIELDA	ADD	L1A	L1A	6	2			
02	C			FIELDB	ADD	L1B	L1B	6	2			
03	C			FIELDC	ADD	L1C	L1C	6	2			
04	C			FIELDD	ADD	L1D	L1D	6	2			
05	C	L1		L1A	ADD	L2A	L2A	6	2			
06	C	L1		L1B	ADD	L2B	L2B	6	2			
07	C	L1		L1C	ADD	L2C	L2C	6	2			
08	C	L1		L1D	ADD	L2D	L2D	6	2			
09	C	L2		L2A	ADD	L3A	L3A	6	2			
10	C	L2		L2B	ADD	L3B	L3B	6	2			
11	C	L2		L2C	ADD	L3C	L3C	6	2			
12	C	L2		L2D	ADD	L3D	L3D	6	2			
13	C											

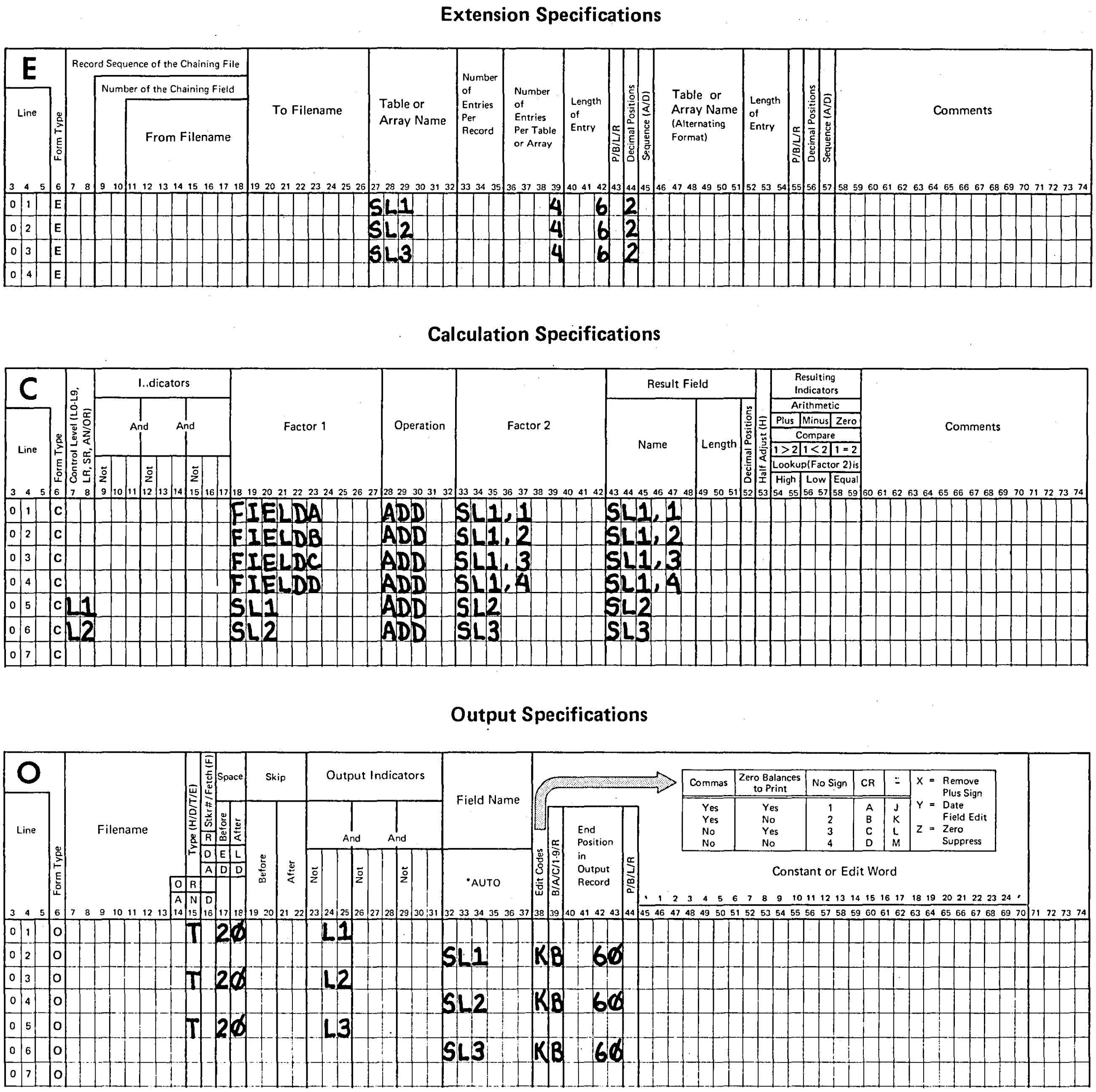
Output Specifications

Commas	Zero Balances to Print	No Sign	CR	-
Yes	Yes	1	A	J
Yes	No	2	B	K
No	Yes	3	C	L
No	No	4	D	M

X = Remove Plus Sign
Y = Date Field Edit
Z = Zero Suppress

Line	Form Type	Filename	Type (H/D/T/E)	Space Before	Space After	Skip Before	Skip After	Output Indicators	Field Name	Edit Codes	B/A/C/1-9/R	End Position in Output Record	P/B/L/R	Constant or Edit Word
01	O		T	2	Ø			L1						
02	O								L1A	K	B	15		
03	O								L1B	K	B	3Ø		
04	O								L1C	K	B	45		
05	O								L1D	K	B	6Ø		
06	O		T	2	Ø			L2						
07	O								L2A	K	B	15		
08	O								L2B	K	B	3Ø		
09	O								L2C	K	B	45		
10	O								L2D	K	B	6Ø		
11	O		T	2	Ø			L3						
12	O								L3A	K	B	15		
13	O								L3B	K	B	3Ø		
14	O								L3C	K	B	45		
15	O								L3D	K	B	6Ø		
16	O													

Figure 14-13. Calculating Totals Without Arrays



Extension Specifications

Line	Form Type	From Filename	To Filename	Table or Array Name	Number of Entries Per Record	Number of Entries Per Table or Array	Length of Entry	P/B/L/R	Decimal Positions	Sequence (A/D)	Table or Array Name (Alternating Format)	Length of Entry	P/B/L/R	Decimal Positions	Sequence (A/D)	Comments
01	E			SL1		4	6		2							
02	E			SL2		4	6		2							
03	E			SL3		4	6		2							
04	E															

Calculation Specifications

Line	Form Type	Control Level (L0-L9, LR, SR, AN/OR)	Indicators	Factor 1	Operation	Factor 2	Result Field Name	Length	Decimal Positions	Half Adjust (H)	Resulting Indicators	Comments
01	C			FIELDA	ADD	SL1,1	SL1,1					
02	C			FIELDB	ADD	SL1,2	SL1,2					
03	C			FIELDC	ADD	SL1,3	SL1,3					
04	C			FIELDD	ADD	SL1,4	SL1,4					
05	C	L1		SL1	ADD	SL2	SL2					
06	C	L2		SL2	ADD	SL3	SL3					
07	C											

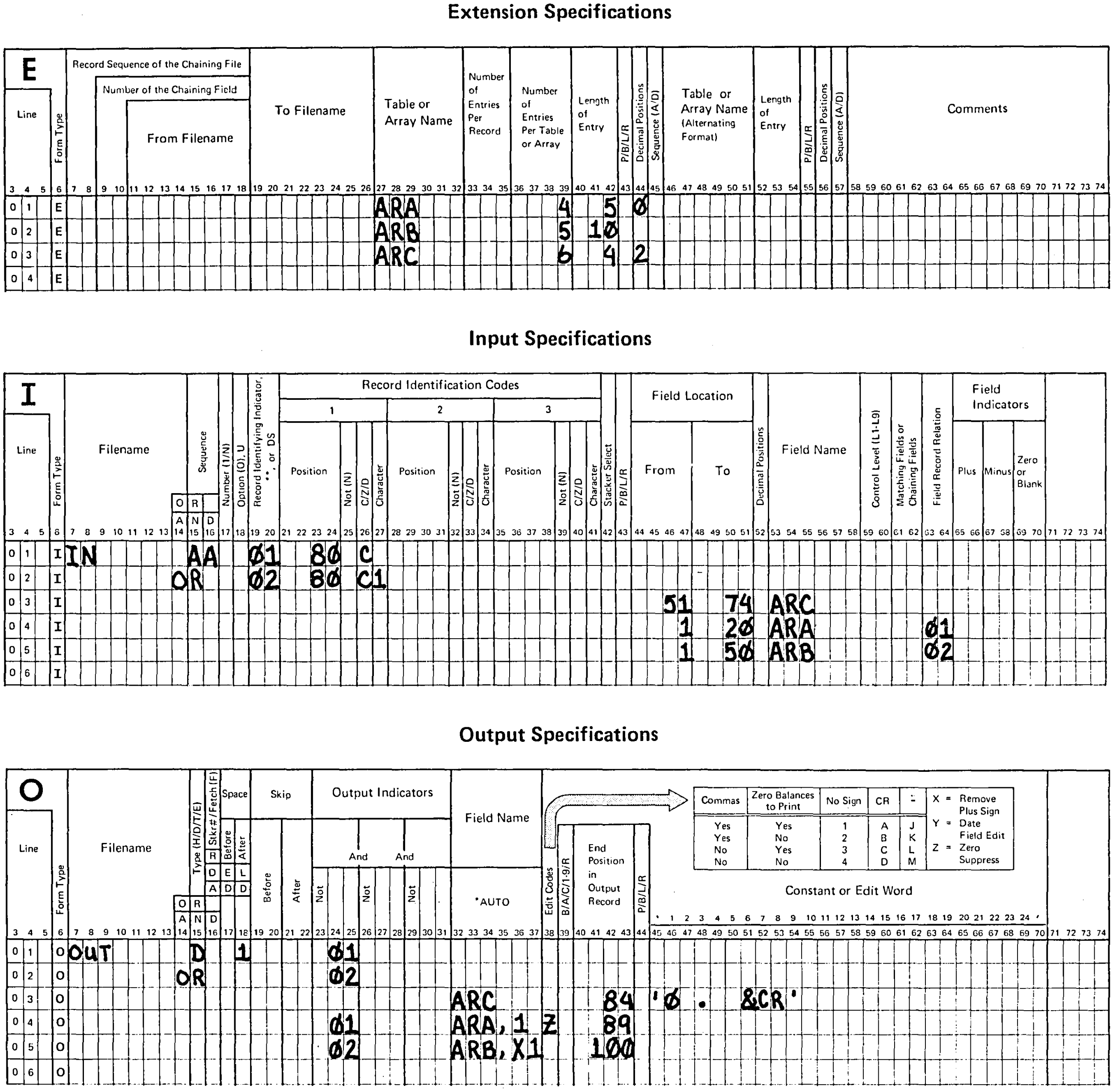
Output Specifications

Line	Form Type	Filename	Type (H/D/T/E)	Stkr #/Fetch (F)	Space Before	Space After	Skip Before	Skip After	Output Indicators	Field Name	Edit Codes	B/A/C/1-9/R	End Position in Output Record	P/B/L/R	Constant or Edit Word
01	O		T		2	Ø			L1						
02	O									SL1	K	B	6Ø		
03	O		T		2	Ø			L2						
04	O									SL2	K	B	6Ø		
05	O		T		2	Ø			L3						
06	O									SL3	K	B	6Ø		
07	O														

Commas	Zero Balances to Print	No Sign	CR	-
Yes	Yes	1	A	J
Yes	No	2	B	K
No	Yes	3	C	L
No	No	4	D	M

X = Remove Plus Sign
Y = Date Field Edit
Z = Zero Suppress

Figure 14-14. Calculating Totals With Arrays



Extension Specifications

```
  01 E                    ARA         4  5 Ø
  02 E                    ARB         5 1Ø
  03 E                    ARC         6  4 2
  04 E
```

Input Specifications

```
  01 IIN      AA  Ø1  8Ø C
  02 I       OR   Ø2  8Ø C1
  03 I                                       51  74 ARC
  04 I                                        1  2Ø ARA       Ø1
  05 I                                        1  5Ø ARB       Ø2
  06 I
```

Output Specifications

```
  01 OOUT     D  1     Ø1
  02 O       OR        Ø2
  03 O                         ARC       84 'Ø .  &CR'
  04 O                 Ø1      ARA,1 Z   89
  05 O                 Ø2      ARB,X1   1ØØ
  06 O
```

Figure 14-15. Using Arrays to Format Field Output

In the first output record, the location and contents of the arrays are (ƀ represents a blank):

Array	Location	Contents
ARA (first element)	85-89	12345
ARC	37-84	ƀ1.23ƀƀƀ45.67ƀƀƀ 89.01ƀƀƀ23.45ƀƀƀ 67.89ƀƀƀ87.65ƀCR

For the second output record assume that the content of field X1 is 4; the locations and contents of the arrays are:

Array	Location	Contents
ARB (fourth element)	91-100	JIMƀKNOTSƀ
ARC	37-84	ƀ1.23ƀƀƀ45.67ƀƀƀ 89.01ƀƀƀ23.45ƀƀƀ 67.89ƀƀƀ87.65ƀCR

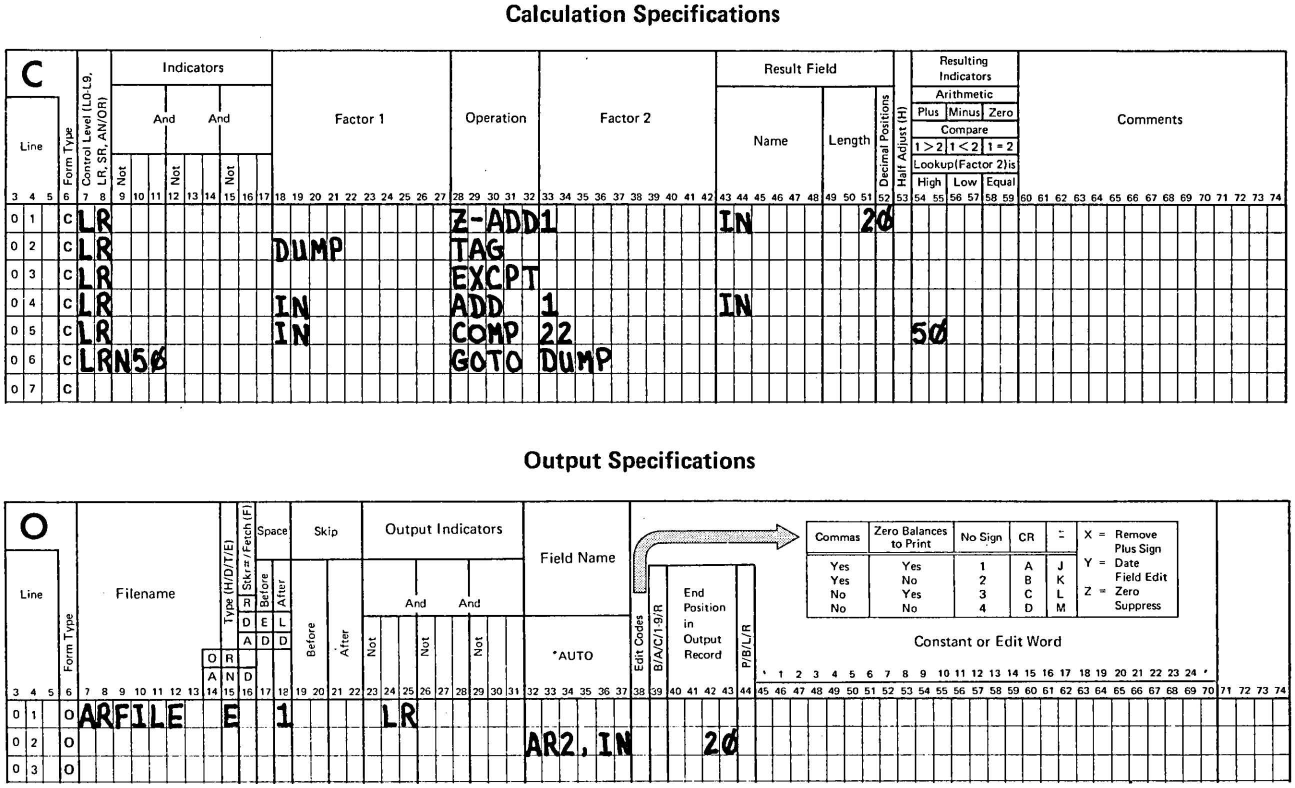
Printing Array Elements

One Element Per Line

Figure 14-16 shows a method of printing one array element per line on the printer output device. The contents of one element of a 22-element array, AR2, are written to the output file ARFILE each time the specification in line 3 of the calculation specifications is performed.

More Than One Element Per Line

Figure 14-17 shows a method of printing more than one array element per line on the printer output device. The number of elements printed on a line depends on the value assigned to the compare on line 10 of the calculation specifications sheet. If an edit code is used, each array element is separated by two spaces. You must take these spaces into consideration when you compute the end position for the output specifications.



Calculation Specifications

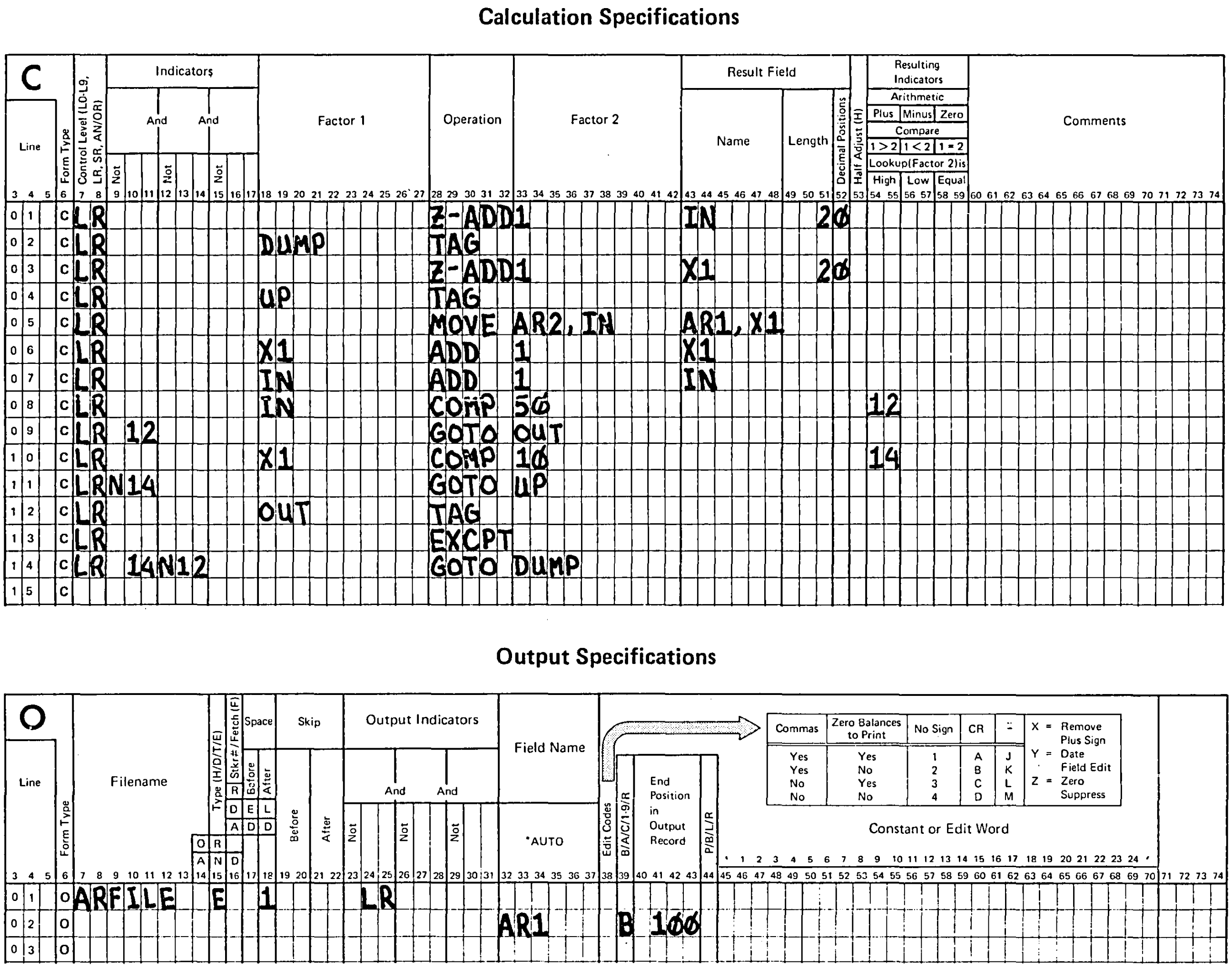
Line	Form Type	Control Level (L0-L9, LR, SR, AN/OR)	Indicators	Factor 1	Operation	Factor 2	Result Field Name	Length	Decimal Positions	Half Adjust (H)	Resulting Indicators High	Low	Equal	Comments
01	C	LR			Z-ADD	1	IN	2	ø					
02	C	LR		DUMP	TAG									
03	C	LR			EXCPT									
04	C	LR		IN	ADD	1	IN							
05	C	LR		IN	COMP	22					5ø			
06	C	LR	N5ø		GOTO	DUMP								
07	C													

Output Specifications

Line	Form Type	Filename	Type (H/D/T/E)	Stacker/Fetch	Space Before	Space After	Skip Before	Skip After	Output Indicators	Field Name	Edit Codes	B/A/C/1-9/R	End Position in Output Record	P/B/L/R	Constant or Edit Word
01	O	ARFILE	E		1				LR						
02	O									AR2,IN			2ø		
03	O														

Commas	Zero Balances to Print	No Sign	CR	–	
Yes	Yes	1	A	J	X = Remove Plus Sign
Yes	No	2	B	K	Y = Date Field Edit
No	Yes	3	C	L	Z = Zero Suppress
No	No	4	D	M	

Figure 14-16. Printing One Array Element Per Line



Calculation Specifications

```
01 CLR                   Z-ADD1         IN      2Ø
02 CLR         DUMP      TAG
03 CLR                   Z-ADD1         X1      2Ø
04 CLR         UP        TAG
05 CLR                   MOVE AR2,IN    AR1,X1
06 CLR         X1        ADD  1         X1
07 CLR         IN        ADD  1         IN
08 CLR         IN        COMP 5Ø                   12
09 CLR 12                GOTO OUT
10 CLR         X1        COMP 1Ø                   14
11 CLRN14                GOTO UP
12 CLR         OUT       TAG
13 CLR                   EXCPT
14 CLR 14N12             GOTO DUMP
15 C
```

Output Specifications

```
01 OARFILE  E  1     LR
02 O                         AR1    B 1ØØ
03 O
```

Figure 14-17. Printing More Than One Array Element Per Line

Chapter 15. Auto Report Function

The RPG II auto report function is a program that operates prior to the RPG II compiler. Auto report accepts special, simplified specifications and standard RPG II source specifications and uses them to generate a complete RPG II source program. Special auto report statements control the three separate functions of auto report, which can be used in any combination:

- *AUTO page headings provide a simplified method of coding page headings.
- *AUTO output provides a simplified method of coding output specifications.
- /COPY statement provides a method of copying cataloged specifications from a library to include them in an RPG II source program.

AUTO REPORT GENERATED SPECIFICATIONS

Auto report generates a complete RPG II source program that is ready to be compiled from the following input:

- Auto renort option specifications
- *AUTO page headings and *AUTO output specifications in the source program
- Standard RPG II specifications in the source program
- Auto report /COPY statements in the source program, with or without modifier statements
- Standard RPG II specifications, including tables and arrays, and *AUTO specifications that are copied from the library by the auto report copy function

Figure 15-1 shows an example of the RPG II specifications that are generated by auto report, and Figure 15-2 shows the general method of operation of the auto report function.

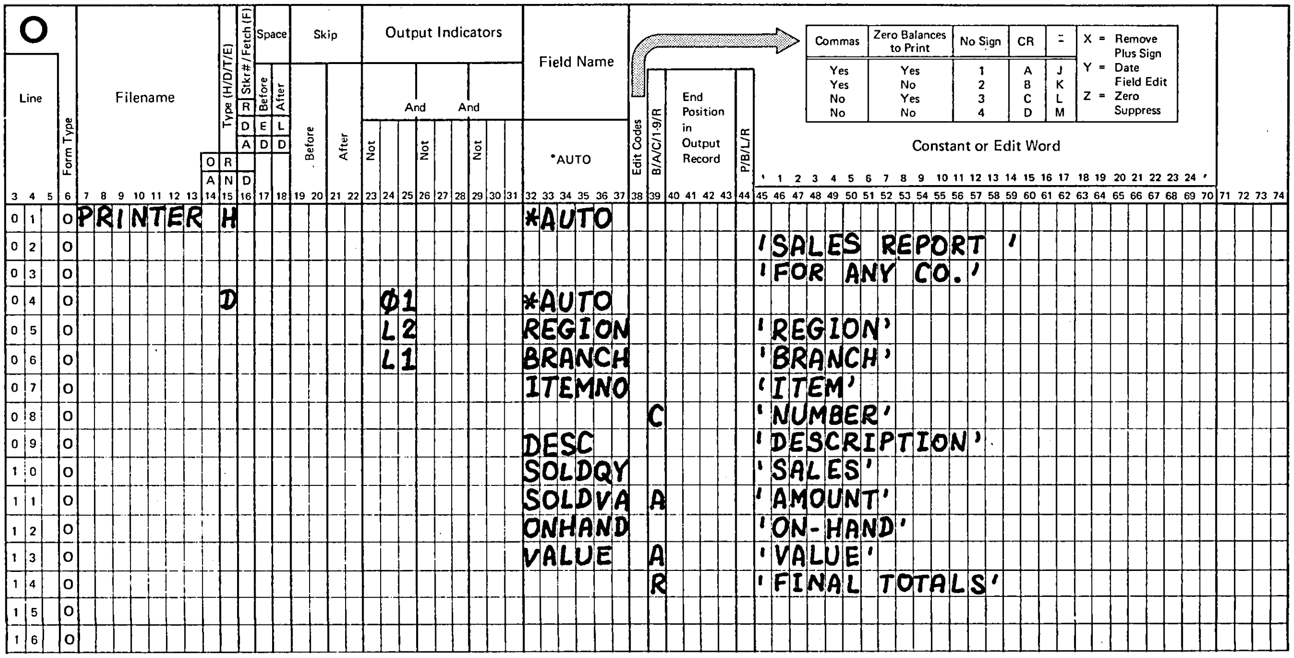
Format of the Generated Specifications

The generated specifications are in the following format:

Column	Contents	
1-4	Sequence number of the specification. This number starts as 0010 on the RPG II control specification and is incremented by 0010 on each specification that follows. If more than 999 specifications are present in the program, the sequence is restarted at 0000.	
5	Code that identifies the specification as follows:	
	Blank	Standard RPG II specification present in the auto report program.
	C	Specification copied from the library.
	M	Specification copied from the library and modified.
	E	Specification generated by auto report.
6-80	Standard RPG II specification.	

Compile-time tables and arrays are not changed by auto report; they remain in standard table/array record format.

Output Specifications



Commas	Zero Balances to Print	No Sign	CR	-
Yes	Yes	1	A	J
Yes	No	2	B	K
No	Yes	3	C	L
No	No	4	D	M

X = Remove Plus Sign
Y = Date Field Edit
Z = Zero Suppress

Line	Form Type	Filename	Type (H/D/T/E)	Stkr#/Fetch (F)	Space Before	Space After	Skip Before	Skip After	Output Indicators	Field Name / *AUTO	Edit Codes	B/A/C/1-9/R	End Position in Output Record	P/B/L/R	Constant or Edit Word
01	O	PRINTER	H							*AUTO					
02	O														'SALES REPORT '
03	O														'FOR ANY CO.'
04	O		D						Ø1	*AUTO					
05	O								L2	REGION					'REGION'
06	O								L1	BRANCH					'BRANCH'
07	O									ITEMNO					'ITEM'
08	O											C			'NUMBER'
09	O									DESC					'DESCRIPTION'
10	O									SOLDQY					'SALES'
11	O									SOLDVA		A			'AMOUNT'
12	O									ONHAND					'ON-HAND'
13	O									VALUE		A			'VALUE'
14	O											R			'FINAL TOTALS'
15	O														
16	O														

From the *AUTO specifications in this example, auto report generates the standard RPG II calculation and output specifications shown in Part 2 of this figure.

Figure 15-1 (Part 1 of 2). Using *AUTO Specifications, Auto Report Generates Standard RPG II Specifications

Generated Specifications

```
0012  0140EC   01               EXSR A$$SUM
0013  0150ECL1         SOLDV2   ADD  SOLDV1   SOLDV2  92
0014  0160ECL1         VALUE2   ADD  VALUE1   VALUE2  92
0015  0170ECL2         SOLDVR   ADD  SOLDV2   SOLDVR  92
0016  0180ECL2         VALUER   ADD  VALUE2   VALUER  92
0017  0190ECSR         A$$SUM   BEGSR
0018  0200ECSR         SOLDV1   ADD  SOLDVA   SOLDV1  92
0019  0210ECSR         VALUE1   ADD  VALUE    VALUE1  92
0020  0220ECSR                  ENDSR
```

Calculations to roll totals for SOLDVA and VALUE fields

```
0021  0230EOPRINTER H 206    1P
0022  0240EO       OR         OA
0023  0250EO                                 45 'SALES REPORT '
0024  0260EO                                 56 'FOR ANY CO.'
0025  0270EO                        UDATE Y   8
0026  0280EO                        PAGE  Z  89
0027  0290EO                                 85 'PAGE '
```

Page heading (includes date and page number)

```
0028  0300EOPRINTER H 1      1P
0029  0310EO       OR         OA
0030  0320EO                                  6 'REGION'
0031  0330EO                                 14 'BRANCH'
0032  0340EO                                 21 'ITEM'
0033  0350EO                                 36 'DESCRIPTION'
0034  0360EO                                 47 'SALES'
0035  0370EO                                 62 'AMOUNT'
0036  0380EO                                 71 'ON-HAND'
0037  0390EO                                 86 'VALUE'
0038  0400EOPRINTER H 2      1P
0039  0410EO       OR         OA
0040  0420EO                                 22 'NUMBER'
0041  0430EOPRINTER D 1      01
```

Column headings

```
0042  0440EO                 L2     REGION    3

0043  0450EO                 L1     BRANCH   12
0044  0460EO                        ITEMNO   23
0045  0470EO                        DESC     40
0046  0480EO                        SOLDQYK  46
0047  0490EO                        SOLDVAKB 62
0048  0500EO                        ONHANDK  69
0049  0510EO                        VALUE KB 86
```

Detail output specifications

```
0050  0520EOPRINTER T 12     L1
0051  0530EO                        SOLDV1KB 62
0052  0540EO                        VALUE1KB 86
0053  0550EO                                 87 '*'
0054  0560EOPRINTER T 2      L2
0055  0570EO                        SOLDV2KD 62
0056  0580EO                        VALUE2KB 86
0057  0590EO                                 88 '**'
0058  0600EOPRINTER T 12     LR
0059  0610EO                        SOLDVRKB 62
0060  0620EO                        VALUERKB 86
0061  0630EO                                 47 'FINAL TOTALS'
0062  0640EO                                 89 '***'
```

Total output specifications

Figure 15-1 (Part 2 of 2). Using *AUTO Specifications, Auto Report Generates Standard RPG II Specifications

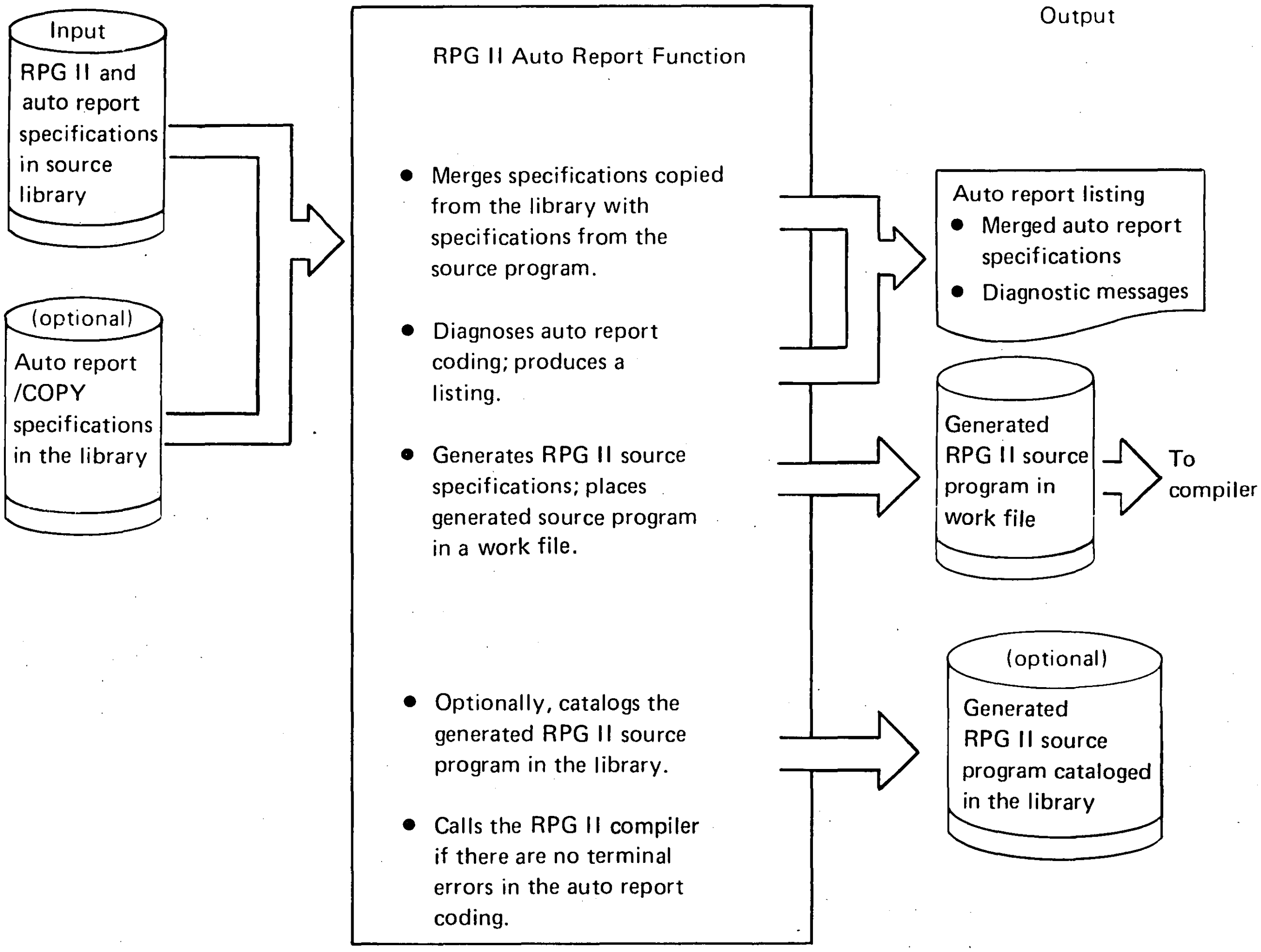

Input
RPG II and auto report specifications in source library
(optional)
Auto report /COPY specifications in the library
RPG II Auto Report Function
● Merges specifications copied from the library with specifications from the source program.
● Diagnoses auto report coding; produces a listing.
● Generates RPG II source specifications; places generated source program in a work file.
● Optionally, catalogs the generated RPG II source program in the library.
● Calls the RPG II compiler if there are no terminal errors in the auto report coding.
Output
Auto report listing
● Merged auto report specifications
● Diagnostic messages
Generated RPG II source program in work file
To compiler
(optional)
Generated RPG II source program cataloged in the library

Figure 15-2. Operations of the Auto Report Function

Order of Generated Specifications

Auto report generates the specifications in the order required by the RPG II compiler. When specifications are included by means of a /COPY statement, those specifications are initially placed immediately after the /COPY statement. After all specifications are copied and before auto report generates RPG II specifications from the H-*AUTO and D/T-*AUTO specifications, the entire auto report source program is sorted into the following order:

1. Control specifications
2. File description specifications
3. Extension specifications
4. Line counter specifications
5. Telecommunications specifications
6. Input specifications
7. Calculation specifications (in the order: detail, L0, L1 through L9, LR, and subroutines)
8. Output specifications
9. Tables and arrays loaded at compilation time, which must be placed last among the input statements to auto report

Calculation Specifications

Generated RPG II calculation specifications are placed in the following order by auto report:

1. Detail calculations specified by the programmer
2. EXSR statement for the generated subroutine
3. Total calculations generated by auto report, grouped in order by level (all L0 calculations, then all L1 calculations, and so on)
4. Total calculations specified by the programmer
5. Subroutines specified by the programmer
6. Generated RPG II subroutine that accumulates the lowest level total

Output Specifications

Output heading specifications generated for H-*AUTO specifications appear in the same order they are coded on the output specifications in relation to other RPG II and *AUTO output specifications for the file.

Normally, RPG II output specifications generated from a D/T-*AUTO specification are in the following order:

1. Heading specifications generated for column headings
2. Detail specifications
3. Total specifications, with the lowest level first and LR last

This group of specifications is placed in the same relative position in the program as the original D/T-*AUTO specification. All other RPG II output specifications remain in their original order.

If, however, the programmer specifies a normal RPG II total output specification conditioned by a positive control level indicator (no N in column 23) in columns 24 and 25 for the file that has a D/T-*AUTO specification, all output specifications in the program are sorted into the following format:

1. All heading, detail, and exception output specifications remain in the same order as they are in the generated RPG II source program. Total specifications that are not conditioned by a positive control level indicator in columns 24 and 25 remain as they were in the program.
2. Total specifications, which are conditioned by a positive control level indicator in columns 24 and 25, are sorted into ascending order according to the control level indicator in columns 24 and 25, with LR last.

See *Examples of Using Auto Report* in this chapter and *Sample Auto Report Program (EXAUT2)* in Chapter 19 for examples of generated specifications.

Comment Statements

Comment statements (identified by an asterisk in column 7) are allowed among the statements read by auto report. However, because the sorting of RPG II specifications is based on the contents of column 6, comments may not occur in the expected order. To ensure that comments remain with the correct specification, place them after that specification and put the same entry in column 6.

Restriction

The order of tables and arrays is not altered when the source specifications are sorted. Therefore, when tables and arrays are included from the library, they may not occur in the correct order after the sort. For example, if a file translation or alternate collating sequence table is present in the auto report source specifications, then any compile-time tables or arrays included from a library member are out of order. That is, the included tables or arrays are placed ahead of the file translation table. Compile-time tables and arrays must be loaded in the following order:

1. File translation specifications
2. Alternate collating sequence specifications
3. Compile-time tables and arrays in the order described on the extension specifications sheet

A solution to this restriction is to place the file translation and alternate collating sequence tables in the library and copy them from the library before any other compile-time tables and arrays are copied. This procedure ensures that the file translation and alternate collating sequence tables are the first compile-time tables in the generated RPG II source program.

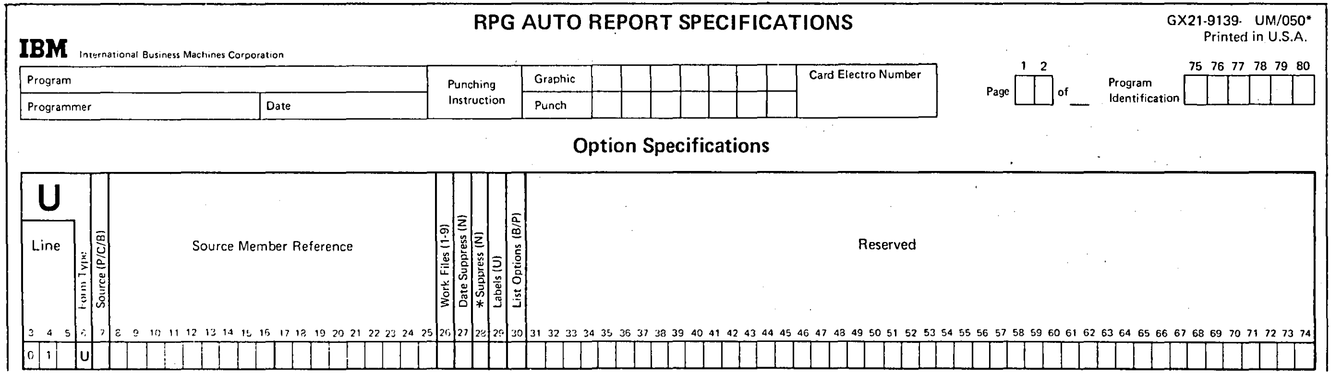
OPTION SPECIFICATIONS

Specify options for the auto report program on the RPG Auto Report Specifications sheet (see Figure 15-3). The auto report option specifications are not required in the auto report program. If present, they must appear as the first specifications in the program. If they are not present, auto report assumes the options that correspond to blank entries (see individual entries for the meanings of the blank entries). Option specifications cannot be contained in a library member that is copied by a /COPY statement, but they can be cataloged with statements that are compiled when you use the COMPILE OCL statement.

If RPG II control specifications (H in column 6) are not present either in the auto report source program or in a copied library member (see */COPY Statement Specifications*), auto report generates control specifications with blank entries. See Chapter 2 for the meaning of blank entries for the control specifications.

The following columns on the auto report specifications sheet are used in the same way as corresponding columns on standard RPG specifications sheets. See *Common Entries* in Chapter 1 for descriptions of these columns:

- Columns 1-2 (page)
- Columns 3-5 (line)
- Columns 75-80 (program identification)



RPG AUTO REPORT SPECIFICATIONS

GX21-9139- UM/050*
Printed in U.S.A.

IBM International Business Machines Corporation

Program		Punching Instruction	Graphic
Programmer	Date		Punch

Card Electro Number

Page 1 2 of ___

Program Identification 75 76 77 78 79 80

Option Specifications

U Line	Form Type	Source (P/C/B)	Source Member Reference	Work Files (1-9)	Date Suppress (N)	*Suppress (N)	Labels (U)	List Options (B/P)	Reserved
3 4 5	6	7	8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25	26	27	28	29	30	31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74
0 1	U								

Figure 15-3. RPG Auto Report Specifications

Column 6 (Form Type)

Enter a U in column 6 to identify this line as an auto report option specification.

Column 7 (Source)

Entry	Explanation
Blank	The generated source program is not cataloged.
C	The generated source program is cataloged in a library on disk.

Use column 7 to specify whether the generated source program is to be cataloged in a library. Whether or not the source program is cataloged, the generated source program is written to a disk work file from which it is immediately compiled.

Generated source programs that are cataloged become permanent library members (RETAIN-P). These members can be deleted only by the DELETE function of the library maintenance utility program ($MAINT) or by the SSP REMOVE procedure. See the *System Support Reference Manual* for information on the $MAINT DELETE function and on the REMOVE procedure. A library member cataloged by auto report, however, is replaced by any other library member cataloged under the same name.

The generated source program is not cataloged when terminal errors exist in the auto report specifications.

Columns 8-24 (Source Member Reference)

Entry	Explanation
library, member	Identifies the library member to be cataloged. Specify the library name, which can be up to 8 characters long, beginning in column 8. Use a comma to separate the library name and the member name, which can also be up to 8 characters long.

Make an entry in columns 8 through 24 if the generated source program is to be cataloged in a library (C in column 7). The first character of the library name and the member name must be alphabetic (any of the letters A through Z or one of the 3 special characters #, $, or @). The remaining characters can be alphabetic or numeric.

If F1 is entered for the library name or the library name is not specified, the library name defaults to the member name in the system library. If the member name is not specified or is specified incorrectly, an error results.

If the name used by auto report to catalog the generated source program is the same as the name of an existing permanent member in the library, the old member is replaced by the new member.

Columns 25-26

Columns 25 and 26 are not used. Leave them blank.

Column 27 (Date Suppress)

Entry	Explanation
Blank	Page number and date are included on the first *AUTO page heading line.
N	Date and page number on the first *AUTO page heading line are suppressed.

To suppress the generated date and page number from printing on the first *AUTO heading line, enter an N in column 27. When these fields are suppressed, the page title and any other fields specified can occupy the entire line. See *AUTO Page Heading Specifications* for further information on the generated date and page numbers.

*Column 28 (*Suppress)*

Entry	Explanation
Blank	Asterisks are generated for total output lines.
N	Asterisk indication is suppressed from generated total output lines.

To suppress asterisks from printing beside generated totals, enter an N in column 28. See **AUTO Output Specifications* for rules used in generating asterisk indication.

Column 29

Column 29 is not used. Leave it blank.

Column 30 (List Options)

Entry	Explanation
Blank	Source program listing, headings, and diagnostics are printed, and a source program is produced if no severe errors are found.
B	The program listing is not printed; however, a source program is produced.
P	A partial program listing is printed, which includes appropriate headings and diagnostics.

Column 30 provides for listing options at the time the auto report function is generating RPG II source specifications. If any severe errors in auto report specifications are found, the listing is completed (provided a listing is to be printed) and the system halts.

The auto report source listing consists of the RPG II specifications included in the input to auto report, RPG II specifications generated by auto report, and specifications copied from the library.

Use the B entry to produce a source program for which you already have a listing, and use the P entry to determine whether minor modifications to a previously tested program have generated any errors.

Columns 31-74

Columns 31 through 74 are not used. Leave them blank.

*AUTO SPECIFICATIONS

The *AUTO page heading function and the *AUTO output function provide simplified methods of describing printed output. These functions of auto report are requested when the characters *AUTO are present in columns 32 through 36 of a record description specification on the standard RPG output specifications sheet. *AUTO can be entered on a heading, detail, or total specification (H, D, or T in column 15), but not on an exception output specification (E in column 15). Use *AUTO with only one printer file in the program.

Standard RPG II output specifications are divided into two general types (see Figure 15-4).

- *Record description specifications* (columns 7 through 31) describe when and where the output line is to be printed. One record description specification is required for each different type of line to be printed. Only the first record description for a file need contain a filename in columns 7 through 14.
- *Field description specifications* (columns 23 through 74) following a record description specification tell when, where, and how each item of data (field or literal) is to be printed on the output record. There can be several field description specifications following a record description specification.

Auto report page headings and auto report output specifications are also divided into the two general categories: record description specifications and field description specifications. However, the use of entries on these specifications is different from that of entries for standard RPG II specifications.

The following output specifications are not changed when they are used with *AUTO. See *Common Entries* in Chapter 1 for descriptions of these entries:

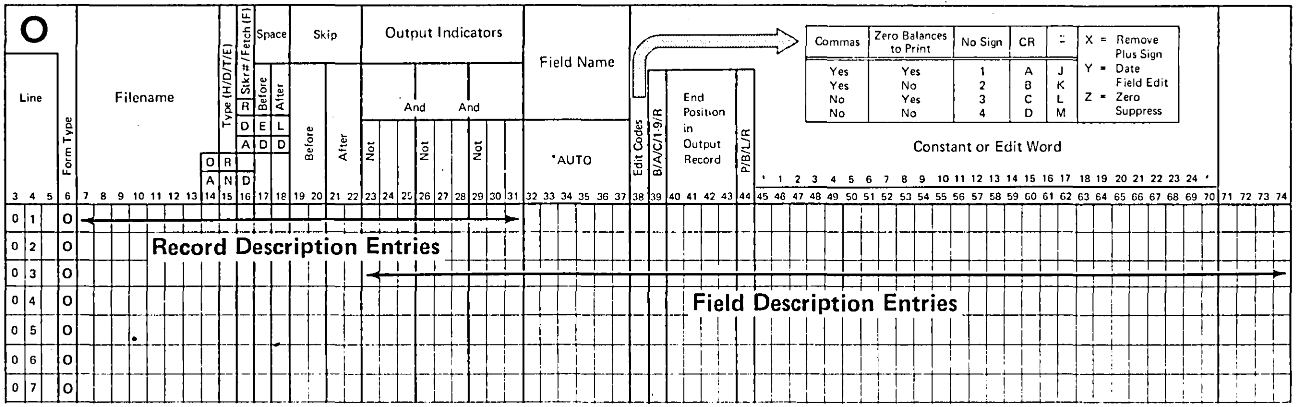
- Columns 1-2 (page)
- Columns 3-5 (line)
- Column 6 (form type)
- Columns 75-80 (program identification)

Columns 71 through 74 must always be blank on auto report output specifications.

*AUTO PAGE HEADING SPECIFICATIONS

The *AUTO page heading specifications provide an easy way to produce a page heading at the top of every page of a printed report (see Figure 15-5). Up to five *AUTO page heading specifications can be used for a multiple-line page heading. If both standard RPG II heading lines and *AUTO page headings are specified in combination for a file, they are printed in the order specified by the output specifications. The *AUTO page headings can be specified for only one printer file per program.

Output Specifications

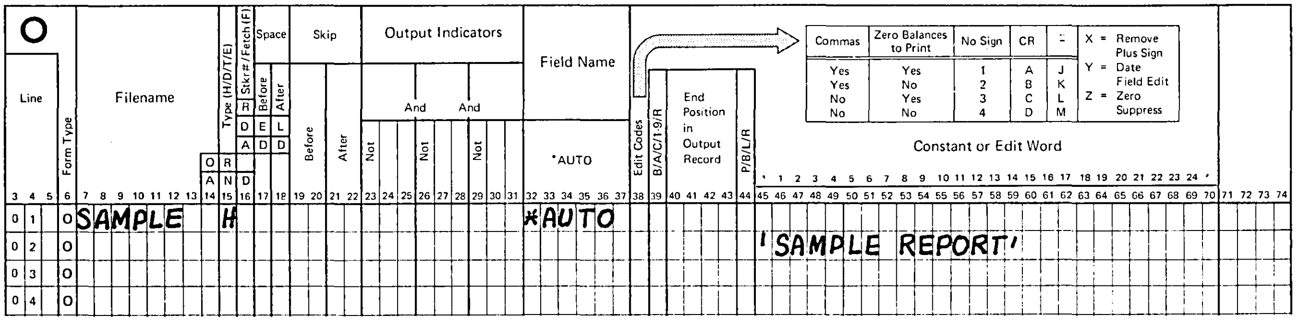

O
Line
Form Type
Filename
Type (H/D/T/E)
Stkr #/Fetch (F)
Space
Before
After
Skip
Before
After
Output Indicators
And
And
Not
Not
Not
Field Name
*AUTO
Edit Codes
B/A/C/1-9/R
End Position in Output Record
P/B/L/R
Constant or Edit Word
Commas
Zero Balances to Print
No Sign
CR
X = Remove Plus Sign
Y = Date Field Edit
Z = Zero Suppress

Commas	Zero Balances to Print	No Sign	CR	-
Yes	Yes	1	A	J
Yes	No	2	B	K
No	Yes	3	C	L
No	No	4	D	M

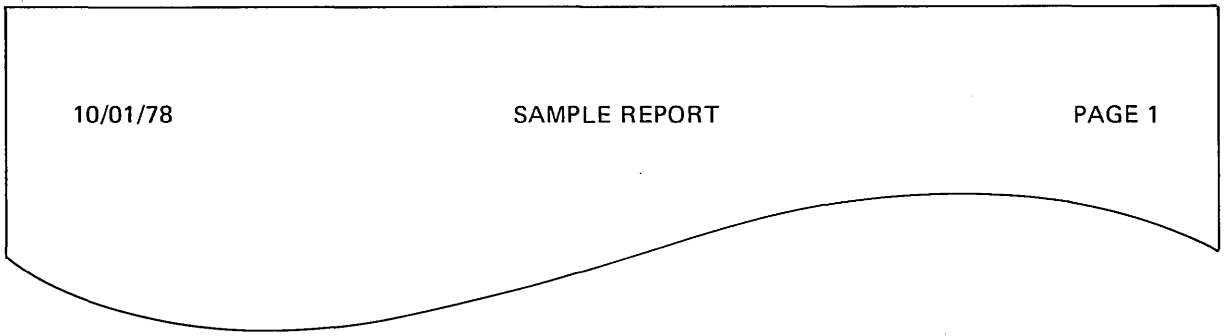
0 1 O ← →
0 2 O Record Description Entries
0 3 O ← →
0 4 O Field Description Entries
0 5 O
0 6 O
0 7 O

Figure 15-4. Two Categories of Output Specifications

Output Specifications



0 1 O SAMPLE H *AUTO
0 2 O 'SAMPLE REPORT'
0 3 O
0 4 O



10/01/78 SAMPLE REPORT PAGE 1

Figure 15-5. Specifications and Results—*AUTO Heading Line

The heading line generated by the first *AUTO page heading (H-*AUTO) specification contains a date and page number. The first heading line can also contain a title. (See *Field Description Specifications* in this section for information on entering a title.) The generated date is left-justified and prints with slashes as mm/dd/yy unless the format is altered by the RPG II date or inverted print option (columns 19 through 21 of the control specifications). The generated page number is right-justified and is preceded by the word PAGE. The page number field is four digits long and is zero suppressed. Auto report uses one of the unused PAGE fields (PAGE, PAGE1 through PAGE7) for page numbering. If all PAGE fields are used in the program, auto report does not number pages. To suppress the date and page number on the first heading line, enter an N in column 27 of the auto report option specifications.

Record Description Specifications

Each H-*AUTO record description defines a separate heading line. The record description entries allow the programmer to specify spacing and skipping and the conditions under which the line is printed.

Columns 7-14 (Filename)

Enter the valid filename of the printer file on which the heading is to be printed.

Column 15 (Type)

Enter an H in column 15 on each record description specification line that defines a page heading line. The H and the entry *AUTO in columns 32 through 36 define this as an H-*AUTO heading specification (see Figure 15-5). Up to five H-*AUTO specifications are allowed.

Column 16

Column 16 is not used. Leave it blank.

Columns 17-22 (Spacing and Skipping)

Enter spacing and skipping values in these columns according to the rules given under *Columns 17-22 (Spacing and Skipping)* in Chapter 9. If these columns do not contain spacing and skipping values, auto report skips to line 06 before the first line is printed and spaces two after the last H-*AUTO line is printed. If multiple H-*AUTO lines are used, auto report spaces one after each line except the last. For additional information on generated spacing and skipping values, see *Report Format* in this chapter.

Columns 23-31 (Output Indicators)

On the first H-*AUTO specification, either leave columns 23 through 31 blank or enter output indicators according to the rules given under *Columns 23-31 (Output Indicators)* in Chapter 9.

If these columns are blank, auto report causes the corresponding output line to be printed at first page (1P) time in the program cycle and when overflow occurs. Thus, the heading is printed at the top of each page of the printed report. Indicators can be assigned to subsequent H-*AUTO specifications. If columns 23 through 31 are blank on any H-*AUTO specification after the first, that specification is assigned the same indicators as the first.

If an overflow indicator is specified on the file description specifications for the printer file, that indicator conditions the generated heading specifications. Otherwise, auto report defines an unused overflow indicator for the printer file and conditions the line with that indicator.

AND and OR lines can be used with H-*AUTO output indicators if an output indicator is used with the first specification. Standard RPG II rules for AND and OR lines apply.

*Columns 32-37 (*AUTO)*

Enter *AUTO in columns 32 through 36. This entry and an H in column 15 of the output specifications (see Figure 15-5) indicate that this is an auto report heading line.

Columns 38-70

Columns 38 through 70 are not used on the record description line. Leave them blank.

Field Description Specifications

Each H-*AUTO record description specification can be followed by one or more field description specifications. The field description specifications specify the title to be printed on the heading line and describe any other fields and literals to be printed on the line.

Columns 7-31

Columns 7 through 31 are not used on field description specifications. Leave them blank. Output indicators in columns 23 through 31 cannot be used to condition a field on an H-*AUTO specification.

Columns 32-37 (Field Name)

Entry	Explanation
Blank	A constant (enclosed in apostrophes) must be entered in columns 45 through 70. The constant is printed on the heading line.
Field name	Field defined in the program is printed on the heading line.
Table name	A table element is printed on the heading line.
Indexed array name	An array element is printed on the heading line.

Use columns 32 through 37 to enter a field name, table name, or indexed array name defined elsewhere in the program that is to print on the heading line. If a name is entered, an edit word, not a constant, can be entered in columns 45 through 70. A constant must be entered in columns 45 through 70 if columns 32 through 37 are blank.

If output indicators (columns 23 through 31) are left blank on the record description specification, auto report conditions all fields and table/array elements included on the heading line with N1P in columns 23 through 25. Therefore, the field or table/array element does not print on the first page. (If printed on the first page, the field may not contain meaningful data because the first record is not read.) N1P is not generated for the following RPG II reserved words: PAGE, PAGE1 through PAGE7, UDATE, UDAY, UMONTH, UYEAR.

For information on formatting and centering *AUTO heading lines, see *Report Format* in this chapter.

Column 38 (Edit Codes)

An edit code can be entered in column 38 if a numeric field, numeric array element, or numeric element is named in columns 32 through 37. If an edit code is used, columns 45 through 70 must be blank unless asterisk protection or a floating currency symbol is specified. If column 38 is blank, no editing is done by auto report unless an edit word is used.

Column 39 (Blank After)

Enter a B in column 39 to reset a numeric field to zeros after it is printed or to reset an alphameric field to blanks after it is printed on the heading line.

Columns 40-44

Columns 40 through 44 are not used with *AUTO heading specifications. Leave them blank. For information on the positioning of fields and literals in the title line and centering of heading lines in relation to the body of the report, see *Report Format* in this chapter.

Columns 45-70 (Constant or Edit Word)

Entry	Explanation
Blank	Columns 32 through 37 contain the name of a field that either is not edited or is edited by an edit code.
Constant	Title or other constant (enclosed in apostrophes) that is to appear on the printed line.
Edit word	The edit pattern used to edit the numeric field named in columns 32 through 37 of the same field description line.

Use columns 45 through 70 to specify the title and other information that is to appear on the output line and to edit numeric fields that are to appear on the line. Rules for specifying constants and edit words are identical to those given under *Columns 45-70 (Constant or Edit Word)* in Chapter 9, except that no end positions can be specified.

For information on the positioning of fields and constants in the title line and centering of heading lines in relation to the body of the report, see *Report Format* in this chapter.

*AUTO OUTPUT SPECIFICATIONS

Detail reports (where a line is printed for each individual record that is read) and group printed reports (where only totals are printed) can be specified by the *AUTO output function alone or in combination with standard RPG II specifications. The *AUTO output function generates totals and formats columns and column headings.

A single detail or total *AUTO record description (D/T-*AUTO) specification and its associated field description specifications can specify:

- Up to three lines of column headings to appear above a field
- Accumulation of several levels of totals, including a final total (known as *total rolling*)
- Generation by auto report of end positions for column headings and fields
- Generation by auto report of the K edit code for numeric fields
- Fields or constants to be printed next to generated totals

Four types of description specifications can be associated with the *AUTO record description specification. The four types are distinguished by entries in column 39. The remaining entries on a field description specification have different meanings depending on the entry in column 39.

The valid entries in column 39 of the field description specifications and their meanings are:

- *Blank or B:* Indicates the associated field or constant appears on the detail line.
- *A:* Indicates the associated numeric field is printed on the detail line and accumulated. A total is printed for each control level defined in columns 59 and 60 of the input specifications for the program. A final total is also printed (when the LR indicator is on).
- *C:* Indicates the associated constant is printed on the second or third line of column headings.
- *1, 2, 3, 4, 5, 6, 7, 8, 9, R:* Indicates the associated field or constant appears on the total line generated for the respective control level indicator (L1 through L9, LR).

See *Group Printing* in this chapter for the effect of these entries in a group printed report.

See *Examples of Using Auto Report* in this chapter for examples of the four types of field descriptions.

Record Description Specifications

An auto report record description specification must contain the entry *AUTO in columns 32 through 36. *AUTO can appear only on a record description specification. This entry indicates that the record description and the following field descriptions are redefined according to their use by auto report.

Columns 7-14 (Filename)

Enter the valid filename of the printer file on which the report is to be printed. This must be the same file named on H-*AUTO specifications, if any.

Column 15 (Type)

Entry	Explanation
D	The auto report specifications describe a report containing detail lines.
T	The auto report specifications describe a report containing total lines, but no detail lines (group printed report).

Enter a D in column 15 and *AUTO in columns 32 through 36 for auto report to generate a report that contains detail lines. The field description specifications associated with the D-*AUTO record description specify:

- Fields to appear on the detail line
- Column headings
- Total rolling
- Constants to appear on total lines

See *Examples of Using Auto Report* in this chapter for examples of D-*AUTO specifications.

Enter a T in column 15 and *AUTO in columns 32 through 36 for auto report to generate a group printed report (see *Group Printing* in this chapter).

Only one detail or one total *AUTO (D/T-*AUTO) record description specification can be used in a program.

Column 16 (Fetch Overflow)

Enter an F in column 16 to specify fetch overflow. See *Column 16 (Fetch Overflow)* in Chapter 9 for the rules on using fetch overflow.

When used with the *AUTO output function, fetch overflow applies only to the detail line. If group printing is specified (T in column 15), fetch overflow applies to the lowest level total line to be printed.

Columns 17-22 (Spacing and Skipping)

Enter spacing and skipping values in columns 17 through 22 according to the standard RPG II rules. Entries specified apply only to the detail line generated by a D-*AUTO specification or to the first total line generated by a T-*AUTO specification.

Leave columns 17 through 22 blank to single space after each detail line printed or, if group printing is specified, after the first total line printed. For information on spacing and skipping for generated column heading and total lines, see *Report Format* in this chapter.

Columns 23-31 (Output Indicators)

Enter any valid output indicators in columns 23 through 31 to condition the detail or group print line generated by this *AUTO specification. If these columns are left blank on a D-*AUTO specification, the generated detail line is conditioned by N1P. Therefore, it is not printed at first page (1P) time in the RPG II program cycle. If these columns are left blank for a T-*AUTO specification, the first generated total line is conditioned by the lowest control level indicator defined in the program. (See *Group Printing* for additional information about the use of this entry with a T-*AUTO specification.)

AND and OR can be used with *AUTO output indicators if an output indicator is specified on the first record description specification. Standard RPG II rules for AND and OR lines apply.

Indicators specified in columns 23 through 31 of the record description specification (and its associated AND/OR lines) apply only to the detail line generated by a D-*AUTO specification or the group print line (lowest level total specification) generated by a T-*AUTO specification.

It column headings are specified in the field description specifications that follow this *AUTO record description, they are conditioned by one of the following:

- The same indicators that are specified for the first H-*AUTO specification.
- The first page (1P) indicator in an OR relationship with the overflow indicator specified for the file on the file description specifications. If no overflow indicator is specified, auto report defines an unused overflow indicator and uses it to condition the lines.

Restriction: If N1P is specified on a D-*AUTO record description specification that is followed by field description specifications for totaling fields (A in column 39), the calculations generated for the totaling fields are also conditioned by N1P. This causes a terminal diagnostic in the RPG II compiler.

*Columns 32-37 (*AUTO)*

To indicate that this is an auto report specification, enter *AUTO in columns 32 through 36 on the record description line. Column 15 must contain a D or a T to indicate a detail or total *AUTO specification. Only one D/T-*AUTO specification can be used in a program.

Columns 38-70

Columns 38 through 70 are not used on a D/T-*AUTO record description specification. Leave them blank.

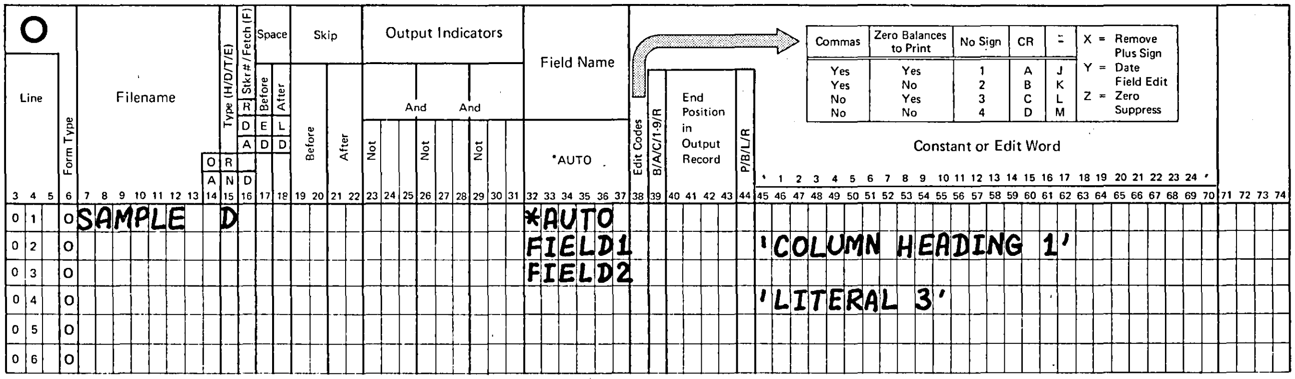
Field Description (Blank or B in Column 39)

D-*AUTO and T-*AUTO field description specifications containing a blank or B in column 39 describe:

- An alphameric field such as an item description
- A numeric field that is not totaled
- A constant
- A field with a literal to be used as a column heading (see Figure 15-6)

A field named on the line (or a constant when no field is named) following a D-*AUTO record description specification is printed only on the detail report line. If the field (or constant when no field is named) on the line follows a T-*AUTO record description, it appears only on the first total line generated.

Output Specifications



Commas	Zero Balances to Print	No Sign	CR	-	X = Remove Plus Sign
Yes	Yes	1	A	J	Y = Date Field Edit
Yes	No	2	B	K	Z = Zero Suppress
No	Yes	3	C	L	
No	No	4	D	M	

Line	Form Type	Filename	Type (H/D/T/E)	Field Name	Constant or Edit Word
01	O	SAMPLE	D	*AUTO	
02	O			FIELD1	'COLUMN HEADING 1'
03	O			FIELD2	
04	O				'LITERAL 3'
05	O				
06	O				

As a result of these specifications, FIELD1 prints on each detail line under the heading COLUMN HEADING 1. FIELD2 and LITERAL 3 print on each detail line without a column heading.

Figure 15-6. Auto Report Field Description Specifications (Blank in Column 39)

Columns 7-22

Columns 7 through 22 are not used on the field description lines. Leave them blank.

Columns 23-31 (Output Indicators)

Enter any valid RPG II output indicators in columns 23 through 31 or leave them blank. If these columns are left blank, the field (or constant when no field is named on the line) is printed on each detail line conditioned by the indicator. When group printing is specified (T-*AUTO specification), the field (or constant when no field is named on the line) is printed each time the lowest level total line is printed. If a column heading is specified in columns 45 through 70 to appear over a field named in columns 32 through 37, the column heading is not affected by output indicators entered in columns 23 through 31.

Columns 32-37 (Field Name)

Enter a field name, data structure name, indexed array name, table name, or blanks in columns 32 through 37. If columns 32 through 37 are blank, a constant must be entered in columns 45 through 70 of the same field description specification. If a field name, data structure name, indexed array name, or table name is entered, the value of the field or element is printed on the detail line (or on the first total line if group printing is specified).

Column 38 (Edit Codes)

Enter a valid RPG II edit code in column 38 if columns 32 through 37 contain the name of a numeric field, a numeric array element, or a numeric table. This column must be blank for alphameric fields, data structures, and table/array elements, and for literals. If column 38 is left blank on a field description line for a numeric field or table/array element, the auto report program provides a K edit code. The K edit code causes a numeric field or element to be printed with commas and a decimal point, such as 3,489.13. It also causes zero suppression; zero balances are not printed, and negative balances are printed with a minus sign on the right.

Column 39 (Blank After)

Entry	Explanation
Blank	Field is not to be reset to zeros or blank after printing.
B	Numeric field is reset to zeros after it is printed; alphameric field is reset to blanks.

Enter a B in column 39 to reset alphameric fields or data structures to blanks or to reset numeric fields to zeros after they are printed. Blank after cannot be used for constants. This entry applies only to the detail line (or the first total line if group printing is specified).

Columns 40-43 (End Position in Output Record)

Either leave columns 40 through 43 blank or enter the print position of the rightmost character of the field (or constant if no field is named in columns 32 through 37) to be printed. If this column is blank, auto report generates end positions for fields, constants, and column headings. See *Report Format* in this chapter for additional information and considerations.

Column 44

Column 44 is not used because packed and binary data cannot be specified. Leave this column blank.

Columns 45-70 (Constant)

Enter a constant or blanks in columns 45 through 70 when column 39 contains a blank. Constants are enclosed in apostrophes according to the standard RPG II rules for coding constants. If these columns are left blank, a field name, data structure name, indexed array name, or table name must be entered in columns 32 through 37. Column heading continuation lines can follow this field description line, but the first line of the printed column heading will be blank. See *Field Description (C in Column 39).*

If a constant is entered in these columns along with a field name in columns 32 through 37, the constant is printed on the first column heading line over the field value. When a column heading is used, the length used to space the column on the report is the greater of the longest column heading length or the field length, adjusted for editing. See *Report Format* in this chapter for additional information on how columns and fields are centered and spaced by auto report.

If a constant is entered in columns 45 through 70 and field name (columns 32 through 37) is blank, the constant is printed each time the detail report line is printed. In group printing, the constant is printed each time the first generated total line is printed.

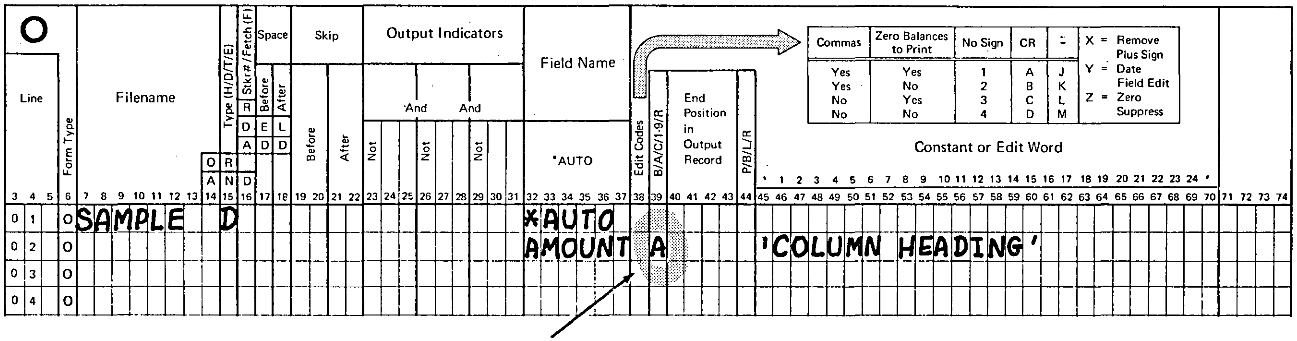
Field Description (A in Column 39)

Enter an A in column 39 of a field description specification following a D/T-*AUTO specification to accumulate and print totals for the field named in columns 32 through 37 (see Figure 15-7). As many levels of totals are printed as are defined in the control level entry (columns 59 and 60) on input specifications. A final total is also printed when the LR indicator is on. (This process is called *total rolling.*)

If group printing is specified and a control level indicator higher than the lowest defined control level is specified in columns 23 through 31 on the record description specification, totals are generated for the indicator entered, all higher defined indicators, and LR.

The total output record generated by auto report if you have entered an A in column 39 of a field description specification is conditioned by the associated control level indicator defined in the input specifications. One total output record is generated for each control level indicator defined in the program.

Output Specifications


O
Line
Form Type
Filename
Type (H/D/T/E)
Stkr #/Fetch (F)
Space
Before
After
Skip
Output Indicators
And
And
Not
Field Name
*AUTO
Edit Codes
B/A/C/1-9/R
End Position in Output Record
P/B/L/R
Constant or Edit Word
Commas
Zero Balances to Print
No Sign
CR
X = Remove Plus Sign
Y = Date Field Edit
Z = Zero Suppress

```
01 OSAMPLE  D                *AUTO
02 O                         AMOUNT A     'COLUMN HEADING'
03 O
04 O
```

The A in column 39 causes the AMOUNT field to be accumulated. Totals are printed for each control level and a final total is printed. A column heading is specified in columns 45 through 70.

Figure 15-7. Describing a Field That is to be Accumulated

Generated Total Fields

When an A is specified in column 39 of a detail or total *AUTO field description specification, auto report generates and names total fields to be used in accumulating the required levels of totals. Auto report generates the field names for the total fields based on the name in columns 32 through 37 of the A-type field description. Names are generated in the following way:

- If the specified field name has fewer than 6 characters, 1 character is added to the name to create a name for the total field. The added character is 1 through 9 or R, corresponding to the total indicators L1 through L9 and LR, respectively. For example, if ITEM is the specified field name and all nine control levels are defined, the generated field names are ITEM1, ITEM2, ... ITEM9, and ITEMR.
- If the specified field name has 6 characters, the last character is replaced by one of the characters, 1 through 9, or R. For example, if AMOUNT is the specified field name and all nine control levels are defined, the generated field names are AMOUN1, AMOUN2, ... AMOUN9, and AMOUNR.

Total fields are generated and named for all control level indicators defined in the program and for LR. (For an exception to this rule, see Figure 15-11 under *Group Printing*.) For example, if L1 and L3 are assigned to control fields on the input specifications and the field QTY is specified, three total fields, QTY1, QTY3, and QTYR are generated and named by auto report. All total fields generated for the same level, such as QTY1 and AMOUN1, are printed on the same total line, and that line is conditioned by the corresponding control level indicator.

Generated total fields are two digits longer than the original field. For example, if the field QTY is defined with a length of three, QTY1, QTY3, and QTYR all have lengths of five. The number of decimal positions remains the same in the generated fields. You can redefine a field name previously defined in a program that is the same as a generated field name, giving that field whatever length and number of decimal positions you want. If you do this, the generated field is assigned the previously defined length and number of decimal positions (if the previous field is numeric).

Considerations

Generated field names can be referenced in RPG II specifications that are included in the program. The programmer must be aware, however, that the use of generated fields in this way can interfere with the automatic accumulation of totals performed by auto report.

Field names ending in 1 through 9 or R should not be used in an auto report program that accumulates totals, because auto report generates total fields ending in those characters. This is especially important for 6-character field names because auto report forms total field names by replacing the last character with 1 through 9 or R. No field name can be used more than once with an A in column 39. Also, if a 5- or 6-character field name is specified with an A in column 39, a second 5- or 6-character field name in which the first 5 characters are identical cannot be specified with an A in column 39. For example, if the following four field names are specified with an A in column 39 in an auto report specification, all but the first are invalid:

FIELD	
FIELDX	Invalid because the first 5 characters duplicate the first 5 characters of the first field.
FIELDY	Invalid for the same reason as that for FIELDX.
FIELD	Invalid because it is a duplicate of the first field.

Columns 7-22

Columns 7 through 22 must remain blank on the field description lines.

Columns 23-31 (Output Indicators)

Enter any valid RPG II output indicators in columns 23 through 31 or leave them blank. If these columns are blank, the field described is printed on each detail line. If indicators are entered in columns 23 through 31, the field is printed only when the conditions represented by those indicators are met. Leave these columns blank for group printing.

If a column heading is specified in columns 45 through 70 to appear over a field named in columns 32 through 37, the column heading is not affected by output indicators entered in these columns. Also, output indicators specified when column 39 contains an A do not affect the generation of calculations for the field.

Output indicators specified on an A-type field description specification following a D-*AUTO specification condition the calculations generated for the field. If the A-type field description follows a T-*AUTO specification, however, a specified indicator does not condition calculations generated for the field.

Columns 32-37 (Field Name)

When column 39 contains an A, the name of a numeric field that is to be accumulated must be entered in columns 32 through 37. These columns cannot identify an array, array element, or table. The field named is printed on each detail line of the report. If group printing is specified, the total field for the lowest control level indicator defined (L1, L2, ... L9, LR, in that order) is printed on the generated total line. (For an exception to this rule, see Figure 15-11 under *Group Printing.*) Totaling for any particular field by means of an A entry in column 39 can be specified only once in each program.

To generate calculation and output specifications that accumulate and print the various levels of totals required, auto report creates and names additional totaling fields. Names for the fields are constructed based on the field name specified in these positions according to a set of rules (see *Generated Total Fields*).

Column 38 (Edit Codes)

Enter an edit code in column 38 or leave it blank. If this column is blank, auto report generates a K edit code for the field named in columns 32 through 37. This causes the field to be edited with commas and a decimal point, such as 1,234,567.89. The field is also zero suppressed. Zero balances are not printed; negative balances are printed with a minus sign on the right. The edit code specified, or the generated K edit code, applies to all generated total fields as well as to the field named in columns 32 through 37.

Column 39

Enter an A in column 39 to indicate that totals are to be accumulated for the field named in columns 32 through 37 of this field description. A total is printed for every control level indicator defined in the input specifications and for the LR indicator. When column 39 contains an A, columns 32 through 37 must contain the name of a numeric field. Columns 45 through 70 can contain a constant to be used as the first line of a column heading. (See *Generated Specifications* for additional information.)

When the lowest control level indicator used for a T-*AUTO specification is higher than the lowest control level indicator defined in the input specifications, auto report generates only the total lines corresponding to the lowest control level indicator used for the T-*AUTO specification, the higher defined control levels, and LR (see *Group Printing*).

Resetting Total Fields to Zero: When column 39 contains an A, the auto report program generates a B (blank after) in column 39 of all the detail and total field description specifications generated from the field name specified. Thus, the value in the specified field and in any generated fields is reset to zero after the field value is printed.

If group printing is specified, auto report generates a calculation to reset the specified field to zero on each cycle. This prevents the same value from being accumulated more than once. An unconditioned total calculation operation (Z-ADD) sets the field value to zero. This calculation is the first total calculation in the generated RPG II source program.

Asterisk Indication: To indicate that a printed line is a generated total line, asterisks are printed on the line to the right of the highest end position generated from the D/T-*AUTO specification. One asterisk is printed to the right on the lowest level total line generated. One additional asterisk is printed on each higher level line, including the final total.

For example, if L1 and L3 are defined control level indicators in a program, one asterisk is printed to the right of the L1 line, two asterisks are printed on the L3 line, and three are printed on the LR line. As many as 10 asterisks are printed on the LR line if all nine control level indicators are defined in the program.

To suppress the generation of asterisks on total lines, enter an N in column 28 of the auto report specifications sheet.

Columns 40-43 (End Position in Output Record)

Enter the print position of the rightmost character of the field to be printed, or leave these positions blank. If this entry is blank, auto report generates end positions for fields and column headings. See *Report Format* in this chapter for additional information and considerations.

Column 44

Column 44 is not used with auto report because packed and binary data cannot be used. Leave these columns blank.

Columns 45-70 (Constant)

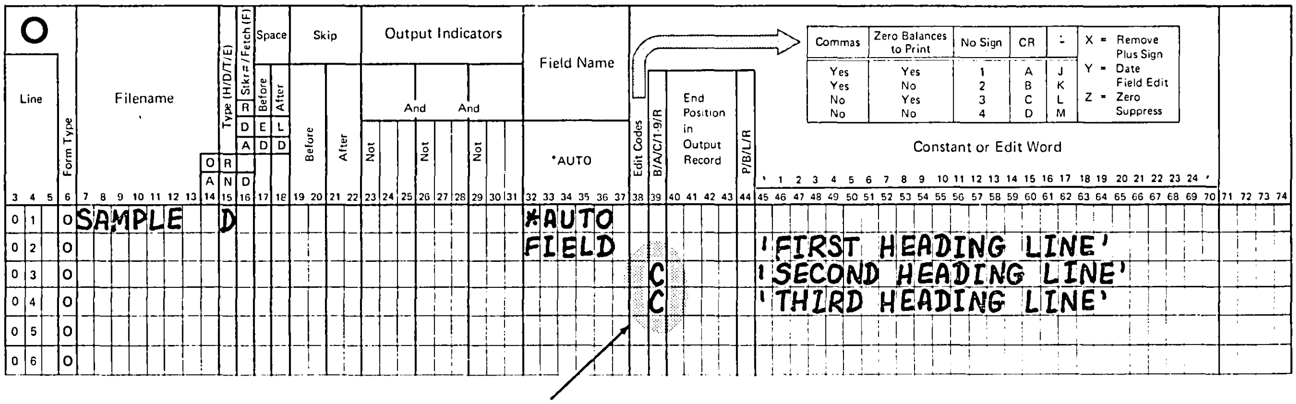
Either leave columns 45 through 70 blank or enter a literal. Do not enter an edit word; editing is accomplished by an edit code. If a literal is entered when column 39 contains an A, the literal becomes the first line of the column heading over the accumulated field.

If these columns are left blank, the first line of the column heading is blank, but column heading continuation lines can specify the second and third lines of the column heading. See *Field Description (C in Column 39)*. Also see *Report Format* for information on how column heading and fields are centered and spaced by auto report.

Field Description (C in Column 39)

Enter a C in column 39 of the *AUTO field descriptions to specify a second and third column heading line. At times you may want more information in a column heading than can be contained on one line. Auto report enables you to specify the second and third lines of column headings by simply specifying the literals to appear on those lines. No additional heading output lines need be coded; no end position need be calculated. The special field description specification that allows you to do this is identified by a C in column 39 (see Figure 15-8).

Output Specifications


O
Line
Form Type
Filename
Type (H/D/T/E)
Stkr #/Fetch (F)
Space
Before
After
Skip
Before
After
Output Indicators
And
And
Not
Not
Not
Field Name
*AUTO
Edit Codes
B/A/C/1-9/R
End Position in Output Record
P/B/L/R
Constant or Edit Word
Commas
Zero Balances to Print
No Sign
CR
-
Yes
Yes
1
A
J
Yes
No
2
B
K
No
Yes
3
C
L
No
No
4
D
M
X = Remove Plus Sign
Y = Date Field Edit
Z = Zero Suppress
OR
AND
ADD
DEL

```
0 1 O SAMPLE  D   *AUTO
0 2 O             FIELD      'FIRST HEADING LINE'
0 3 O                   C    'SECOND HEADING LINE'
0 4 O                   C    'THIRD HEADING LINE'
0 5 O
0 6 O
```

C in column 39 is used to specify second and third column heading lines. A maximum of three column heading lines (two C-type field descriptions) can be used.

Figure 15-8. Specifying Second and Third Column Heading Lines

Columns 7-38

Columns 7 through 38 must be blank on a field description that has a C in column 39.

Column 39

Enter a C in column 39. One or two C-type specifications can follow a field description specification that has an A, B, or blank in column 39 and an entry in columns 32 through 37. The first C-type specification causes a second column heading line to be generated. The second C-type specification causes a third column heading line to be generated (see Figure 15-8).

Columns 40-44

Columns 40 through 44 must be blank on a C-type field description specification.

Columns 45-70 (Constant)

Enter a constant, up to 24 positions long including blanks, enclosed in apostrophes. The constant becomes the second or third line of column headings, depending on whether it is on the first or second C-type specification. If two or three column heading lines are specified, the shorter literals are centered on the longest.

Field Description (1-9 or R in Column 39)

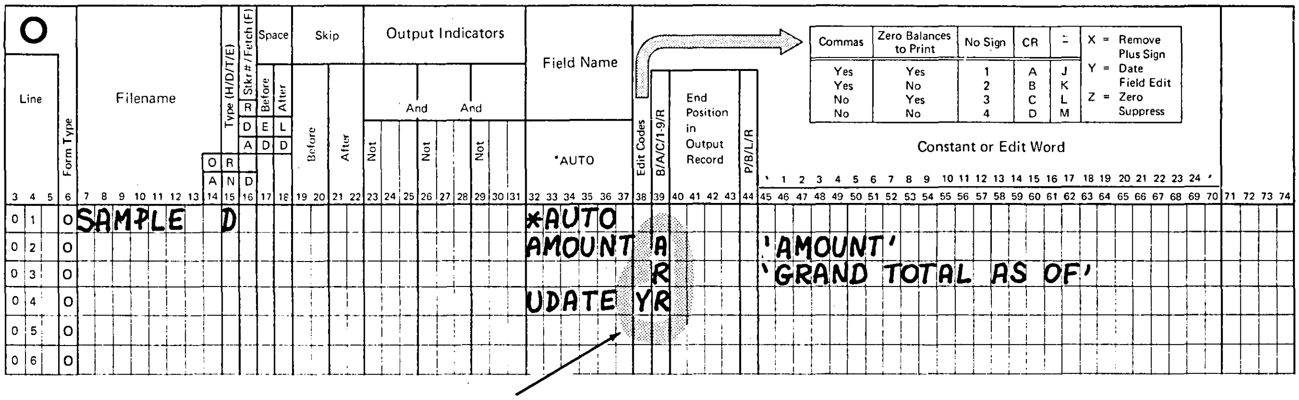
Enter a digit (1 through 9) or R in column 39 of a field description to specify a field or constant to be printed on a specific total line.

Auto report allows you to print other information on generated total lines in addition to the generated totals resulting from A-type field descriptions. The value entered in column 39 corresponds to the level of the total line on which the information is to be printed (the corresponding control level must be defined in columns 59 and 60 in the input specifications). For example, a 3 in column 39 indicates that the information is printed on the L3 total line; an R indicates that the information appears on the final total, or LR, line (see Figure 15-9). Fields and constants specified in this way are printed to the left of the leftmost generated total on the line. See *Report Format* for exact placement.

This type of field description can print information such as DISTRICT TOTAL, GRAND TOTAL, or other literal information. It can also print a field and specify an edit word, floating dollar sign, or asterisk protection for the field.

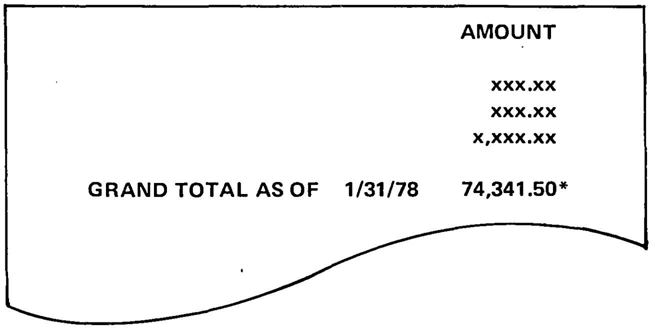
If none of the *AUTO output fields is defined with an A in column 39, then 1 through 9 or R cannot be used in column 39. In group printing, only specify numbers that are higher than the lowest control level indicator used to condition the T-*AUTO specification. If the T-*AUTO specification is not conditioned by a control level indicator, use only numbers that are higher than the lowest control level defined in columns 59 and 60 on the input specifications.

Output Specifications



Line	Form Type	Filename	Type (H/D/T/E)	Field Name / *AUTO	Edit Codes	B/A/C/1-9/R	Constant or Edit Word
01	O	SAMPLE	D	*AUTO			
02	O			AMOUNT		A	'AMOUNT'
03	O					R	'GRAND TOTAL AS OF'
04	O			UDATE	Y	R	
05	O						
06	O						

In this example, the literal 'GRAND TOTAL AS OF' followed by the current date prints on the left of the generated final total line, as shown below.



```
                                   AMOUNT

                                    XXX.XX
                                    XXX.XX
                                  X,XXX.XX

    GRAND TOTAL AS OF   1/31/78   74,341.50*
```

Figure 15-9. Specifying a Literal and a Field to Print on a Generated Total Line

Columns 7-31

Columns 7 through 31 must be blank on a field description line with 1 through 9 or R in column 39.

Columns 32-37 (Field Name)

Enter the name of a field, an indexed array name, or a table name. The corresponding field or element value prints on the total line indicated by the entry in column 39. If columns 32 through 37 are blank, a constant must be entered in columns 45 through 70.

Column 38 (Edit Code)

Enter an edit code in column 38 to edit a numeric field named in columns 32 through 37, or leave column 38 blank. If column 38 is left blank, an edit word can be entered in columns 45 through 70. If column 38 is blank, no edit code is assumed by auto report.

Column 39

Enter a digit (1 through 9) or R. These entries correspond to the indicators L1, L2, ... L9, and LR. The entry identifies a specific total line on which the field or literal described is to be printed. The entry in column 39 must correspond to a control level that is defined by the input specifications. In group printing, the entry in this column must be higher than the control level of the first total line generated.

Columns 40-43 (End Position in Output Record)

Do not make an entry in columns 40 through 43 on field description specifications with 1 through 9 or R in column 39. See *Report Format* for additional information and considerations.

Column 44

Leave column 44 blank.

Columns 45-70 (Constant or Edit Word)

Leave columns 45 through 70 blank, or enter a constant or edit word. If field name (columns 32 through 37) on this specification line contains an entry, then columns 45 through 70 can contain any of the following:

- Blanks, if no editing is needed for the field or if the field is already edited by an edit code in column 38
- Edit word, if special editing is desired
- Floating currency symbol or asterisk protection entry used with an edit code

Columns 45 through 70 cannot contain a constant when field name contains an entry. However, when field name is blank, columns 45 through 70 must contain a constant.

Group Printing

In group printing, data is summarized for a group of input records and only totals are printed on the report. Totals can have subtotals with a final total or only a final total.

Specifications

To specify group printing using auto report, enter a T in column 15 and *AUTO in columns 32 through 36. A control level indicator can be specified in columns 23 through 31. When a T-*AUTO specification is used, a line is not printed for each individual record that is read, but only after a complete control group is read.

Fields and literals defined by field description specifications that have a blank or B in column 39 and follow a T-*AUTO record description are printed on the lowest level total line. Fields defined with an A in column 39 are not printed on the total lines, but the total fields created by auto report are. Generated calculations are printed on their associated total lines. Continued column headings (C in column 39) and total-indicated fields (1 through 9 or R in column 39) can also be specified by field descriptions following a T-*AUTO record description.

Output indicators can be entered in columns 23 through 31 of a field description specification following a T-*AUTO record description if column 39 of the field description specifications contains a blank or a B. If output indicators are used in a field description that has an A in column 39 following a T-*AUTO specification, those indicators are ignored by auto report. Output indicators cannot be used in a field description that contains C, 1 through 9, or R in column 39.

Examples

Figure 15-10 shows the file description and input specifications for the group printed reports shown in Figures 15-11 and 15-12. BRANCH and REGION are defined as control fields.

Figure 15-11 shows the output specifications and the group printed report showing sales totals for a company. Since the T-*AUTO specification is conditioned by L2, only the totals for REGION (L2) and for the entire company (LR) are printed on the report. The totals for BRANCH (L1) are not printed.

A disk summary file, DISKSUM, is also produced by this program. The summary file contains a summary record of the sales data for each branch. The output specifications for DISKSUM illustrate the use of standard RPG II output specifications in the same program with *AUTO specifications. The output record described is written on the disk file, DISKSUM, when there is an L1 control break (BRANCH field changes). Since the T-*AUTO specification is conditioned by L2, auto report does not generate fields for the L1 control level. Therefore, standard RPG II calculation specifications must be used to calculate the L1 totals. The L1 total fields that are written on the DISKSUM file (SOLDQ1, SOLDV1, and VALUE1) must be defined in the calculations.

Figure 15-12 shows a group printed report similar to the one shown in Figure 15-11. However, the T-*AUTO specifications are not conditioned by a control level indicator, so totals are printed for all defined control levels and for LR.

/COPY STATEMENT SPECIFICATIONS

The auto report copy function provides a way to include cataloged RPG II source specifications in an RPG II program. The source specifications that are included must reside as a library member on disk. Create the library member by using the library maintenance disk utility program. Use the copy function to include source specifications that are identical or nearly identical in several different programs, thereby reducing the need to repeatedly code specifications that are used in several programs. For example, if file description and input specifications for a particular file are similar in different programs, these specifications can be placed in the library by the library maintenance program or the source entry utility (SEU) and included in any program by the copy function.

Auto report specifications and any valid RPG II specifications, including tables and arrays, can be copied in this manner. The auto report option specifications and other copy statements cannot be copied. See *Examples of Using Auto Report* in this chapter for an example of using the copy function.

The specifications included in an auto report program by the copy function are initially placed in the program immediately following the /COPY statement. When all specifications are copied from the library, the entire auto report program is sorted into the order required by the RPG II compiler (see *Order of Generated Specifications* in this chapter).

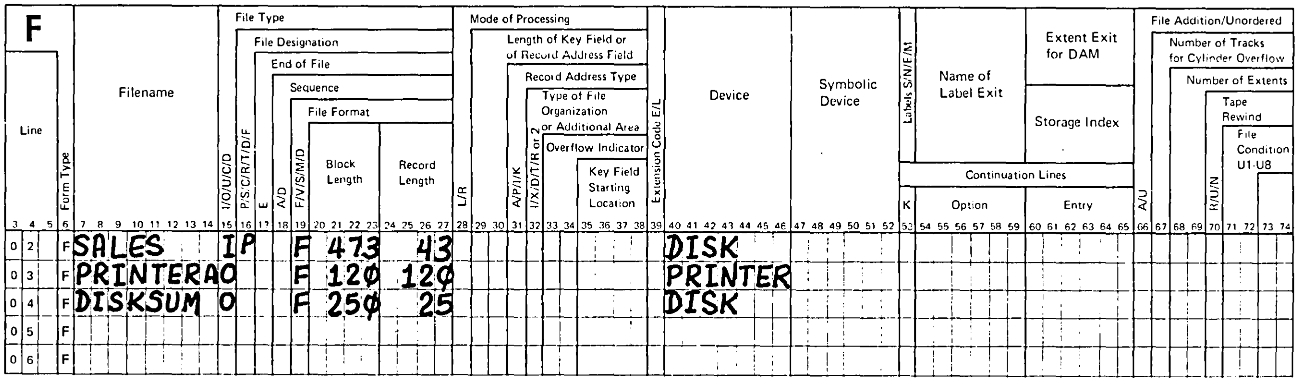
To request the copy function, use the /COPY statement. This statement identifies the library and library member containing the RPG II specifications to be included in the source program generated by auto report. /COPY statements must follow the auto report option specifications, and they must precede source tables and arrays (file translation tables, alternate collating sequence tables, and compile-time tables and arrays).

The format of the /COPY statement is:

Column	**Entry**
1-5	Page and line number indicating the placement of the statement in the sequence of auto report source specifications.
6	This column can contain any entry except H or U, or can be blank.
7-11	Enter the characters /COPY.
12	Blank.
13-29	Identifies the library member to be included. Specify the library name, which can be up to 8 characters long, beginning in column 13. Use a comma to separate the library name and the member name, which can also be up to 8 characters long. If an entry is not made for the library name or if F1 is specified, the default is to the system library.
30-49	Blank.
50-80	Enter any information or comments. The contents of these columns are not read by auto report.

Figure 15-13 shows an example of the /COPY statement.

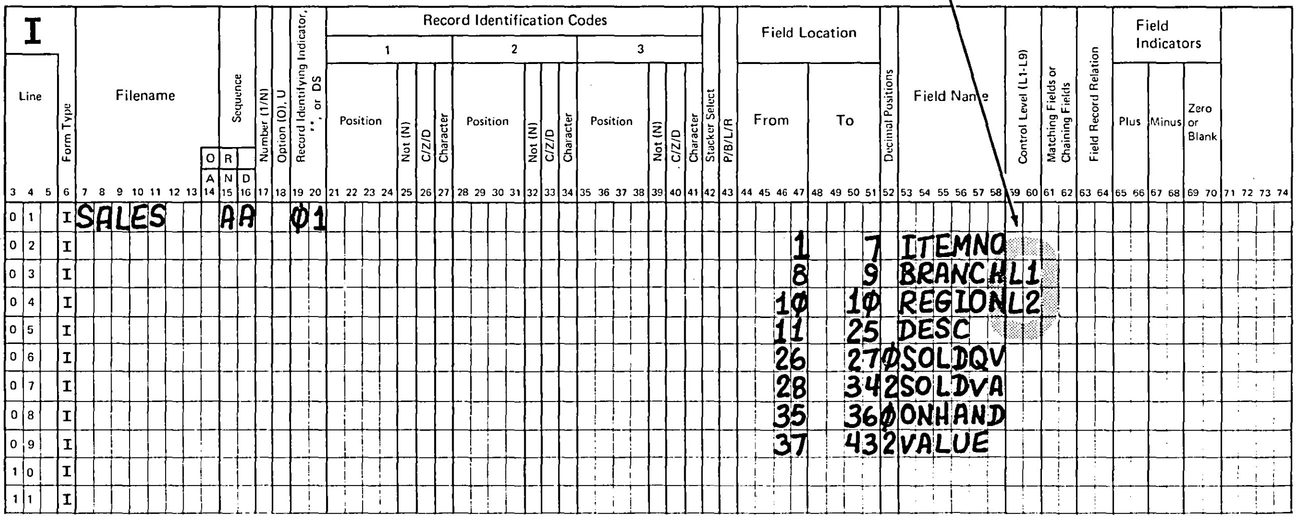
File Description Specifications



Line	Form Type	Filename	File Type	File Designation	File Format	Block Length	Record Length	Device
02	F	SALES	I	P	F	473	43	DISK
03	F	PRINTERA	O		F	12Ø	12Ø	PRINTER
04	F	DISKSUM	O		F	25Ø	25	DISK
05	F							
06	F							

L1 and L2 are the defined control levels.

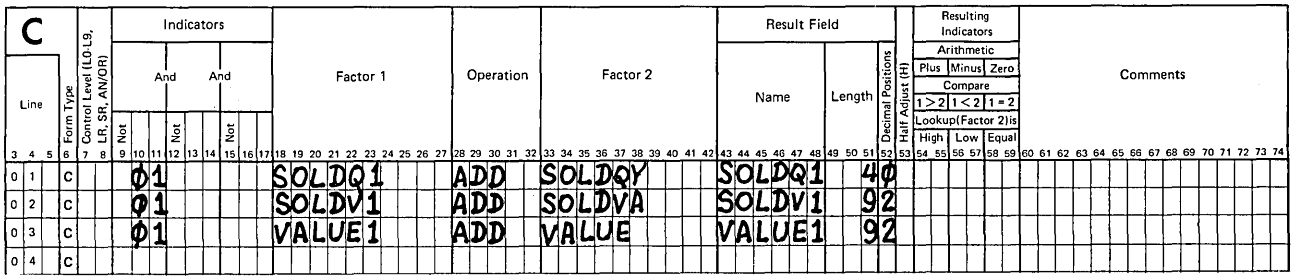
Input Specifications



Line	Form Type	Filename	Sequence	Record Identifying Indicator	Field Location From	Field Location To	Decimal Positions	Field Name	Control Level (L1-L9)
01	I	SALES	AA	Ø1					
02	I				1	7		ITEMNO	
03	I				8	9		BRANCH	L1
04	I				1Ø	1Ø		REGION	L2
05	I				11	25		DESC	
06	I				26	27	Ø	SOLDQV	
07	I				28	34	2	SOLDVA	
08	I				35	36	Ø	ONHAND	
09	I				37	43	2	VALUE	
10	I								
11	I								

Figure 15-10. File Description and Input Specifications for the Group Printed Reports Shown in Figures 15-11 and 15-12

Calculation Specifications

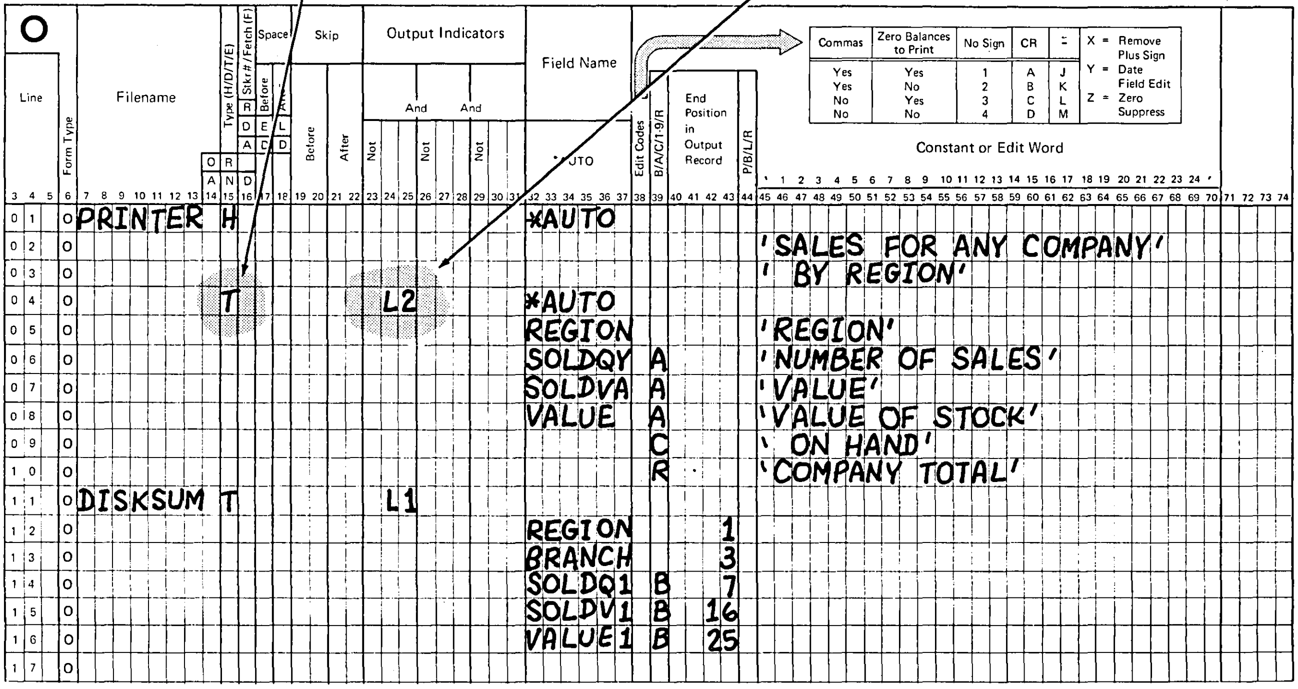


Line	Form Type	Control Level (L0-L9, LR, SR, AN/OR)	Indicators	Factor 1	Operation	Factor 2	Result Field Name	Length	Decimal Positions	Half Adjust (H)	Resulting Indicators	Comments
01	C		Ø1	SOLDQ1	ADD	SOLDQY	SOLDQ1	4	Ø			
02	C		Ø1	SOLDV1	ADD	SOLDVA	SOLDV1	9	2			
03	C		Ø1	VALUE1	ADD	VALUE	VALUE1	9	2			
04	C											

T in column 15 with *AUTO in columns 32 through 37 specifies a group printed report.

Because L2 is entered under output indicators, total lines are printed only for L2 and LR, although L1 is also a defined control level.

Output Specifications

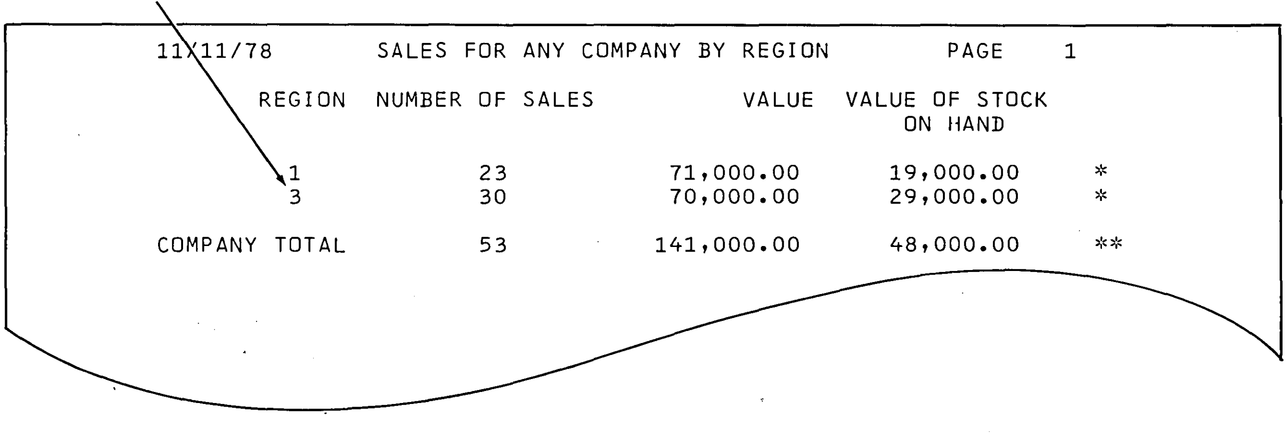


Commas	Zero Balances to Print	No Sign	CR	-
Yes	Yes	1	A	J
Yes	No	2	B	K
No	Yes	3	C	L
No	No	4	D	M

X = Remove Plus Sign
Y = Date Field Edit
Z = Zero Suppress

Line	Form Type	Filename	Type (H/D/T/E)	Space/Skip	Output Indicators	Field Name / *AUTO	Edit Codes	B/A/C/1-9/R	End Position in Output Record	Constant or Edit Word
01	O	PRINTER	H			*AUTO				
02	O									'SALES FOR ANY COMPANY'
03	O									' BY REGION'
04	O		T		L2	*AUTO				
05	O					REGION				'REGION'
06	O					SOLDQY	A			'NUMBER OF SALES'
07	O					SOLDVA	A			'VALUE'
08	O					VALUE	A			'VALUE OF STOCK'
09	O						C			' ON HAND'
10	O						R			'COMPANY TOTAL'
11	O	DISKSUM	T		L1					
12	O					REGION			1	
13	O					BRANCH			3	
14	O					SOLDQ1	B		7	
15	O					SOLDV1	B		16	
16	O					VALUE1	B		25	
17	O									

In group printing, the lowest level total lines printed (L2, in this case) are single-spaced, like detail lines.



```
      11/11/78        SALES FOR ANY COMPANY BY REGION          PAGE    1

               REGION  NUMBER OF SALES          VALUE  VALUE OF STOCK
                                                             ON HAND

                    1           23         71,000.00       19,000.00     *
                    3           30         70,000.00       29,000.00     *

      COMPANY TOTAL             53        141,000.00       48,000.00     **
```

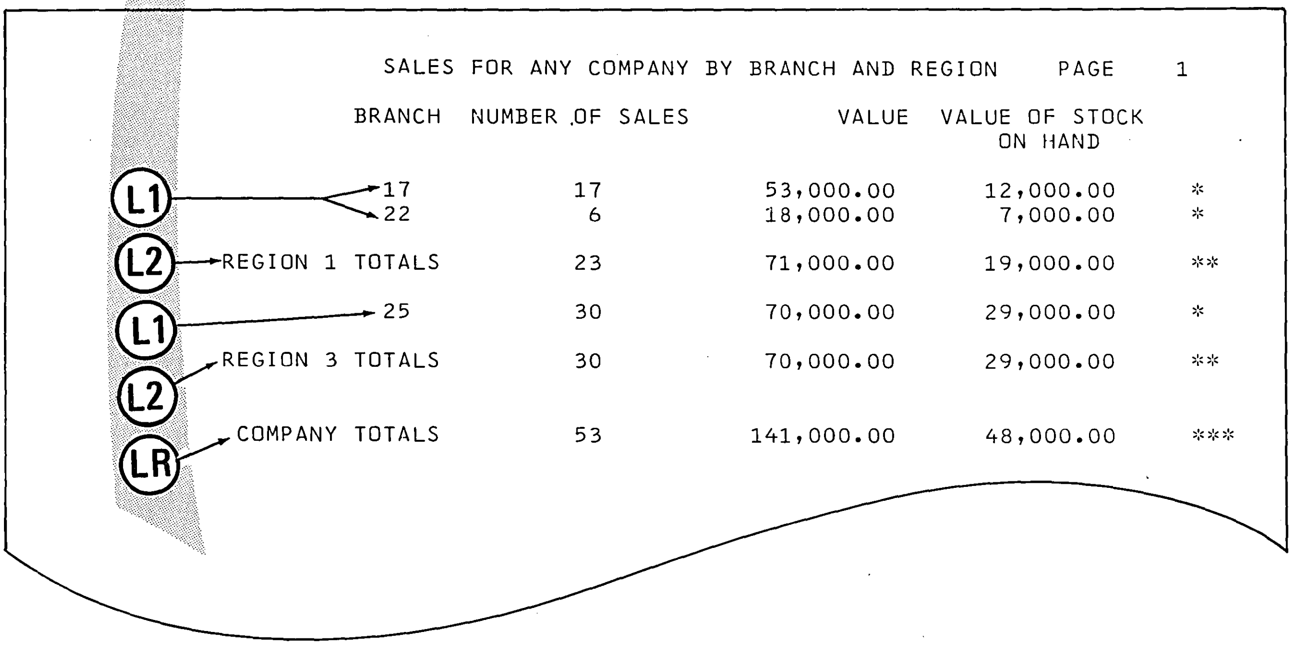
Figure 15-11. Using *AUTO to Produce a Group Printed Report Showing Region and Final Totals

Output Specifications

Commas	Zero Balances to Print	No Sign	CR	-	
Yes	Yes	1	A	J	X = Remove Plus Sign
Yes	No	2	B	K	Y = Date Field Edit
No	Yes	3	C	L	Z = Zero Suppress
No	No	4	D	M	

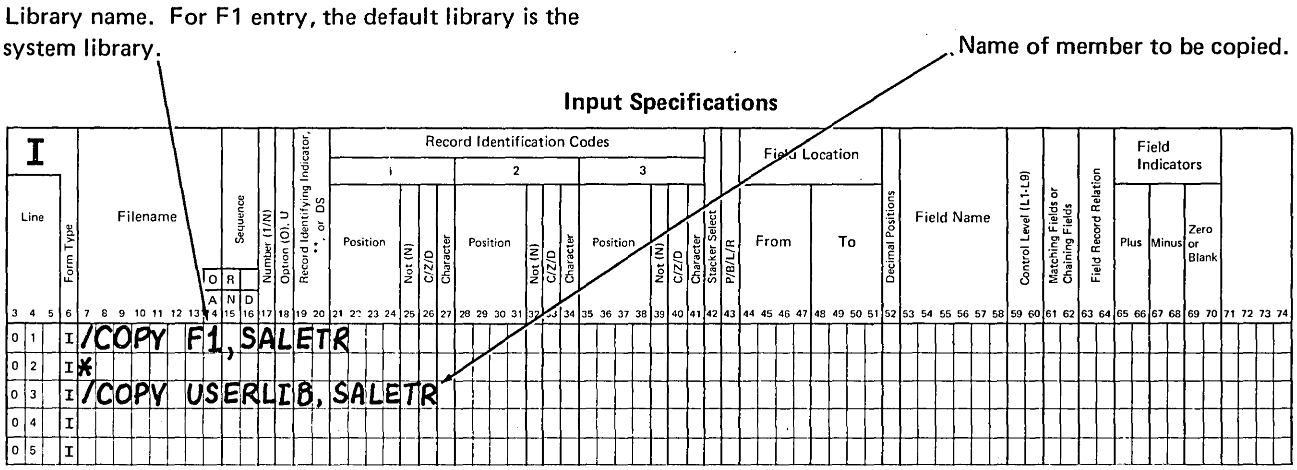
Line	Form Type	Filename	Type (H/D/T/E)	Space Before	Space After	Skip Before	Skip After	Output Indicators	Field Name	Edit Codes	B/A/C/1-9/R	End Position in Output Record	P/B/L/R	Constant or Edit Word
01	O	PRINTER	H						*AUTO					
02	O													'SALES FOR ANY COMPANY'
03	O													' BY BRANCH AND REGION'
04	O		T						*AUTO					
05	O								BRANCH					'BRANCH'
06	O								SOLDQY		A			'NUMBER OF SALES'
07	O								SOLDVA		A			'VALUE'
08	O								VALUE		A			'VALUE OF STOCK'
09	O										C			' ON HAND'
10	O										2			'REGION'
11	O								REGION		2			
12	O										2			'TOTALS'
13	O										R			'COMPANY TOTALS'
14	O													
15	O													

When no control level indicators are entered under output indicators, a total line is generated for each defined control level indicator (L1 and L2, in this case) and for LR.


L1
L2
L1
L2
LR

```
                SALES FOR ANY COMPANY BY BRANCH AND REGION      PAGE     1

              BRANCH  NUMBER OF SALES          VALUE  VALUE OF STOCK
                                                            ON HAND

                  17             17        53,000.00       12,000.00     *
                  22              6        18,000.00        7,000.00     *

       REGION 1 TOTALS           23        71,000.00       19,000.00     **

                  25             30        70,000.00       29,000.00     *

       REGION 3 TOTALS           30        70,000.00       29,000.00     **

         COMPANY TOTALS          53       141,000.00       48,000.00     ***
```

Figure 15-12. Using *AUTO to Produce a Group Printed Report Showing Branch, Region, and Final Totals


Library name. For F1 entry, the default library is the system library.

Name of member to be copied.

Input Specifications

I
Line
Form Type
Filename
Sequence
OR
AND
Number (1/N)
Option (O), U
Record Identifying Indicator, **, or DS
Record Identification Codes
1
2
3
Position
Not (N)
C/Z/D
Character
Stacker Select
P/B/L/R
Field Location
From
To
Decimal Positions
Field Name
Control Level (L1-L9)
Matching Fields or Chaining Fields
Field Record Relation
Field Indicators
Plus
Minus
Zero or Blank

```
0 1 I/COPY F1,SALETR
0 2 I*
0 3 I/COPY USERLIB,SALETR
0 4 I
0 5 I
```

Note: It is convenient to code the /COPY statement on the input specifications sheet if input specifications are to be modified as they are copied.

Figure 15-13. Example of the /COPY Auto Report Statement

Modifying Copied Specifications

Statements can be included in the auto report specifications to modify file description and input field specifications as they are copied from the library. No other types of specification can be modified. /COPY modifier statements from the source program that add, change, or delete entries on cataloged input field specifications are identified by an X in print position 6 of the auto report listing.

Modifying File Description Specifications

To modify a file description specification that is copied from the library, enter the filename in columns 7 through 14 of a file description specification (F in column 6). Then make only those entries on the line that are to replace existing entries in the copied specification or that are to be included as new entries. Blank entries in the modifier statement do not affect the copied statement.

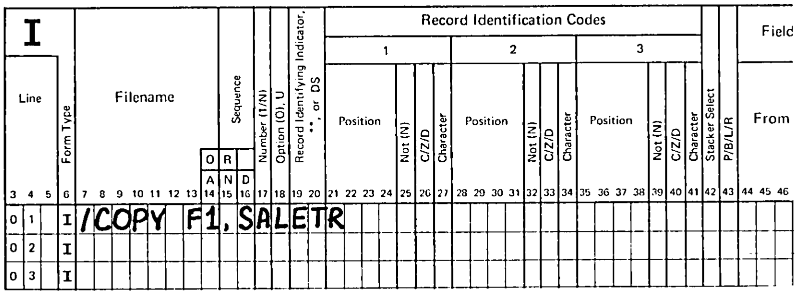
For example, the file description specifications for a frequently used file named SALES are to be copied from the system library. The original specification contains an I in file type (column 15), defining SALES as an input file (see Figure 15-14). To update the sales file, change column 15 to a U by including a modifier file description specification in the auto report source program. The modifier statement must contain the filename, SALES, and the new file type entry, U. As a result of the modifier statement, the file type on the copied file description specification is changed from I to U.

To set an entry to blanks, enter an ampersand (&) in the first position of that entry on the modifier statement, and leave the remaining positions blank. For example, to remove the block length entry (columns 20 through 23) from the cataloged specification shown in Figure 15-14, add an ampersand to the modifier statement in column 20, as shown in Figure 15-15, and leave columns 21 through 23 blank.

Modifier statements for file description specifications do not have to be in any particular order in the auto report source program, except that they cannot immediately follow the /COPY statement if input field specifications are also being modified.

/COPY statement to copy specifications for SALES file from the system library member named SALETR

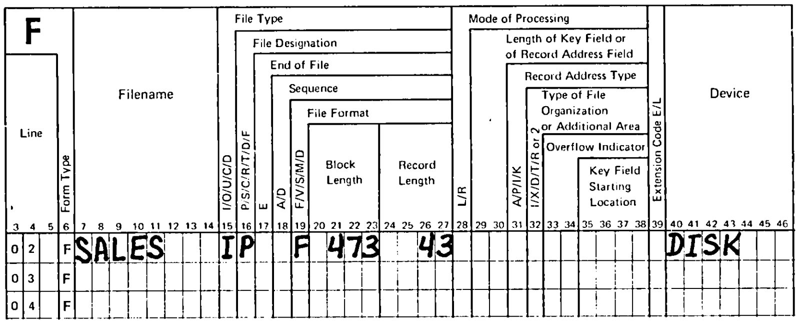
Input Specifications



```
01 I/COPY F1,SALETR
02 I
03 I
```

File description specification as it is cataloged in the system library

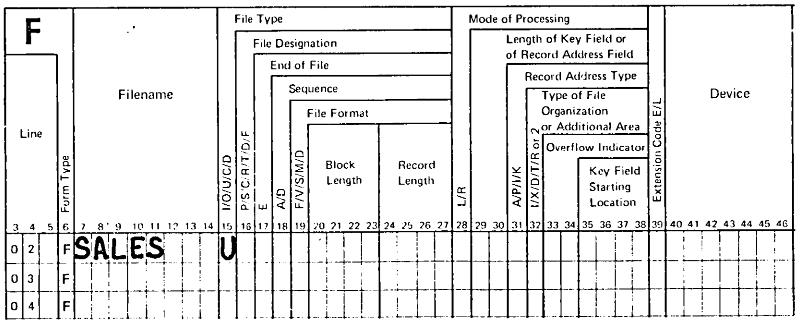
File Description Specifications



```
02 FSALES   IP  F 473  43            DISK
03 F
04 F
```

Copy function modifier statement

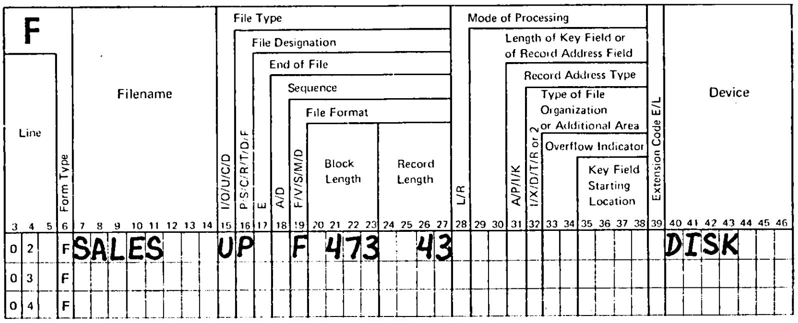
File Description Specifications



```
02 FSALES   U
03 F
04 F
```

Resulting file description specification that is included in the RPG II source program

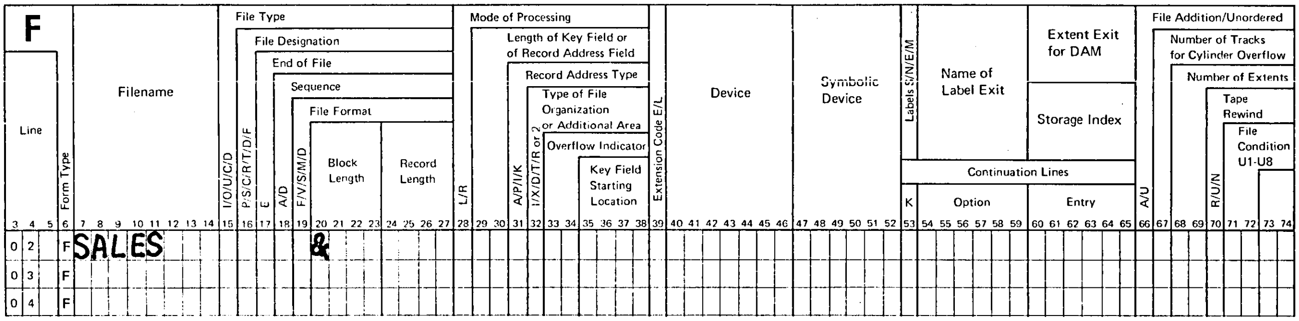
File Description Specifications



```
02 FSALES   UP  F 473  43            DISK
03 F
04 F
```

Figure 15-14. Modifying a Copied File Description Specification

File Description Specifications



Line	Form Type	Filename	Block Length
02	F	SALES	&
03	F		
04	F		

F
Line
Form Type
Filename
File Type
File Designation
End of File
Sequence
File Format
I/O/U/C/D
P/S/C/R/T/D/F
E
A/D
F/V/S/M/D
Block Length
Record Length
L/R
Mode of Processing
Length of Key Field or of Record Address Field
Record Address Type
Type of File Organization or Additional Area
Overflow Indicator
Key Field Starting Location
A/P/I/K
I/X/D/T/R or 2
Extension Code E/L
Device
Symbolic Device
Labels S/N/E/M
Name of Label Exit
Extent Exit for DAM
Storage Index
Continuation Lines
K
Option
Entry
A/U
File Addition/Unordered
Number of Tracks for Cylinder Overflow
Number of Extents
Tape Rewind
R/U/N
File Condition U1-U8

Figure 15-15. Setting a Copied File Description Entry to Blank

Only one file description specification with a particular filename is allowed to come from the library entries, and a particular filename can be used only once on a modifier statement.

No modifications are allowed to the file description continuation specifications that accompany a copied file description. To add new continuation specifications, place them after a file description modifier statement for the file. A maximum of five continuation specifications are allowed to follow a file description specification (combined total of original and added continuation specifications).

Modifying Input Field Specifications

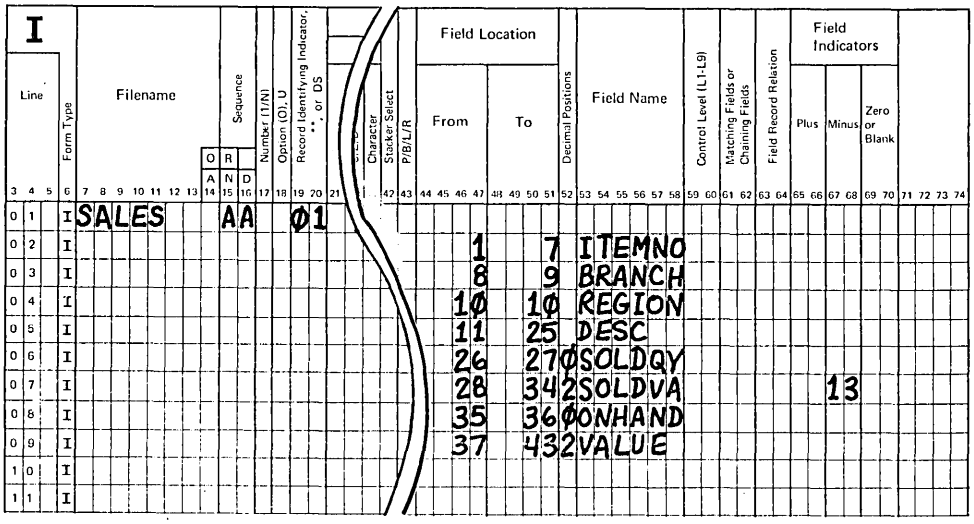
Only input field specifications (specifications describing individual fields on the input record) can be modified. To modify an input field specification copied from the library, enter the field name in columns 53 through 58 of an input field modifier statement (I in column 6). Modifier statements for input field specifications must immediately follow the /COPY statement in the auto report program that copies those specifications. The first specification following the /COPY statement that is not an input field specification is considered the end of the input field modifier statements for the /COPY statement. (A comment statement with an I in position 6 is not considered the end of the input field modifier statements.)

The fields that can be modified are:

• Column 43	(packed/binary)
• Columns 44-51	(field location)
• Column 52	(decimal positions)
• Columns 59-60	(control levels)
• Columns 61-62	(matching or chaining fields)
• Columns 63-64	(field record relation)
• Columns 65-70	(field indicators)

The method of replacing, adding, or blanking entries is similar to the method used to modify file description specifications. To replace or add entries, code the new entry in the proper location in the modifier statement; to set an entry to blank, place an ampersand (&) in the first position of that entry in the modifier statement. Figure 15-16 shows examples of modifying input specifications.

The modifier statement modifies all copied input field specifications that have the same field name. If there is no input field by the same name, the modifier statement is added to the program as a new input field specification. Modifier statements with duplicate field names are allowed (length and number of decimal positions must also be the same), but only the first is used to modify a copied specification. Other field names are added as new input field specifications. Up to 20 input field modifier statements are allowed per /COPY statement.


I
Line
Form Type
Filename
Sequence
Number (1/N)
Option (O), U
Record Identifying Indicator, ••, or DS
Character
Stacker Select
P/B/L/R
Field Location
From
To
Decimal Positions
Field Name
Control Level (L1-L9)
Matching Fields or Chaining Fields
Field Record Relation
Field Indicators
Plus
Minus
Zero or Blank

```
0 1 ISALES    AA  Ø1
0 2 I                   1    7 ITEMNO
0 3 I                   8    9 BRANCH
0 4 I                  1Ø   1Ø REGION
0 5 I                  11   25 DESC
0 6 I                  26   27ØSOLDQY
0 7 I                  28   342SOLDVA            13
0 8 I                  35   36ØONHAND
0 9 I                  37   432VALUE
1 0 I
1 1 I
```

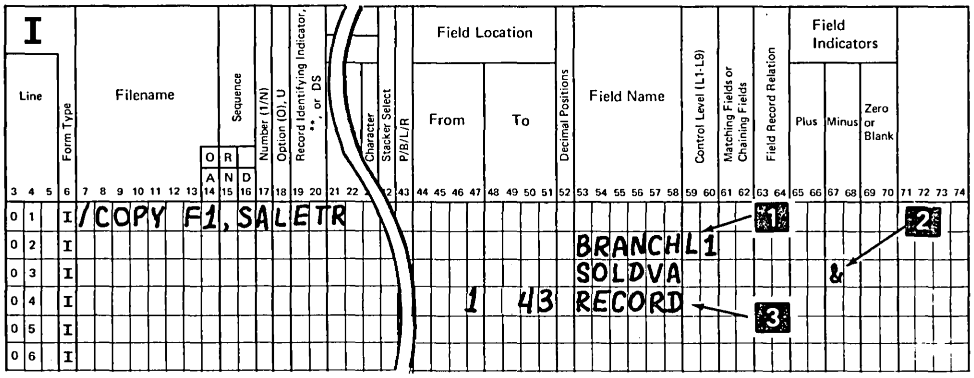
Input specifications as cataloged in the library.

/COPY statement and modifier statements:

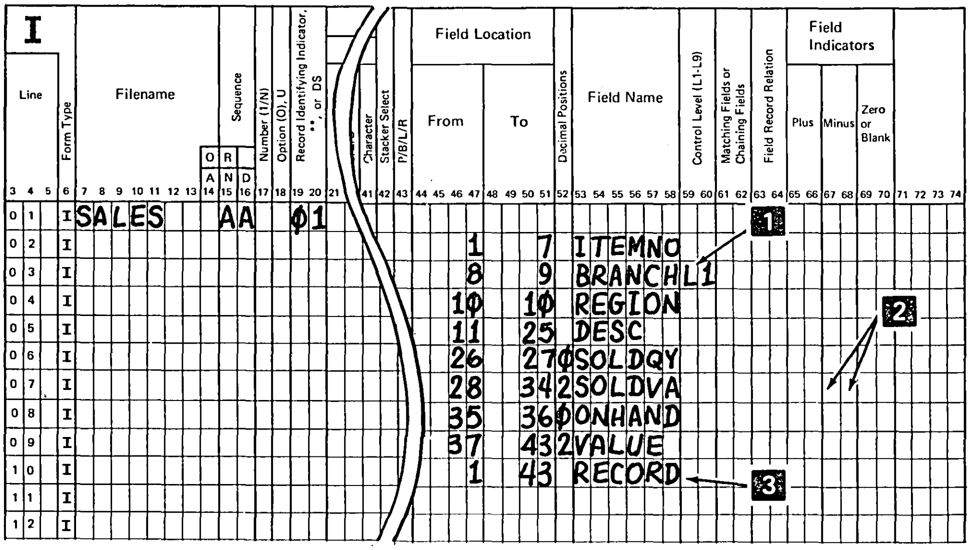
1 Add an entry to BRANCH field description

2 Blank out minus field indicator on SOLDVA description

3 Add a new field description


I
Line
Form Type
Filename
Sequence
Field Location
From
To
Decimal Positions
Field Name
Control Level (L1-L9)
Matching Fields or Chaining Fields
Field Record Relation
Field Indicators
Plus
Minus
Zero or Blank
1
2
3

```
0 1 I/COPY F1,SALETR
0 2 I                          BRANCHL1
0 3 I                          SOLDVA           &
0 4 I                  1   43  RECORD
0 5 I
0 6 I
```


I
Line
Form Type
Filename
Sequence
Field Location
From
To
Decimal Positions
Field Name
Control Level (L1-L9)
Matching Fields or Chaining Fields
Field Record Relation
Field Indicators
Plus
Minus
Zero or Blank
1
2
3

```
0 1 ISALES    AA  Ø1
0 2 I                   1    7 ITEMNO
0 3 I                   8    9 BRANCHL1
0 4 I                  1Ø   1Ø REGION
0 5 I                  11   25 DESC
0 6 I                  26   27ØSOLDQY
0 7 I                  28   342SOLDVA
0 8 I                  35   36ØONHAND
0 9 I                  37   432VALUE
1 0 I                   1   43 RECORD
1 1 I
1 2 I
```

Resulting input specifications for SALES file showing:

1 Added L1 indicator

2 Blanks in place of minus field indicator

3 Added field description

Figure 15-16. Modifying Copied Input Field Specifications

For best results, first place those statements that modify existing input field specifications; then place those that are to be added as new input field specifications. This procedure is suggested because input field modifier statements that do not fit into the special main storage table for modifier statements are added to the RPG II source program as new input field specifications. This order of specifying modifier statements increases the likelihood that excess statements, if any, will be valid field descriptions.

REPORT FORMAT

One of the advantages of auto report is that it frees the programmer from the task of specifying the format of his report on the output specifications sheet. Auto report can completely format the report by spacing, skipping, centering lines, and calculating end positions for fields and constants.

Spacing and Skipping

Spacing and skipping can be either left to auto report or specified by the programmer. Figure 15-17 shows spacing and skipping generated by auto report. For the specifications used to produce the report, see *Auto Report Generated Specifications* in this chapter. If columns 17 through 22 are left blank on an H-*AUTO specification, a skip to line 06 is done before the first heading line is printed and space-two-after is done for the last heading line. If more than one heading line is specified, space-one-after is done for the first and all succeeding lines except the last. To specify spacing and skipping, follow the standard RPG II rules for spacing and skipping.

Column heading lines are spaced like page headings. Space-one-after is done for all except the last. Space-two-after is done for a single heading line, or for the last heading line if more than one is specified. Spacing and skipping entries cannot be specified for column headings. If spacing and skipping entries are made on a D-*AUTO record description specification, the entries apply to the detail line generated. The entries do not apply to column headings or total lines generated by auto report from the D-*AUTO specification. Standard RPG II rules for spacing and skipping must be followed. Space-one-after is assumed for the generated detail line if spacing and skipping entries are not made.

Space-two-after is generated for all total lines produced by auto report from a D-*AUTO specification. In addition, the lowest level total line and the final total line are also generated with a space-one-before.

If spacing and skipping entries are made on a T-*AUTO specification, the entries apply to the lowest level total line generated, but not to column headings or higher level total lines. If spacing and skipping are not made, the lowest level total lines are generated with space-one-after; all higher levels are generated with space-two-after. Space-one-before is always generated for the second-to-the-lowest level total and the final total (see Figure 15-12 for an example).

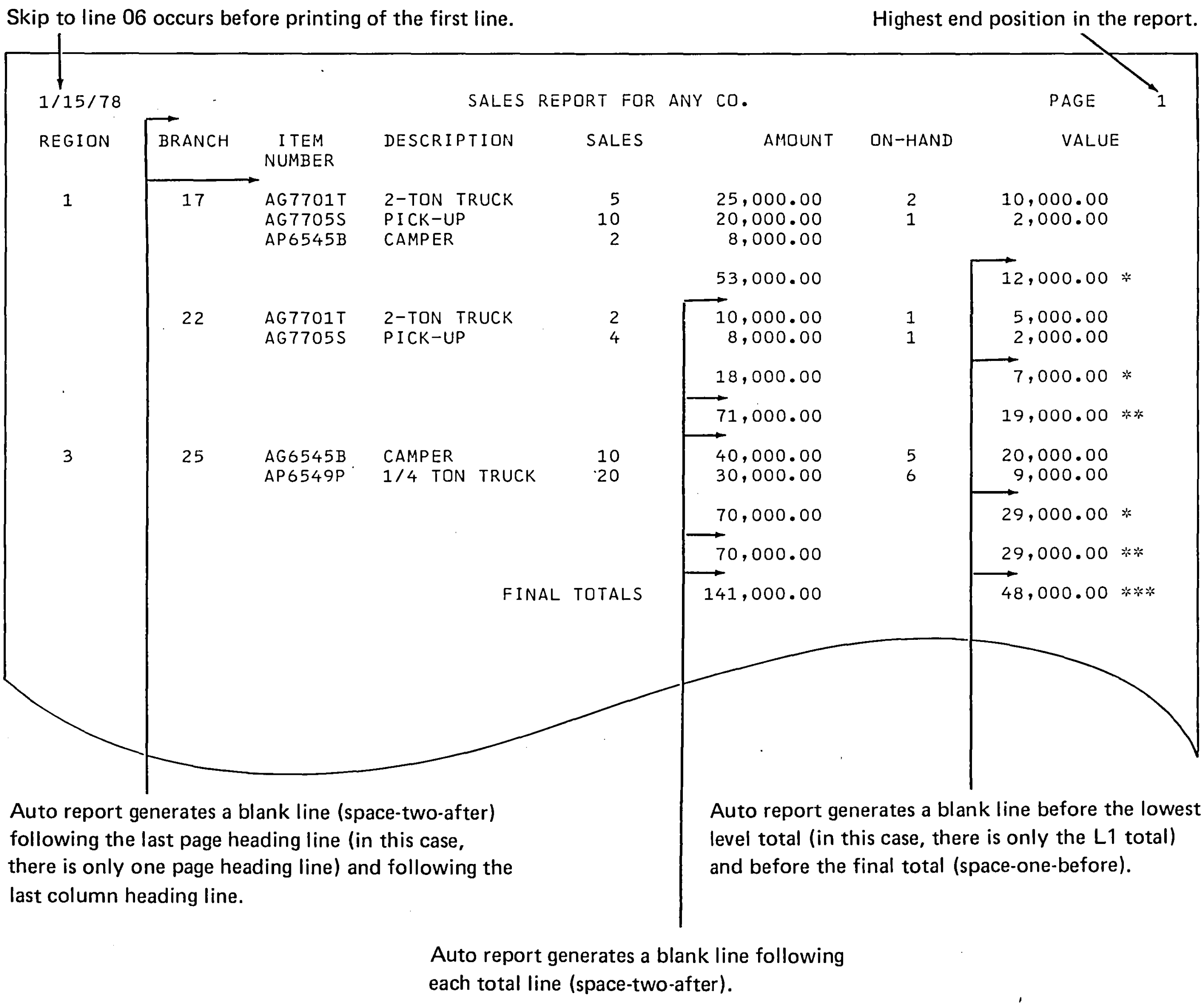
Placement of Headings and Fields

Auto report generates end positions for fields and constants and centers column headings, columns, and report lines (see Figure 15-17 for an example). However, if an end position is specified for a field or constant on a D/T-*AUTO field description line, that end position is used on all column heading, detail, and total specifications generated from the field description. (The specified end position may be altered slightly by auto report when the line is centered or when the column heading and field are positioned in relation to each other.) If the specified end position causes an overlay with a previous field or constant, auto report generates a new end position.

Specify end positions only to eliminate the automatic spacing between fields or to spread out or expand a report on the page.

Page Headings

If the date and page number are printed on the first *AUTO page heading line (that is, if they are not suppressed by an N in column 27 of the option specifications sheet), the date is always printed in positions 1 through 8. The page number is printed with an end position equal to the highest end position of the longest line in the report. When the first *AUTO page heading (including date, title, and page number) is the longest line in the report, one blank space separates the title from the date and the word PAGE from the title. If the resulting line exceeds the record length of the printer file, the excess information on the right of the line is not printed.



Skip to line 06 occurs before printing of the first line.

Highest end position in the report.

```
1/15/78                   SALES REPORT FOR ANY CO.                          PAGE    1

REGION   BRANCH   ITEM     DESCRIPTION     SALES        AMOUNT   ON-HAND        VALUE
                  NUMBER

   1       17     AG7701T  2-TON TRUCK       5       25,000.00      2       10,000.00
                  AG7705S  PICK-UP          10       20,000.00      1        2,000.00
                  AP6545B  CAMPER            2        8,000.00

                                                     53,000.00              12,000.00 *

           22     AG7701T  2-TON TRUCK       2       10,000.00      1        5,000.00
                  AG7705S  PICK-UP           4        8,000.00      1        2,000.00

                                                     18,000.00               7,000.00 *

                                                     71,000.00              19,000.00 **

   3       25     AG6545B  CAMPER           10       40,000.00      5       20,000.00
                  AP6549P  1/4 TON TRUCK    20       30,000.00      6        9,000.00

                                                     70,000.00              29,000.00 *

                                                     70,000.00              29,000.00 **

                                  FINAL TOTALS      141,000.00              48,000.00 ***
```

Auto report generates a blank line (space-two-after) following the last page heading line (in this case, there is only one page heading line) and following the last column heading line.

Auto report generates a blank line before the lowest level total (in this case, there is only the L1 total) and before the final total (space-one-before).

Auto report generates a blank line following each total line (space-two-after).

Figure 15-17. Report Illustrating Format Generated by the Auto Report

If a line generated from a D/T-*AUTO specification is the longest report line, that line is printed starting in print position 1, and the title portion of the first page heading line is centered in relation to that line. Additional *AUTO page headings are then centered on the first *AUTO page heading line.

If an *AUTO page heading is the longest line in the report and a D/T-*AUTO specification is present, any other *AUTO page heading lines and the line generated from the D/T-*AUTO specification are centered on the longest page heading.

Fields and constants appear in the order specified in the *AUTO output specifications from left to right. Auto report provides one blank space before and after fields on the heading line. No spacing is provided between constants.

*Reformatting *AUTO Page Headings*

You can reformat an *AUTO page heading line if you do not want to use the end positions for fields and constants that are generated by auto report. If you want to find what end positions are generated for page, date, and title information, see the listing of the generated source program that is produced by the RPG II compiler (see *Generated Specifications*).

Catalog the generated RPG II source program in a library by specifying the C option in column 7 of the auto report option specifications, and change the end positions on the generated source statements by using the library maintenance program or the source entry utility.

Body of the Report

Placement of column headings above columns depends on which is longer, the heading or the associated field (including edit characters). If any column heading is longer than the associated field, the field is centered under the longest column heading constant. If, however, the field is longer than the longest column heading constant, the column heading is left-justified over an alphameric field and right-justified over a numeric field. When more than one column heading line is specified, shorter column headings are always centered on the longest column heading.

Fields and constants appear from left to right on a line in the order they are specified by the output specifications. At least two blank spaces appear before each field on the line. However, no spaces are provided before a constant; the programmer must incorporate blanks within constants to provide for additional spacing.

Total indication information (fields and constants specified with 1 through 9 or R in column 39) is placed to the left of the first total field (A in column 39) on the corresponding total line, followed by two spaces. If two or more such fields or constants are specified for a total line, they appear from left to right in the order specified on the left of the first total on the line. Each field is preceded and followed by one space. No spacing is provided for constants.

*Overflow of the D/T-*AUTO Print Lines*

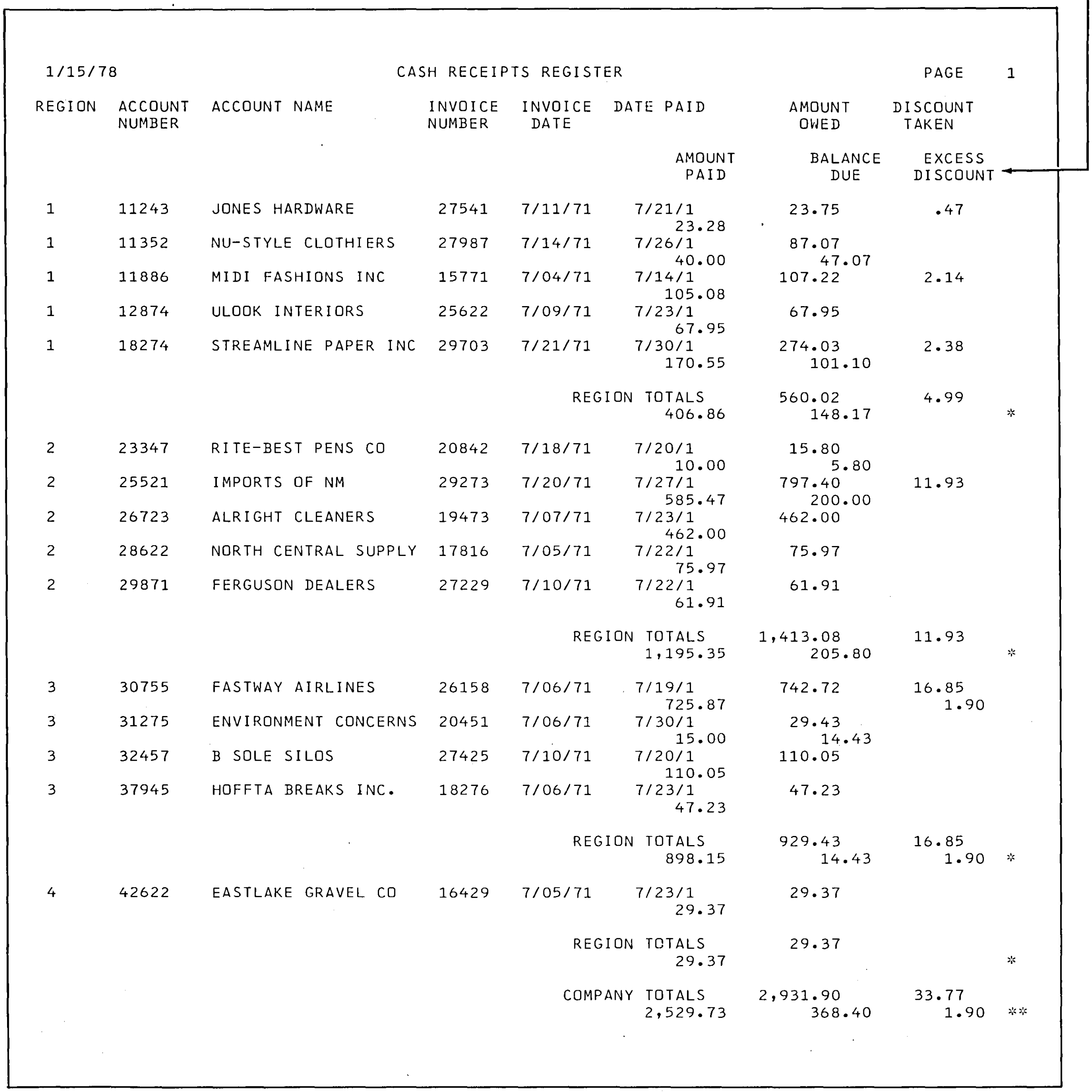
If the lines generated from a D/T-*AUTO specification are longer than the record length specified for the printer file, a second print line (overflow line) is generated for each column heading line, detail (or group print) line, and total line. (Remember, a second print line is not generated for *AUTO page heading lines.) The excess information is placed on the overflow line in the order specified, right-justified.

Figure 15-18 shows the result of an overflow condition.

In the output specifications for the report shown in Figure 15-18, no spacing or skipping is specified. If spacing and skipping were specified, however, auto report spaces the report as follows:

- Column heading lines and total lines are spaced as shown in Figure 15-18.
- The space-before and skip-before entries specified are for the original detail (or group print) line. Auto report generates space-one-after for this line.
- The space-after and skip-after entries specified are for the overflow line. Auto report generates blanks for space-before and skip-before for the overflow line.

Auto report prints those columns that cannot be completely contained on the original line on overflow lines.



```
1/15/78                        CASH RECEIPTS REGISTER                          PAGE   1

REGION  ACCOUNT  ACCOUNT NAME         INVOICE  INVOICE  DATE PAID        AMOUNT    DISCOUNT
        NUMBER                        NUMBER    DATE                      OWED      TAKEN

                                                              AMOUNT       BALANCE    EXCESS
                                                               PAID          DUE     DISCOUNT

  1     11243    JONES HARDWARE        27541   7/11/71     7/21/1          23.75        .47
                                                               23.28
  1     11352    NU-STYLE CLOTHIERS    27987   7/14/71     7/26/1          87.07
                                                               40.00          47.07
  1     11886    MIDI FASHIONS INC     15771   7/04/71     7/14/1         107.22       2.14
                                                              105.08
  1     12874    ULOOK INTERIORS       25622   7/09/71     7/23/1          67.95
                                                               67.95
  1     18274    STREAMLINE PAPER INC  29703   7/21/71     7/30/1         274.03       2.38
                                                              170.55         101.10

                                              REGION TOTALS               560.02       4.99
                                                              406.86         148.17           *

  2     23347    RITE-BEST PENS CO     20842   7/18/71     7/20/1          15.80
                                                               10.00           5.80
  2     25521    IMPORTS OF NM         29273   7/20/71     7/27/1         797.40      11.93
                                                              585.47         200.00
  2     26723    ALRIGHT CLEANERS      19473   7/07/71     7/23/1         462.00
                                                              462.00
  2     28622    NORTH CENTRAL SUPPLY  17816   7/05/71     7/22/1          75.97
                                                               75.97
  2     29871    FERGUSON DEALERS      27229   7/10/71     7/22/1          61.91
                                                               61.91

                                              REGION TOTALS             1,413.08      11.93
                                                            1,195.35         205.80           *

  3     30755    FASTWAY AIRLINES      26158   7/06/71     7/19/1         742.72      16.85
                                                              725.87                    1.90
  3     31275    ENVIRONMENT CONCERNS  20451   7/06/71     7/30/1          29.43
                                                               15.00          14.43
  3     32457    B SOLE SILOS          27425   7/10/71     7/20/1         110.05
                                                              110.05
  3     37945    HOFFTA BREAKS INC.    18276   7/06/71     7/23/1          47.23
                                                               47.23

                                              REGION TOTALS               929.43      16.85
                                                              898.15          14.43     1.90  *

  4     42622    EASTLAKE GRAVEL CO    16429   7/05/71     7/23/1          29.37
                                                               29.37

                                              REGION TOTALS                29.37
                                                               29.37                          *

                                             COMPANY TOTALS             2,931.90      33.77
                                                            2,529.73         368.40     1.90  **
```

Figure 15-18. Report Illustrating Overflow of D-*AUTO Print Lines

GENERATED SPECIFICATIONS

Standard RPG II specifications are generated by auto report and are combined with RPG II specifications included in the input to auto report and specifications copied from the library to produce the final RPG II source program. This section describes the generated RPG II specifications and the order of those specifications in the RPG II source program.

Figures 15-19 and 15-20 show auto report specifications for a sales report and the resulting RPG II source specifications that are generated for the report. Numbers are inserted in the figures to identify the auto report functions and to show the specifications that are generated by each function.

Generated Calculations

Calculations are generated to accumulate totals for fields named on *AUTO field description specifications that have an A in column 39 (see Figure 15-21).

An RPG II subroutine is generated to accumulate the values from these fields into the lowest-level generated total fields. The name of the subroutine is always A$$SUM. The subroutine specifications are conditioned differently, depending on whether detail or group printing is specified:

- If detail printing is specified, as in Figure 15-21, the EXSR statement is conditioned by the same indicator(s) that conditions the D-*AUTO specification (01 in this example). Each ADD statement in the subroutine is conditioned by the field indicator(s) specified with the field in its field description specification (none in this example).
- If group printing is specified, the EXSR statement and all ADD statements in the subroutine are unconditioned.

Total calculations are generated to roll the total from the lowest-level defined total field through the higher-level defined total fields and the final total. The total calculation to add the total from one level to that of the next higher level is conditioned by the control level indicator corresponding to the field name of the lower level. As shown in Figure 15-21, total calculations to accumulate L2 and LR totals are followed by the subroutine to accumulate the lowest level total, L1.

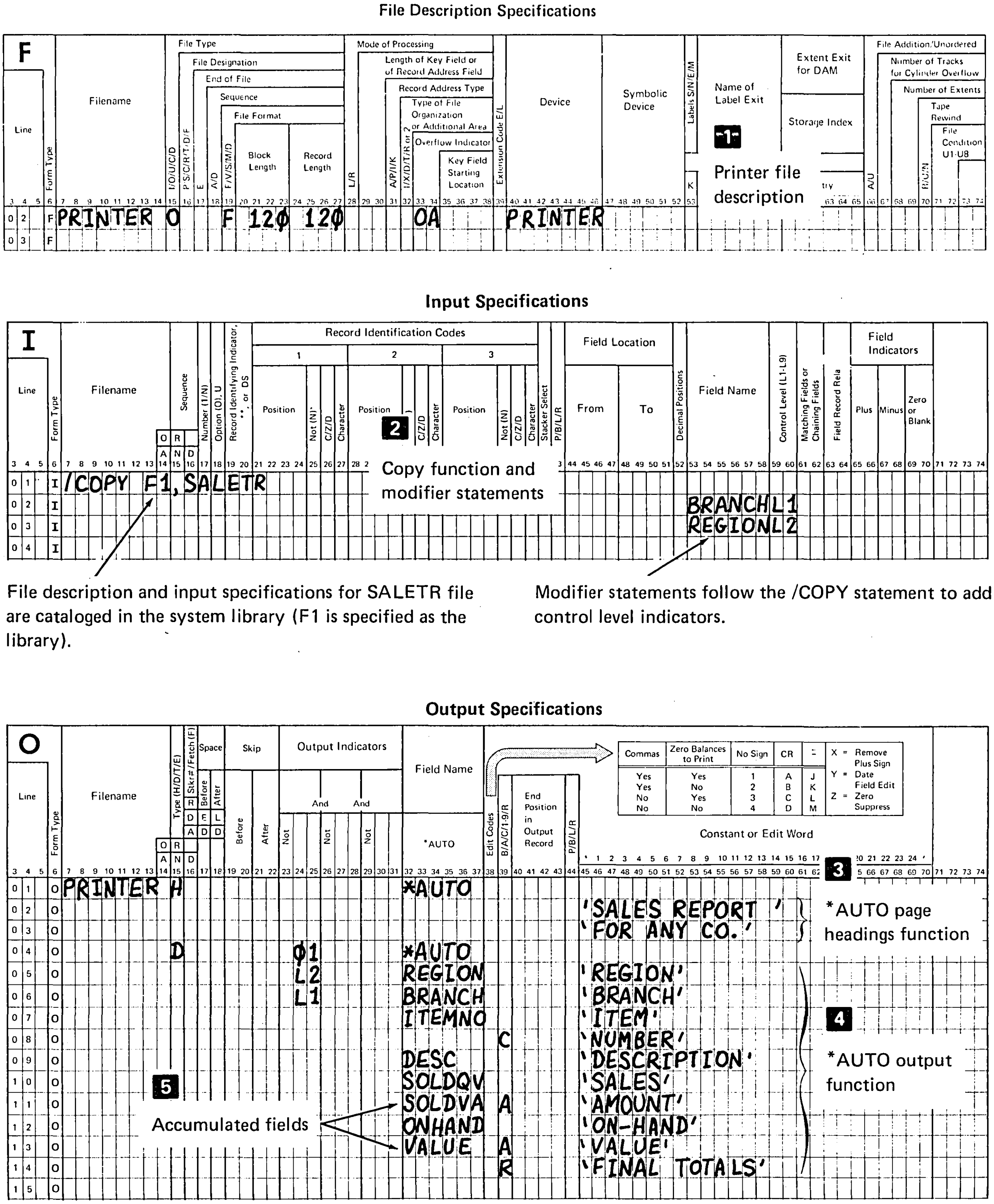
Generated total fields are defined (given length and number of decimal positions) when the total field is the result field in a generated calculation. In the input specifications, SOLDVA and VALUE are numeric fields defined with a length of seven and two decimal positions. Figure 15-21 shows that the total fields generated from SOLDVA and VALUE are defined as two positions longer than the original fields, with the same number of decimal positions.

When group printing is specified (T-*AUTO specification), auto report generates total calculations to reset each of the accumulated fields (A in column 39) on the lowest level total line to zero on each cycle. A Z-ADD calculation, conditioned by L0, is generated for each accumulated field. These calculations are the first total calculations in the generated RPG II source program.

Generated Output Specifications

Figure 15-22 shows the output specifications generated by auto report. To identify specifications supplied by auto report (column heading specifications, total specifications, conditioning indicators, spacing and skipping values, end position values, blank after) compare the listing with the auto report specifications.

Auto report generates specifications to reset accumulated fields to zero after they are printed. See *Field Description (A in Column 39)* for a discussion of resetting fields to zero. In this example, blank after is generated for accumulated fields.



File Description Specifications

```
Line  Form  Filename   I/O  ...  F  Block  Record  ...  Device
02    F     PRINTER    O         F  12Ø    12Ø     OA   PRINTER
03    F
```

Printer file description

Input Specifications

```
Line  Form  Filename                         Field Name  Control Level
01    I     /COPY F1,SALETR
02    I                                      BRANCH      L1
03    I                                      REGION      L2
04    I
```

Copy function and modifier statements

File description and input specifications for SALETR file are cataloged in the system library (F1 is specified as the library).

Modifier statements follow the /COPY statement to add control level indicators.

Output Specifications

```
Line  Form  Filename  Type  Output      Field Name  Edit   B/A/C/1-9/R  Constant or Edit Word
                            Indicators              Codes
01    O     PRINTER   H                 *AUTO
02    O                                                                 'SALES REPORT '
03    O                                                                 'FOR ANY CO.'
04    O               D     Ø1          *AUTO
05    O                     L2          REGION                          'REGION'
06    O                     L1          BRANCH                          'BRANCH'
07    O                                 ITEMNO                          'ITEM'
08    O                                             C                   'NUMBER'
09    O                                 DESC                            'DESCRIPTION'
10    O                                 SOLDQV                          'SALES'
11    O                                 SOLDVA             A            'AMOUNT'
12    O                                 ONHAND                          'ON-HAND'
13    O                                 VALUE              A            'VALUE'
14    O                                                    R            'FINAL TOTALS'
15    O
```

*AUTO page headings function

*AUTO output function

Accumulated fields

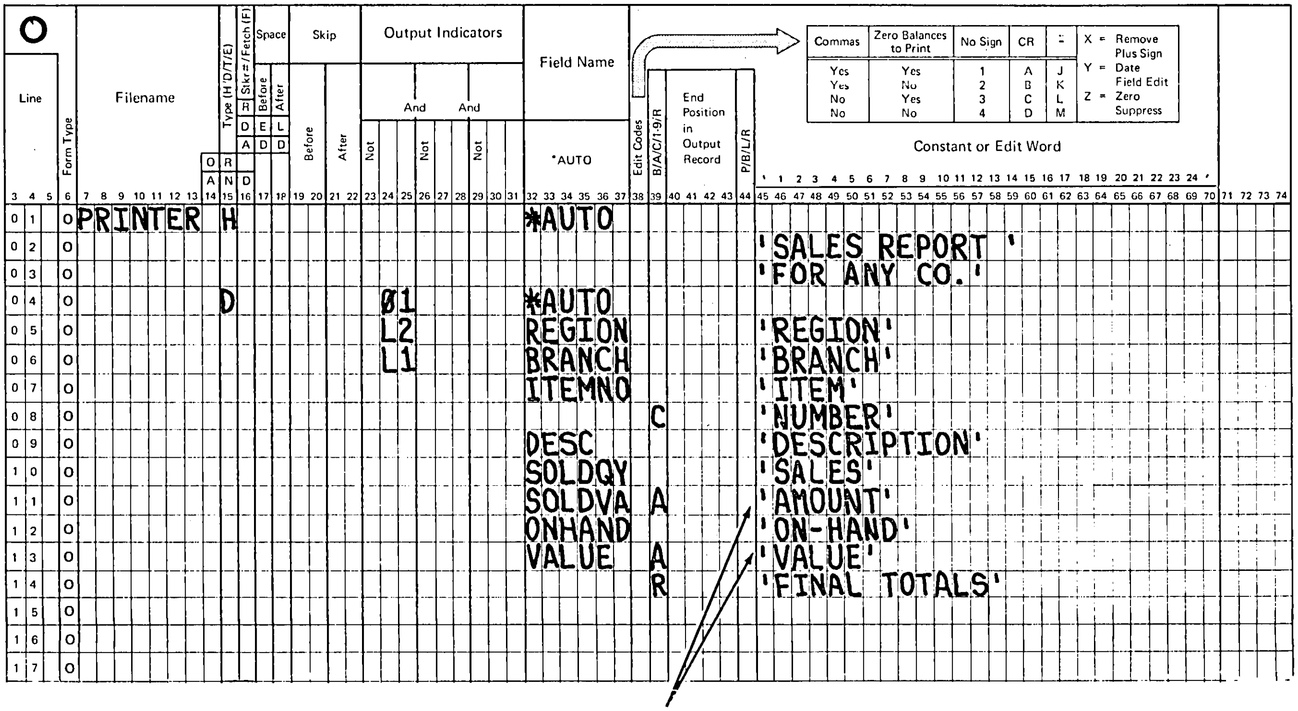
Figure 15-19. Auto Report Specifications for a Sales Transaction Report

If you do not specify a control specification, auto report generates an all-blank control specification for you.

```
                0010 H
     RG 004
1 --> 0001      0020 FPRINTER O   F 120 120      OA     PRINTER
      0002      0030CFSALES   IP  F 473  43             DISK
                0040 I*/COPY F1,SALETR
      0003      0050CISALES   AA  01
      0004      0060CI                                      1    7 ITEMNO
      0005      0070MI                                      8    9 BRANCHL1
2     0006      0080MI                                     10   10 REGIONL2
      0007      0090CI                                     11   25 DESC
      0008      0100CI                                     26   270SOLDQY
      0009      0110CI                                     28   342SOLDVA
      0010      0120CI                                     35   360ONHAND
      0011      0130CI                                     37   432VALUE
      0012      0140EC   01                 EXSR A$$SUM
      0013      0150ECL1          SOLDV2    ADD  SOLDV1    SOLDV2   92
      0014      0160ECL1          VALUE2    ADD  VALUE1    VALUE2   92
      0015      0170ECL2          SOLDVR    ADD  SOLDV2    SOLDVR   92
5     0016      0180ECL2          VALUER    ADD  VALUE2    VALUER   92
      0017      0190ECSR          A$$SUM    BEGSR
      0018      0200ECSR          SOLDV1    ADD  SOLDVA    SOLDV1   92
      0019      0210ECSR          VALUE1    ADD  VALUE     VALUE1   92
      0020      0220ECSR                    ENDSR
      0021      0230EOPRINTER H   206   1P
      0022      0240EO       OR         OA
      0023      0250EO                                    45 'SALES REPORT '
3     0024      0260EO                                    56 'FOR ANY CO.'
      0025      0270EO                          UDATE Y    8
      0026      0280EO                          PAGE  Z   89
      0027      0290EO                                    85 'PAGE '
      0028      0300EOPRINTER H   1     1P
      0029      0310EO       OR         OA
      0030      0320EO                                     6 'REGION'
      0031      0330EO                                    14 'BRANCH'
      0032      0340EO                                    21 'ITEM'
      0033      0350EO                                    36 'DESCRIPTION'
      0034      0360EO                                    47 'SALES'
      0035      0370EO                                    62 'AMOUNT'
      0036      0380EO                                    71 'ON-HAND'
      0037      0390EO                                    86 'VALUE'
      0038      0400EOPRINTER H   2     1P
      0039      0410EO       OR         OA
      0040      0420EO                                    22 'NUMBER'
      0041      0430EOPRINTER D   1     01
      0042      0440EO                  L2      REGION     3
      0043      0450EO                  L1      BRANCH    12
      0044      0460EO                          ITEMNO    23
4     0045      0470EO                          DESC      40
      0046      0480EO                          SOLDQYK   46
      0047      0490EO                          SOLDVAKB  62
      0048      0500EO                          ONHANDK   69
      0049      0510EO                          VALUE KB  86
      0050      0520EOPRINTER T  12     L1
      0051      0530EO                          SOLDV1KB  62
      0052      0540EO                          VALUE1KB  86
      0053      0550EO                                    87 '*'
      0054      0560EOPRINTER T   2     L2
      0055      0570EO                          SOLDV2KB  62
      0056      0580EO                          VALUE2KB  86
      0057      0590EO                                    88 '**'
      0058      0600EOPRINTER T  12     LR
      0059      0610EO                          SOLDVRKB  62
      0060      0620EO                          VALUERKB  86
      0061      0630EO                                    47 'FINAL TOTALS'
      0062      0640EO                                    89 '***'
```

Figure 15-20. RPG II Source Program Generated from Auto Report Specifications

Output Specifications



Commas	Zero Balances to Print	No Sign	CR	-
Yes	Yes	1	A	J
Yes	No	2	B	K
No	Yes	3	C	L
No	No	4	D	M

X = Remove Plus Sign
Y = Date Field Edit
Z = Zero Suppress

Line	Form Type	Filename	Type (H/D/T/E)	Output Indicators	Field Name	Edit Codes	B/A/C/1-9/R	End Position in Output Record	Constant or Edit Word
01	O	PRINTER	H		*AUTO				
02	O								'SALES REPORT '
03	O								'FOR ANY CO.'
04	O		D	Ø1	*AUTO				
05	O			L2	REGION				'REGION'
06	O			L1	BRANCH				'BRANCH'
07	O				ITEMNO				'ITEM'
08	O						C		'NUMBER'
09	O				DESC				'DESCRIPTION'
10	O				SOLDQY				'SALES'
11	O				SOLDVA		A		'AMOUNT'
12	O				ONHAND				'ON-HAND'
13	O				VALUE		A		'VALUE'
14	O						R		'FINAL TOTALS'
15	O								
16	O								
17	O								

Calculations are generated for fields with an A in column in 39.

Total calculations roll higher level totals.

```
0012   0140EC    01          EXSR  A$$SUM
0013   0150ECL1    SOLDV2    ADD   SOLDV1     SOLDV2    92
0014   0160ECL1    VALUE2    ADD   VALUE1     VALUE2    92
0015   0170ECL2    SOLDVR    ADD   SOLDV2     SOLDVR    92
0016   0180ECL2    VALUER    ADD   VALUE2     VALUER    92
0017   0190ECSR    A$$SUM    BEGSR
0018   0200ECSR    SOLDV1    ADD   SOLDVA     SOLDV1    92
0019   0210ECSR    VALUE1    ADD   VALUE      VALUE1    92
0020   0220ECSR              ENDSR
```

Subroutine accumulates the lowest level totals (L1, in this example).

Length and decimal position of generated total fields.

Note: Placement of the generated calculations in the RPG II source program is shown in Figure 15-20.

Figure 15-21. Calculations Generated from Auto Report Coding for Sales Transaction Report

Output Specifications

Commas	Zero Balances to Print	No Sign	CR	–
Yes	Yes	1	A	J
Yes	No	2	B	K
No	Yes	3	C	L
No	No	4	D	M

X = Remove Plus Sign
Y = Date Field Edit
Z = Zero Suppress

Line	Form Type	Filename	Type (H/D/T/E)	Space Before	Space After	Skip Before	Skip After	Output Indicators	Field Name / *AUTO	Edit Codes	B/A/C/1-9/R	End Position in Output Record	P/B/L/R	Constant or Edit Word
01	O	PRINTER	H						*AUTO					
02	O													'SALES REPORT '
03	O													'FOR ANY CO.'
04	O		D					Ø1	*AUTO					
05	O							L2	REGION					'REGION'
06	O							L1	BRANCH					'BRANCH'
07	O								ITEMNO					'ITEM'
08	O										C			'NUMBER'
09	O								DESC					'DESCRIPTION'
10	O								SOLDQY					'SALES'
11	O								SOLDVA		A			'AMOUNT'
12	O								ONHAND					'ON-HAND'
13	O								VALUE		A			'VALUE'
14	O										R			'FINAL TOTALS'
15	O													

```
0021  0230EOPRINTER H  206   1P
0022  0240EO       OR        0A
0023  0250EO                                  45 'SALES REPORT '
0024  0260EO                                  56 'FOR ANY CO.'
0025  0270EO                         UDATE Y   8
0026  0280EO                         PAGE  Z  89
0027  0290EO                                  85 'PAGE '
0028  0300EOPRINTER H  1      1P
0029  0310EO       OR         0A
0030  0320EO                                   6 'REGION'
0031  0330EO                                  14 'BRANCH'
0032  0340EO                                  21 'ITEM'
0033  0350EO                                  36 'DESCRIPTION'
0034  0360EO                                  47 'SALES'
0035  0370EO                                  62 'AMOUNT'
0036  0380EO                                  71 'ON-HAND'
0037  0390EO                                  86 'VALUE'
0038  0400EOPRINTER H  2      1P
0039  0410EO       OR         0A
0040  0420EO                                  22 'NUMBER'
0041  0430EOPRINTER D  1      01
0042  0440EO                 L2      REGION    3
0043  0450EO                 L1      BRANCH   12
0044  0460EO                         ITEMNO   23
0045  0470EO                         DESC     40
0046  0480EO                         SOLDQYK  46
0047  0490EO                         SOLDVAKB 62
0048  0500EO                         ONHANDK  69
0049  0510EO                         VALUE KB 86
0050  0520EOPRINTER T 12      L1
0051  0530EO                         SOLDV1KH 62
0052  0540EO                         VALUE1KB 86
0053  0550EO                                  87 '*'
0054  0560EOPRINTER T  2      L2
0055  0570EO                         SOLDV2KB 62
0056  0580EO                         VALUE2KB 86
0057  0590EO                                  88 '**'
0058  0600EOPRINTER T 12      LR
0059  0610EO                         SOLDVRKB 62
0060  0620EO                         VALUERKB 86
0061  0630EO                                  47 'FINAL TOTALS'
0062  0640EO                                  89 '***'
```

Two heading specifications are generated for column headings because ITEM NUMBER is a two-line heading.

Auto report generates total specifications to print accumulated totals for SOLDVA and VALUE fields.

Figure 15-22. Output Specifications Generated from Auto Report Coding for Sales Transaction Report

PROGRAMMING AIDS

The chart shown in Figure 15-23 should be helpful in determining valid *AUTO output entries depending on the contents of column 39.

The following programming suggestions may be helpful in specific programming situations:

- One column heading can be printed over two or more fields if automatic column spacing is taken into consideration. For example, if the heading DATE is to print over a month field and a day field as follows:

```
 D A   T  E
  MON   DAY

   XX    XX
   XX    XX
```

Code the output specifications as follows:

Output Specifications

Line	Form Type	Filename	Field Name (32-37)	B/A/C/1-9/R (39)	End Position in Output Record (40-43)	Constant or Edit Word (45-70)
01	O					
02	O		MONTH			'D A'
03	O			C		'MON'
04	O		DAY			'T E'
05	O			C		'DAY'
06	O					
07	O					

Commas	Zero Balances to Print	No Sign	CR	-
Yes	Yes	1	A	J
Yes	No	2	B	K
No	Yes	3	C	L
No	No	4	D	M

X = Remove Plus Sign
Y = Date Field Edit
Z = Zero Suppress

- To print a constant on only the first detail line under a column heading, move the constant to a field in calculation specifications and print that field as shown in Figure 15-24.

- If group printing is being done and more than one record type is present in the input file, certain precautions must be taken. If a field to be accumulated is present in all record types, but only one record type is to be processed, the correct total is not generated unless additional coding is used. The specifications shown in Figure 15-25 give incorrect results because the T-*AUTO specification causes an unconditioned ADD subroutine to be generated if a field is to be added. Therefore, QTY is added when indicator 10 is on and when indicator 11 or 12 is on. Figure 15-26 shows a method of obtaining the correct results.

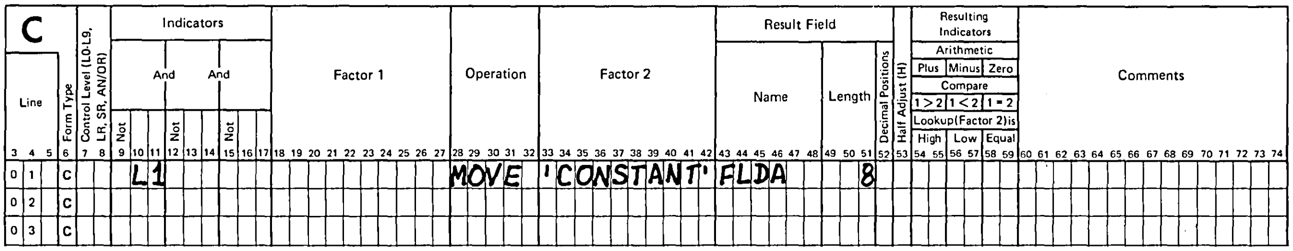
- Figure 15-27 shows the specifications for counting records. This method is especially applicable when you want to print a detail list, to take totals by control level, or to eliminate 1's from being listed down the page.

39	7-22	23-31	32-37	38	40-43	44	45-70
Blank	Blank	Blank or indicators	Field name	Blank or edit code	Blank or end position	Blank	Blank or column heading
	Blank	Blank or indicators	Blank	Blank	Blank or end position	Blank	Literal
B	Blank	Blank or indicators	Field name	Blank or edit code	Blank or end position	Blank	Blank or column heading
A	Blank	Blank or indicators	Field name	Blank or edit code	Blank or end position	Blank	Blank or column heading
C	Blank	Blank	Blank	Blank	Blank	Blank	Column heading
1-9, R	Blank	Blank	Field name	Blank or edit code	Blank	Blank	Blank or edit word
	Blank	Blank	Blank	Blank	Blank	Blank	Literal

Figure 15-23. Valid *AUTO Entries Depending on the Contents of Column 39

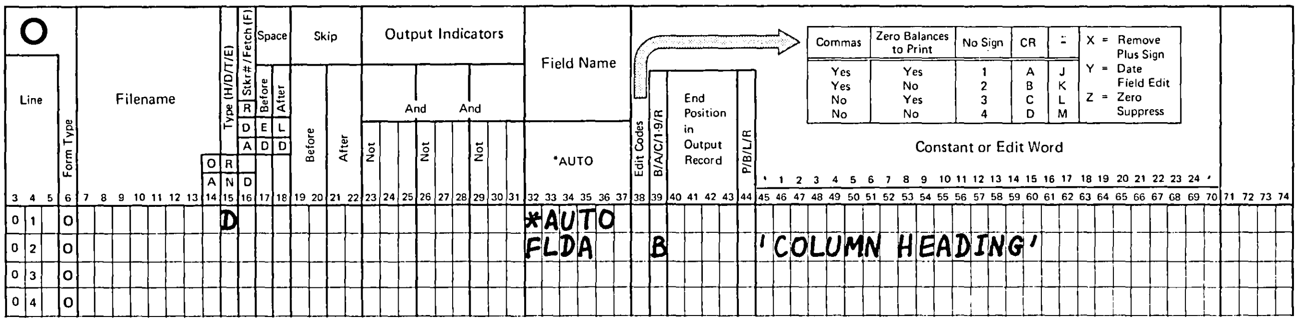
Assume L1 is defined in columns 59 and 60 on input specifications.

Calculation Specifications



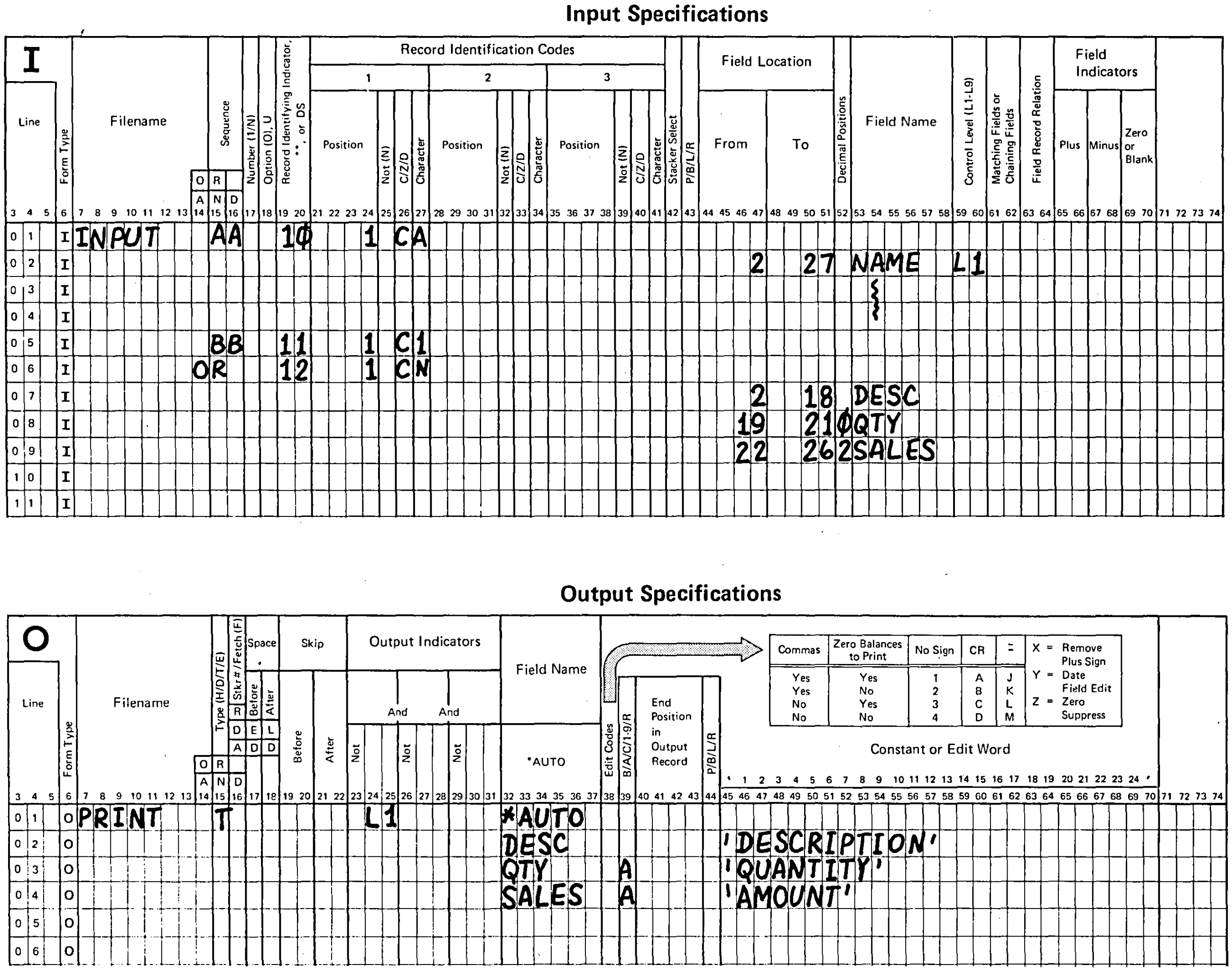
Line	Form Type	Indicators	Operation	Factor 2	Result Field Name	Length
01	C	L1	MOVE	'CONSTANT'	FLDA	8
02	C					
03	C					

Output Specifications



Line	Form Type	Type (H/D/T/E)	Field Name	B/A/C/1-9/R	Constant or Edit Word
01	O	D	*AUTO		
02	O		FLDA	B	'COLUMN HEADING'
03	O				
04	O				

Figure 15-24. Printing a Constant Only on the First Detail Line



Input Specifications

Line	Form Type	Filename	Sequence	Number (1/N)	Option (O), U	Record Identifying Indicator, ** or DS	Position (1)	Not (N)	C/Z/D	Character	Field Location From	To	Decimal Positions	Field Name	Control Level (L1-L9)
01	I	INPUT	AA			1Ø	1		C	A					
02	I										2	27		NAME	L1
03	I														
04	I														
05	I		BB			11	1		C	1					
06	I	OR				12	1		C	N					
07	I										2	18		DESC	
08	I										19	21	Ø	QTY	
09	I										22	26	2	SALES	
10	I														
11	I														

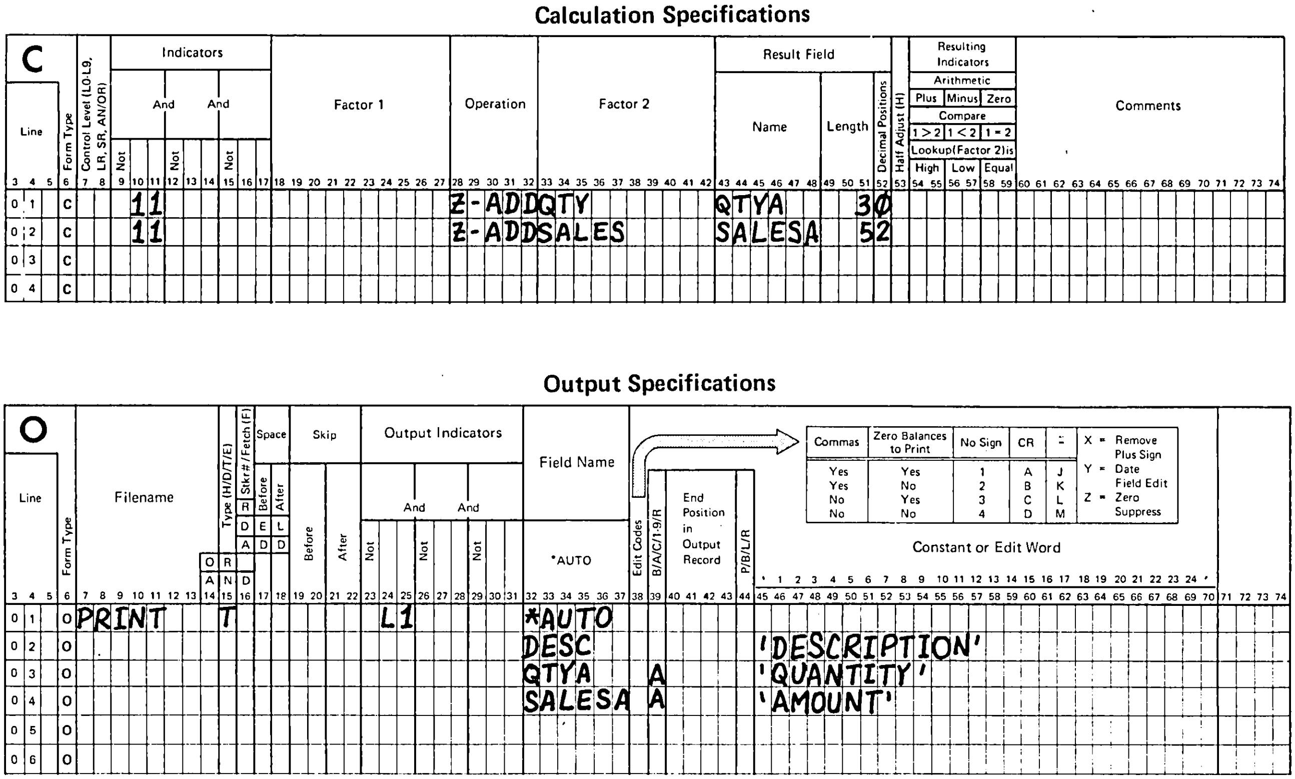
Output Specifications

Line	Form Type	Filename	Type (H/D/T/E)	Space Before	Space After	Skip Before	Skip After	Output Indicators	Field Name / *AUTO	Edit Codes	B/A/C/1-9/R	End Position in Output Record	P/B/L/R	Constant or Edit Word
01	O	PRINT	T					L1	*AUTO					
02	O								DESC					'DESCRIPTION'
03	O								QTY		A			'QUANTITY'
04	O								SALES		A			'AMOUNT'
05	O													
06	O													

Commas	Zero Balances to Print	No Sign	CR	-
Yes	Yes	1	A	J
Yes	No	2	B	K
No	Yes	3	C	L
No	No	4	D	M

X = Remove Plus Sign
Y = Date Field Edit
Z = Zero Suppress

Figure 15-25. Incorrect *AUTO Specifications for More Than One Record Type



Calculation Specifications

Line	Form Type	Control Level (L0-L9, LR, SR, AN/OR)	Indicators	Factor 1	Operation	Factor 2	Result Field Name	Length	Decimal Positions	Half Adjust (H)	Resulting Indicators	Comments
01	C		11		Z-ADD	QTY	QTYA	3	Ø			
02	C		11		Z-ADD	SALES	SALESA	5	2			
03	C											
04	C											

Output Specifications

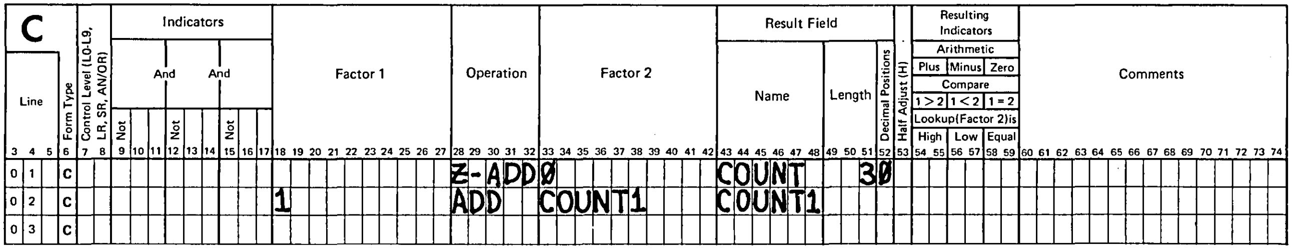
Line	Form Type	Filename	Type (H/D/T/E)	Stkr #/Fetch (F)	Space Before	Space After	Skip Before	Skip After	Output Indicators	Field Name / *AUTO	Edit Codes	B/A/C/1-9/R	End Position in Output Record	P/B/L/R	Constant or Edit Word
01	O	PRINT	T						L1	*AUTO					
02	O									DESC					'DESCRIPTION'
03	O									QTYA		A			'QUANTITY'
04	O									SALESA		A			'AMOUNT'
05	O														
06	O														

Commas	Zero Balances to Print	No Sign	CR	-
Yes	Yes	1	A	J
Yes	No	2	B	K
No	Yes	3	C	L
No	No	4	D	M

X = Remove Plus Sign
Y = Date Field Edit
Z = Zero Suppress

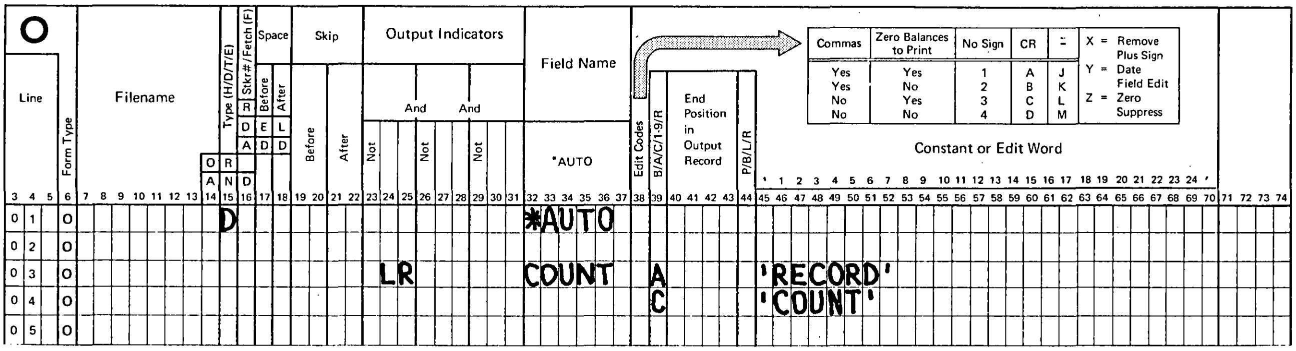
Figure 15-26. Correct *AUTO Specifications for More Than One Record Type

Calculation Specifications



Line	Form Type	Control Level	Indicators	Factor 1	Operation	Factor 2	Result Field Name	Length	Decimal Positions	Half Adjust (H)	Resulting Indicators	Comments
01	C				Z-ADD	Ø	COUNT	3	Ø			
02	C			1	ADD	COUNT1	COUNT1					
03	C											

Output Specifications



Line	Form Type	Filename	Type (H/D/T/E)	Space	Skip	Output Indicators	Field Name	Edit Codes	B/A/C/1-9/R	End Position in Output Record	P/B/L/R	Constant or Edit Word
01	O		D				*AUTO					
02	O											
03	O					LR	COUNT		A			'RECORD'
04	O								C			'COUNT'
05	O											

Calculation Specifications

Line 01 This instruction is needed only to define the field COUNT for accumulation.

Line 02 This instruction accumulates the total for the first control level.

Output Specifications

Line 03 This instruction causes the generation of calculation and output specifications for the detail and total lines. The LR conditioning indicator prevents the generated detail calculation from occurring. It also prevents printing at detail time.

Note: If no control levels are specified in the program, a 1 is added to COUNTR rather than to COUNT1 on the calculation specifications.

Figure 15-27. Method of Using *AUTO for Counting Records

EXAMPLES OF USING AUTO REPORT

Examples 1 through 4 explain how auto report is used to generate report page headings and such output specifications as column headings, detail lines, and total lines. Examples 5 and 6 illustrate the use of the auto report copy function to copy specifications from the library and to modify copied specifications for a particular job.

EXAMPLE 1

***AUTO Page Headings**

***AUTO Output**

Problem

Produce the sales report shown below using the *AUTO page headings and *AUTO output functions of auto report.

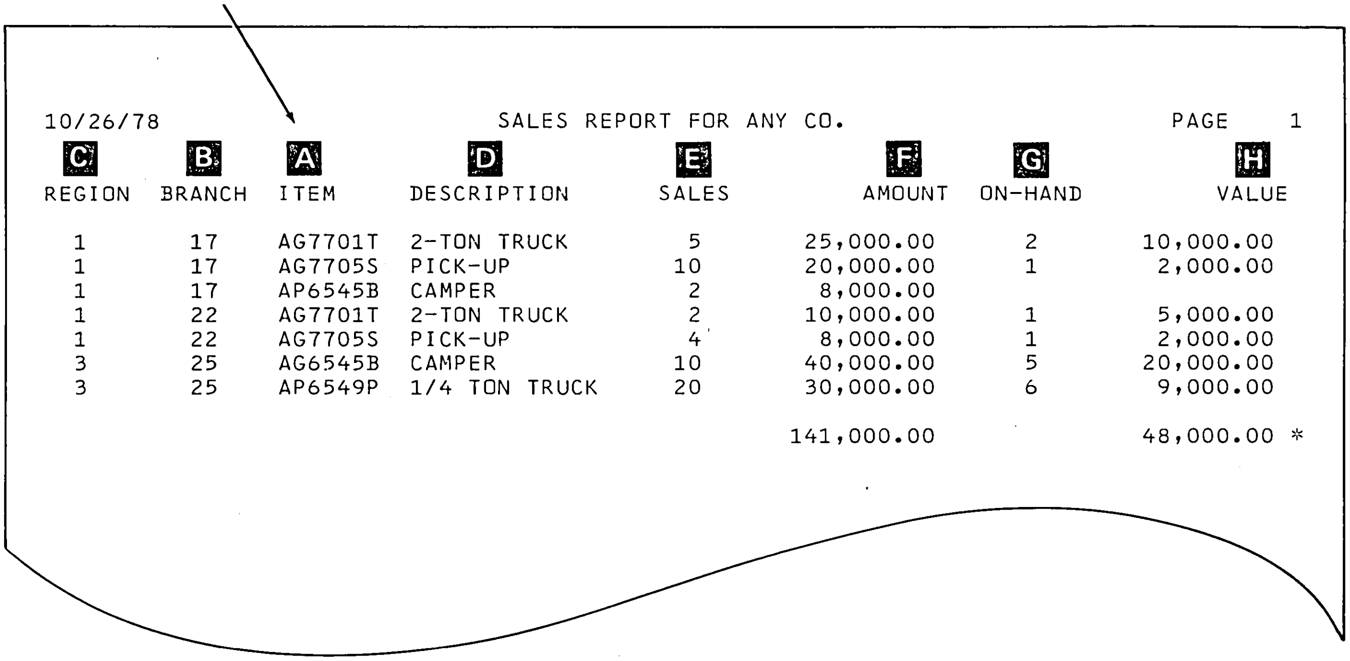
Procedure

1. Code normal RPG II file description and input specifications for the job.

2. Code *AUTO page headings to produce a one-line page heading that includes date and page number.

3. Code *AUTO output to produce one-line column headings, detail report lines, and final totals.

Letters refer to fields on the following page.



```
10/26/78                  SALES REPORT FOR ANY CO.                  PAGE    1
   C        B        A           D             E           F        G          H
REGION  BRANCH  ITEM     DESCRIPTION      SALES      AMOUNT  ON-HAND      VALUE

   1       17   AG7701T  2-TON TRUCK        5     25,000.00     2      10,000.00
   1       17   AG7705S  PICK-UP           10     20,000.00     1       2,000.00
   1       17   AP6545B  CAMPER             2      8,000.00
   1       22   AG7701T  2-TON TRUCK        2     10,000.00     1       5,000.00
   1       22   AG7705S  PICK-UP            4      8,000.00     1       2,000.00
   3       25   AG6545B  CAMPER            10     40,000.00     5      20,000.00
   3       25   AP6549P  1/4 TON TRUCK     20     30,000.00     6       9,000.00

                                                 141,000.00             48,000.00 *
```

1 Code RPG II file description and input specifications.

File Description Specifications

Line	Form Type	Filename	File Type (I/O/U/C/D)	File Designation (P/S/C/R/T/D/F)	File Format (F/V/S/M/D)	Block Length	Record Length	Device
02	F	SALES	I	P	F	473	43	DISK
03	F	PRINTERA	O		F	12Ø	12Ø	PRINTER
04	F							

Input Specifications

Line	Form Type	Filename	Sequence	Record Identifying Indicator, **, or DS	Field Location From	Field Location To	Decimal Positions	Field Name
01	I	SALES	AA	Ø1				
02	I				1	7		ITEMNO
03	I				8	9		BRANCH
04	I				1Ø	1Ø		REGION
05	I				11	25		DESC
06	I				26	27	Ø	SOLDQY
07	I				28	34	2	SOLDVA
08	I				35	36	Ø	ONHAND
09	I				37	43	2	VALUE
10	I							

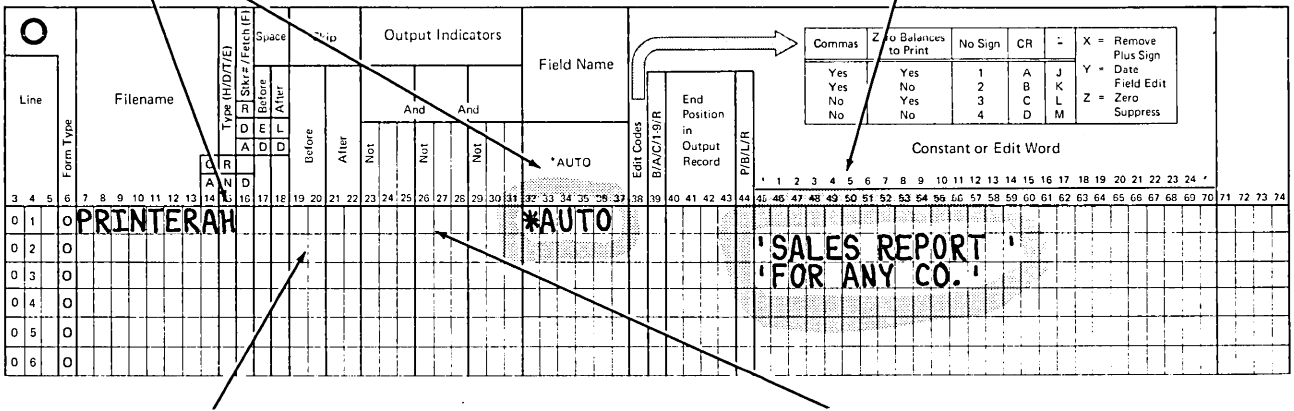
	Field Name	Contents
A	ITEMNO	Item number
B	BRANCH	Number of the branch office where the item was sold
C	REGION	Sales region in which tne branch office is located
D	DESC	Description of the sales item
E	SOLDQY	Quantity of the item sold
F	SOLDVA	Total value of the items sold
G	ONHAND	Quantity of the item remaining on hand
H	VALUE	Total value of the items remaining on hand

2 Code *AUTO page heading specifications.

A Enter an H in column 15 and *AUTO in columns 32 through 36 to request an auto report page heading. Up to five page heading lines can be described. The system date is printed on the left and the page number on the right of the first heading line on each page. To suppress the date and page, enter an N in column 27 of the auto report option specifications sheet.

B The title information is centered by auto report; do not enter end positions in columns 40 through 43. Fields and table/array elements can also be used.

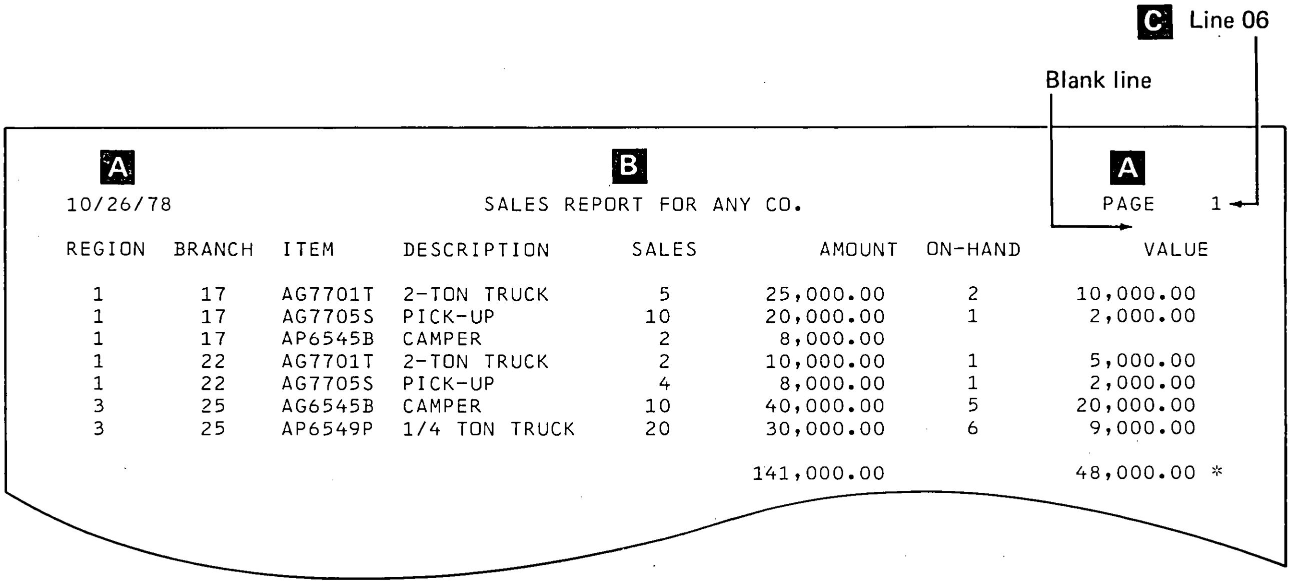
Output Specifications


O
Line
Form Type
Filename
Type (H/D/T/E)
Stkr #/Fetch (F)
Space
Skip
Before
After
Output Indicators
And
And
Not
Field Name
*AUTO
Edit Codes
B/A/C/1-9/R
End Position in Output Record
P/B/L/R
Commas
Zero Balances to Print
No Sign
CR
X = Remove Plus Sign
Y = Date Field Edit
Z = Zero Suppress
Constant or Edit Word

Line	Form Type	Filename	Type	Field Name	Constant or Edit Word
01	O	PRINTERA	H	*AUTO	
02	O				'SALES REPORT '
03	O				'FOR ANY CO.'
04	O				
05	O				
06	O				

C When space and skip entries (columns 17 through 22) are left blank, skip to line 06 is assumed for the first heading line; single spacing is done between heading lines, double spacing after the last heading line. (See *Example 4* for an example of multiple page heading lines.)

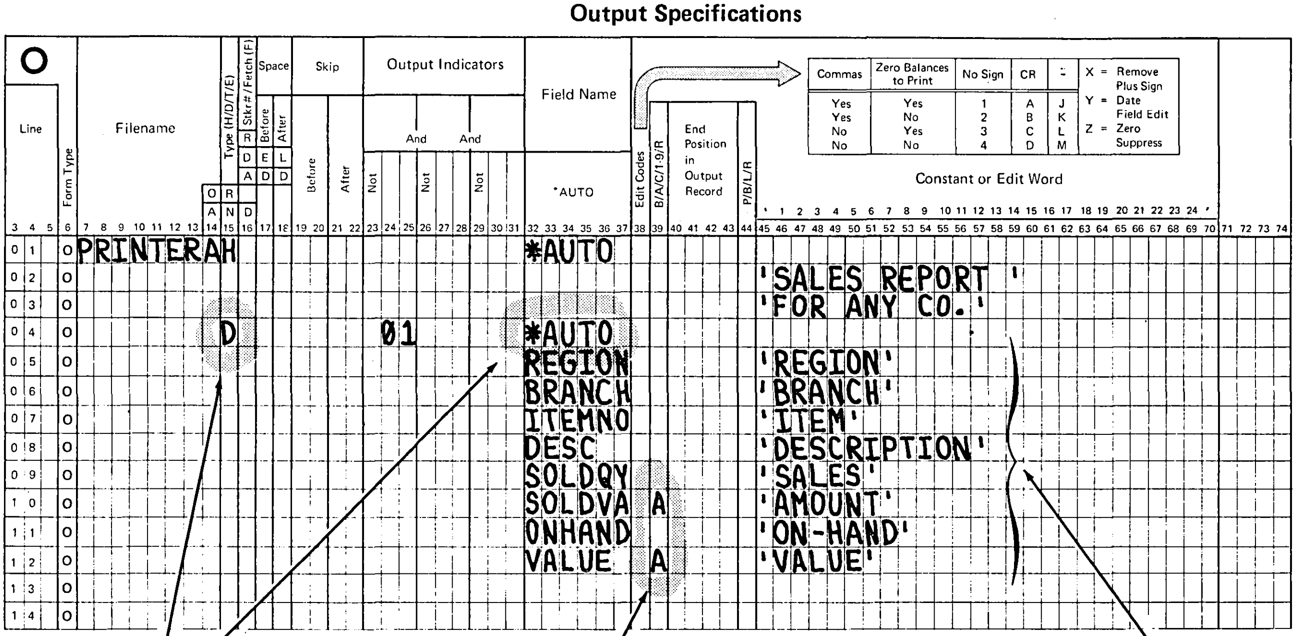
D When output indicators (columns 23 through 31) are left blank, auto report page headings are printed on each page (conditioned by 1P or overflow). If no overflow indicator is defined for the printer file, auto report assigns an unused overflow indicator to the printer line.


C Line 06
Blank line
A
B
A

```
10/26/78                      SALES REPORT FOR ANY CO.                    PAGE    1

REGION   BRANCH   ITEM      DESCRIPTION       SALES        AMOUNT   ON-HAND        VALUE

   1       17     AG7701T   2-TON TRUCK          5      25,000.00      2       10,000.00
   1       17     AG7705S   PICK-UP             10      20,000.00      1        2,000.00
   1       17     AP6545B   CAMPER               2       8,000.00
   1       22     AG7701T   2-TON TRUCK          2      10,000.00      1        5,000.00
   1       22     AG7705S   PICK-UP              4       8,000.00      1        2,000.00
   3       25     AG6545B   CAMPER              10      40,000.00      5       20,000.00
   3       25     AP6549P   1/4 TON TRUCK       20      30,000.00      6        9,000.00

                                                       141,000.00              48,000.00 *
```

3 Code *AUTO output specifications to produce:

- **A** Detail report lines
- **B** Column headings
- **C** Final totals



Output Specifications

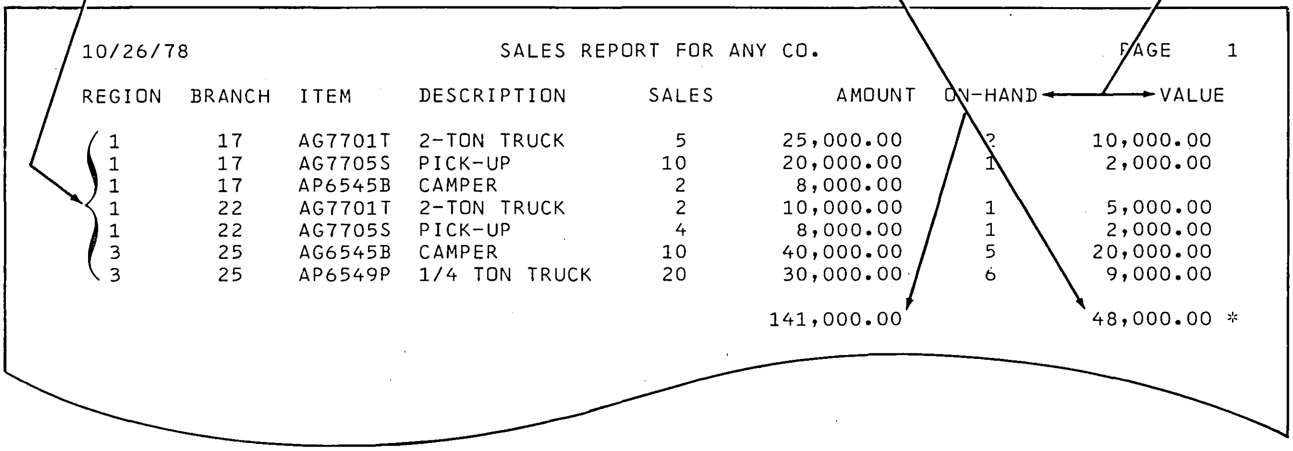
```
01 OPRINTERAH                *AUTO
02 O                                      'SALES REPORT '
03 O                                      'FOR ANY CO.'
04 O        D       01       *AUTO
05 O                         REGION       'REGION'
06 O                         BRANCH       'BRANCH'
07 O                         ITEMNO       'ITEM'
08 O                         DESC         'DESCRIPTION'
09 O                         SOLDQY       'SALES'
10 O                         SOLDVA A     'AMOUNT'
11 O                         ONHAND       'ON-HAND'
12 O                         VALUE  A     'VALUE'
13 O
14 O
```

O
Line
Form Type
Filename
Type (H/D/T/E)
Stkr #/Fetch (F)
Space
Before
After
Skip
Before
After
Output Indicators
And
And
Not
Not
Not
OR
AND
ADD
DEL
Field Name
*AUTO
Edit Codes
B/A/C/1-9/R
End Position in Output Record
P/B/L/R
Commas
Zero Balances to Print
No Sign
CR
-
Yes Yes 1 A J
Yes No 2 B K
No Yes 3 C L
No No 4 D M
X = Remove Plus Sign
Y = Date Field Edit
Z = Zero Suppress
Constant or Edit Word

B Column headings are entered on the same line as the fields over which they appear in the report.

A Enter D in column 15 and *AUTO in columns 32 through 36 to describe an auto report with detail lines. The record identification indicator 01 conditions printing of the detail lines.

C Enter an A in column 39 to cause fields to be accumulated. Auto report generates (1) total fields and calculations to accumulate the totals and (2) total output specifications to print the totals.



```
10/26/78                       SALES REPORT FOR ANY CO.                        PAGE     1

REGION  BRANCH   ITEM      DESCRIPTION       SALES          AMOUNT  ON-HAND          VALUE

  1        17    AG7701T   2-TON TRUCK         5         25,000.00     2         10,000.00
  1        17    AG7705S   PICK-UP            10         20,000.00     1          2,000.00
  1        17    AP6545B   CAMPER              2          8,000.00
  1        22    AG7701T   2-TON TRUCK         2         10,000.00     1          5,000.00
  1        22    AG7705S   PICK-UP             4          8,000.00     1          2,000.00
  3        25    AG6545B   CAMPER             10         40,000.00     5         20,000.00
  3        25    AP6549P   1/4 TON TRUCK      20         30,000.00     6          9,000.00

                                                        141,000.00                48,000.00 *
```

Auto report formats the report so that column headings and data are neatly spaced and centered on each other.

All numeric fields for which a blank, B, or A is specified in column 39 are edited by the K edit code unless a different edit code is specified.

EXAMPLE 2

***AUTO Output**

Problem

Expand sales report from *Example 1* to include three levels of totals:

1. Total for each branch
2. Total for each region
3. Final total

Procedure

1. Code file description and *AUTO specifications as in *Example 1.*
2. Add control level indicators to the input fields BRANCH and REGION.

Note: The *AUTO output function can also be used to produce a group printed report. See *Group Printing* in this chapter for a discussion and examples of group printing.

Input Specifications

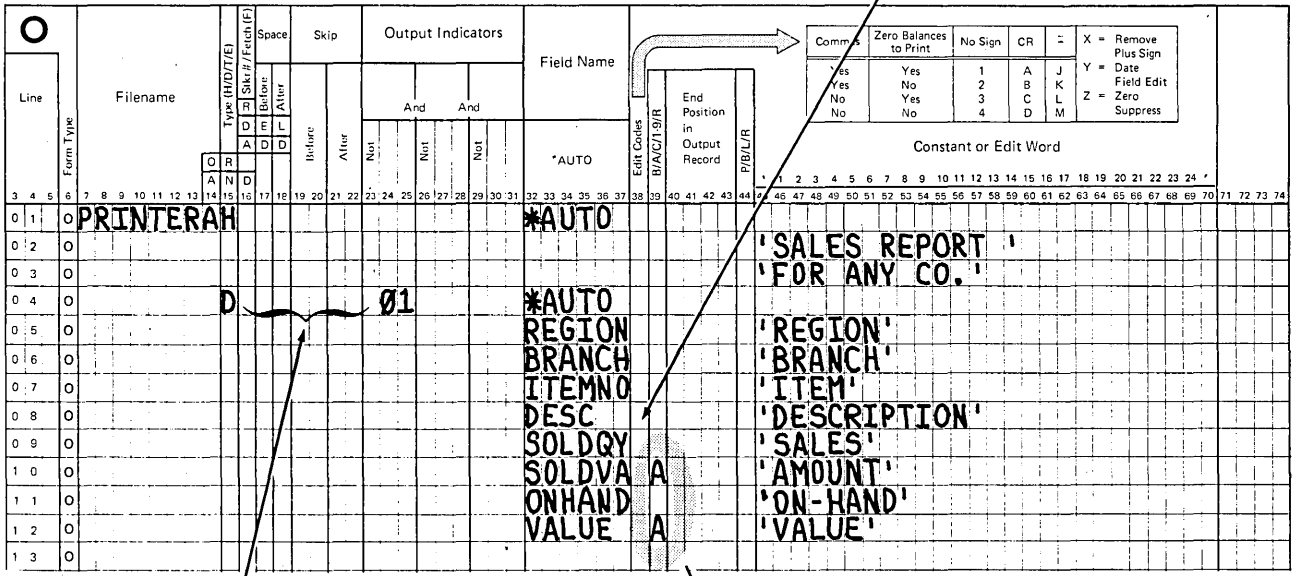
Line	Form Type	Filename	Sequence	Record Identifying Indicator	Field Location From	Field Location To	Decimal Positions	Field Name	Control Level (L1-L9)
01	I	SALES	AA	Ø1					
02	I				1	7		ITEMNO	
03	I				8	9		BRANCH	L1
04	I				1Ø	1Ø		REGION	L2
05	I				11	25		DESC	
06	I				26	27	Ø	SOLDQY	
07	I				28	34	2	SOLDVA	
08	I				35	36	Ø	ONHAND	
09	I				37	43	2	VALUE	
10	I								

2

Because two control levels are defined, the SOLDVA and VALUE fields (see following page) are accumulated to two levels of totals (branch and region) and a final total (LR).

As in *Example 1,* an A in column 39 of the output specification causes SOLDVA and VALUE to be accumulated.

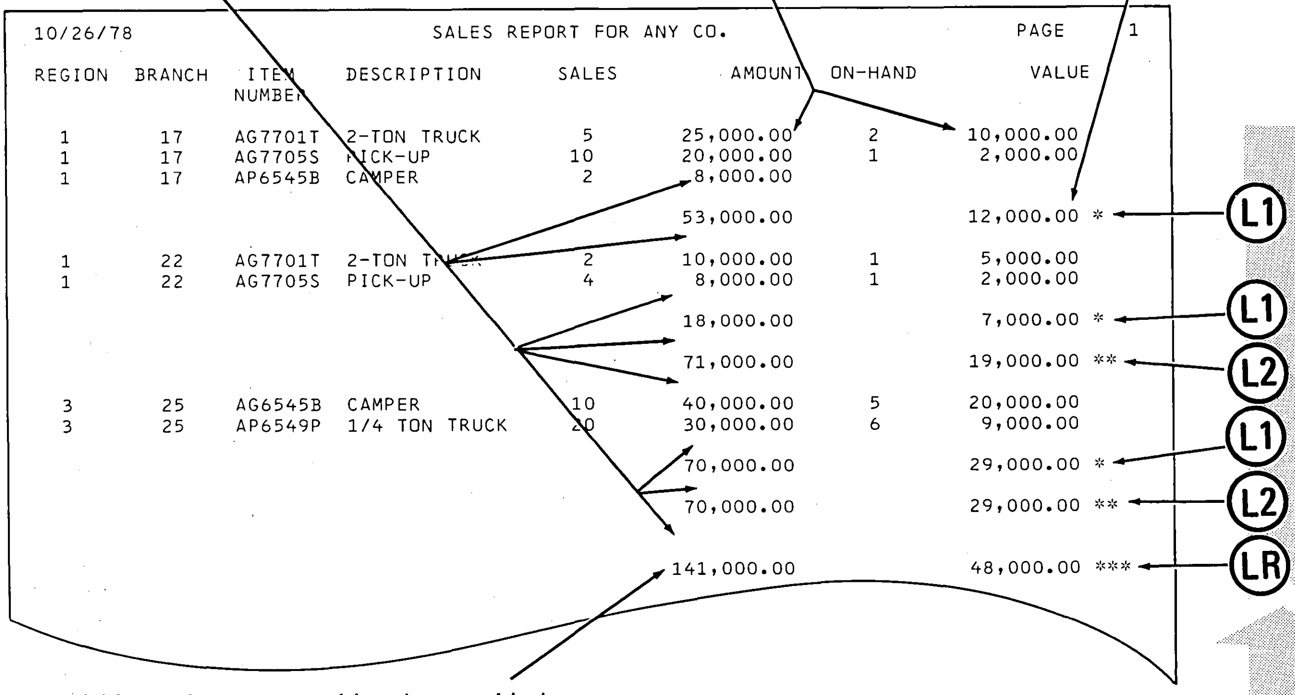
Output Specifications



Line	Form Type	Filename	Type (H/D/T/E)	Output Indicators	Field Name / *AUTO	B/A/C/1-9/R	Constant or Edit Word
01	O	PRINTERA	H		*AUTO		
02	O						'SALES REPORT '
03	O						'FOR ANY CO.'
04	O		D	01	*AUTO		
05	O				REGION		'REGION'
06	O				BRANCH		'BRANCH'
07	O				ITEMNO		'ITEM'
08	O				DESC		'DESCRIPTION'
09	O				SOLDQY		'SALES'
10	O				SOLDVA	A	'AMOUNT'
11	O				ONHAND		'ON-HAND'
12	O				VALUE	A	'VALUE'
13	O						

Auto report places a blank line after each total line and an additional blank line before the lowest level total and before the final total. If you enter spacing and skipping values on the D-*AUTO specification, they apply to the detail print line only.

Auto report prints asterisks (*) to the right of generated total lines to aid in identifying them. If you want to suppress the asterisks, enter N in column 28 of the auto report option specifications sheet.



```
10/26/78                      SALES REPORT FOR ANY CO.                          PAGE    1

REGION  BRANCH   ITEM     DESCRIPTION     SALES        AMOUNT   ON-HAND         VALUE
                 NUMBER

  1       17     AG7701T  2-TON TRUCK       5       25,000.00      2         10,000.00
  1       17     AG7705S  PICK-UP          10       20,000.00      1          2,000.00
  1       17     AP6545B  CAMPER            2        8,000.00

                                                    53,000.00                12,000.00 *

  1       22     AG7701T  2-TON TRUCK       2       10,000.00      1          5,000.00
  1       22     AG7705S  PICK-UP           4        8,000.00      1          2,000.00

                                                    18,000.00                 7,000.00 *

                                                    71,000.00                19,000.00 **

  3       25     AG6545B  CAMPER           10       40,000.00      5         20,000.00
  3       25     AP6549P  1/4 TON TRUCK    20       30,000.00      6          9,000.00

                                                    70,000.00                29,000.00 *

                                                    70,000.00                29,000.00 **

                                                   141,000.00                48,000.00 ***
```

Total fields are always two positions longer with the same number of decimal positions as the original fields.

EXAMPLE 3

***AUTO Output**

Problem

Expand the sales report from *Examples 1* and *2* to contain:

A Group indication for REGION and BRANCH fields

B Second column heading line

C Literal (constant) on the final total line

Procedure

1 Code file description and input specifications as for *Example 2.*

2 Code *AUTO output with:

A Output indicator on field description specifications

B C in column 39 and a literal in columns 45 through 70.

C R in column 39 and a literal in columns 45 through 70.

1

File Description Specifications

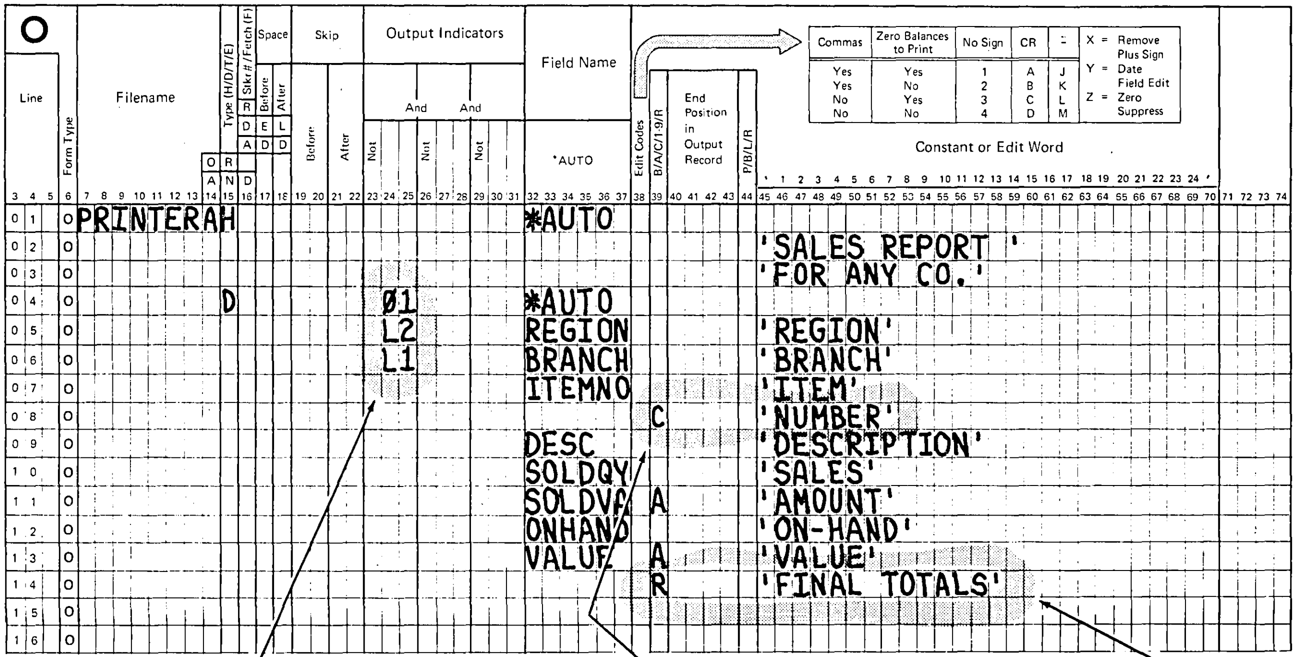
Line	Form Type	Filename	File Type (I/O/U/C/D)	File Designation (P/S/C/R/T/D/F)	End of File (E)	Sequence (A/D)	File Format (F/V/S/M/D)	Block Length	Record Length	Device
02	F	SALES	I	P			F	473	43	DISK
03	F	PRINTERA	O				F	12Ø	12Ø	PRINTER
04	F									

Input Specifications

Line	Form Type	Filename	Sequence	Record Identifying Indicator	Field Location From	Field Location To	Decimal Positions	Field Name	Control Level (L1-L9)
01	I	SALES	AA	Ø1					
02	I				1	7		ITEMNO	
03	I				8	9		BRANCH	L1
04	I				1Ø	1Ø		REGION	L2
05	I				11	25		DESC	
06	I				26	27		SOLDQY	
07	I				28	34	2	SOLDVA	
08	I				35	36	Ø	ONHAND	
09	I				37	43	2	VALUE	
10	I								
11	I								

2

Output Specifications



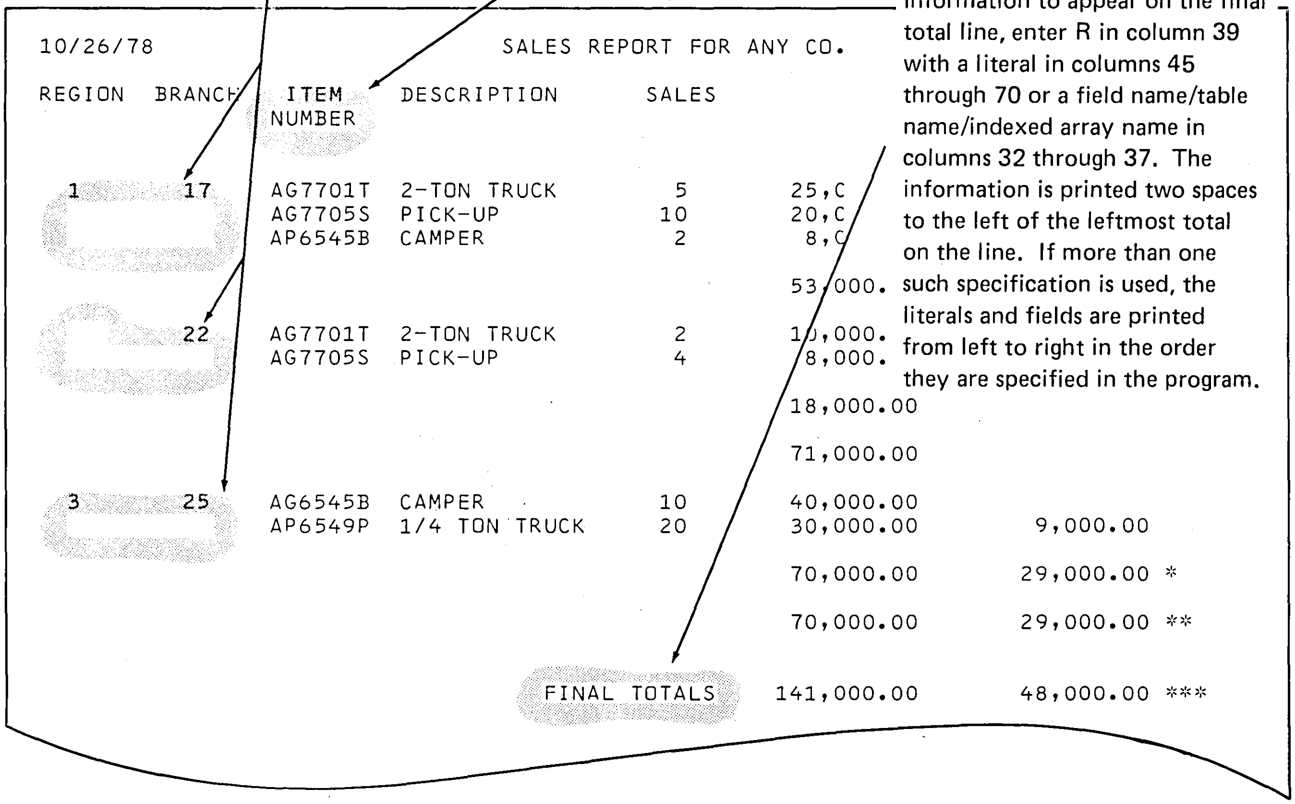
Commas	Zero Balances to Print	No Sign	CR	-	X = Remove Plus Sign
Yes	Yes	1	A	J	Y = Date Field Edit
Yes	No	2	B	K	Z = Zero Suppress
No	Yes	3	C	L	
No	No	4	D	M	

Line	Form Type	Filename	Type (H/D/T/E)	Output Indicators (23–31)	Field Name / *AUTO (32–37)	Edit Codes (38)	B/A/C/1-9/R (39)	End Position in Output Record (40–43)	P/B/L/R (44)	Constant or Edit Word (45–70)
01	O	PRINTERA	H		*AUTO					
02	O									'SALES REPORT '
03	O									'FOR ANY CO.'
04	O		D	Ø1	*AUTO					
05	O			L2	REGION					'REGION'
06	O			L1	BRANCH					'BRANCH'
07	O				ITEMNO					'ITEM'
08	O						C			'NUMBER'
09	O				DESC					'DESCRIPTION'
10	O				SOLDQY					'SALES'
11	O				SOLDVA		A			'AMOUNT'
12	O				ONHAND					'ON-HAND'
13	O				VALUE		A			'VALUE'
14	O						R			'FINAL TOTALS'
15	O									
16	O									

A Output indicators can be used on field description specifications. In this example, control level indicators condition BRANCH and REGION so that they are printed only for the first record of the corresponding control group. This print suppressing of common fields (group indication) reduces repetitive information.

B One or two additional column heading lines can be specified by a C entry in column 39 with the heading information in columns 45 through 70.

C The literal FINAL TOTALS makes that line easy to find. To specify information to appear on the final total line, enter R in column 39 with a literal in columns 45 through 70 or a field name/table name/indexed array name in columns 32 through 37. The information is printed two spaces to the left of the leftmost total on the line. If more than one such specification is used, the literals and fields are printed from left to right in the order they are specified in the program.



```
10/26/78                     SALES REPORT FOR ANY CO.

REGION   BRANCH   ITEM     DESCRIPTION       SALES
                  NUMBER

   1        17    AG7701T  2-TON TRUCK          5       25,C
                  AG7705S  PICK-UP             10       20,C
                  AP6545B  CAMPER               2        8,C

                                                        53,000.

            22    AG7701T  2-TON TRUCK          2       10,000.
                  AG7705S  PICK-UP              4        8,000.

                                                        18,000.00

                                                        71,000.00

   3        25    AG6545B  CAMPER              10       40,000.00
                  AP6549P  1/4 TON TRUCK       20       30,000.00          9,000.00

                                                        70,000.00         29,000.00 *

                                                        70,000.00         29,000.00 **

                                    FINAL TOTALS       141,000.00         48,000.00 ***
```

EXAMPLE 4

Problem

Expand the sales report from *Examples 1, 2,* and *3* to include a cross-totals column and:

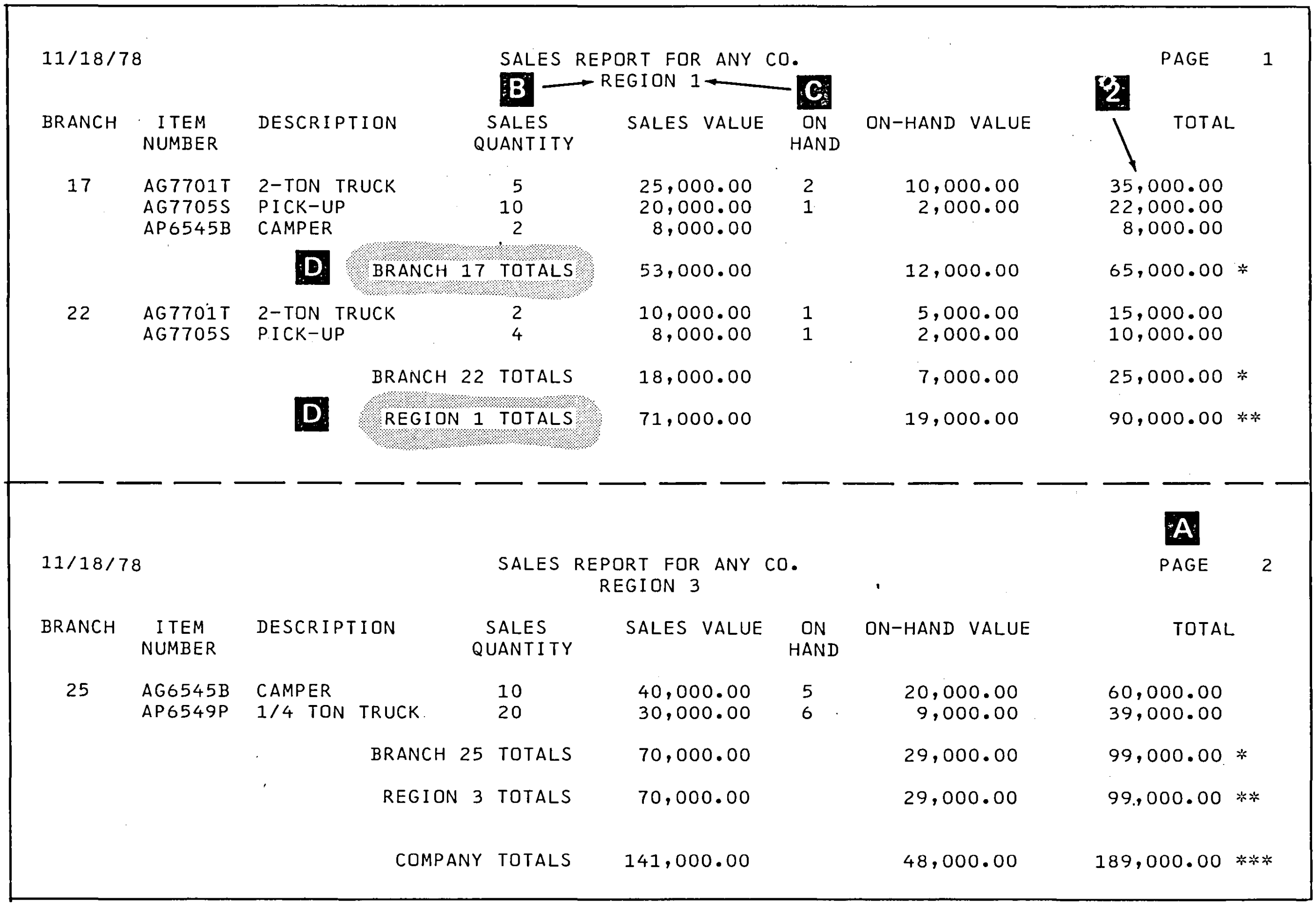
- **A** A new report page for each region
- **B** Two heading lines on each page
- **C** A field in a page heading line
- **D** Identification of branch and region totals

***AUTO Page Headings**

***AUTO Output**

Procedure

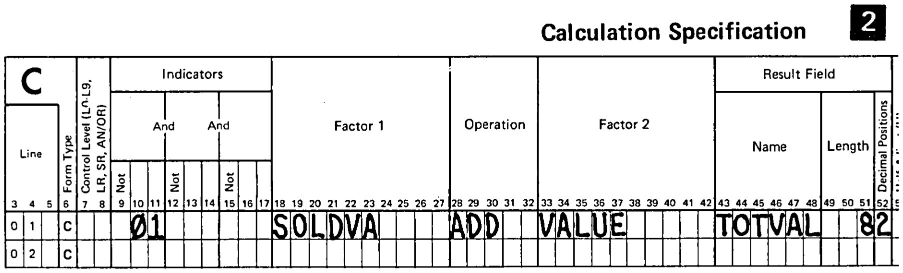
- **1** Code file description and input specifications as in *Example 3;* add an overflow indicator to the printer file.
- **2** Code RPG II calculation specifications for cross-total.
- **3** Code *AUTO specifications:
  - **A** Output indicators on page heading specifications
  - **B** Two heading lines per page
  - **C** Use of a field in an *AUTO page heading specification
  - **D** Fields and literals on L1 through L9 total lines (1 through 9 in column 39)



```
11/18/78                           SALES REPORT FOR ANY CO.                            PAGE    1
                                             REGION 1

BRANCH   ITEM    DESCRIPTION         SALES      SALES VALUE   ON   ON-HAND VALUE              TOTAL
        NUMBER                      QUANTITY                 HAND

  17    AG7701T  2-TON TRUCK            5        25,000.00    2      10,000.00          35,000.00
        AG7705S  PICK-UP               10        20,000.00    1       2,000.00          22,000.00
        AP6545B  CAMPER                 2         8,000.00                               8,000.00

                       BRANCH 17 TOTALS          53,000.00           12,000.00          65,000.00 *

  22    AG7701T  2-TON TRUCK            2        10,000.00    1       5,000.00          15,000.00
        AG7705S  PICK-UP                4         8,000.00    1       2,000.00          10,000.00

                       BRANCH 22 TOTALS          18,000.00            7,000.00          25,000.00 *

                        REGION 1 TOTALS          71,000.00           19,000.00          90,000.00 **
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

11/18/78                           SALES REPORT FOR ANY CO.                            PAGE    2
                                             REGION 3

BRANCH   ITEM    DESCRIPTION         SALES      SALES VALUE   ON   ON-HAND VALUE              TOTAL
        NUMBER                      QUANTITY                 HAND

  25    AG6545B  CAMPER                10        40,000.00    5      20,000.00          60,000.00
        AP6549P  1/4 TON TRUCK         20        30,000.00    6       9,000.00          39,000.00

                       BRANCH 25 TOTALS          70,000.00           29,000.00          99,000.00 *

                        REGION 3 TOTALS          70,000.00           29,000.00          99,000.00 **

                         COMPANY TOTALS         141,000.00           48,000.00         189,000.00 ***
```

Note: Compare matching letters (**B**) on this and the following pages to see the auto report coding to obtain this report.

Calculation Specification 2

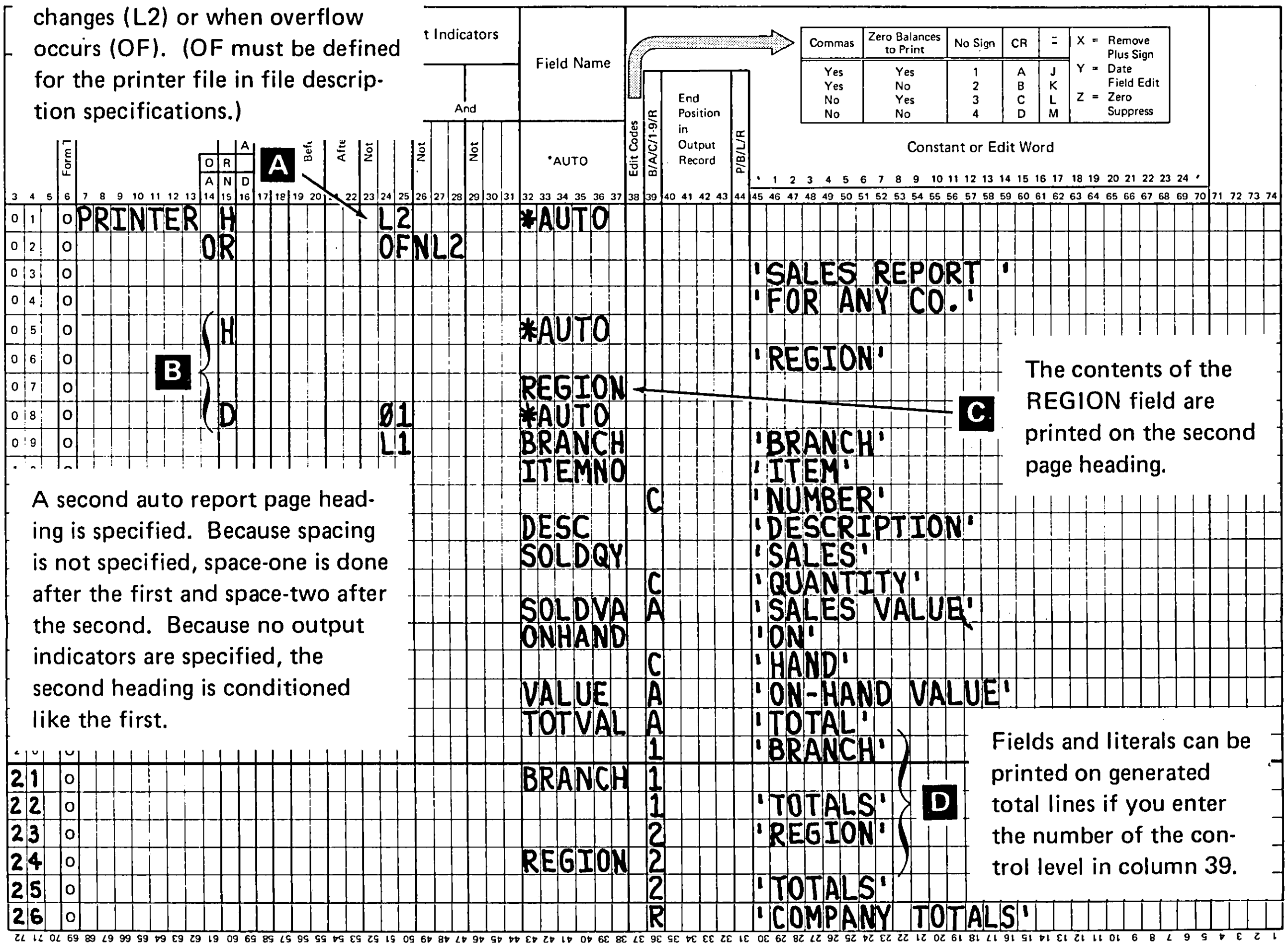


```
  01 C   01      SOLDVA    ADD  VALUE     TOTVAL   82
  02 C
```

RPG II calculations can be among the input statements for auto report. This specification calculates a cross-total of the sales and on-hand values. (The placement of the calculation in relation to calculations generated by auto report is described under *Generated Specifications.*)

Output Specifications 3

The headings are printed on a new page when the region number changes (L2) or when overflow occurs (OF). (OF must be defined for the printer file in file description specifications.)



```
  01 OPRINTER H        L2      *AUTO
  02 O        OR       OFNL2
  03 O                                      'SALES REPORT '
  04 O                                      'FOR ANY CO.'
  05 O        H                *AUTO
  06 O                                      'REGION'
  07 O                         REGION
  08 O        D        01      *AUTO
  09 O                 L1      BRANCH       'BRANCH'
                               ITEMNO       'ITEM'
                                      C     'NUMBER'
                               DESC         'DESCRIPTION'
                               SOLDQY       'SALES'
                                      C     'QUANTITY'
                               SOLDVA A     'SALES VALUE'
                               ONHAND       'ON'
                                      C     'HAND'
                               VALUE  A     'ON-HAND VALUE'
                               TOTVAL A     'TOTAL'
                                      1     'BRANCH'
  21 O                         BRANCH 1
  22 O                                1     'TOTALS'
  23 O                                2     'REGION'
  24 O                         REGION 2
  25 O                                2     'TOTALS'
  26 O                                R     'COMPANY TOTALS'
```

A second auto report page heading is specified. Because spacing is not specified, space-one is done after the first and space-two after the second. Because no output indicators are specified, the second heading is conditioned like the first.

The contents of the REGION field are printed on the second page heading.

Fields and literals can be printed on generated total lines if you enter the number of the control level in column 39.

EXAMPLE 5

COPY

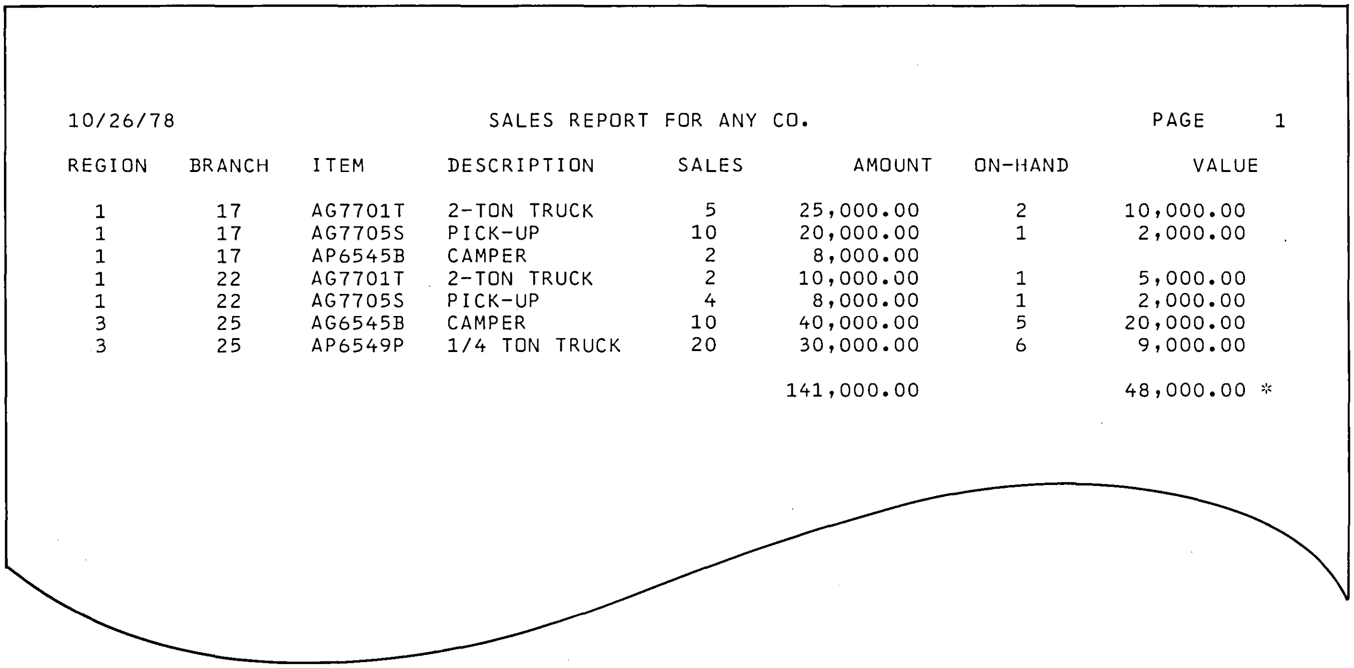
Problem

Use the copy function to obtain specifications for the sales report below (same as in *Example 1*).

Procedure

1. Catalog the file description and input specifications for the SALES file in a library.

2. Code the /COPY statement in the specifications for auto report.



```
10/26/78                     SALES REPORT FOR ANY CO.                          PAGE     1

REGION    BRANCH    ITEM      DESCRIPTION      SALES       AMOUNT   ON-HAND          VALUE

   1        17      AG7701T   2-TON TRUCK        5      25,000.00      2        10,000.00
   1        17      AG7705S   PICK-UP           10      20,000.00      1         2,000.00
   1        17      AP6545B   CAMPER             2       8,000.00
   1        22      AG7701T   2-TON TRUCK        2      10,000.00      1         5,000.00
   1        22      AG7705S   PICK-UP            4       8,000.00      1         2,000.00
   3        25      AG6545B   CAMPER            10      40,000.00      5        20,000.00
   3        25      AP6549P   1/4 TON TRUCK     20      30,000.00      6         9,000.00

                                                       141,000.00               48,000.00 *
```

1 Catalog specifications for the SALES file in a library using the library maintenance utility program.

File Description Specifications

Line	Form Type	Filename	File Type (I/O/U/C/D)	File Designation (P/S/C/R/T/D/F)	End of File	Sequence (A/D)	File Format (F/V/S/M/D)	Block Length	Record Length	Device
02	F	SALES	I	P			F	473	43	DISK
03	F	PRINTER	O				F	12Ø	12Ø	PRINTER
04	F									

These specifications could be replaced by a single statement as shown on the following page if they were cataloged in a library.

Input Specifications

Line	Form Type	Filename	Sequence	Number (1/N)	Option (O), U	Record Identifying Indicator, **, or DS	Field Location From	Field Location To	Decimal Positions	Field Name
01	I	SALES	AA			Ø1				
02	I						1	7		ITEMNO
03	I						8	9		BRANCH
04	I						1Ø	1Ø		REGION
05	I						11	25		DESC
06	I						26	27	Ø	SOLDQY
07	I						28	34	2	SOLDVA
08	I						35	36	Ø	ONHAND
09	I						37	43	2	VALUE
10	I									

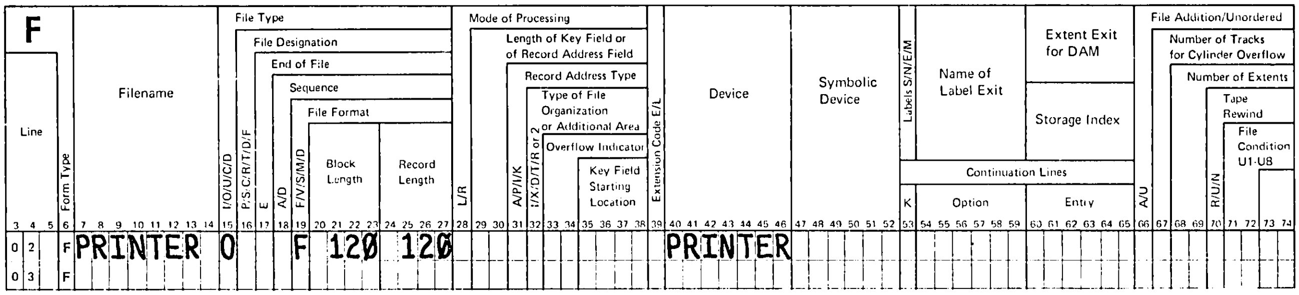
Output Specifications

Line	Form Type	Filename	Type (H/D/T/E)	Output Indicators	Field Name / *AUTO	Edit Codes	Constant or Edit Word
01	O	PRINTER	H		*AUTO		
02	O						'SALES REPORT '
03	O						'FOR ANY CO.'
04	O		D	Ø1	*AUTO		
05	O				REGION		'REGION'
06	O				BRANCH		'BRANCH'
07	O				ITEMNO		'ITEM'
08	O				DESC		'DESCRIPTION'
09	O				SOLDQY		'SALES'
10	O				SOLDVA	A	'AMOUNT'
11	O				ONHAND		'ON-HAND'
12	O				VALUE	A	'VALUE'
13	O						

Commas	Zero Balances to Print	No Sign	CR	-	
Yes	Yes	1	A	J	X = Remove Plus Sign
Yes	No	2	B	K	Y = Date Field Edit
No	Yes	3	C	L	Z = Zero Suppress
No	No	4	D	M	

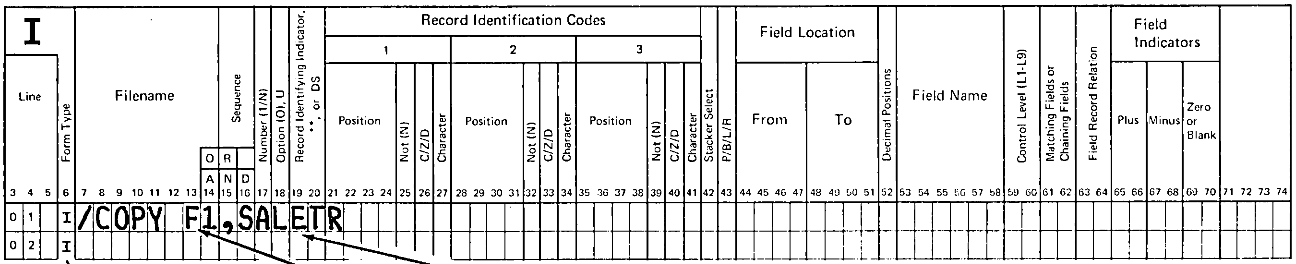
2 Code the /COPY statement to include the file description and input specifications. (For a detailed description of the copy function, see *Statement Specifications.)*

File Description Specifications



```
  02 FPRINTER O   F 12Ø 12Ø            PRINTER
  03 F
```

Input Specifications



```
  01 I/COPY F1,SALETR
  02 I
```

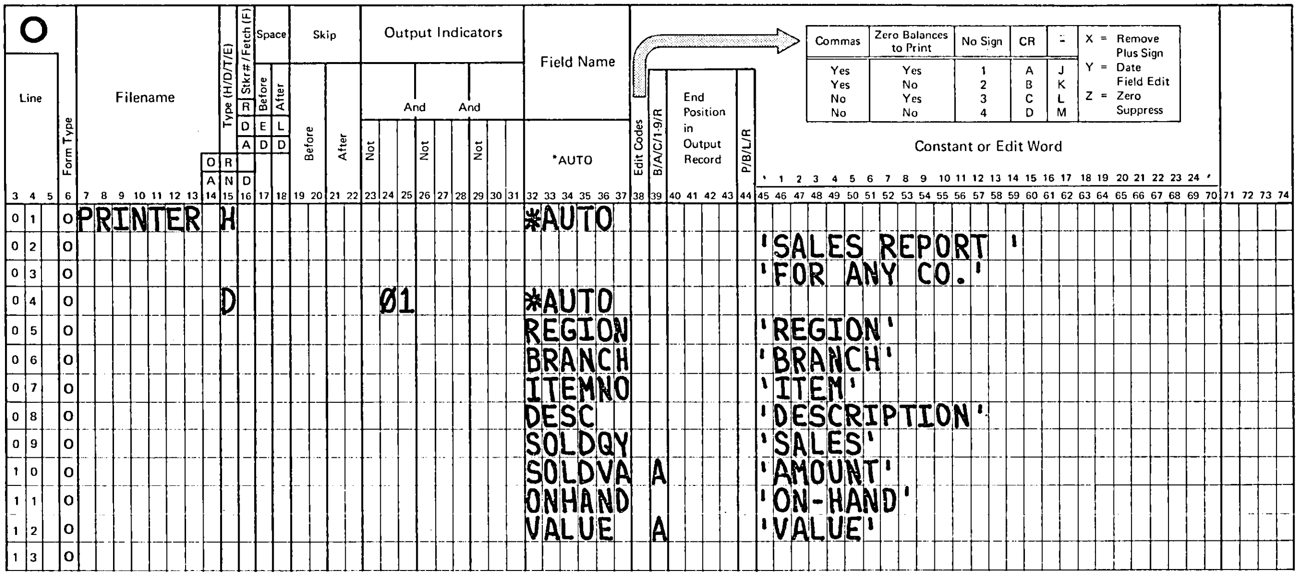
Column 6 of a /COPY statement must not contain a U or an H.

The source member is in the system library.

The /COPY statement copies file description and input specifications for the SALES file from the library member named SALETR.

The /COPY statement can appear anywhere among the auto report specifications following the auto report option statement and preceding table and array input records. It is convenient to code the /COPY on the input sheet when you want to override copied input specificaitons, as in *Example 6.* After specifications are copied, all specifications are sorted into the order required by the RPG II compiler.

Output Specifications



```
  01 OPRINTER H                *AUTO
  02 O                                      'SALES REPORT '
  03 O                                      'FOR ANY CO.'
  04 O        D        Ø1      *AUTO
  05 O                         REGION       'REGION'
  06 O                         BRANCH       'BRANCH'
  07 O                         ITEMNO       'ITEM'
  08 O                         DESC         'DESCRIPTION'
  09 O                         SOLDQY       'SALES'
  10 O                         SOLDVAA      'AMOUNT'
  11 O                         ONHAND       'ON-HAND'
  12 O                         VALUE A      'VALUE'
  13 O
```

EXAMPLE 6

COPY

Problem

Override copied input specifications to produce a report (below) that includes subtotals for branch and region.

Procedure

1 Catalog specifications for the SALES file, as in *Example 5.*

2 Code the /COPY statement.

3 Code /COPY modifier statements to add control level indicators to BRANCH and REGION fields on copied specifications.

```
10/26/78                    SALES REPORT FOR ANY CO.                        PAGE     1

REGION   BRANCH    ITEM       DESCRIPTION      SALES        AMOUNT    ON-HAND         VALUE
                  NUMBER

   1       17     AG7701T     2-TON TRUCK        5       25,000.00       2        10,000.00
                  AG7705S     PICK-UP           10       20,000.00       1         2,000.00
                  AP6545B     CAMPER             2        8,000.00

                                                         53,000.00                12,000.00 *

           22     AG7701T     2-TON TRUCK        2       10,000.00       1         5,000.00
                  AG7705S     PICK-UP            4        8,000.00       1         2,000.00

                                                         18,000.00                 7,000.00 *

                                                         71,000.00                19,000.00 **

   3       25     AG6545B     CAMPER            10       40,000.00       5        20,000.00
                  AG6549P     1/4 TON TRUCK     20       30,000.00       6         9,000.00
```

Cataloged input specifications for the SALES file.

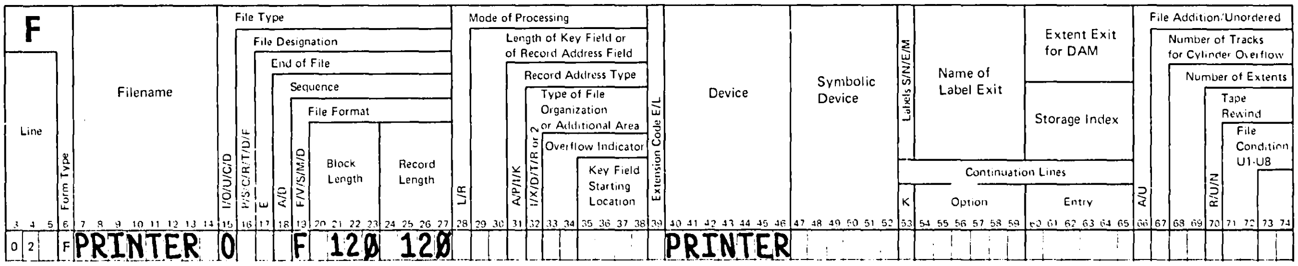
To produce a report that has subtotals for branch and region, L1 must be assigned to BRANCH and L2 to REGION as the specifications are copied from the library.

Input Specifications

Line	Form Type	Filename	Sequence	Number (1/N)	Option (O), U	Record Identifying Indicator, ** or DS	Record Identification Codes 1	Record Identification Codes 2	Record Identification Codes 3	Stacker Select	P/B/L/R	Field Location From	Field Location To	Decimal Positions	Field Name	Control Level (L1-L9)	Matching Fields or Chaining Fields	Field Record Relation	Field Indicators Plus	Field Indicators Minus	Field Indicators Zero or Blank
01	I	SALES	AA			Ø1															
02	I											1	7		ITEMNO						
03	I											8	9		BRANCH						
04	I											1Ø	1Ø		REGION						
05	I											11	25		DESC						
06	I											26	27	Ø	SOLDQY						
07	I											28	34	2	SOLDVA						
08	I											35	36	Ø	ONHAND						
09	I											37	43	2	VALUE						
10	I																				
11	I																				

2 and **3** Code /COPY and modifier statements. As a result of the modifier statements, three levels of totals are accumulated for the SOLDVA and VALUE fields (L1, L2, and LR).

File Description Specifications

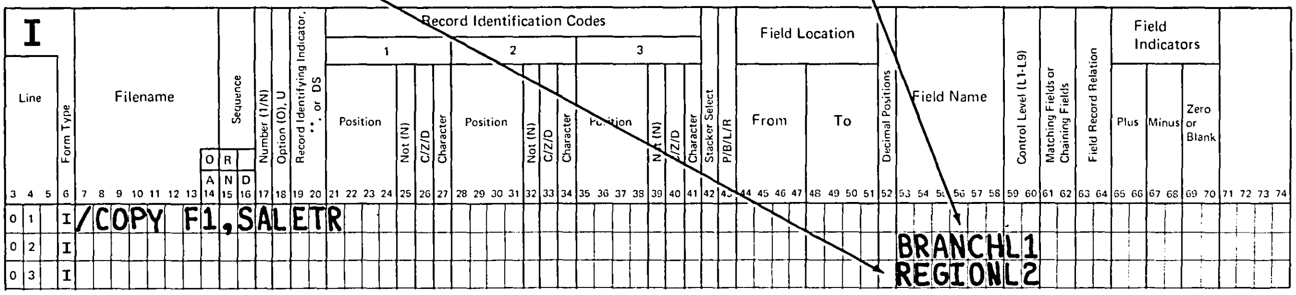


```
02 FPRINTER O   F 12Ø 12Ø            PRINTER
```

Entries on the modifier statements override the corresponding entries in the copied specifications.

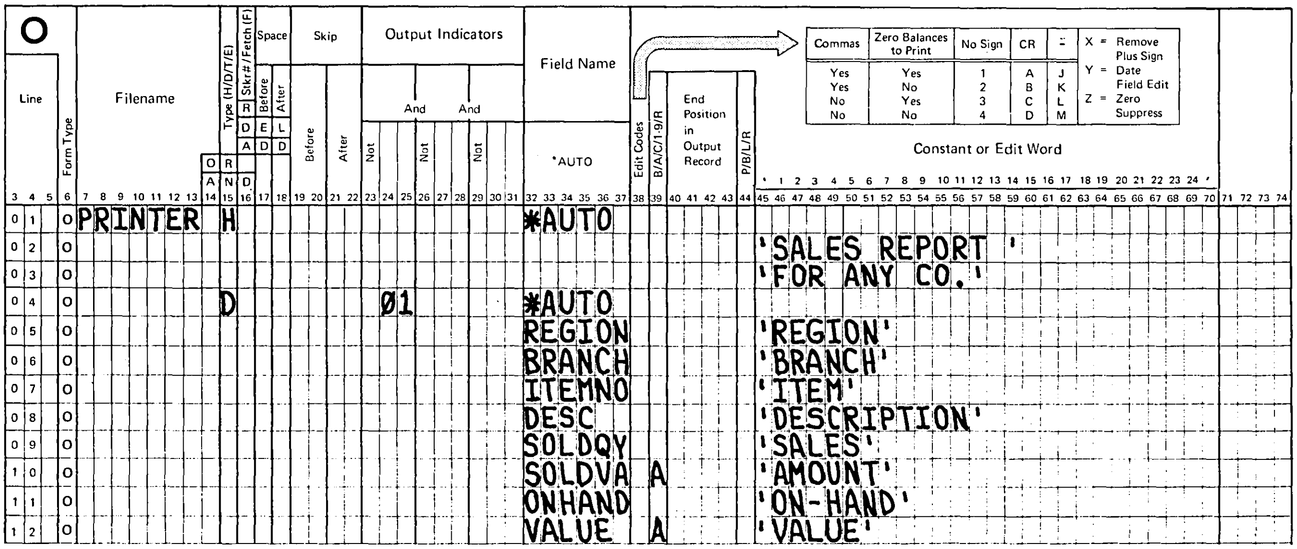
The field names, BRANCH and REGION, identify the input field specifications that are to be modified.

Input Specifications



```
01 I/COPY F1,SALETR
02 I                                              BRANCHL1
03 I                                              REGIONL2
```

Output Specifications



```
01 OPRINTER H                *AUTO
02 O                                      'SALES REPORT '
03 O                                      'FOR ANY CO.'
04 O        D        Ø1      *AUTO
05 O                         REGION       'REGION'
06 O                         BRANCH       'BRANCH'
07 O                         ITEMNO       'ITEM'
08 O                         DESC         'DESCRIPTION'
09 O                         SOLDQY       'SALES'
10 O                         SOLDVAA      'AMOUNT'
11 O                         ONHAND       'ON-HAND'
12 O                         VALUE A      'VALUE'
```

Cataloged file description or input specifications are overridden as follows (see */COPY Statement Specifications* for examples):

- Entries in a modifier statement override corresponding entries in a copied file description or input field specification.
- Blank entries in a modifier statement remain unchanged in a copied specification.
- Ampersand (&) in the leftmost position of an entry in the modifier statement sets the entry to blanks in the copied specification.
- New fields can be added to input specifications by new input field specifications added as modifier statements.
- Modifier statements do not change the cataloged specifications. The modification is only for the program into which the specifications are copied.

Chapter 16. Programming Considerations

STORAGE SAVING TECHNIQUES

When a program is too large to fit into the execution storage size (specified in columns 12 through 14 of the control specifications), some storage saving techniques can be used to help reduce the program size. Before you can use these techniques effectively, however, you need to understand: (1) how the RPG II compiler creates overlays to make a program fit into the storage area available for execution and (2) how the compiler determines when a program is too large to fit into the storage available for execution. The following text discusses the overlay process and provides some suggestions for saving storage.

Note: The maximum object program size before the overlay structure is created is 64K.

Overlay Process

When a program exceeds the storage size available for program execution, the compiler places some RPG II object program routines on disk. These routines are then called into main storage as they are needed by the program. This is known as the *overlay process.*

When the overlay process is used, main storage is divided into two main parts: the *root segment* and the *overlay area.* The root segment contains constants and data used more than once during the program execution. For this reason, the root segment always remains in main storage. The root segment can be used by routines in the overlay area and can call a routine in the overlay area by using a branch instruction. The overlay area contains the major routines of the RPG II object program. Routines in this area can be called by the root segment or by other routines in the same overlay area.

Some large programs require that storage be divided into two additional parts: the secondary root segment and the suboverlay area. The secondary root segment supplements the root segment. If the root segment and the overlay area fill main storage, the secondary root segment is not created. The suboverlay area, created by the RPG II compiler, contains subroutines and other RPG II code needed to support a routine in the overlay area. Figure 16-1 shows the location of the main storage areas.

Creating the Overlays

To create overlays, the compiler first determines which routines go into the overlay area and which routines go into the suboverlay area. Then it calculates the size of the largest overlay and the size of the largest suboverlay. The compiler rounds off these sizes upward in increments of 256 bytes (1 sector) and then adds the sizes of the root segment, the largest overlay, and the largest suboverlay. If the sum is larger than the available storage, the program is too large and storage saving techniques must be used if the program is to be run in the available execution storage size.

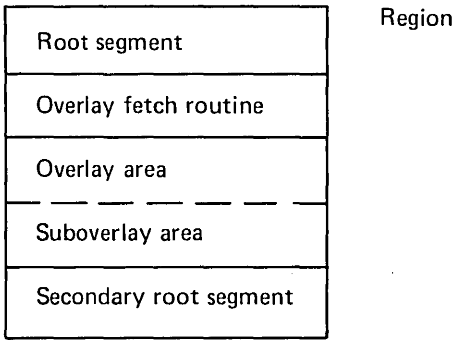


Figure 16-1. RPG II Storage Map

Special Open/Close

The compiler uses a special open/close when the overlay requirements for open and close exceed the overlay requirements for the rest of the program. Special open/close can be easily identified because overlay $##002 is the first overlay identified in the main storage usage map (see Figure 16-2).

The initial load of the program brings in the code specified in the main storage usage map between and including the root segment and the overlay fetch area. Open is completely self-contained and does not need any of the nonoverlay code. When open is complete, overlay code is loaded.

Overlay code consists of all code that is identified as nonoverlay and was not loaded during the first load. (Overlay code is also identified as overlay $##002 in the overlay map following the storage usage map. The other overlay numbers correspond to their respective overlay numbers as they appear in the storage usage map shown in Figure 16-2.) The program then executes as a normal overlay program until close is needed. At this time, close is brought into main storage starting at the overlay fetch area and using as much main storage as is needed. To find the overlay fetch area size for the rest of the program, subtract the start of the overlay fetch area from the lowest start address of the nonoverlay code that was not included in the first load. For example, in Figure 16-2 the input control routine starts at hex 2340 so hex 2340 minus hex 2340 equals hex 0000–therefore, in this example there is no overlay fetch area.

After identifying the root segment and the largest overlay and suboverlay, you can determine whether they contain routines that can be manipulated to reduce the overlay size. Remove routines that can be manipulated from the largest overlay and place them in the smaller overlay. When the root segment, largest overlay, and largest suboverlay are calculated, the storage area needed is less. Storage saving techniques can be used to control the following routines:

- Input records
- Detail calculations
- Total calculations
- Detail output
- Total output

The following storage saving techniques can be used for these routines; however, they might not work for all programs.

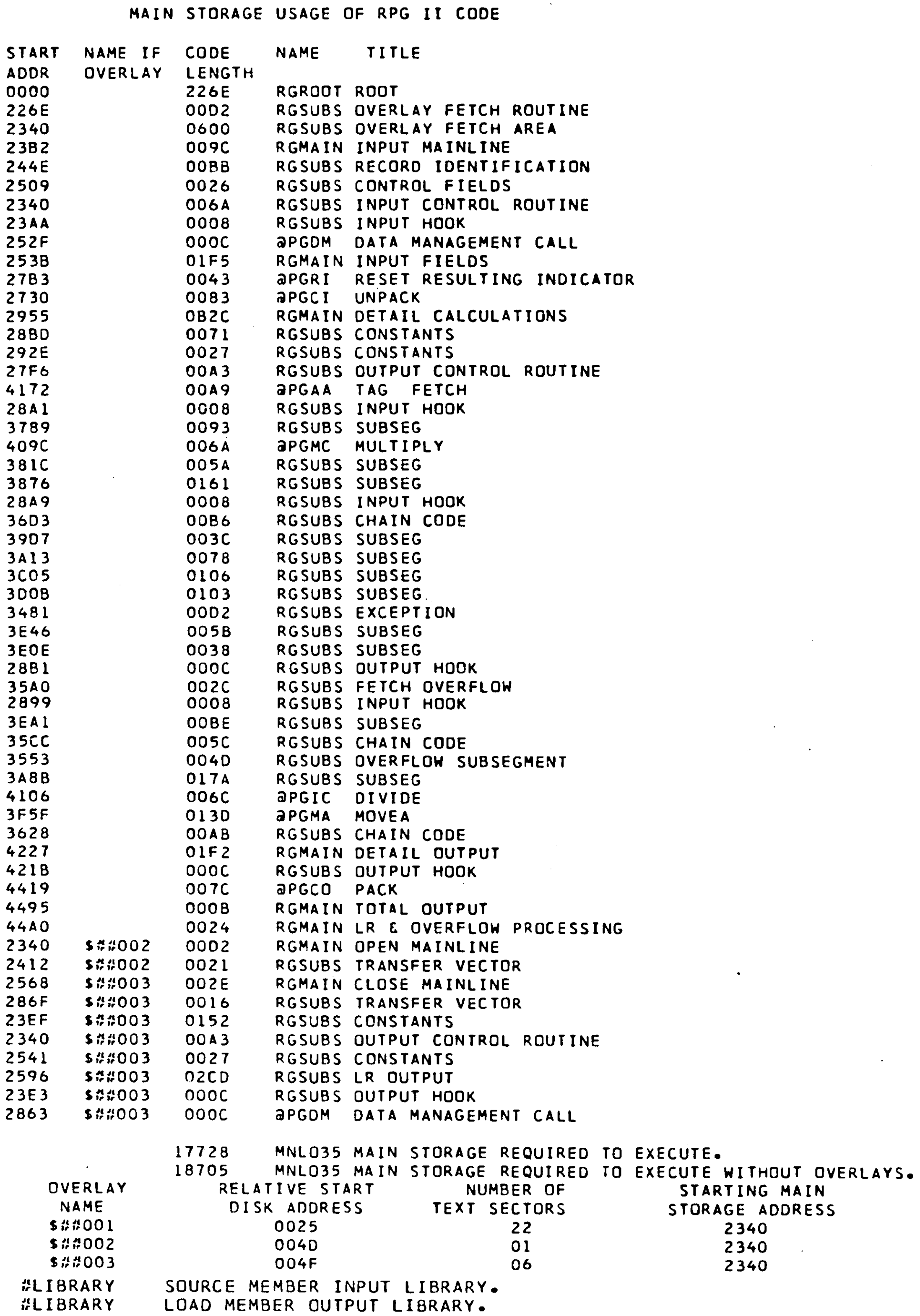
Input Records: You can process one or more of the input or update files as a demand or chained file using the READ or CHAIN operation code. With a demand or chained file, the instructions to read the file can be moved into the total or detail calculation routines. Remember that total calculations are not done on the first cycle.

Detail or Total Calculations: Use the following techniques:

- Use subroutine calculations. In some instances using subroutines can increase, rather than decrease, the storage required because of the nature of the existing calculation routines. However, it can reduce the overall storage requirements. If one subroutine calls another subroutine, both subroutines must be in storage at the same time. This can increase the size of the suboverlay area and the total storage required. To ensure the smallest requirement, do not call a subroutine from another subroutine.
- Eliminate exception output if possible. This moves the logic for those output operations to either total or detail output routines.
- Eliminate READ and/or CHAIN operations by using matching records and consecutive processing. This moves the logic to input records routine.
- Move part of the detail calculations to total calculations (or total calculation logic to detail calculations). Total calculations are not done on the first cycle.

Detail or Total Output: Use the following techniques:

- Use exception output. This moves part of the output logic to detail or total calculation routines.
- Do some of the output at total (or detail) output time. This moves logic to the total (or detail) output routine.
- Do not specify blank after for fields; instead, clear them at the beginning of detail or total calculations.



```
              MAIN STORAGE USAGE OF RPG II CODE

START   NAME IF  CODE     NAME    TITLE
ADDR    OVERLAY  LENGTH
0000             226E     RGROOT ROOT
226E             00D2     RGSUBS OVERLAY FETCH ROUTINE
2340             0600     RGSUBS OVERLAY FETCH AREA
23B2             009C     RGMAIN INPUT MAINLINE
244E             00BB     RGSUBS RECORD IDENTIFICATION
2509             0026     RGSUBS CONTROL FIELDS
2340             006A     RGSUBS INPUT CONTROL ROUTINE
23AA             0008     RGSUBS INPUT HOOK
252F             000C     @PGDM  DATA MANAGEMENT CALL
253B             01F5     RGMAIN INPUT FIELDS
27B3             0043     @PGRI  RESET RESULTING INDICATOR
2730             0083     @PGCI  UNPACK
2955             0B2C     RGMAIN DETAIL CALCULATIONS
28BD             0071     RGSUBS CONSTANTS
292E             0027     RGSUBS CONSTANTS
27F6             00A3     RGSUBS OUTPUT CONTROL ROUTINE
4172             00A9     @PGAA  TAG  FETCH
28A1             0008     RGSUBS INPUT HOOK
3789             0093     RGSUBS SUBSEG
409C             006A     @PGMC  MULTIPLY
381C             005A     RGSUBS SUBSEG
3876             0161     RGSUBS SUBSEG
28A9             0008     RGSUBS INPUT HOOK
36D3             00B6     RGSUBS CHAIN CODE
39D7             003C     RGSUBS SUBSEG
3A13             0078     RGSUBS SUBSEG
3C05             0106     RGSUBS SUBSEG
3D0B             0103     RGSUBS SUBSEG
3481             00D2     RGSUBS EXCEPTION
3E46             005B     RGSUBS SUBSEG
3E0E             0038     RGSUBS SUBSEG
28B1             000C     RGSUBS OUTPUT HOOK
35A0             002C     RGSUBS FETCH OVERFLOW
2899             0008     RGSUBS INPUT HOOK
3EA1             00BE     RGSUBS SUBSEG
35CC             005C     RGSUBS CHAIN CODE
3553             004D     RGSUBS OVERFLOW SUBSEGMENT
3A8B             017A     RGSUBS SUBSEG
4106             006C     @PGIC  DIVIDE
3F5F             013D     @PGMA  MOVEA
3628             00AB     RGSUBS CHAIN CODE
4227             01F2     RGMAIN DETAIL OUTPUT
421B             000C     RGSUBS OUTPUT HOOK
4419             007C     @PGCO  PACK
4495             000B     RGMAIN TOTAL OUTPUT
44A0             0024     RGMAIN LR & OVERFLOW PROCESSING
2340    $##002   00D2     RGMAIN OPEN MAINLINE
2412    $##002   0021     RGSUBS TRANSFER VECTOR
2568    $##003   002E     RGMAIN CLOSE MAINLINE
286F    $##003   0016     RGSUBS TRANSFER VECTOR
23EF    $##003   0152     RGSUBS CONSTANTS
2340    $##003   00A3     RGSUBS OUTPUT CONTROL ROUTINE
2541    $##003   0027     RGSUBS CONSTANTS
2596    $##003   02CD     RGSUBS LR OUTPUT
23E3    $##003   000C     RGSUBS OUTPUT HOOK
2863    $##003   000C     @PGDM  DATA MANAGEMENT CALL

                17728     MNL035 MAIN STORAGE REQUIRED TO EXECUTE.
                18705     MNL035 MAIN STORAGE REQUIRED TO EXECUTE WITHOUT OVERLAYS.
   OVERLAY        RELATIVE START        NUMBER OF        STARTING MAIN
    NAME           DISK ADDRESS       TEXT SECTORS      STORAGE ADDRESS
   $##001              0025                22                2340
   $##002              004D                01                2340
   $##003              004F                06                2340
  #LIBRARY      SOURCE MEMBER INPUT LIBRARY.
  #LIBRARY      LOAD MEMBER OUTPUT LIBRARY.
```

Figure 16-2. Main Storage Usage Map

Influencing the Overlay Structure

You can influence the overlay structure that is created for a particular program if you have subroutines in the detail calculations.

As the calculation subroutines are generated, the compiler assigns a priority to each subroutine for going into the overlay. The priorities are assigned from 1 to 6 with the first subroutine in the calculations being assigned priority 1. The remaining subroutines in the calculations are assigned priority 2 through 6 as they are encountered. All subroutines after the sixth are also assigned priority 6. The subroutines are placed into the overlay, if required, according to the assigned priority. Priority 6 subroutines go into the overlay first, and the priority 1 subroutine goes in last. You should place the most frequently used subroutine first in your calculations and the least frequently used one last. This may reduce the number of overlay or suboverlay loads.

General Storage Saving Techniques

When the compiler finds that a program is too large for the storage available for execution, an error message is written. You can reduce the main storage needed for your program either by using some general storage saving techniques or by reducing the size of the overlays.

You can use some of the following techniques:

- Divide the program into separate tasks, creating a separate program for each task. For example, if you want to update a file and print a listing of the updated file, you can save storage by updating the file with one program and printing the listing with another program.
- Eliminate unreferenced indicators. Eliminating unreferenced indicators can eliminate the instructions required to set the indicators on and off.
- Eliminate unnecessary conditioning indicators. Two possible forms of unnecessary indicator tests are as follows:
  - If only one type of input record is to be processed, the indicator associated with that record is always on except during the first detail output time. It is, therefore, not necessary for any calculation to be conditioned with this indicator.
  - When two subsequent operations on the same result field are conditioned on opposite indicator conditions, one of the conditions is not necessary. For instance, the N09 conditioning is not required in this example:

  ```
  N09  Z-ADD  FLD    FLDB
  09   Z-ADD  FLDC   FLDB
  ```

  This technique might not work for certain operations if the same field is used as the result field and as factor 1 or factor 2.
- Reuse calculation work areas and temporary hold areas. Once the data stored in these areas is used for the last time in a given cycle, the area is available. Reusing these areas can eliminate the need for two or more additional areas to be defined. However, the areas must be used for the same type of data.
- Reuse input field names. You can reuse input field areas by using the same name for fields in two or more files. This can be done only if the fields have the same attributes (length, alphameric/numeric, packed/binary) and each field is used only in the cycle in which the record is processed. Both files cannot be used in the same cycle.
- Reduce calculation result field sizes. Be sure that no result field is defined any larger than is necessary. Reducing the result field size can cause a warning that the result field may not be large enough. If you know that the largest possible number fits into the result field specified, you can continue compiling the program.
- Include the necessary intervening blanks when describing alphameric fields and constants for output. This makes the fields adjacent. The output optimization phase moves all adjacent fields and constants with one instruction instead of using one instruction to move each line:

Not Optimized	**Optimized**
5'DAILY'	18'DAILY TRANSACTIONƀ'
17'TRANSACTION'	26'REGISTER'
26'REGISTER'	

- Use data structures to define the same internal storage area for multiple record types and to reduce the use of MOVE and MOVEL operations.
- Design files to contain record lengths that are an even multiple of 256 bytes or that divide into 256 bytes an even number of times.
- Design files so that match fields and control fields are assigned the same position within all record types.
- Do not designate a field as numeric unless the field is used in an arithmetic operation in the program. This saves on the amount of storage required to store the field and allows the input and output fields' transfer routine to be optimized.
- Use the shared input/output access method (SIAM) to process disk files. This may reduce the storage required for programs using input disk files; however, using SIAM can decrease program throughput.
- Group calculation statements that are conditioned by the same indicators. When a large number of indicators are required, try to use GOTO or EXSR to reduce the number of indicator tests required on each statement.
- Use the actual bit pattern in factor 2 when using TESTB, BITON, or BITOF.
- Do not use half adjust unless absolutely necessary.
- Try to use either factor 1 or factor 2 as the result field whenever possible.
- Try to use numeric fields of the same length and with the same number of decimal positions. If the fields cannot be the same length, try to have the number of decimal positions the same.
- Do not sequence check your records unless absolutely necessary.
- Use OR lines rather than multiple record lines because OR lines require less code.
- Specify the fields in a record in ascending order by record position.
- Do not use halt indicators unless absolutely necessary.

Reduce the Overlay Size

To reduce the size of the overlay, you can reduce the size of the root segment or the overlay areas. First, however, you must identify the contents of the root segment and the largest overlays in main storage. Then you can determine whether the contents of these areas can be reduced to fit into the storage available for execution.

Use the program listing to find the contents of the root segment, overlay area, and suboverlay area. The root segment contains the data and routines that are not given an overlay name in the main storage usage of the RPG II code section of the program listing (see Figure 16-3).

Two sections of the program listing determine the contents of the overlay and suboverlay areas. The section shown in Figure 16-4 gives the:

- Overlay name
- Number of sectors in the overlay
- Start address of the overlay

The start address separates overlays and suboverlays. Two start addresses appear in the start address column. The lower address (1868 in Figure 16-4) identifies an overlay; the higher address (1B68 in Figure 16-4) identifies a suboverlay.

The text sectors column indicates the largest overlays. In Figure 16-4, overlay 002 is the largest suboverlay; overlay 004 is the largest overlay.

Relate the name given in the overlay name columns shown in Figure 16-4 to the main storage usage of RPG II code section shown in Figure 16-5. The name and title columns in this section identify the routines or subroutines in the overlay.

Note: If overlay 001 does not appear in the overlay name column, a special open/close overlay construction took place. When this occurs, overlay 001 is not treated as an overlay, but remains in main storage.

```
                      MAIN STORAGE USAGE OF RPG II CODE

        START  NAME IF   CODE    NAME    TITLE
        ADDR   OVERLAY   LENGTH
        0000             1743    RGROOT  ROOT
        1743             0125    RGSUBS  OVERLAY FETCH ROUTINE
        1868             0600    RGSUBS  OVERLAY FETCH AREA
        1EBB             0099    RGMAIN  INPUT MAINLINE
        1FCB             0016    RGSUBS  TRANSFER VECTOR
        1F54             0051    RGSUBS  RECORD IDENTIFICATION
        1FA5             0026    RGSUBS  CONTROL FIELDS
        1E68             0053    RGSUBS  INPUT CONTROL ROUTINE
        1B68   $##001    01FC    @PGTA   CONSOLE - IDE DATA MANAGEMENT
        1B68   $##002    0008    RGSUBS  INPUT HOOK
        1B70   $##002    01FC    @PGTA   CONSOLE - IDE DATA MANAGEMENT
        1868   $##003    0049    RGMAIN  INPUT FIELDS
        1990   $##004    00B1    RGMAIN  DETAIL CALCULATIONS
        194D   $##004    0043    RGSUBS  CONSTANTS
Root    1FE1             0002    RGSUBS  CONSTANTS
        1868   $##004    00B5    RGSUBS  OUTPUT CONTROL ROUTINE
        1B31   $##004    0043    @PGRI   RESET RESULTING INDICATOR
        1A41   $##004    007B    RGSUBS  EXCEPTION
        1935   $##004    000C    RGSUBS  OUTPUT HOOK
        1941   $##004    000C    RGSUBS  OUTPUT HOOK
        1AF8   $##004    002C    RGSUBS  FETCH OVERFLOW
        191D   $##004    000C    RGSUBS  OUTPUT HOOK
        1929   $##004    000C    RGSUBS  OUTPUT HOOK
        1B74   $##004    000C    @PGDM   DATA MANAGEMENT CALL
        1ABC   $##004    003C    RGSUBS  OVERFLOW SUBSEGMENT
        1B24   $##004    000D    RGSUBS  SUBSEG
        1978   $##005    00A4    RGMAIN  DETAIL OUTPUT
        1868   $##005    00B5    RGSUBS  OUTPUT CONTROL ROUTINE
        1935   $##005    0043    RGSUBS  CONSTANTS
        1929   $##005    000C    RGSUBS  OUTPUT HOOK
        1A98   $##005    000C    @PGDM   DATA MANAGEMENT CALL
        191D   $##005    000C    RGSUBS  OUTPUT HOOK
        1A1C   $##005    007C    @PGCO   PACK
        1FE3             000B    RGMAIN  TOTAL OUTPUT
        19A8   $##006    0024    RGMAIN  LR & OVERFLOW PROCESSING
        1868   $##006    00B5    RGSUBS  OUTPUT CONTROL ROUTINE
        1929   $##006    0043    RGSUBS  CONSTANTS
        196C   $##006    003C    RGSUBS  OVERFLOW SUBSEGMENT
        191D   $##006    000C    RGSUBS  OUTPUT HOOK
        19CC   $##006    000D    RGSUBS  SUBSEG
        19D9   $##006    000C    @PGDM   DATA MANAGEMENT CALL
        19A1   $##007    002E    RGMAIN  CLOSE MAINLINE
        1A06   $##007    0016    RGSUBS  TRANSFER VECTOR
        1868   $##007    00B5    RGSUBS  OUTPUT CONTROL ROUTINE
        195E   $##007    0043    RGSUBS  CONSTANTS
        1929   $##007    0035    RGSUBS  CONSTANTS
        19CF   $##007    002B    RGSUBS  LR OUTPUT
        191D   $##007    000C    RGSUBS  OUTPUT HOOK
        19FA   $##007    000C    @PGDM   DATA MANAGEMENT CALL
        19A8   $##008    00FA    RGMAIN  OPEN MAINLINE
        1ABB   $##008    0021    RGSUBS  TRANSFER VECTOR
        1965   $##008    0043    RGSUBS  CONSTANTS
        1868   $##008    00B5    RGSUBS  OUTPUT CONTROL ROUTINE
        1929   $##008    003C    RGSUBS  CONSTANTS
        191D   $##008    000C    RGSUBS  OUTPUT HOOK
        1AAF   $##008    000C    @PGDM   DATA MANAGEMENT CALL
        1AA2   $##008    000D    RGSUBS  SUBSEG

                         08171   RPF148 MAIN STORAGE REQUIRED TO EXECUTE.
                         08492   RPF148 MAIN STORAGE REQUIRED TO EXECUTE WITHOUT OVERLAYS.
                      #LIBRARY   SOURCE MEMBER INPUT LIBRARY.
                      #LIBRARY   LOAD MEMBER OUTPUT LIBRARY.
```

Figure 16-3. Overlay Usage Map

```
OVERLAY             RELATIVE START          NUMBER OF           STARTING MAIN
 NAME      Largest   DISK ADDRESS         TEXT SECTORS         STORAGE ADDRESS
$##001                  0021                  02                 {1B68
$##002<-- Suboverlay    0024                  03     Suboverlays {1B68
$##003                  0028                  01                 (1868
$##004<-- Largest       002A                  04                 {1868
$##005    Overlay       002F                  03                 }1868
$##006                  0033                  02        Overlays <1868
$##007                  0036                  02                 }1868
$##008                  0039                  03                 (1868
```

Figure 16-4. Overlay Identification Area

```
                               MAIN STORAGE USAGE OF RPG II CODE

                  START   NAME IF   CODE     NAME    TITLE
                  ADDR    OVERLAY   LENGTH
                  0000              1743     RGROOT  ROOT
                  1743              0125     RGSUBS  OVERLAY FETCH ROUTINE
                  1868              0600     RGSUBS  OVERLAY FETCH AREA
                  1EBB              0099     RGMAIN  INPUT MAINLINE
                  1FCB              0016     RGSUBS  TRANSFER VECTOR
                  1F54              0051     RGSUBS  RECORD IDENTIFICATION
                  1FA5              0026     RGSUBS  CONTROL FIELDS
                  1E68              0053     RGSUBS  INPUT CONTROL ROUTINE
Suboverlay 001    1B68    $##001    01FC     @PGTA   CONSOLE - IDE DATA MANAGEMENT
Suboverlay 002   {1B68    $##002    0008     RGSUBS  INPUT HOOK
                 }1B70    $##002    01FC     @PGTA   CONSOLE - IDE DATA MANAGEMENT
Overlay 003       1868    $##003    0049     RGMAIN  INPUT FIELDS
                 {1990    $##004    00B1     RGMAIN  DETAIL CALCULATIONS
                 }194D    $##004    0043     RGSUBS  CONSTANTS
                  1FE1              0002     RGSUBS  CONSTANTS
                 (1868    $##004    00B5     RGSUBS  OUTPUT CONTROL ROUTINE
                 |1B31    $##004    0043     @PGRI   RESET RESULTING INDICATOR
                 |1A41    $##004    007B     RGSUBS  EXCEPTION
                 |1935    $##004    000C     RGSUBS  OUTPUT HOOK
                 |1941    $##004    000C     RGSUBS  OUTPUT HOOK
Overlay 004     < 1AF8    $##004    002C     RGSUBS  FETCH OVERFLOW
                 |191D    $##004    000C     RGSUBS  OUTPUT HOOK
                 |1929    $##004    000C     RGSUBS  OUTPUT HOOK
                 |1B74    $##004    000C     @PGDM   DATA MANAGEMENT CALL
                 |1ABC    $##004    003C     RGSUBS  OVERFLOW SUBSEGMENT
                 (1B24    $##004    000D     RGSUBS  SUBSEG
                 (1978    $##005    00A4     RGMAIN  DETAIL OUTPUT
                 |1868    $##005    00B5     RGSUBS  OUTPUT CONTROL ROUTINE
                 |1935    $##005    0043     RGSUBS  CONSTANTS
Overlay 005     < 1929    $##005    000C     RGSUBS  OUTPUT HOOK
                 |1A98    $##005    000C     @PGDM   DATA MANAGEMENT CALL
                 |191D    $##005    000C     RGSUBS  OUTPUT HOOK
                 (1A1C    $##005    007C     @PGCO   PACK
                  1FE3              000B     RGMAIN  TOTAL OUTPUT
                 (19A8    $##006    0024     RGMAIN  LR & OVERFLOW PROCESSING
                 |1868    $##006    00B5     RGSUBS  OUTPUT CONTROL ROUTINE
                 |1929    $##006    0043     RGSUBS  CONSTANTS
Overlay 006     < 196C    $##006    003C     RGSUBS  OVERFLOW SUBSEGMENT
                 |191D    $##006    000C     RGSUBS  OUTPUT HOOK
                 |19CC    $##006    000D     RGSUBS  SUBSEG
                 (19D9    $##006    000C     @PGDM   DATA MANAGEMENT CALL
                 (19A1    $##007    002E     RGMAIN  CLOSE MAINLINE
                 |1A06    $##007    0016     RGSUBS  TRANSFER VECTOR
                 |1868    $##007    00B5     RGSUBS  OUTPUT CONTROL ROUTINE
                 |195E    $##007    0043     RGSUBS  CONSTANTS
Overlay 007     < 1929    $##007    0035     RGSUBS  CONSTANTS
                 |19CF    $##007    002B     RGSUBS  LR OUTPUT
                 |191D    $##007    000C     RGSUBS  OUTPUT HOOK
                 (19FA    $##007    000C     @PGDM   DATA MANAGEMENT CALL
                 (19A8    $##008    00FA     RGMAIN  OPEN MAINLINE
                 |1ABB    $##008    0021     RGSUBS  TRANSFER VECTOR
                 |1965    $##008    0043     RGSUBS  CONSTANTS
Overlay 008     < 1868    $##008    00B5     RGSUBS  OUTPUT CONTROL ROUTINE
                 |1929    $##008    003C     RGSUBS  CONSTANTS
                 |191D    $##008    000C     RGSUBS  OUTPUT HOOK
                 |1AAF    $##008    000C     @PGDM   DATA MANAGEMENT CALL
                 (1AA2    $##008    000D     RGSUBS  SUBSEG

                                    08171    RPF148 MAIN STORAGE REQUIRED TO EXECUTE.
                                    08492    RPF148 MAIN STORAGE REQUIRED TO EXECUTE WITHOUT OVERLAYS.
                                 #LIBRARY    SOURCE MEMBER INPUT LIBRARY.
                                 #LIBRARY    LOAD MEMBER OUTPUT LIBRARY.
```

Figure 16-5. Overlay Usage Map

PERFORMANCE IMPROVEMENT TECHNIQUES

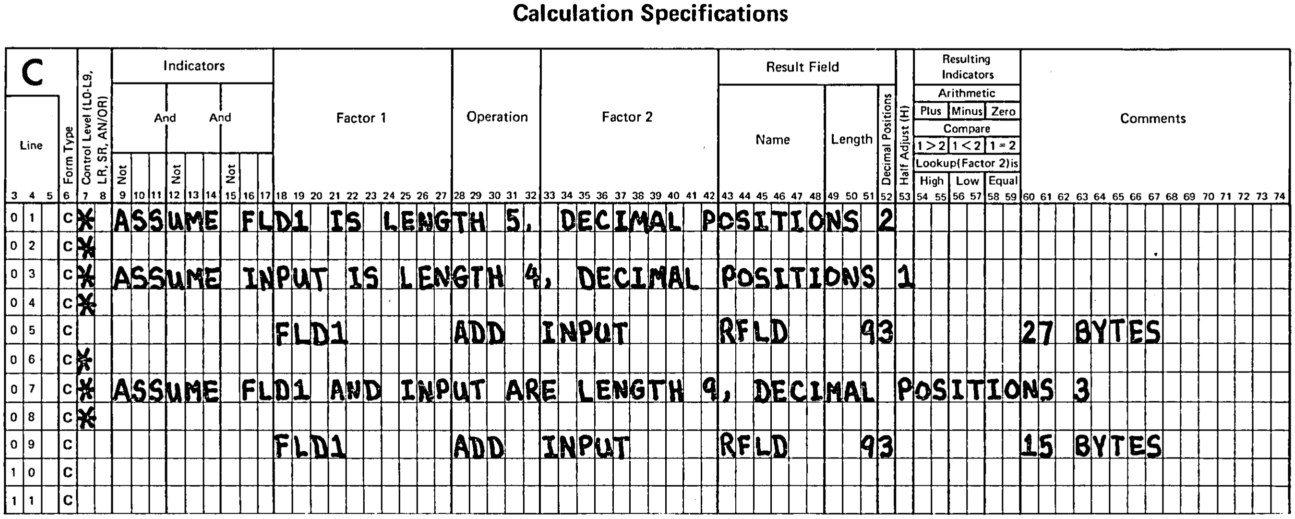
Some relatively simple program changes can make significant improvements in a program's performance. However, these performance techniques do not improve performance in all programs. Therefore, study these techniques and determine whether they can improve your program's performance. The performance improvement techniques are:

- Unblock all randomly processed indexed files. Blocking is not necessary because each record has its own index entry with the direct address of the record.
- Block all sequentially processed indexed files.
- Use the storage index. For a minimum cost in main storage, this allows the system to read the single track of indexes it needs rather than reading the entire index to look for an entry.
- Reduce or eliminate blocking of consecutive files and double the buffer instead. For example, instead of using a block of 1,600 bytes with 80-byte records, use a block of 800 bytes and a double buffer.

CODING TECHNIQUES FOR CALCULATION OPERATIONS

This section contains the number of bytes of object code generated for RPG II operation codes. When used with the preceding information in this chapter, this information helps you determine the amount of storage that you can save by using certain coding practices. For example, one storage saving technique is to use numeric fields of the same length and with the same number of decimal positions. If the fields cannot be the same length, try to have the number of decimal positions the same.

For example, Figure 16-6 shows that if the decimal positions of factor 1, factor 2, and the result field are all different, an ADD operation generates 27 bytes. However, if all the fields are defined as having the same number of decimal positions, the same ADD operation generates only 15 bytes. Uniformity of fields saves main storage not only for ADD and SUB, but also for most of the other arithmetic operations as well.



Calculation Specifications

```
01 C* ASSUME FLD1 IS LENGTH 5, DECIMAL POSITIONS 2
02 C*
03 C* ASSUME INPUT IS LENGTH 4, DECIMAL POSITIONS 1
04 C*
05 C           FLD1      ADD  INPUT     RFLD    93       27 BYTES
06 C*
07 C* ASSUME FLD1 AND INPUT ARE LENGTH 9, DECIMAL POSITIONS 3
08 C*
09 C           FLD1      ADD  INPUT     RFLD    93       15 BYTES
10 C
11 C
```

Figure 16-6. Bytes of Code Generated for ADD Operation

Abbreviations and symbols used in the following text are:

F1	Factor 1
F2	Factor 2
RF	Result field
L1	Total length of factor 1
L2	Total length of factor 2
LR	Total length of result field
D1	Number of decimal positions in factor 1
D2	Number of decimal positions in factor 2
DR	Number of decimal positions in result field
H/A	Half adjust
RAF	Record address file
=	Equal
≠	Not equal
-	Minus
>	Greater than
<	Less than
+	Plus

Operation	**Bytes**
SETON (each indicator set on)	3
SETOF (each indicator set off)	3
BITON	4
BITOF	4
TESTB	
Test bit off	10
Test bit mixed	17
Test bit on	10
Test bit off and mixed	23
Test bit off and on	23
Test bit mixed and on	23
Test bit off, mixed, and on	29
SUB	
F1 = RF and D1 = D2 = DR and L1 ≥ L2	6
F1 ≠ RF and D1 = D2 = DR	15
F1 ≠ RF and D2 = DR	23
F1 ≠ RF and D2 = DR H/A	27
All other combinations	31
All other combinations H/A	39
Z-SUB	
D2 = DR	14
D2 ≠ DR	18
D2 ≠ DR H/A	22

Operation	**Bytes**
ADD	
F1 = RF and D1 = D2 = DR	6
F2 = RF and D1 = D2 = DR	6
F1 ≠ F2 ≠ RF and D1 = D2 = DR	15
F1 = RF and D2 > DR	14
F2 = RF and D1 > DR	14
F1 = RF and D2 > DR H/A	18
F2 = RF and D1 > DR H/A	18
F1 = RF and D2 < DR	18
F2 = RF and D1 < DR	18
D1 = D2 < DR	23
F1 = RF and L2-D2 > L1-D1	27
F2 = RF and L1-D1 > L2-D2	27
F1 = RF and L2-D2 > L1-D1 H/A	35
F2 = RF and L1-D1 > L2-D2 H/A	35
All other combinations	27
All other combinations H/A	35
Z-ADD	
D2 = DR	6
D2 > DR	14
D2 > DR H/A	18
D2 < DR	18
COMP	
F1 and F2 are numeric and D1 = D2	10
F1 and F2 are numeric and D1 ≠ D2	18
F1 and F2 are alphameric and L1 = L2	6
F1 and F2 are alphameric and L1 ≠ L2	22
F1 and F2 are alphameric and F1 is a table	26
Alternate collating sequence (add these bytes to the appropriate COMP listed previously)	10
TESTZ	
RF is a field	9
RF is a table	20
MULT	23
with H/A	27
DIV	
D1 - D2 = DR	23
D1 - D2 ≠ DR	27
D1 - D2 = DR + 1 H/A	31
D1 - D2 ≠ DR + 1 H/A	35

Operation	Bytes
MVR	
D2 = DR	5
D2 ≠ DR	9
XFOOT	
D2 = DR	9
D2 ≠ DR	13
FORCE	
with external indicator	13 + 7 = 20
Conditioning indicators (does not apply to CHAIN, FORCE, LOKUP, and READ)	
each indicator	3
each AND type	3
Resulting indicators (does not apply to CHAIN, FORCE, LOKUP, and READ)	5
with each resulting indicator	3
CHAIN (base = 16)	
With external indicator	6
When F1 has a variable index	11
When key is not packed	14
When key is packed	23
When key is packed and F1 is a table element	6
When key is a record number	8
When key is a record number and F1 is a table element	6
When record-not-found indicator is given	1
When record-not-found indicator not given	16
READ (base = 29)	
With external indicator	6
With EOF indicator with BSCA	6
With EOF indicator without BSCA	12
With BSCA without EOF indicator	6
Without BSCA without EOF indicator	19
With RAF limits	6
LOKUP (base = 15)	
When F1 is a table	6
When F1 is a variable	11
With each resulting indicator	12
SORTA	
Inline calculation code	7
Subroutine	428

Operation	Bytes
KEYnn (base = 27)	
When RF is a variable indexed array	11
When RF is numeric, and a table element	6
with each resulting indicator	14
When RF is alphameric, and	0
with resulting indicator and field	
length > 1	23
length = 1	7
When F1 is numeric and with resulting indicator and field	
length > 1	8
length = 1	6
SETnn (base = 27)	
With ERASE function	4
When F1 is numeric and with resulting indicators and field	
length > 1	8
length = 1	6
SETnn/KEYnn combination (base = 27)	
See KEYnn operation for code in addition to base. If F1 code appears on both SET and KEY instructions, both counts should be included.	
SETLL (base = 18)	
When key is packed	12
EXSR	4
GOTO	4
TIME	
Time only	21
Time and system date	21
ACQ	
Inline calculation code	12
Subroutine	350
REL	
Inline calculation code	12
Subroutine	402
NEXT	
Inline calculation code	12
Subroutine	231
POST	
Inline calculation code	8
Subroutine	437

Operation		**Bytes**
SHTDN		22
MOVEA		
Inline calculation code		14
Subroutine		367
MOVE, MOVEL, MHHZO, MHLZO, MLHZO, MLLZO	The number of bytes specified includes all array control code lengths (see Figure 16-7).	

Array control code (initialization and processing) is generated for all calculations except LOKUP, CHAIN, READ, and FORCE.

Array initialization	
F1 or F2 is an array	6 bytes
F1 or F2 is a table	4 bytes
F1 or F2 is an array with variable index	11 bytes
Array processing	
F1, F2, RF are arrays	28 bytes
F1 and RF, F2 and RF arrays	22 bytes
RF arrays	16 bytes

If a SUB operation code is specified and has the following conditions:

F1 = RF
D1 = D2 = DR
F1 and RF = full array
F2 = table

the length of object code generated is as follows:

Array initialization	
F1 is an array	6 bytes
F2 is a table	4 bytes
RF is an array	6 bytes
SUB	6 bytes
Array processing	
F1 and RF are arrays	22 bytes

Thus, the total bytes of code generated for a SUB operation code is 44 bytes.

Whenever an array with a variable index is specified in a program (except with a MOVEA operation), the following are also generated:

Inline code	11 bytes
Subroutine	173 bytes

	MOVE Alphameric/Numeric	MOVEL LR < L2 Numeric	MOVEL LR > L2 Numeric	MOVEL LR = L2 Alphameric/Numeric	MOVEL LR < L2 Alphameric	MOVEL LR > L2 Alphameric	MLHZO	MHLZO	MHHZO	MLLZO
Field to Field	6	26	10	6	6	6	20	20	20	20
Array to Array	42	55	45	42	42	42	42	42	42	42
Field to Array	29	43	32	29	29	29	29	29	29	29
Table to Array	35	53	38	35	40	35	35	41	40	35
Array, Variable Index to Array	40	66	43	40	52	40	40	52	52	40
Array, Variable Index to Array, Variable Index	28	57	38	28	35	35	35	35	47	42
Field to Array, Variable Index	17	34	27	17	17	24	24	31	24	31
Table to Array, Variable Index	20	52	33	20	24	30	30	24	36	20
Array, Variable Index to Table	20	46	27	20	30	24	24	24	36	20
Field to Table	9	23	16	9	9	13	13	9	13	9
Table to Table	15	41	22	15	19	19	19	19	25	15
Array, Variable Index to Field	17	40	21	17	24	17	31	24	36	31
Table to Field	9	29	13	9	13	9	9	13	13	9

Figure 16-7. Bytes of Code Generated for MOVE Operations

Chapter 17. Detailed RPG II Program Logic

For each record that is processed, the RPG II object program goes through the same general cycle of operations. Within each program cycle, calculation and output operations can be performed at two different times: total time and detail time. First, total calculation and total output operations (those conditioned by control level indicators) are performed. Second, all detail calculation and detail output operations are performed. (Detail calculation and output operations are those not conditioned by control level indicators in columns 7 and 8 of the calculation specifications or a T in column 15 of the output specifications.) Total calculation and total output operations are performed on data accumulated for a control group. Detail calculation and detail output operations are performed for individual records as they are read, provided conditioning indicators are satisfied. See *RPG II Program Cycle* in Chapter 1, *Introduction*, for a general description of the logic flow.

The specific steps taken in one program cycle are shown in Figure 17-1. The item numbers in the following description refer to the numbers in the figure. The program cycle, which occurs for each record read, begins with step 3 and continues through step 26.

1. The system reads in the external indicators and the display station local data area, if specified, and *opens* all data files to be used by the RPG II object program; that is, the files are prepared to be processed by the object program. Data structures are blanked, and preexecution-time tables and arrays are loaded before the first program cycle.

2. The object program performs all output conditioned by the 1P (first page) indicator. This output is performed only once per job and does not fall within the program cycle (steps 3 through 26).

3. The object program performs all specified heading and detail output operations whose conditions are satisfied. This includes specifications that are conditioned by the overflow indicator if the overflow routine has been fetched.

4. The object program tests to determine whether the overflow line was encountered during detail calculations in the previous cycle or when heading and detail records were written in the current cycle. If so, the overflow indicator is set on. Otherwise, the indicator is set off unless the overflow routine was fetched in step 3.

5. The object program tests the halt indicators. If the halt indicators are off, the program branches to step 6.
   a. The execution of the program is stopped once for each halt indicator that is on. The operator selects one of three options: continue, controlled cancel, or immediate cancel.
   b. If the operator selects the option to continue the job, the program returns to step 5 to test for other halt indicators. If the operator selects one of the cancel options, the program branches to step 35.

6. All record identifying indicators and indicators 1P, L1 through L9, and H1 through H9 are set off.

7. The program tests to determine whether the LR indicator is on. If it is, the program branches to step 27.

8. The program tests to determine whether the KEYBORD is the primary file or if no primary file is specified. For either condition, the program branches to step 18.

9. The program reads (and translates, if necessary) the next input record. At the beginning of processing, one record from each input file (except forced files and demand files) is read. If the file has look-ahead fields, the file is read only on the first cycle. After that, records with look-ahead fields are identified only. If this is a WORKSTN file and the SAVDS and/or IND option is specified, the common SAVDS and/or IND area is moved to the active display station's SAVDS and/or IND hold area. The next record is accepted, and the current display station's SAVDS and/or IND area is moved from its hold area to the common SAVDS and/or IND area.

10. The program tests to determine whether the record is an end-of-file record. If an end-of-file condition has occurred, the program branches to step 12.

11. If end of file has not occurred, a test determines whether the input records are in the sequence specified on the input specifications sheet. If the sequence is incorrect, the program branches to step 33. The program also branches to step 33 if nonsequential input records are specified and the record cannot be identified.

12. If end-of-job conditions have been met, the program branches to step 27. All files for which an E is specified in column 17 of the file description specifications must be at end of file.

13. When multiple input files are used, the program must select the next record to process, so it branches to step 28.

14. If there is only one input file, no record selection is needed. A test determines whether sequence checking is requested. If so, the program branches to step 31.

15. The record identifying indicator specified for the current record type is set on. Data from the current record type is not available for processing until step 25.

16. If the record contains control fields, the object program tests to determine whether a control break has occurred (the contents of the control field are not equal to the contents of the previously stored control field). If a control break has not occurred or control fields are not specified, the program branches to step 18.

17. If a control break has occurred, the control level indicator reflecting the condition is set on. All lower level indicators also are set on.

18. A test is made to determine whether the total time calculations and total time output should be performed. If no control levels are specified on the input specifications, the totals are bypassed only on the first cycle. If control levels are specified on the input specifications, totals are bypassed until after the first record containing control fields is processed. Totals are always processed when the LR indicator is on.

19. All calculations conditioned by control level indicators (columns 7 and 8 of calculation specifications) are performed, and resulting indicators are set on or off as specified. If the LR indicator is on, calculations conditioned by LR are done after other total calculations. File translation, if specified, is done for exception output and CHAIN and READ operations. Fetch overflow is performed if it is required by exception output. If the overflow line has been reached because of the exception output, the overflow indicator is set on.

20. All total output that is not conditioned by an overflow indicator is performed. The program tests to determine whether an overflow condition has occurred. If an overflow condition has occurred at any time during this cycle, the overflow indicator is set on. If the LR indicator is on, output conditioned by LR is done after other total output. File translation, if specified, is done for total output. Fetch overflow is performed if required.

21. The program tests to determine whether the LR indicator is on. If the indicator is on, the program branches to step 38.

22. The program tests to determine whether any overflow indicators are on. If no overflow indicators are on, the program branches to step 24.

23. All output operations conditioned by a positive overflow indicator (no N preceding the indicator) are performed. File translation, if specified, is done for overflow output.

24. The MR indicator is set on if this is a multifile job and the record to be processed is a matching record. Otherwise, the MR indicator is set off.

25. Field indicators are set on or off as specified. Data from the last record read and from specified look-ahead fields is made available for processing. Command key indicators (KA through KN, KP through KY), for a WORKSTN file only, are set off, and if a command key is pressed for the WORKSTN file being processed, that command key indicator is set on.

26. Any calculations not conditioned by control level indicators (columns 7 and 8 of the calculation specifications) are performed, and resulting indicators are set on or off as specified. File translation, if specified, is done for exception output and CHAIN and READ operations. Fetch overflow is performed if it is required by exception output. If the overflow line is reached because of the exception output, the overflow indicator is set on. Processing continues with step 3.

27. The LR indicator and all control level indicators (L1 through L9) are set on and processing continues with step 19.

28. If a file was forced or if NEXT was specified, the next record in that file is selected for processing and the program branches to step 15.

29. If a record with no match fields is found in a normal input file that is not at end of file, the record is selected for processing.

30. When match fields are specified, the normal file with the highest priority matching record field is selected. If two or more files have equal and highest priority matching record fields, the highest priority file is selected. (The primary file has the highest file priority, the first specified secondary file is next, and so forth.)

31. The match field value is compared to the match field value of the last record. If it is in sequence, the record is accepted, and processing continues with step 15.

32. The execution of the program stops because a file with match fields is out of sequence. The operator's options, indicated in step 34, are to bypass (read the next record from the same file) or cancel the job.

33. The execution of the program stops because of a record type sequence error or an unidentified record.

34. The program tests the operator's decision either to bypass the record that caused the error condition (branch to step 4) or to cancel the job.

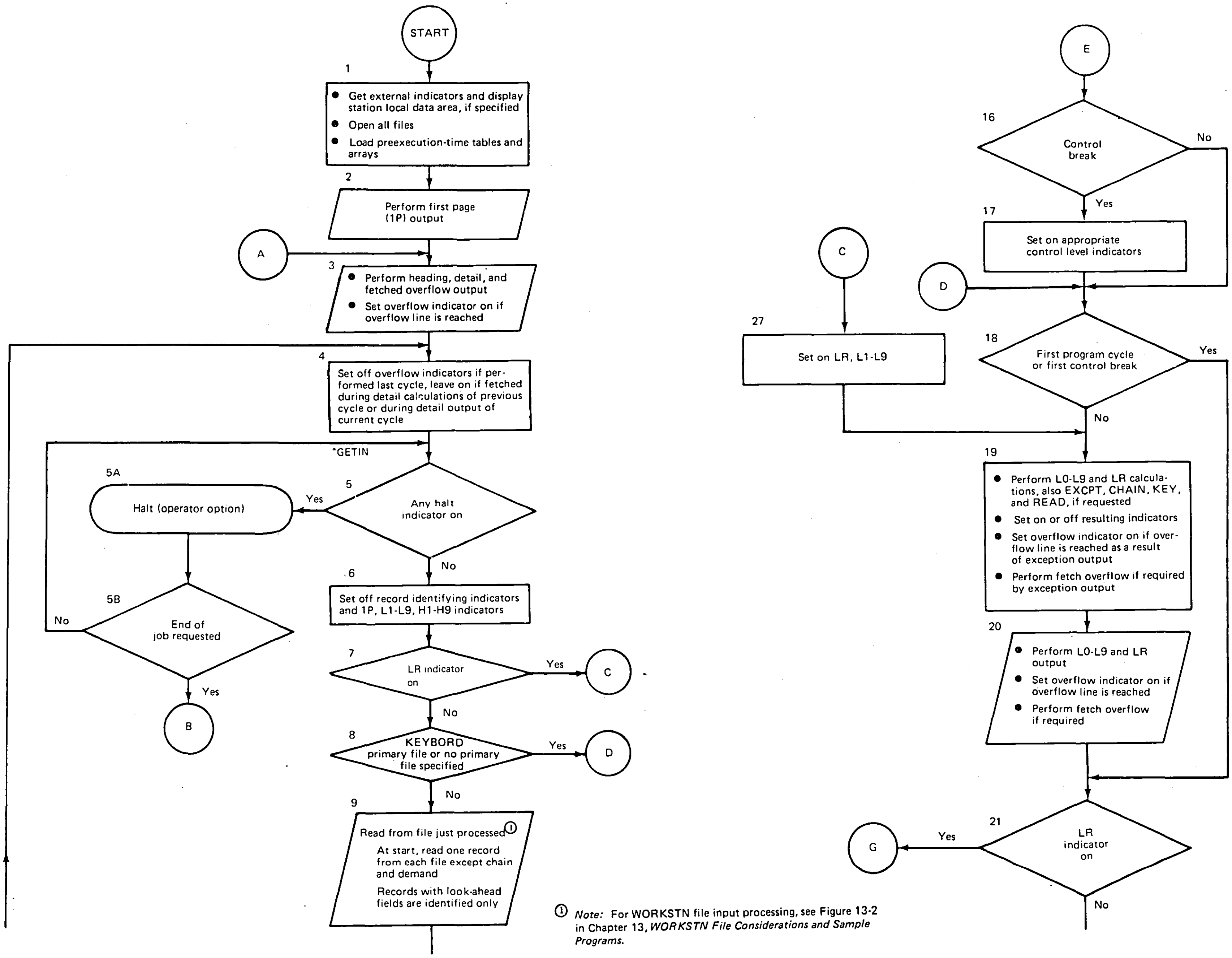
35. If the operator elects to terminate the job by means of a controlled cancel, steps 36 through 40 are performed. If the operator selects an immediate cancel, the job is terminated.

36 and 37. All operations conditioned by the LR indicator are done.

38. The program writes any tables or arrays for which a filename entry is specified on the extension specifications (columns 19 through 26). Output tables or arrays are translated, if necessary.

39. All files used by the program are closed (final termination functions are done). The external indicators and display station local data area, if specified, are written out.

40. End of job occurs.


START
1
Get external indicators and display station local data area, if specified
Open all files
Load preexecution-time tables and arrays
2
Perform first page (1P) output
A
3
Perform heading, detail, and fetched overflow output
Set overflow indicator on if overflow line is reached
4
Set off overflow indicators if performed last cycle, leave on if fetched during detail calculations of previous cycle or during detail output of current cycle
*GETIN
5
Any halt indicator on
Yes
5A
Halt (operator option)
5B
End of job requested
No
Yes
B
No
6
Set off record identifying indicators and 1P, L1-L9, H1-H9 indicators
7
LR indicator on
Yes
C
No
8
KEYBORD primary file or no primary file specified
Yes
D
No
9
Read from file just processed ①
At start, read one record from each file except chain and demand
Records with look-ahead fields are identified only
C
27
Set on LR, L1-L9
E
16
Control break
No
Yes
17
Set on appropriate control level indicators
D
18
First program cycle or first control break
Yes
No
19
Perform L0-L9 and LR calculations, also EXCPT, CHAIN, KEY, and READ, if requested
Set on or off resulting indicators
Set overflow indicator on if overflow line is reached as a result of exception output
Perform fetch overflow if required by exception output
20
Perform L0-L9 and LR output
Set overflow indicator on if overflow line is reached
Perform fetch overflow if required
21
LR indicator on
Yes
G
No

① *Note:* For WORKSTN file input processing, see Figure 13-2 in Chapter 13, *WORKSTN File Considerations and Sample Programs.*

Figure 17-1 (Part 1 of 2). Detailed RPG II Object Program Cycle

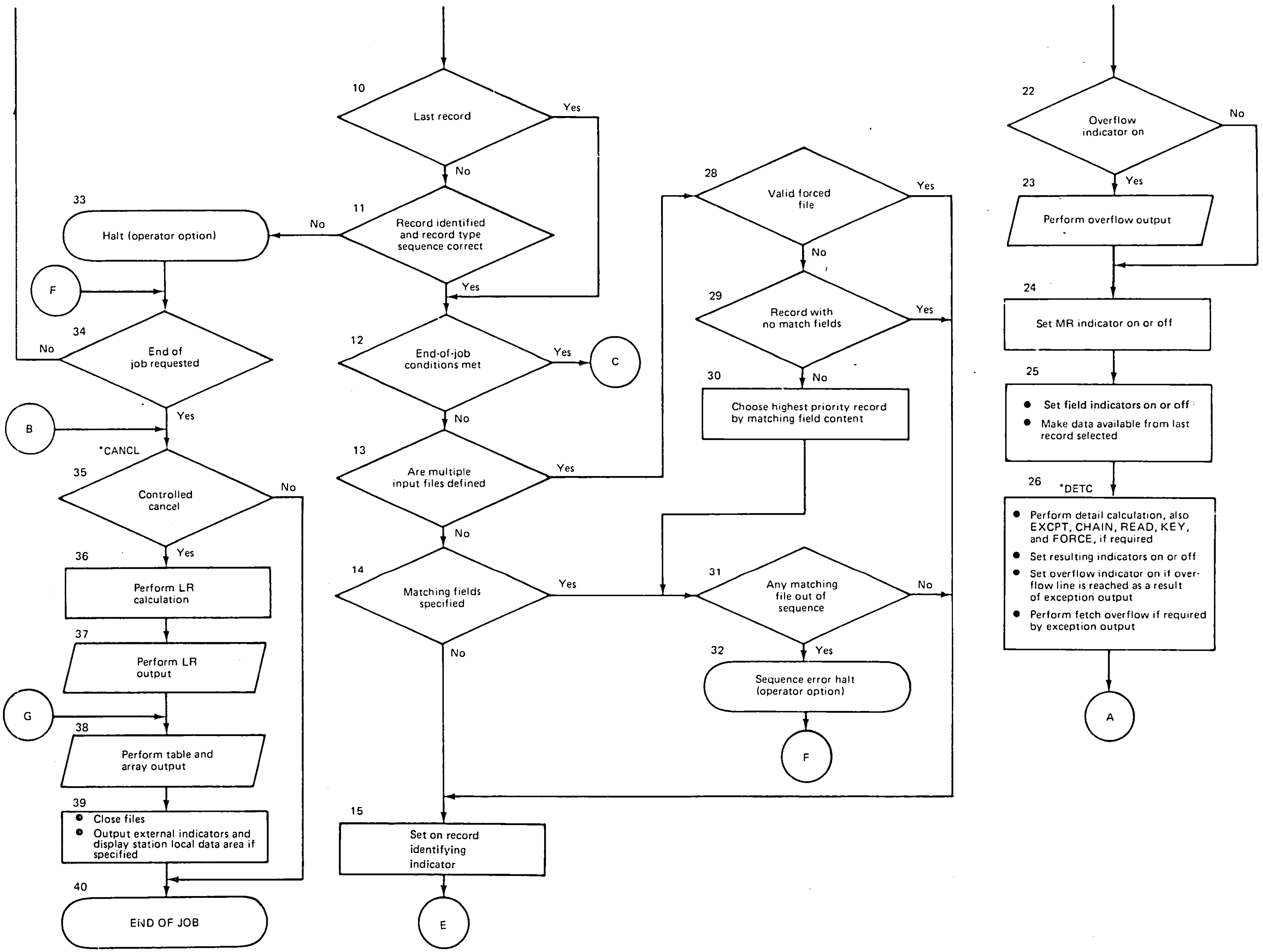

10
Last record
Yes
No
11
Record identified and record type sequence correct
No
Yes
33
Halt (operator option)
F
34
End of job requested
No
Yes
B
*CANCL
35
Controlled cancel
No
Yes
36
Perform LR calculation
37
Perform LR output
G
38
Perform table and array output
39
Close files
Output external indicators and display station local data area if specified
40
END OF JOB
12
End-of-job conditions met
Yes
C
No
13
Are multiple input files defined
Yes
No
14
Matching fields specified
Yes
No
15
Set on record identifying indicator
E
28
Valid forced file
Yes
No
29
Record with no match fields
Yes
No
30
Choose highest priority record by matching field content
31
Any matching file out of sequence
No
Yes
32
Sequence error halt (operator option)
F
22
Overflow indicator on
No
23
Yes
Perform overflow output
24
Set MR indicator on or off
25
Set field indicators on or off
Make data available from last record selected
26
*DETC
Perform detail calculation, also EXCPT, CHAIN, READ, KEY, and FORCE, if required
Set resulting indicators on or off
Set overflow indicator on if overflow line is reached as a result of exception output
Perform fetch overflow if required by exception output
A

Figure 17-1 (Part 2 of 2). Detailed RPG II Object Program Cycle

COMPILING THE RPG II SOURCE PROGRAM

RPG Command Statement

To compile an RPG II source program, the RPG II compiler must be loaded into main storage. To do this, the operator enters a command statement that executes an IBM-supplied library procedure named RPG. The operator can place the job on the input job queue either by specifying the appropriate option on the command statement (see the special options parameter) or by using the job queue command. (For information on the job queue command, see the *System Operator's Guide.*) The RPG command statement is:

```
RPG program name, [source file size] , [work file size] ,
                  [20              ]   [20            ]

[NOSTOP ]
[REPLACE]   [source program library name]
[NOHALT ] , [#LIBRARY                   ] ,
[HALT   ]

[object program library name] , [mrtmax value] ,
[#LIBRARY                   ]   [0           ]

        [NOGEN ]
[YES]   [GEN960] , [special options]
[NO ] , [GEN   ]   [00             ]
```

where:

program name is the name of the source program to be compiled. If this parameter is not specified and no other parameters are specified, the system displays a menu requesting the name of the source program and listing the default values for all the parameters. The default values can be overridden at this time. If this parameter is not specified but other parameters are specified, the system displays a menu requesting the name of the source program and listing the user-specified parameter values and the defaults for all unspecified parameters. These values can be overridden at this time. If the RPG procedure is placed on the input job queue, the program name must be specified. See the description of the special options parameter for an explanation of how to direct the RPG procedure to place the job on the input job queue after you enter values into the menu.

source file size is the number of blocks (each block is 2,560 bytes) for the $SOURCE file. If this is not specified, the default is 20 blocks.

work file size is the number of blocks for the $WORK file. If this is not specified, the default is 20 blocks. If the compiler is assigned a region size greater than or equal to 48K, a minimum of 30 blocks must be specified for the $WORK file for any size program.

HALT, the default for this parameter, specifies that the system halts for terminal diagnostics.

NOHALT specifies that the system does not halt for terminal diagnostics. REPLACE specifies that the system replaces an existing library member with the newly compiled object that has the same name. The replace is done automatically; no message is issued. NOSTOP combines the functions of NOHALT and REPLACE. The system replaces an existing library member with the newly compiled object that has the same name. The system does not halt for terminal diagnostics.

source program library name specifies the name of the library that contains the source program. If it is not specified, the system library, #LIBRARY, is assumed.

object program library name specifies the name of the library that will contain the compiled object program. If it is not specified, the system library, #LIBRARY, is assumed.

mrtmax value specifies the maximum number of active, requesting display stations that can be attached to the program. The mrtmax value can be a decimal number from 0 through 99. If the mrtmax value is 0 or if it is not specified, the object program is not an MRT program. If the value specified here is equal to or greater than 1, it can be overridden by an ATTR statement when the object program is executed.

Note: The mrtmax value must be less than or equal to the value specified for the NUM option on the file description specifications. If the mrtmax value equals the NUM value, the ATTR statement cannot increase the maximum number of allowable requestors unless the NUM value is also increased.

YES/NO specifies whether the object program is an NEP (never-ending program). YES specifies that the program is an NEP. If NO is specified or if the parameter is not used, the program is not executed as an NEP. The NEP attribute can be overridden by an ATTR statement when the object program is executed.

NOGEN specifies that the RPG II format generator will not be executed, and GEN960 specifies that the screen formats generated for the CONSOLE file will be for a 12-line, 960-character screen. If neither NOGEN nor GEN960 is specified, the RPG II format generator is executed to generate source input to the $SFGR utility program, which provides display screen formats for the CONSOLE file for a 24-line, 1920-character screen. The source specifications for the $SFGR utility are not saved when the RPG II format generator is executed by the RPG procedure. To save the source specifications for the 24-line, 1920-character displays, specify NOGEN on the RPG command statement and use the RPGR command statement.

special options (00) specifies which of the following special compile options are chosen:

0n: The job is not placed on the input job queue.

1n: The job is placed on the input job queue.

n0: A cross-reference listing of symbols is not provided.

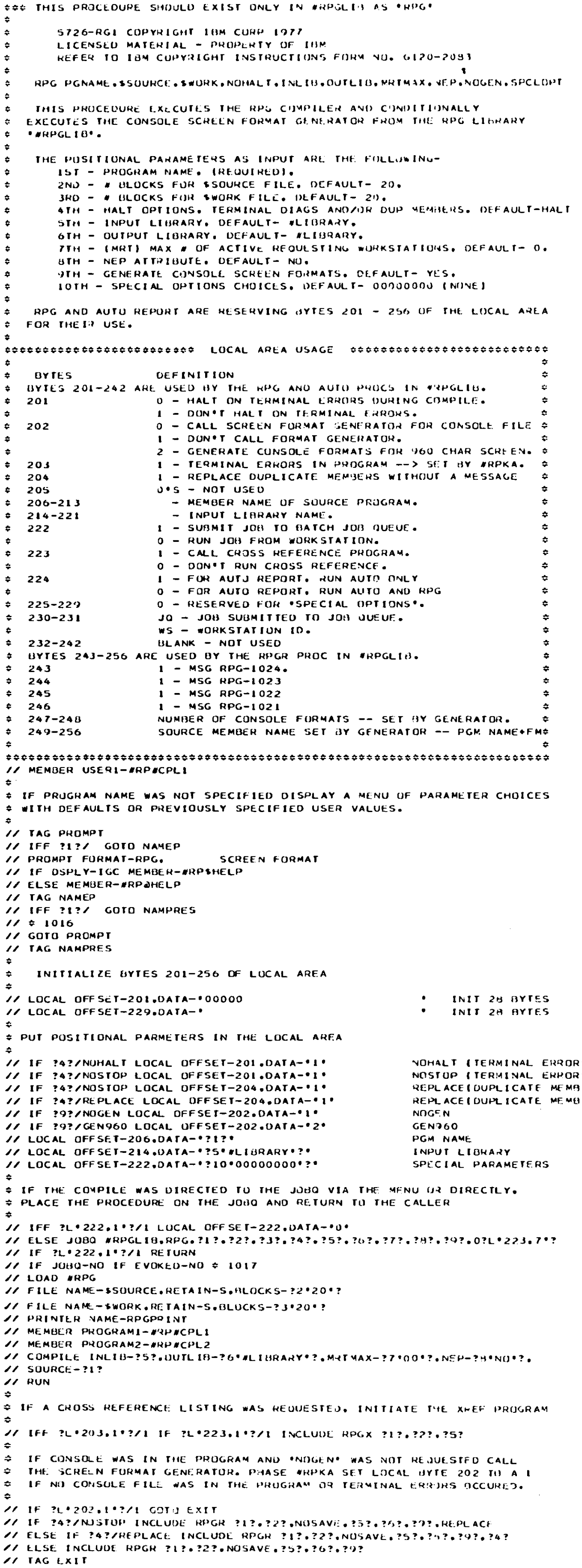
n1: A cross-reference listing of symbols is provided.

If special options are not specified, the default is 00.

For a complete description of the MRTMAX (xx) and NEP (yes/no) parameters, see the *System Support Reference Manual.*

Note: The RPG procedure reserves bytes 201 through 256 of the display station local data area for its use. Therefore, any user data in these bytes will be destroyed.

The RPG procedure exists in RPG's library #RPGLIB and is called by the RPG procedure that exists in the system library. Figure 18-1 shows the OCL statements included in the library procedure named RPG.



```
*** THIS PROCEDURE SHOULD EXIST ONLY IN #RPGLIB AS 'RPG'
*
*       5726-RG1 COPYRIGHT IBM CORP 1977
*       LICENSED MATERIAL - PROPERTY OF IBM
*       REFER TO IBM COPYRIGHT INSTRUCTIONS FORM NO. G120-2083
*
*   RPG PGNAME.$SOURCE.$WORK.NOHALT.INLIB.OUTLIB.MRTMAX.NEP.NOGEN.SPCLOPT
*
*   THIS PROCEDURE EXECUTES THE RPG COMPILER AND CONDITIONALLY
*  EXECUTES THE CONSOLE SCREEN FORMAT GENERATOR FROM THE RPG LIBRARY
*  '#RPGLIB'.
*
*   THE POSITIONAL PARAMETERS AS INPUT ARE THE FOLLOWING-
*       1ST - PROGRAM NAME. (REQUIRED).
*       2ND - # BLOCKS FOR $SOURCE FILE. DEFAULT- 20.
*       3RD - # BLOCKS FOR $WORK FILE. DEFAULT- 20.
*       4TH - HALT OPTIONS. TERMINAL DIAGS AND/OR DUP MEMBERS. DEFAULT-HALT
*       5TH - INPUT LIBRARY. DEFAULT- #LIBRARY.
*       6TH - OUTPUT LIBRARY. DEFAULT- #LIBRARY.
*       7TH - (MRT) MAX # OF ACTIVE REQUESTING WORKSTATIONS. DEFAULT- 0.
*       8TH - NEP ATTRIBUTE. DEFAULT- NO.
*       9TH - GENERATE CONSOLE SCREEN FORMATS. DEFAULT- YES.
*       10TH - SPECIAL OPTIONS CHOICES. DEFAULT- 00000000 (NONE)
*
*   RPG AND AUTO REPORT ARE RESERVING BYTES 201 - 256 OF THE LOCAL AREA
*  FOR THEIR USE.
*
*************************  LOCAL AREA USAGE  *************************
*                                                                     *
*   BYTES              DEFINITION                                     *
*  BYTES 201-242 ARE USED BY THE RPG AND AUTO PROCS IN #RPGLIB.       *
*  201                0 - HALT ON TERMINAL ERRORS DURING COMPILE.     *
*                     1 - DON'T HALT ON TERMINAL ERRORS.              *
*  202                0 - CALL SCREEN FORMAT GENERATOR FOR CONSOLE FILE *
*                     1 - DON'T CALL FORMAT GENERATOR.                *
*                     2 - GENERATE CONSOLE FORMATS FOR 960 CHAR SCREEN. *
*  203                1 - TERMINAL ERRORS IN PROGRAM --> SET BY #RPKA. *
*  204                1 - REPLACE DUPLICATE MEMBERS WITHOUT A MESSAGE  *
*  205                0'S - NOT USED                                  *
*  206-213              - MEMBER NAME OF SOURCE PROGRAM.              *
*  214-221              - INPUT LIBRARY NAME.                         *
*  222                1 - SUBMIT JOB TO BATCH JOB QUEUE.              *
*                     0 - RUN JOB FROM WORKSTATION.                   *
*  223                1 - CALL CROSS REFERENCE PROGRAM.               *
*                     0 - DON'T RUN CROSS REFERENCE.                  *
*  224                1 - FOR AUTO REPORT. RUN AUTO ONLY              *
*                     0 - FOR AUTO REPORT. RUN AUTO AND RPG           *
*  225-229            0 - RESERVED FOR 'SPECIAL OPTIONS'.             *
*  230-231            JQ - JOB SUBMITTED TO JOB QUEUE.                *
*                     WS - WORKSTATION ID.                            *
*  232-242            BLANK - NOT USED                                *
*  BYTES 243-256 ARE USED BY THE RPGR PROC IN #RPGLIB.                *
*  243                1 - MSG RPG-1024.                               *
*  244                1 - MSG RPG-1023                                *
*  245                1 - MSG RPG-1022                                *
*  246                1 - MSG RPG-1021                                *
*  247-248            NUMBER OF CONSOLE FORMATS -- SET BY GENERATOR.  *
*  249-256            SOURCE MEMBER NAME SET BY GENERATOR -- PGM NAME+FM*
*                                                                     *
***********************************************************************
// MEMBER USER1-#RP#CPL1
*
* IF PROGRAM NAME WAS NOT SPECIFIED DISPLAY A MENU OF PARAMETER CHOICES
* WITH DEFAULTS OR PREVIOUSLY SPECIFIED USER VALUES.
*
// TAG PROMPT
// IFF ?1?/  GOTO NAMEP
// PROMPT FORMAT-RPG.        SCREEN FORMAT
// IF DSPLY-IGC MEMBER-#RP$HELP
// ELSE MEMBER-#RP@HELP
// TAG NAMEP
// IFF ?1?/  GOTO NAMPRES
// * 1016
// GOTO PROMPT
// TAG NAMPRES
*
*   INITIALIZE BYTES 201-256 OF LOCAL AREA
*
// LOCAL OFFSET-201.DATA-'00000                            '   INIT 28 BYTES
// LOCAL OFFSET-229.DATA-'                                 '   INIT 28 BYTES
*
* PUT POSITIONAL PARMETERS IN THE LOCAL AREA
*
// IF ?4?/NOHALT LOCAL OFFSET-201.DATA-'1'              NOHALT (TERMINAL ERRORS)
// IF ?4?/NOSTOP LOCAL OFFSET-201.DATA-'1'              NOSTOP (TERMINAL ERRORS)
// IF ?4?/NOSTOP LOCAL OFFSET-204.DATA-'1'              REPLACE(DUPLICATE MEMBR)
// IF ?4?/REPLACE LOCAL OFFSET-204.DATA-'1'             REPLACE(DUPLICATE MEMBR)
// IF ?9?/NOGEN LOCAL OFFSET-202.DATA-'1'               NOGEN
// IF ?9?/GEN960 LOCAL OFFSET-202.DATA-'2'              GEN960
// LOCAL OFFSET-206.DATA-'?1?'                          PGM NAME
// LOCAL OFFSET-214.DATA-'?5'#LIBRARY'?'                INPUT LIBRARY
// LOCAL OFFSET-222.DATA-'?10'00000000'?'               SPECIAL PARAMETERS
*
* IF THE COMPILE WAS DIRECTED TO THE JOBQ VIA THE MENU OR DIRECTLY.
* PLACE THE PROCEDURE ON THE JOBQ AND RETURN TO THE CALLER
*
// IFF ?L'222.1'?/1 LOCAL OFFSET-222.DATA-'0'
// ELSE JOBQ #RPGLIB.RPG.?1?.?2?.?3?.?4?.?5?.?6?.?7?.?8?.?9?.0?L'223.7'?
// IF ?L'222.1'?/1 RETURN
// IF JOBQ-NO IF EVOKED-NO * 1017
// LOAD #RPG
// FILE NAME-$SOURCE.RETAIN-S.BLOCKS-?2'20'?
// FILE NAME-$WORK.RETAIN-S.BLOCKS-?3'20'?
// PRINTER NAME-RPGPRINT
// MEMBER PROGRAM1-#RP#CPL1
// MEMBER PROGRAM2-#RP#CPL2
// COMPILE INLIB-?5?.OUTLIB-?6'#LIBRARY'?.MRTMAX-?7'00'?.NEP-?8'NO'?.
// SOURCE-?1?
// RUN
*
* IF A CROSS REFERENCE LISTING WAS REQUESTED. INITIATE THE XREF PROGRAM
*
// IFF ?L'203.1'?/1 IF ?L'223.1'?/1 INCLUDE RPGX ?1?.?2?.?5?
*
*  IF CONSOLE WAS IN THE PROGRAM AND 'NOGEN' WAS NOT REQUESTED CALL
*  THE SCREEN FORMAT GENERATOR. PHASE #RPKA SET LOCAL BYTE 202 TO A 1
*  IF NO CONSOLE FILE WAS IN THE PROGRAM OR TERMINAL ERRORS OCCURED.
*
// IF ?L'202.1'?/1 GOTO EXIT
// IF ?4?/NOSTOP INCLUDE RPGR ?1?.?2?.NOSAVE.?5?.?6?.?9?.REPLACE
// ELSE IF ?4?/REPLACE INCLUDE RPGR ?1?.?2?.NOSAVE.?5?.?6?.?9?.?4?
// ELSE INCLUDE RPGR ?1?.?2?.NOSAVE.?5?.?6?.?9?
// TAG EXIT
```

Figure 18-1. IBM-Supplied Library Procedure (RPG) for Compiling an RPG II Source Program

RPGX Command Statement

The special compile options parameter (ab) on the RPG command statement allows you to specify that a cross-reference listing of symbols be provided for the program being compiled. The RPGX command statement allows you to request this cross-reference listing for a program that has already been successfully compiled. The RPGX command statement is:

```
                   ┌source and symbol file size┐
RPGX program name, │20                         │ ,
                   └                           ┘
┌source program library name┐
│#LIBRARY                   │
└                           ┘
```

where:

program name is the name of the RPG source program. This parameter is required. If it is not specified, a prompt requests the name of the source program.

source and symbol file size is the number of blocks (each block is 2,560 bytes) for the files used by the cross-reference programs. If the file size is not specified, the default is 20. You should specify the same value for this file size that you specified when the program was compiled.

source program library name specifies the name of the library containing the RPG II source program. If the library name is not specified, the system library, #LIBRARY, is assumed.

Note: No diagnostic checking is provided with the RPGX procedure. Therefore, you should use this command only for RPG II source programs that have been successfully compiled and for which object programs have been produced. Unpredictable or confusing results may occur if auto report source statements or RPG II source statements containing terminal errors are used as input to the RPGX procedure.

The RPGX procedure exists in RPG's library #RPGLIB and is called by the RPGX procedure that exists in the system library. Figure 18-2 shows the OCL statements included in the library procedure named RPGX.

Note: The RPGX procedure is not valid for auto report source. Auto report must be executed, producing RPG specifications, before the RPGX procedure can be executed.

RPGR Command Statement

If the NOGEN parameter is specified for the RPG procedure and the program contains a CONSOLE device, you can use the RPGR procedure to execute the RPG II format generator. The RPG II format generator produces and saves source input for the $SFGR utility program of the system support program product. The $SFGR utility program then produces the display screen formats for a CONSOLE file. The command statement for the RPGR procedure is:

```
                    ┌source and format file size┐
RPGR  program name, │20                         │ ,
                    └                           ┘
┌NOSAVE┐   ┌source program library name┐
│SAVE  │ , │#LIBRARY                   │ ,
└      ┘   └                           ┘
┌load module library name┐   ┌GEN960┐   ┌       ┐
│#LIBRARY                │ , │GEN   │ , │REPLACE│
└                        ┘   └      ┘   └       ┘
```

where:

program name is the name of the RPG II source program. This parameter is required. If it is not specified, a prompt requests the name of the source program.

source and format file size is the number of blocks (each block is 2,560 bytes) for the source file and SFGR file. If the file size is not specified, the default is 20.

NOSAVE specifies that the source statements for the $SFGR utility are not to be saved. If NOSAVE is not specified, the source statements are saved in the library specified as the source library and are assigned the program name plus FM.

source program library name specifies the name of the library that contains the RPG II source program. If it is not specified, the system library, #LIBRARY, is assumed.

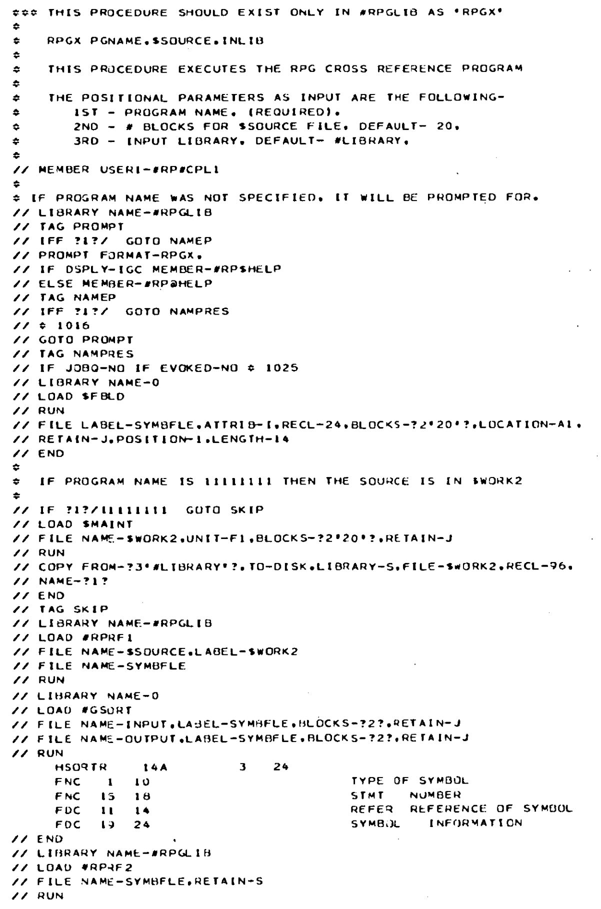
load module library name specifies the name of the library that will contain the load module created by the $SFGR utility program. If it is not specified, the system library, #LIBRARY, is assumed.

GEN960 specifies that the screen formats generated for the CONSOLE file will be for a 12-line, 960-character screen. If GEN960 is not specified, the screen format defaults to a 24-line, 1920-character screen.

REPLACE specifies that the system replaces an existing library member with the newly compiled object that has the same name. The replace is done automatically; no message is issued.

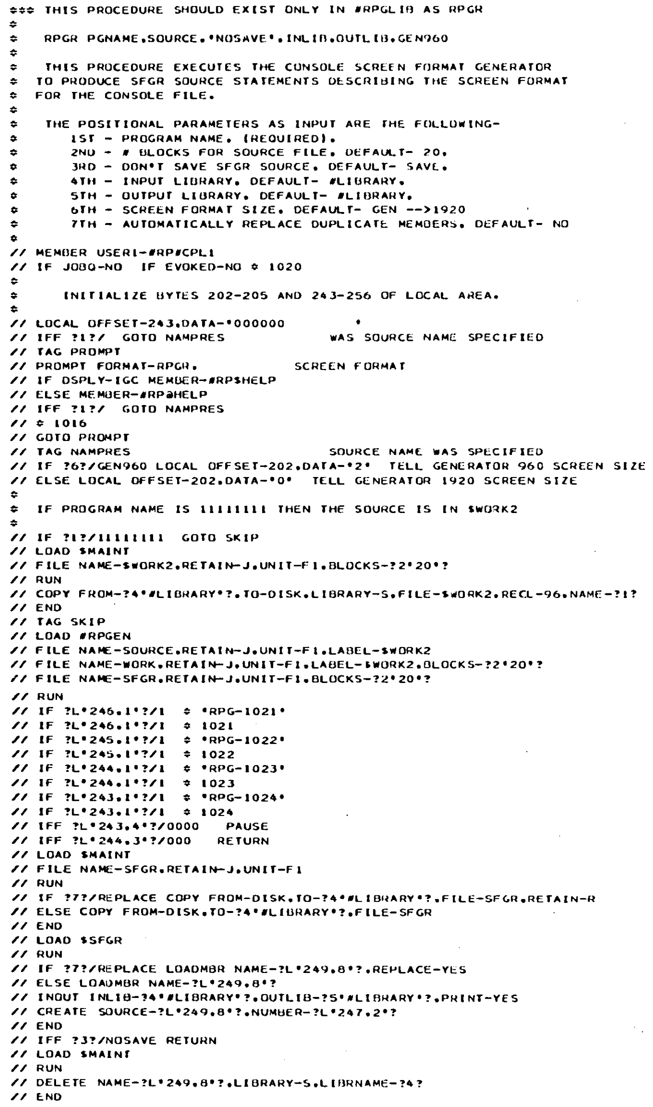
Note: The RPGR procedure reserves bytes 201 through 256 of the display station local data area for its use. Therefore, any user data in these bytes will be destroyed.

The RPGR procedure exists in RPG's library #RPGLIB and is called by the RPGR procedure that exists in the system library. Figure 18-3 shows the OCL statements included in the library procedure named RPGR.



```
*** THIS PROCEDURE SHOULD EXIST ONLY IN #RPGLIB AS 'RPGX'
*
*   RPGX PGNAME.$SOURCE.INLIB
*
*   THIS PROCEDURE EXECUTES THE RPG CROSS REFERENCE PROGRAM
*
*   THE POSITIONAL PARAMETERS AS INPUT ARE THE FOLLOWING-
*      1ST - PROGRAM NAME. (REQUIRED).
*      2ND - # BLOCKS FOR $SOURCE FILE. DEFAULT- 20.
*      3RD - INPUT LIBRARY. DEFAULT- #LIBRARY.
*
// MEMBER USER1-#RP#CPL1
*
* IF PROGRAM NAME WAS NOT SPECIFIED. IT WILL BE PROMPTED FOR.
// LIBRARY NAME-#RPGLIB
// TAG PROMPT
// IFF ?1?/  GOTO NAMEP
// PROMPT FORMAT-RPGX.
// IF DSPLY-IGC MEMBER-#RP$HELP
// ELSE MEMBER-#RP@HELP
// TAG NAMEP
// IFF ?1?/  GOTO NAMPRES
// * 1016
// GOTO PROMPT
// TAG NAMPRES
// IF JOBQ-NO IF EVOKED-NO * 1025
// LIBRARY NAME-0
// LOAD $FBLD
// RUN
// FILE LABEL-SYMBFLE.ATTRIB-I.RECL-24.BLOCKS-?2'20'?.LOCATION-A1.
// RETAIN-J.POSITION-1.LENGTH-14
// END
*
*  IF PROGRAM NAME IS 11111111 THEN THE SOURCE IS IN $WORK2
*
// IF ?1?/11111111  GOTO SKIP
// LOAD $MAINT
// FILE NAME-$WORK2.UNIT-F1.BLOCKS-?2'20'?.RETAIN-J
// RUN
// COPY FROM-?3'#LIBRARY'?.TO-DISK.LIBRARY-S.FILE-$WORK2.RECL-96.
// NAME-?1?
// END
// TAG SKIP
// LIBRARY NAME-#RPGLIB
// LOAD #RPRF1
// FILE NAME-$SOURCE.LABEL-$WORK2
// FILE NAME-SYMBFLE
// RUN
// LIBRARY NAME-0
// LOAD #GSORT
// FILE NAME-INPUT.LABEL-SYMBFLE.BLOCKS-?2?.RETAIN-J
// FILE NAME-OUTPUT.LABEL-SYMBFLE.BLOCKS-?2?.RETAIN-J
// RUN
     HSORTR    14A         3   24
     FNC   1  10                          TYPE OF SYMBOL
     FNC  15  18                          STMT   NUMBER
     FDC  11  14                          REFER  REFERENCE OF SYMBOL
     FDC  19  24                          SYMBOL    INFORMATION
// END
// LIBRARY NAME-#RPGLIB
// LOAD #RPRF2
// FILE NAME-SYMBFLE.RETAIN-S
// RUN
```

Figure 18-2. IBM-Supplied Library Procedure (RPGX) for Producing a Cross-Reference Listing



```
*** THIS PROCEDURE SHOULD EXIST ONLY IN #RPGLIB AS RPGR
*
*   RPGR PGNAME.SOURCE.'NOSAVE'.INLIB.OUTLIB.GEN960
*
*   THIS PROCEDURE EXECUTES THE CONSOLE SCREEN FORMAT GENERATOR
*  TO PRODUCE SFGR SOURCE STATEMENTS DESCRIBING THE SCREEN FORMAT
*  FOR THE CONSOLE FILE.
*
*   THE POSITIONAL PARAMETERS AS INPUT ARE THE FOLLOWING-
*      1ST - PROGRAM NAME. (REQUIRED).
*      2ND - # BLOCKS FOR SOURCE FILE. DEFAULT- 20.
*      3RD - DON'T SAVE SFGR SOURCE. DEFAULT- SAVE.
*      4TH - INPUT LIBRARY. DEFAULT- #LIBRARY.
*      5TH - OUTPUT LIBRARY. DEFAULT- #LIBRARY.
*      6TH - SCREEN FORMAT SIZE. DEFAULT- GEN -->1920
*      7TH - AUTOMATICALLY REPLACE DUPLICATE MEMBERS. DEFAULT- NO
*
// MEMBER USER1-#RP#CPL1
// IF JOBQ-NO  IF EVOKED-NO * 1020
*
*      INITIALIZE BYTES 202-205 AND 243-256 OF LOCAL AREA.
*
// LOCAL OFFSET-243.DATA-'000000         '
// IFF ?1?/  GOTO NAMPRES               WAS SOURCE NAME SPECIFIED
// TAG PROMPT
// PROMPT FORMAT-RPGR.             SCREEN FORMAT
// IF DSPLY-IGC MEMBER-#RP$HELP
// ELSE MEMBER-#RP@HELP
// IFF ?1?/  GOTO NAMPRES
// * 1016
// GOTO PROMPT
// TAG NAMPRES                          SOURCE NAME WAS SPECIFIED
// IF ?6?/GEN960 LOCAL OFFSET-202.DATA-'2'  TELL GENERATOR 960 SCREEN SIZE
// ELSE LOCAL OFFSET-202.DATA-'0'  TELL GENERATOR 1920 SCREEN SIZE
*
*  IF PROGRAM NAME IS 11111111 THEN THE SOURCE IS IN $WORK2
*
// IF ?1?/11111111  GOTO SKIP
// LOAD $MAINT
// FILE NAME-$WORK2.RETAIN-J.UNIT-F1.BLOCKS-?2'20'?
// RUN
// COPY FROM-?4'#LIBRARY'?.TO-DISK.LIBRARY-S.FILE-$WORK2.RECL-96.NAME-?1?
// END
// TAG SKIP
// LOAD #RPGEN
// FILE NAME-SOURCE.RETAIN-J.UNIT-F1.LABEL-$WORK2
// FILE NAME-WORK.RETAIN-J.UNIT-F1.LABEL-$WORK2.BLOCKS-?2'20'?
// FILE NAME-SFGR.RETAIN-J.UNIT-F1.BLOCKS-?2'20'?
// RUN
// IF ?L'246.1'?/1  * 'RPG-1021'
// IF ?L'246.1'?/1  * 1021
// IF ?L'245.1'?/1  * 'RPG-1022'
// IF ?L'245.1'?/1  * 1022
// IF ?L'244.1'?/1  * 'RPG-1023'
// IF ?L'244.1'?/1  * 1023
// IF ?L'243.1'?/1  * 'RPG-1024'
// IF ?L'243.1'?/1  * 1024
// IFF ?L'243.4'?/0000   PAUSE
// IFF ?L'244.3'?/000   RETURN
// LOAD $MAINT
// FILE NAME-SFGR.RETAIN-J.UNIT-F1
// RUN
// IF ?7?/REPLACE COPY FROM-DISK.TO-?4'#LIBRARY'?.FILE-SFGR.RETAIN-R
// ELSE COPY FROM-DISK.TO-?4'#LIBRARY'?.FILE-SFGR
// END
// LOAD $SFGR
// RUN
// IF ?7?/REPLACE LOADMBR NAME-?L'249.8'?.REPLACE-YES
// ELSE LOADMBR NAME-?L'249.8'?
// INOUT INLIB-?4'#LIBRARY'?.OUTLIB-?5'#LIBRARY'?.PRINT-YES
// CREATE SOURCE-?L'249.8'?.NUMBER-?L'247.2'?
// END
// IFF ?3?/NOSAVE RETURN
// LOAD $MAINT
// RUN
// DELETE NAME-?L'249.8'?.LIBRARY-S.LIBRNAME-?4?
// END
```

Figure 18-3. IBM-Supplied Library Procedure (RPGR) for Executing the RPG II Format Generator

COMPILING THE AUTO REPORT SOURCE PROGRAM

AUTO Command Statement

To compile an RPG II source program that includes auto report specifications, the auto report program must be loaded into main storage. To load the auto report program, the operator enters the following command statement that executes an IBM-supplied library procedure named AUTO:

```
AUTO program name, [source file size] ,
                   [20              ]

[work file size] , [NOSTOP ] ,
[20            ]   [REPLACE]
                   [NOHALT ]
                   [HALT   ]

[source program library name] ,
[#LIBRARY                   ]

[object program library name] , [mrtmax value] ,
[#LIBRARY                   ]   [0           ]

[YES] , [NOGEN ] , [special options]
[NO ]   [GEN960]   [000            ]
        [GEN   ]
```

where:

the positional parameters are the same as for the RPG procedure. The last positional parameter, special options, has an additional option for auto report. If the third special options position contains a zero (the default), the RPG compiler is called after the auto report specifications are generated. If the third special options postion contains a one, the RPG compiler is not called after the auto report specifications are generated. The AUTO procedure exists in RPG's library #RPGLIB and is called by the AUTO procedure that exists in the system library.

If the source program name is not specified and no other parameters are specified, the system displays a menu requesting the name of the source program and listing the default values for all the parameters. The default values can be overridden at this time. If the source program name is not specified but other parameters are specified, the system displays a menu requesting the name of the source program and listing the user-specified parameter values and the defaults for all unspecified parameters. These values can be overridden at this time. If the AUTO procedure is placed on the input job queue, the program name must be specified. See the description of the special options parameter for the RPG procedure for an explanation of how to direct the AUTO procedure to place the job on the input job queue after you enter values into the menu.

If the default block size of 20 is not going to be used in the command statement, determine the blocks required as follows:

$$\text{Blocks} = \frac{\text{Number of specifications}}{25}$$

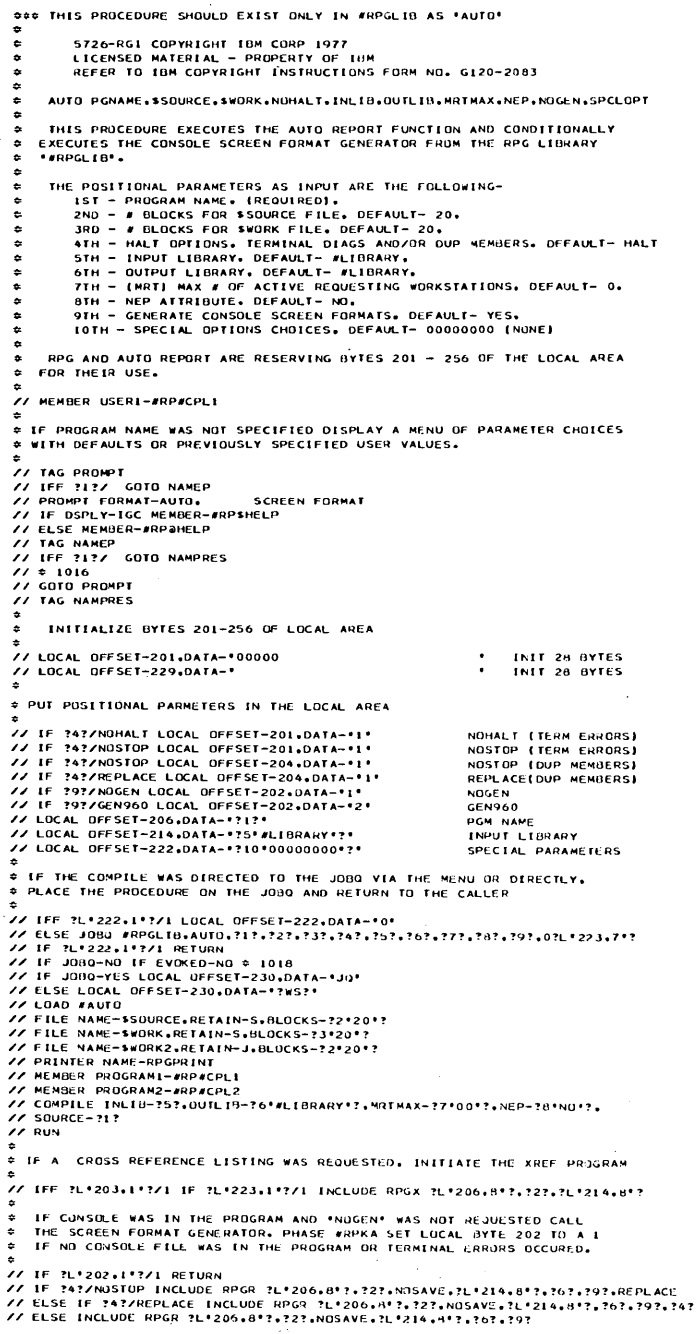
For the number of specifications, use the greater of: the number of specifications read by auto report or the estimated number of specifications in the generated source program. The calculated number of blocks should be used for both $SOURCE and $WORK. If the compiler is assigned a region size greater than or equal to 48K, a minimum of 30 blocks must be specified for the $WORK file for any size program.

The AUTO procedure builds a source file for processing by RPG and RPGR, and provides a cross-reference listing, if requested. This intermediate source is saved if you specify in the option specifications that the intermediate source is to be cataloged. If the RPGR or RPGX procedures were requested from the AUTO procedure and you did not specify that the intermediate source is to be cataloged, the intermediate source is saved in a work file named $WORK2. $WORK2 is a job file and is deleted at the end of the AUTO procedure execution.

Note: If the catalog option is not specified in the option specifications, the library name printed by the compiler is where the auto report source originated. Otherwise, the library name printed is the library name specified in the catalog option.

Figure 18-4 shows the OCL statements included in the library procedure named AUTO.

Note: The AUTO procedure reserves bytes 201 through 256 of the display station local data area for its use. Therefore, any user data in these bytes will be destroyed.



```
*** THIS PROCEDURE SHOULD EXIST ONLY IN #RPGLIB AS 'AUTO'
*
*      5726-RG1 COPYRIGHT IBM CORP 1977
*      LICENSED MATERIAL - PROPERTY OF IBM
*      REFER TO IBM COPYRIGHT INSTRUCTIONS FORM NO. G120-2083
*
*   AUTO PGNAME.$SOURCE.$WORK.NOHALT.INLIB.OUTLIB.MRTMAX.NEP.NOGEN.SPCLOPT
*
*   THIS PROCEDURE EXECUTES THE AUTO REPORT FUNCTION AND CONDITIONALLY
*  EXECUTES THE CONSOLE SCREEN FORMAT GENERATOR FROM THE RPG LIBRARY
*  '#RPGLIB'.
*
*   THE POSITIONAL PARAMETERS AS INPUT ARE THE FOLLOWING-
*      1ST - PROGRAM NAME. (REQUIRED).
*      2ND - # BLOCKS FOR $SOURCE FILE. DEFAULT- 20.
*      3RD - # BLOCKS FOR $WORK FILE. DEFAULT- 20.
*      4TH - HALT OPTIONS. TERMINAL DIAGS AND/OR DUP MEMBERS. DEFAULT- HALT
*      5TH - INPUT LIBRARY. DEFAULT- #LIBRARY.
*      6TH - OUTPUT LIBRARY. DEFAULT- #LIBRARY.
*      7TH - (MRT) MAX # OF ACTIVE REQUESTING WORKSTATIONS. DEFAULT- 0.
*      8TH - NEP ATTRIBUTE. DEFAULT- NO.
*      9TH - GENERATE CONSOLE SCREEN FORMATS. DEFAULT- YES.
*      10TH - SPECIAL OPTIONS CHOICES. DEFAULT- 00000000 (NONE)
*
*   RPG AND AUTO REPORT ARE RESERVING BYTES 201 - 256 OF THE LOCAL AREA
*  FOR THEIR USE.
*
// MEMBER USER1-#RP#CPL1
*
* IF PROGRAM NAME WAS NOT SPECIFIED DISPLAY A MENU OF PARAMETER CHOICES
* WITH DEFAULTS OR PREVIOUSLY SPECIFIED USER VALUES.
*
// TAG PROMPT
// IFF ?1?/  GOTO NAMEP
// PROMPT FORMAT-AUTO.        SCREEN FORMAT
// IF DSPLY-IGC MEMBER-#RP$HELP
// ELSE MEMBER-#RP@HELP
// TAG NAMEP
// IFF ?1?/  GOTO NAMPRES
// * 1016
// GOTO PROMPT
// TAG NAMPRES
*
*   INITIALIZE BYTES 201-256 OF LOCAL AREA
*
// LOCAL OFFSET-201.DATA-'00000                      '    INIT 28 BYTES
// LOCAL OFFSET-229.DATA-'                           '    INIT 28 BYTES
*
* PUT POSITIONAL PARMETERS IN THE LOCAL AREA
*
// IF ?4?/NOHALT LOCAL OFFSET-201.DATA-'1'             NOHALT (TERM ERRORS)
// IF ?4?/NOSTOP LOCAL OFFSET-201.DATA-'1'             NOSTOP (TERM ERRORS)
// IF ?4?/NOSTOP LOCAL OFFSET-204.DATA-'1'             NOSTOP (DUP MEMBERS)
// IF ?4?/REPLACE LOCAL OFFSET-204.DATA-'1'            REPLACE(DUP MEMBERS)
// IF ?9?/NOGEN LOCAL OFFSET-202.DATA-'1'              NOGEN
// IF ?9?/GEN960 LOCAL OFFSET-202.DATA-'2'             GEN960
// LOCAL OFFSET-206.DATA-'?1?'                         PGM NAME
// LOCAL OFFSET-214.DATA-'?5'#LIBRARY'?'               INPUT LIBRARY
// LOCAL OFFSET-222.DATA-'?10'00000000'?'              SPECIAL PARAMETERS
*
* IF THE COMPILE WAS DIRECTED TO THE JOBQ VIA THE MENU OR DIRECTLY.
* PLACE THE PROCEDURE ON THE JOBQ AND RETURN TO THE CALLER
*
// IFF ?L'222.1'?/1 LOCAL OFFSET-222.DATA-'0'
// ELSE JOBQ #RPGLIB.AUTO.?1?.?2?.?3?.?4?.?5?.?6?.?7?.?8?.?9?.0?L'223.7'?
// IF ?L'222.1'?/1 RETURN
// IF JOBQ-NO IF EVOKED-NO * 1018
// IF JOBQ-YES LOCAL OFFSET-230.DATA-'JQ'
// ELSE LOCAL OFFSET-230.DATA-'?WS?'
// LOAD #AUTO
// FILE NAME-$SOURCE.RETAIN-S.BLOCKS-?2'20'?
// FILE NAME-$WORK.RETAIN-S.BLOCKS-?3'20'?
// FILE NAME-$WORK2.RETAIN-J.BLOCKS-?2'20'?
// PRINTER NAME-RPGPRINT
// MEMBER PROGRAM1-#RP#CPL1
// MEMBER PROGRAM2-#RP#CPL2
// COMPILE INLIB-?5?.OUTLIB-?6'#LIBRARY'?.MRTMAX-?7'00'?.NEP-?8'NO'?.
// SOURCE-?1?
// RUN
*
* IF A  CROSS REFERENCE LISTING WAS REQUESTED. INITIATE THE XREF PROGRAM
*
// IFF ?L'203.1'?/1 IF ?L'223.1'?/1 INCLUDE RPGX ?L'206.8'?.?2?.?L'214.8'?
*
*   IF CONSOLE WAS IN THE PROGRAM AND 'NOGEN' WAS NOT REQUESTED CALL
*   THE SCREEN FORMAT GENERATOR. PHASE #RPKA SET LOCAL BYTE 202 TO A 1
*   IF NO CONSOLE FILE WAS IN THE PROGRAM OR TERMINAL ERRORS OCCURED.
*
// IF ?L'202.1'?/1 RETURN
// IF ?4?/NOSTOP INCLUDE RPGR ?L'206.8'?.?2?.NOSAVE.?L'214.8'?.?6?.?9?.REPLACE
// ELSE IF ?4?/REPLACE INCLUDE RPGR ?L'206.8'?.?2?.NOSAVE.?L'214.8'?.?6?.?9?.?4?
// ELSE INCLUDE RPGR ?L'206.8'?.?2?.NOSAVE.?L'214.8'?.?6?.?9?
```

Figure 18-4. IBM-Supplied Library Procedure (AUTO) for Compiling an Auto Report Source Program

Auto Report Halts

Auto report does not diagnose all error conditions in the source program. Diagnostics that are performed by the RPG II compiler are not duplicated by auto report. If a program cannot be successfully generated because of errors in the auto report specifications, auto report halts. Only option 3 (immediate cancel) is available following this halt.

If an RPG II source program is successfully generated, auto report calls the RPG II compiler without halting. Normal RPG II compilation halts can occur after compilation has begun.

CROSS-REFERENCE LISTING

The cross-reference option in the special compile options parameter (the ab parameter) of the RPG and AUTO command statements provides a cross-reference listing of the symbols defined and referenced in the respective RPG II and auto report source programs. The execution of the cross-reference listing step in the RPG or AUTO procedure depends upon the following:

- The listing is provided only when specified by the special options parameter (the ab parameter) of the RPG or AUTO command statement. The default is no cross-reference listing.
- The listing is not provided if terminal errors occur in the RPG or auto report compilation.
- The Sort utility program product is required to sort the symbol entries and provide a cross-reference listing.

The symbols used in an RPG II or auto report program are sorted and placed in the following categories in the cross-reference listing:

- Filenames
- Indicators
- Tables and arrays
- Fields and data structures
- Labels

Listing Format

The format of the cross-reference listing is as follows:

```
SYMBOL   LNG  TYPE  DEC  DEFN  REFERENCES
x──────►x nnnn xx──►x n    nnnn  nnnn nnnn nnnn*
```

where:

SYMBOL is from 1 to 8 characters in length and defines the filenames, indicators, tables/arrays, data structures, fields, and labels used in the RPG II or auto report program. Alphameric and numeric literals are not processed by the cross-reference listing option.

LNG is four positions long and defines the length of the field or data structure, the length of an element in a table or array, or the record length for the file named. LNG is not used for indicators or labels.

TYPE is 2 to 7 positions in length and defines the type of file named (by using columns 15 and 16 from the file description specifications) or the type of label being defined and referenced. TYPE is used only for filenames or labels.

DEC is one position long and defines the number of decimal positions in a numeric field. DEC is not used for filenames, alphameric fields, indicators, data structures, or labels.

DEFN is four positions long and defines the statement number in which the symbol is defined. If the symbol is defined multiple times in the program, the first definition is assumed. The use of a field in a data structure is considered to be the definition of that field; all other uses of that field are considered to be references. The definition of an array is considered to be in the extension specification specifying the array even if the array is also specified in a data structure.

REFERENCES are four positions long and define the statement number in which the symbol is referenced. The number of entries under REFERENCES depends on the number of times the symbol is used in the program. If the symbol is unreferenced, there are no entries. If the symbol is referenced multiple times, multiple lines of references could be printed for the related symbol. An asterisk (*) printed beside a reference indicates that the contents of the symbol are, or could be, altered in this statement. An asterisk indicates that a field is used as a calculation result field, or that an indicator is specified in positions 59 through 70 of the input specifications or in positions 54 through 59 of the calculation specifications.

Figure 18-5 shows an example of the information that is printed in the cross-reference listing for each symbol type.

RUNNING THE OBJECT PROGRAM

To load and run an RPG II object program, the operator must enter certain OCL statements, enter the name of a user-written procedure, or place the job on the input job queue.

LOAD and RUN OCL statements are required. If the program uses disk files, a FILE statement is required for each disk file. A SWITCH statement can be included in the OCL stream to set any external indicators used by the program. A display station can also be attached to an RPG II program that uses the WORKSTN device by the WORKSTN OCL statement. For a complete description of the OCL statements and their parameters and on how to write a procedure, see the *System Support Reference Manual.* For information on how to place the job on the input job queue, see the *System Operator's Guide.*

As an example, the following OCL statements load and run an RPG II object program named PROG1 that uses an input disk file named INPUT and an output disk file named OUTPUT:

```
//ƀLOADƀPROG1

//ƀFILEƀNAME-INPUT

//ƀFILEƀNAME-OUTPUT,BLOCKS-10,RETAIN-P

//ƀRUN
```

RPG II HALT PROCEDURES

Error conditions found in the RPG II program result in a halt during execution or compilation of the program. If the RPG II job is run from the input job queue, the halts go to the system operator. If the job is run from a display station (and not placed on the input job queue), the halts go to the display station operator.

Options available to the operator following each halt are also given. The options are:

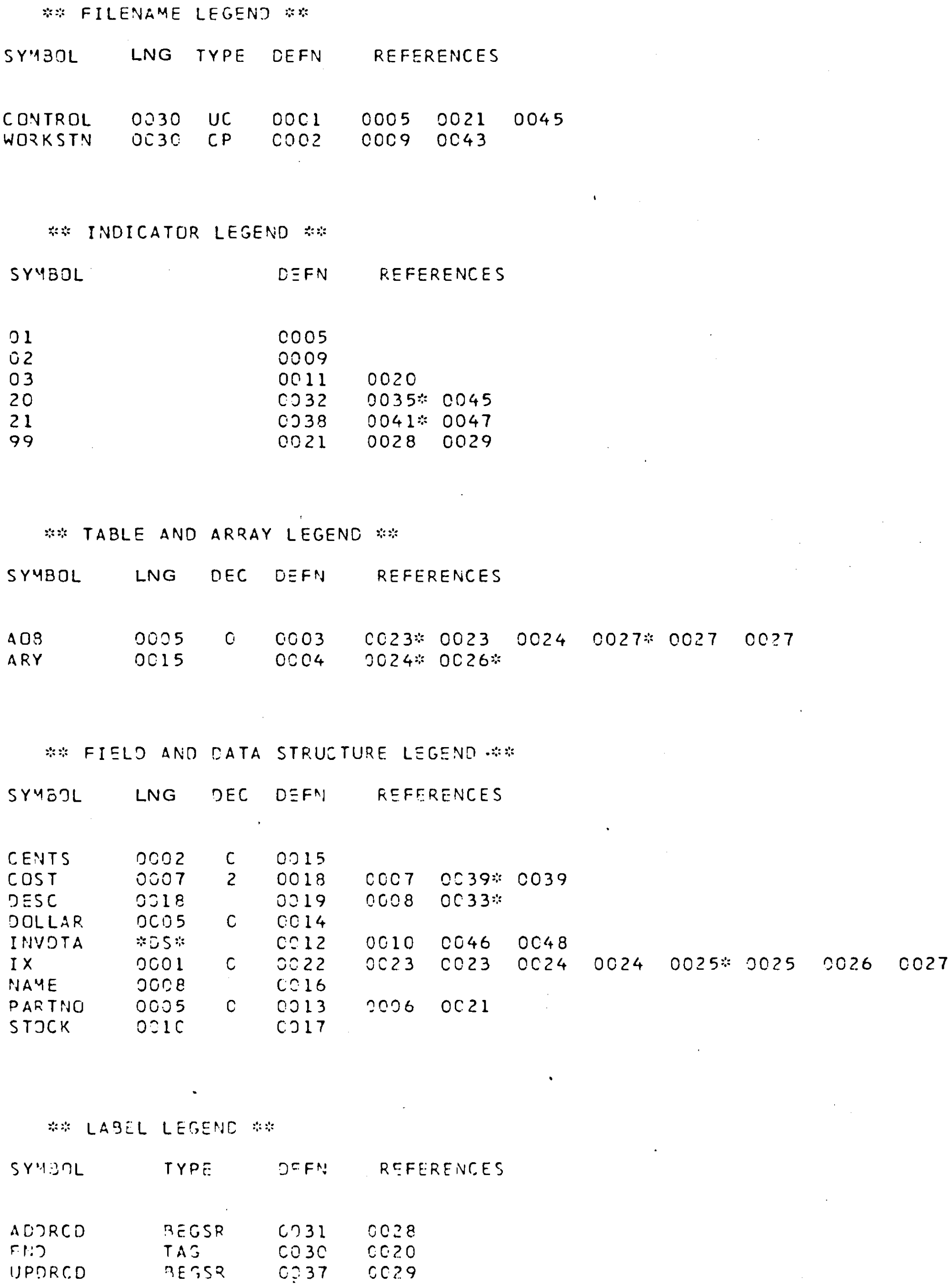
0-Continue: Control is returned to the program, and processing continues.

1-Bypass: The remainder of the program cycle is bypassed, and the next record is read.

2-Controlled Cancel: End-of-job operations specified by the program are done, tables are dumped, and file labels are cataloged.

3-Immediate Cancel: The job is canceled, but control is not returned to the RPG II program. New data entered for this job is not preserved.

A complete discussion of the halts and the necessary operator procedures appears in the *Displayed Messages Guide.*



```
   ** FILENAME LEGEND **

SYMBOL     LNG  TYPE  DEFN     REFERENCES


CONTROL    0030  UC   0001   0005  0021  0045
WORKSTN    0030  CP   0002   0009  0043



   ** INDICATOR LEGEND **

SYMBOL                DEFN     REFERENCES


01                    0005
02                    0009
03                    0011   0020
20                    0032   0035* 0045
21                    0038   0041* 0047
99                    0021   0028  0029



   ** TABLE AND ARRAY LEGEND **

SYMBOL     LNG   DEC  DEFN     REFERENCES


A08        0005   0   0003   0023* 0023  0024  0027* 0027  0027
ARY        0015       0004   0024* 0026*



   ** FIELD AND DATA STRUCTURE LEGEND **

SYMBOL     LNG   DEC  DEFN     REFERENCES


CENTS      0002   0   0015
COST       0007   2   0018   0007  0039* 0039
DESC       0018       0019   0008  0033*
DOLLAR     0005   0   0014
INVDTA     *DS*       0012   0010  0046  0048
IX         0001   0   0022   0023  0023  0024  0024  0025* 0025  0026  0027
NAME       0008       0016
PARTNO     0005   0   0013   0006  0021
STOCK      0010       0017



   ** LABEL LEGEND **

SYMBOL         TYPE      DEFN     REFERENCES


ADDRCD         BEGSR     0031   0028
END            TAG       0030   0020
UPDRCD         BEGSR     0037   0029
```

Figure 18-5. Symbol Information Provided by the Cross-Reference Listing

Chapter 19. Sample Programs

The sample programs contained in this chapter are designed to illustrate some of the functions of RPG II. Although they are relatively simple programs, they are complete jobs that can be run on any System/34. For additional sample programs that use the WORKSTN file, see Chapter 13.

SAMPLE PROGRAM 1 (SAMPL1)

SAMPL1 loads an indexed disk file that consists of 100 data records created by calculation operations. (Each record contains two fields: COUNT and RECNBR.) The program only requires the operator to enter a blank data record at the beginning of the job and press command key 12, that is, the Cmd and = (equal) keys, to end the job when the first prompt appears for the field EOF. SAMPL1 should be followed by SAMPL2, which prints the indexed file loaded in SAMPL1 to verify that the file was loaded properly. Figure 19-1 shows the specification sheets required for SAMPL1.

Control Specifications

Control specifications (see Figure 19-1, Part 1) should be present for every program. They are the first record in the source program and identify the program.

File Description Specifications

All files used in SAMPL1 are first described on the file description specifications sheet (see Figure 19-1, Part 1). The primary input file, INPUT, is assigned to the device CONSOLE. The E in column 17 ensures that the program does not end until after the last record is read from INPUT. At the end of the job, the indexed output file, DISKOUT, consists of 128-position records with a 6-position key field starting in the first record position. Messages indicating that the job was completed successfully are written to the printer output file, OUTPUT, at the end of job.

Input Specifications

The input file, INPUT, is further described on the input specifications sheet (see Figure 19-1, Part 1). The input record contains a 1-position blank field called NODATA (blank in position 1) describing the record identification code. The field NODATA is not prompted. A 1-position field called EOF will be prompted and is described for position 2 of the input record. When the prompt for EOF is made and command key 12 (the Cmd and = keys) is pressed (end of file for CONSOLE file), the LR indicator turns on.

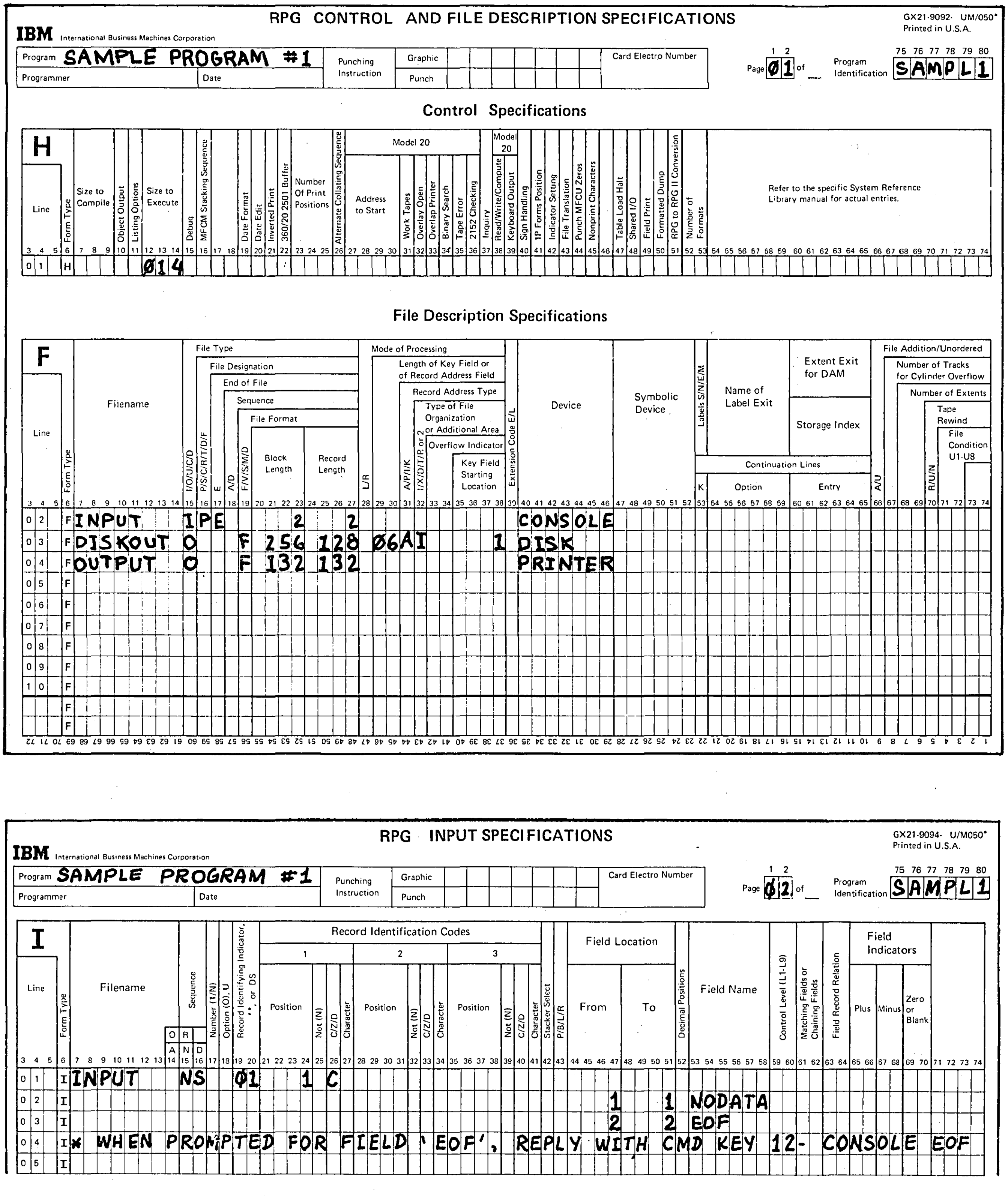
Calculation Specifications

All calculations (see Figure 19-1, Part 2) are conditioned by the LR indicator; therefore, they are executed at LR calculation time (see Chapter 17, *Detailed RPG II Program Logic*). The record number field (RECNBR) keeps track of the number of records written to DISKOUT. The COUNT field accumulates in increments of five to provide a unique key field for each record written to DISKOUT. The EXCPT operation code and exception output (E in column 15 of the output specifications) are used to write the records to disk. These calculations are part of the REPEAT loop and are executed 100 times, until COUNT equals 505 and 100 disk records have been created. At the end of the loop, one is subtracted from RECNBR to indicate the actual number of records that have been loaded.

Output Specifications

The output specifications (see Figure 19-1, Part 2) conditioned by LR cause a message to be written to the OUTPUT file, providing the following information:

- Job completion
- Number of records loaded
- File and key field descriptions
- Brief description of the function of program SAMPL2



IBM International Business Machines Corporation

RPG CONTROL AND FILE DESCRIPTION SPECIFICATIONS

GX21-9092- UM/050* Printed in U.S.A.

Program SAMPLE PROGRAM #1 — Page Ø1 — Program Identification SAMPL1

Control Specifications

```
  01 H     Ø14
```

File Description Specifications

```
  02 FINPUT   IPE     2   2            CONSOLE
  03 FDISKOUT O   F 256 128 Ø6AI     1 DISK
  04 FOUTPUT  O   F 132 132            PRINTER
```

IBM International Business Machines Corporation

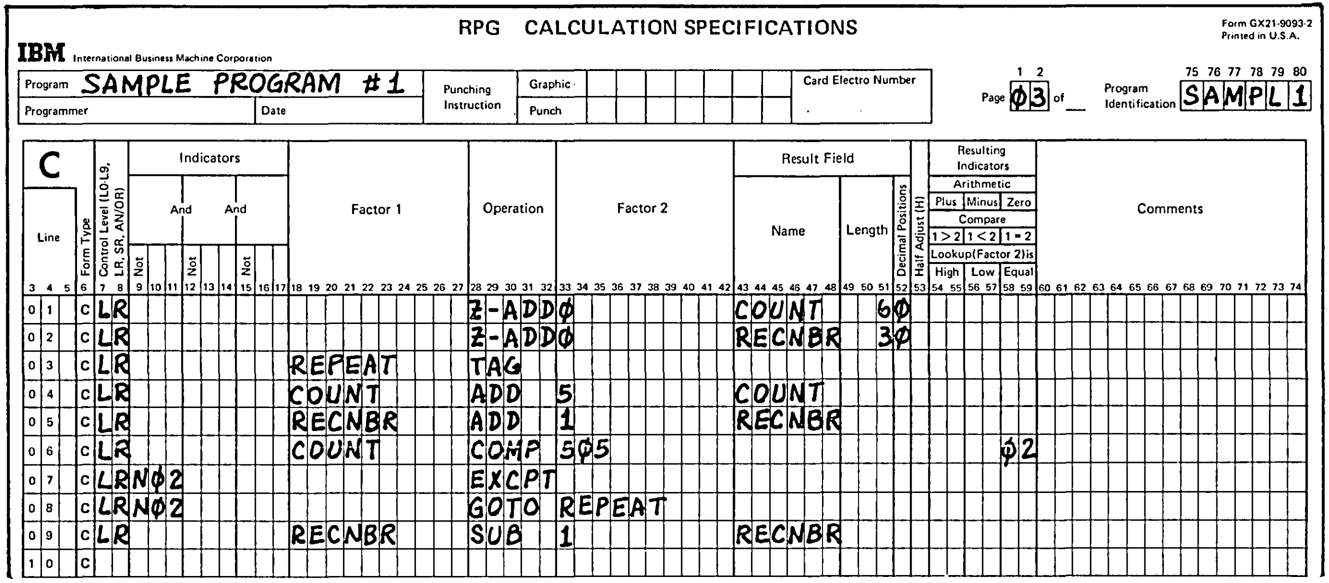
RPG INPUT SPECIFICATIONS

GX21-9094- U/M050* Printed in U.S.A.

Program SAMPLE PROGRAM #1 — Page Ø2 — Program Identification SAMPL1

```
  01 IINPUT   NS  Ø1   1  C
  02 I                                        1   1 NODATA
  03 I                                        2   2 EOF
  04 I* WHEN PROMPTED FOR FIELD 'EOF', REPLY WITH CMD KEY 12- CONSOLE EOF
  05 I
```

Figure 19-1 (Part 1 of 2). Sample Program 1 (SAMPL1)



IBM International Business Machine Corporation

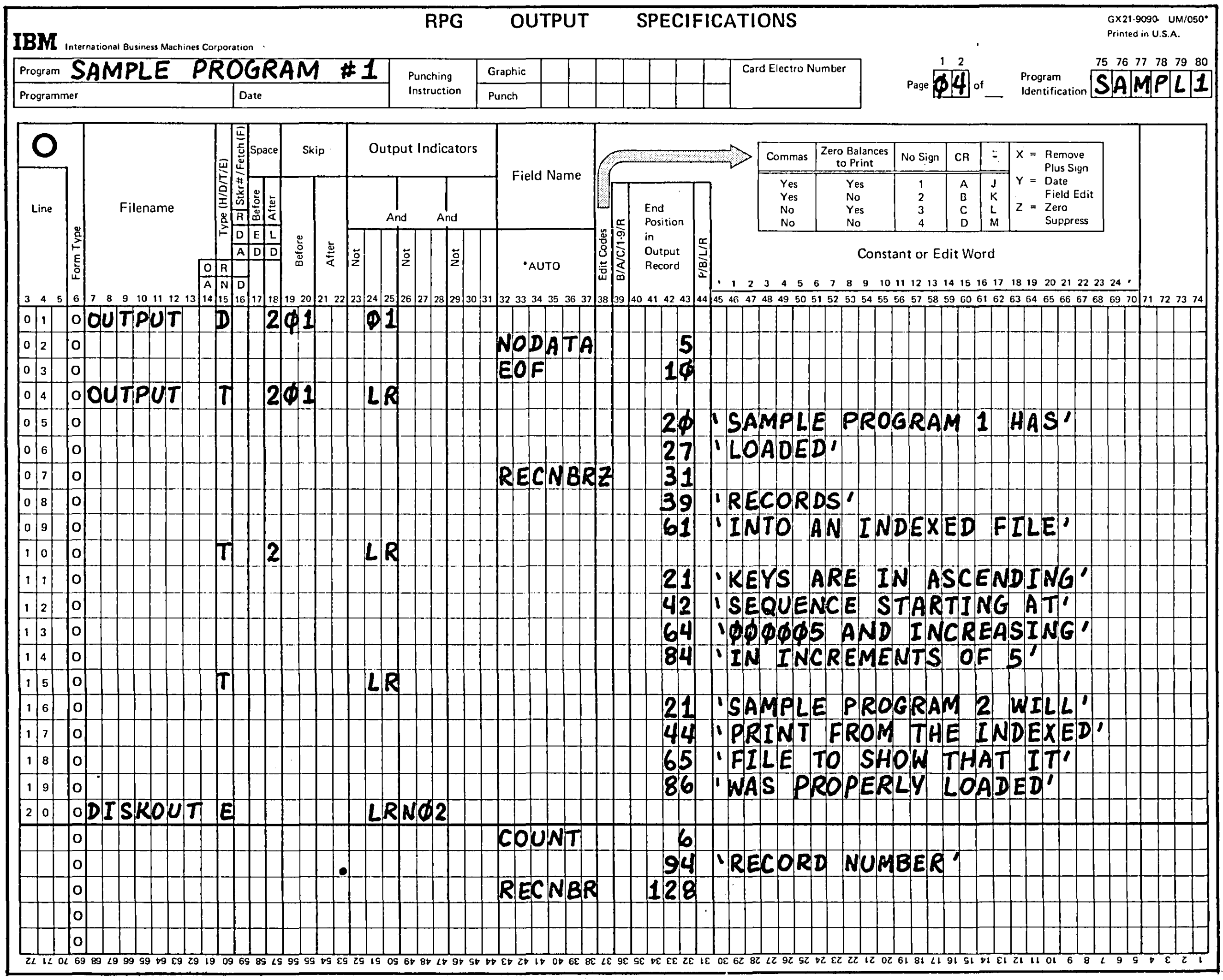
RPG CALCULATION SPECIFICATIONS

Form GX21-9093-2 Printed in U.S.A.

Program SAMPLE PROGRAM #1 | Punching Instruction | Graphic | Punch | Card Electro Number | Page ØЗ of __ | Program Identification SAMPL1

Programmer | Date

Line	Form Type	Control Level (L0-L9, LR, SR, AN/OR)	Indicators	Factor 1	Operation	Factor 2	Result Name	Length	Decimal Positions	Resulting Indicators (Equal)	Comments
01	C	LR			Z-ADD	Ø	COUNT	6	Ø		
02	C	LR			Z-ADD	Ø	RECNBR	3	Ø		
03	C	LR		REPEAT	TAG						
04	C	LR		COUNT	ADD	5	COUNT				
05	C	LR		RECNBR	ADD	1	RECNBR				
06	C	LR		COUNT	COMP	5Ø5				Ø2	
07	C	LR	NØ2		EXCPT						
08	C	LR	NØ2		GOTO	REPEAT					
09	C	LR		RECNBR	SUB	1	RECNBR				
10	C										



IBM International Business Machines Corporation

RPG OUTPUT SPECIFICATIONS

GX21-9090- UM/050* Printed in U.S.A.

Program SAMPLE PROGRAM #1 | Punching Instruction | Graphic | Punch | Card Electro Number | Page Ø4 of __ | Program Identification SAMPL1

Programmer | Date

Commas	Zero Balances to Print	No Sign	CR	-
Yes	Yes	1	A	J
Yes	No	2	B	K
No	Yes	3	C	L
No	No	4	D	M

X = Remove Plus Sign
Y = Date Field Edit
Z = Zero Suppress

Line	Form Type	Filename	Type (H/D/T/E)	Space Before	Space After	Skip Before	Skip After	Output Indicators	Field Name	Edit Codes	End Position in Output Record	Constant or Edit Word
01	O	OUTPUT	D		2	Ø1		Ø1				
02	O								NODATA		5	
03	O								EOF		1Ø	
04	O	OUTPUT	T		2	Ø1		LR				
05	O										2Ø	'SAMPLE PROGRAM 1 HAS'
06	O										27	'LOADED'
07	O								RECNBR	Z	31	
08	O										39	'RECORDS'
09	O										61	'INTO AN INDEXED FILE'
10	O		T		2			LR				
11	O										21	'KEYS ARE IN ASCENDING'
12	O										42	'SEQUENCE STARTING AT'
13	O										64	'ØØØØØ5 AND INCREASING'
14	O										84	'IN INCREMENTS OF 5'
15	O		T					LR				
16	O										21	'SAMPLE PROGRAM 2 WILL'
17	O										44	'PRINT FROM THE INDEXED'
18	O										65	'FILE TO SHOW THAT IT'
19	O										86	'WAS PROPERLY LOADED'
20	O	DISKOUT	E					LRNØ2				
	O								COUNT		6	
	O										94	'RECORD NUMBER'
	O								RECNBR		128	
	O											
	O											

Figure 19-1 (Part 2 of 2). Sample Program 1 (SAMPL1)

SAMPLE PROGRAM 2 (SAMPL2)

SAMPL2 must be preceded by sample program 1 (SAMPL1). SAMPL2 reads the indexed file created by SAMPL1 and prints fields from each record read. Thus, SAMPL2 allows you to verify that SAMPL1 loaded the indexed file properly. The specifications required for SAMPL2 are shown in Figure 19-2.

Control Specifications

Control specifications (see Figure 19-2, Part 1) should be present in every program. They are the first record in the source program and identify the program.

File Description Specifications

The files in SAMPL2 are described on the file description specifications sheet (see Figure 19-2, Part 1). The indexed file, DISKOUT, loaded in SAMPL1 is defined as the primary input file for SAMPL2. The E in column 17 ensures that SAMPL2 does not end until end of file is reached on DISKOUT. The records read from DISKOUT are printed on the output file, OUTPUT. An overflow indicator is specified in columns 33 and 34 so that subsequent operations can be conditioned on overflow.

Input Specifications

The primary input file, DISKOUT, is further described on the input specifications sheet (see Figure 19-2, Part 2). DISKOUT records are 128 positions long and are identified by a zero in position 1. When an input record containing a zero in position 1 is read, indicator 01 turns on.

Calculation Specifications

The calculation specifications (see Figure 19-2, Part 2) add one to COUNT when indicator 01 is on. The COUNT field keeps track of the number of records read from the DISKOUT file.

Output Specifications

In the output specifications (see Figure 19-2, Part 2) the 1P and OF indicators, specified in an OR relationship, cause a heading line to be printed on the first output page and on each succeeding page. Conditioned by indicator 01, the disk record just read is printed.

The next record is read from DISKOUT, and the same calculation and output operations are repeated until there are no more records in the disk file. When end of file is reached on DISKOUT, the LR indicator turns on.

Conditioned by LR, a total line is printed indicating how many records are read from DISKOUT. If the number printed (COUNT) is 100, SAMPL1 and SAMPL2 were executed properly.

SAMPLE PROGRAM 3 (SAMPL3)

The programs SAMPL3, SAMPL4, and SAMPL5 are designed to be run in sequence.

SAMPL3 loads master records into an indexed file and creates a consecutive file of transactions. The transaction file is processed against the master file in SAMPL4. SAMPL4 should follow SAMPL3. Figure 19-3 shows the completed specifications for SAMPL3.

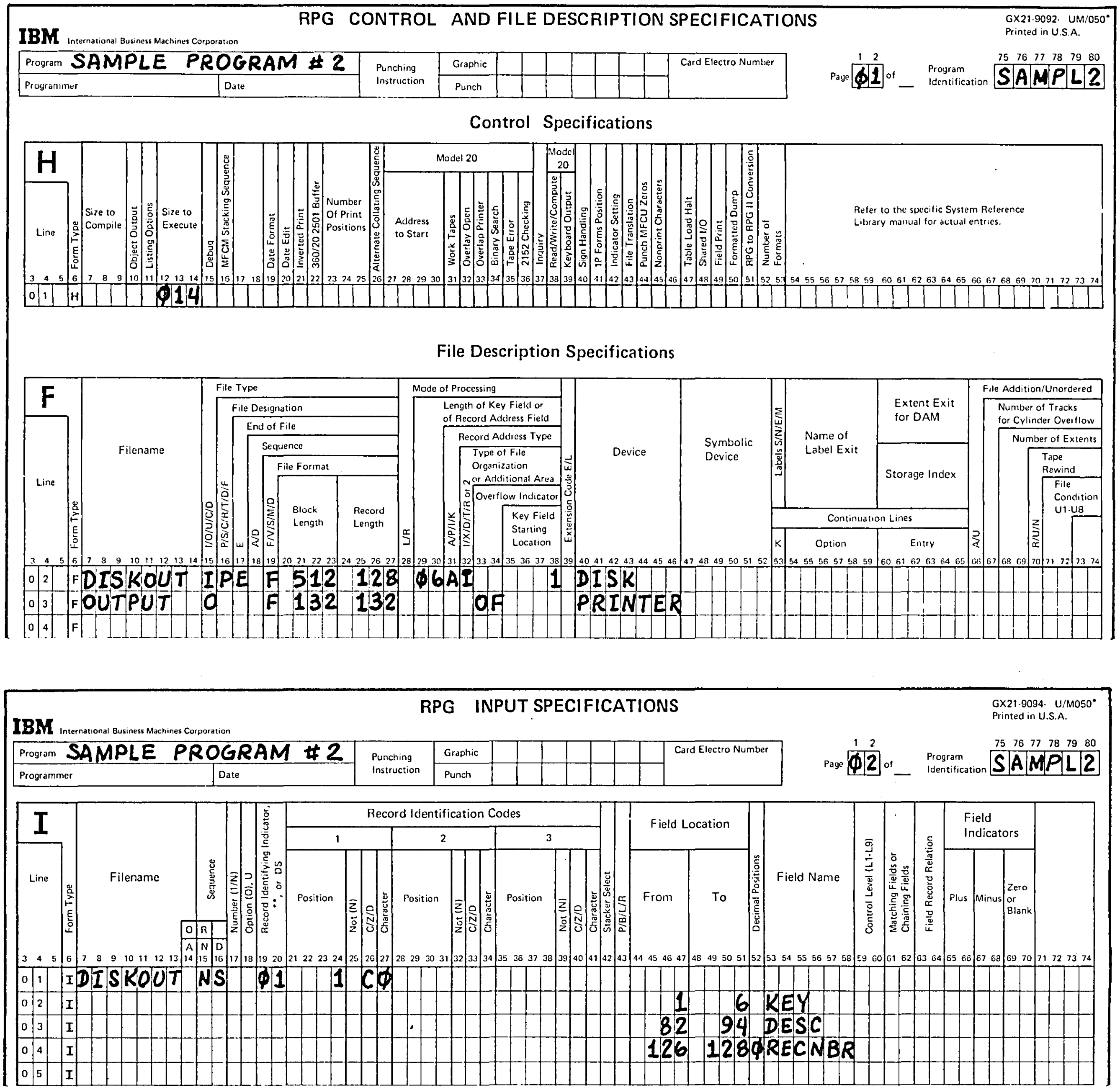
Control Specifications

Control specifications should be present in every program. They are the first record in the source program and identify the program.

File Description Specifications

The file description specifications describe the files used in the program. The input record file, INPUT, is read from CONSOLE. An E in column 17 indicates that the program ends when the last data record keyed in is processed. The indexed output file, MASTER, consists of 26-position records with a 5-position key field starting in the second record position.

A consecutive output file, TRANS, with a 10-position record length is specified by the file description specifications. A printer output file, PRINT, with a record length of 78 is also defined by the file description specifications.



IBM International Business Machines Corporation

RPG CONTROL AND FILE DESCRIPTION SPECIFICATIONS

GX21-9092- UM/050*
Printed in U.S.A.

Program: SAMPLE PROGRAM # 2
Programmer:
Date:
Punching Instruction: Graphic / Punch
Card Electro Number
Page Ø1 of __
Program Identification (75–80): SAMPL2

Control Specifications

```
01 H     Ø14
```

File Description Specifications

```
02 FDISKOUT IPE F 512 128 Ø6AI     1 DISK
03 FOUTPUT  O   F 132 132     OF     PRINTER
04 F
```

IBM International Business Machines Corporation

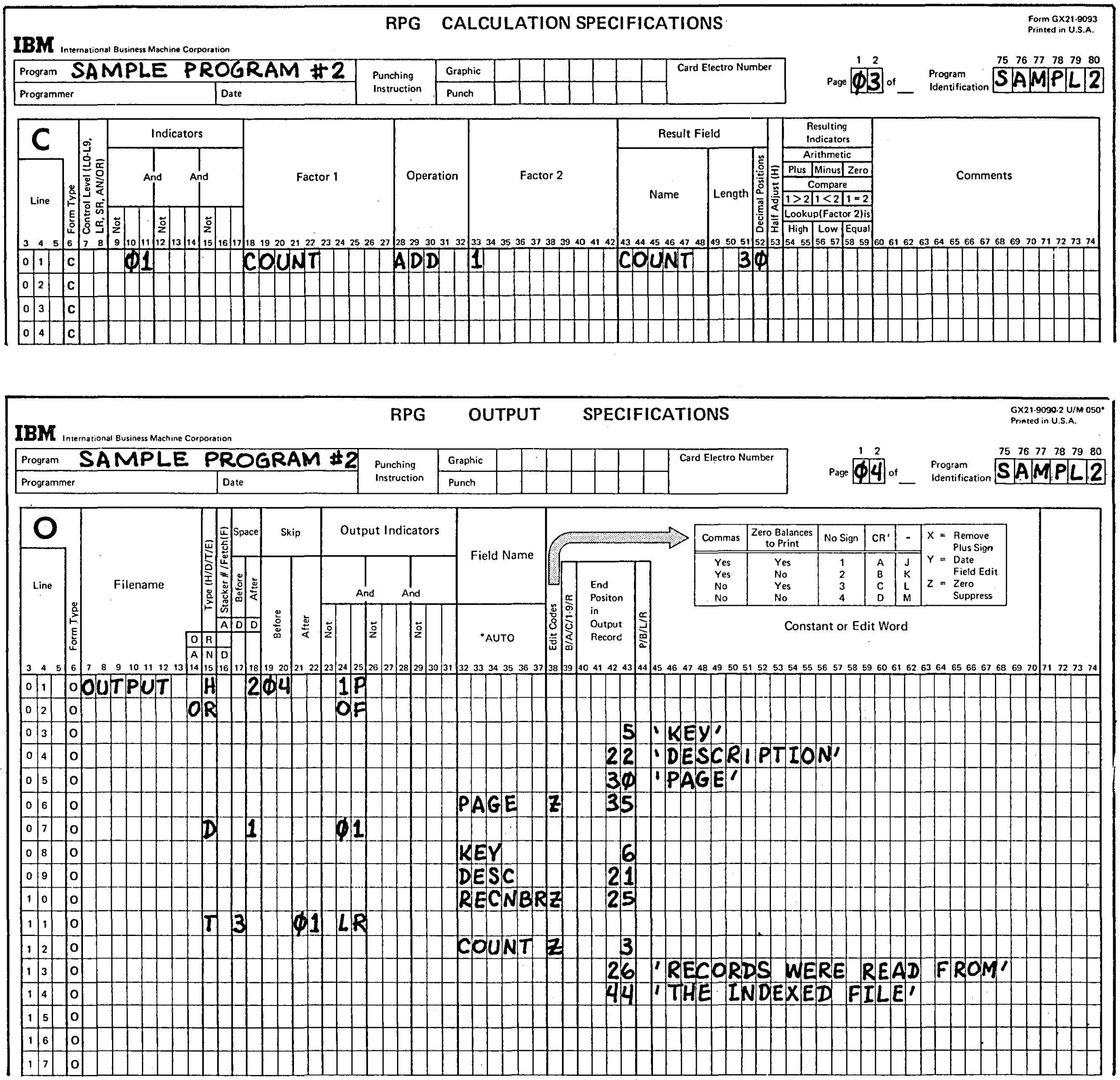
RPG INPUT SPECIFICATIONS

GX21-9094- U/M050*
Printed in U.S.A.

Program: SAMPLE PROGRAM #2
Programmer:
Date:
Punching Instruction: Graphic / Punch
Card Electro Number
Page Ø2 of __
Program Identification (75–80): SAMPL2

```
01 IDISKOUT NS  Ø1   1 CØ
02 I                                        1   6 KEY
03 I                                       82  94 DESC
04 I                                      126 1280RECNBR
05 I
```

Figure 19-2 (Part 1 of 2). Sample Program 2 (SAMPL2)



IBM International Business Machine Corporation

RPG CALCULATION SPECIFICATIONS

Form GX21-9093
Printed in U.S.A.

Program SAMPLE PROGRAM #2
Programmer
Date
Punching Instruction: Graphic, Punch
Card Electro Number
Page 03 of
Program Identification SAMPL2

```
  01 C   01      COUNT     ADD  1         COUNT   30
  02 C
  03 C
  04 C
```

IBM International Business Machine Corporation

RPG OUTPUT SPECIFICATIONS

GX21-9090-2 U/M 050*
Printed in U.S.A.

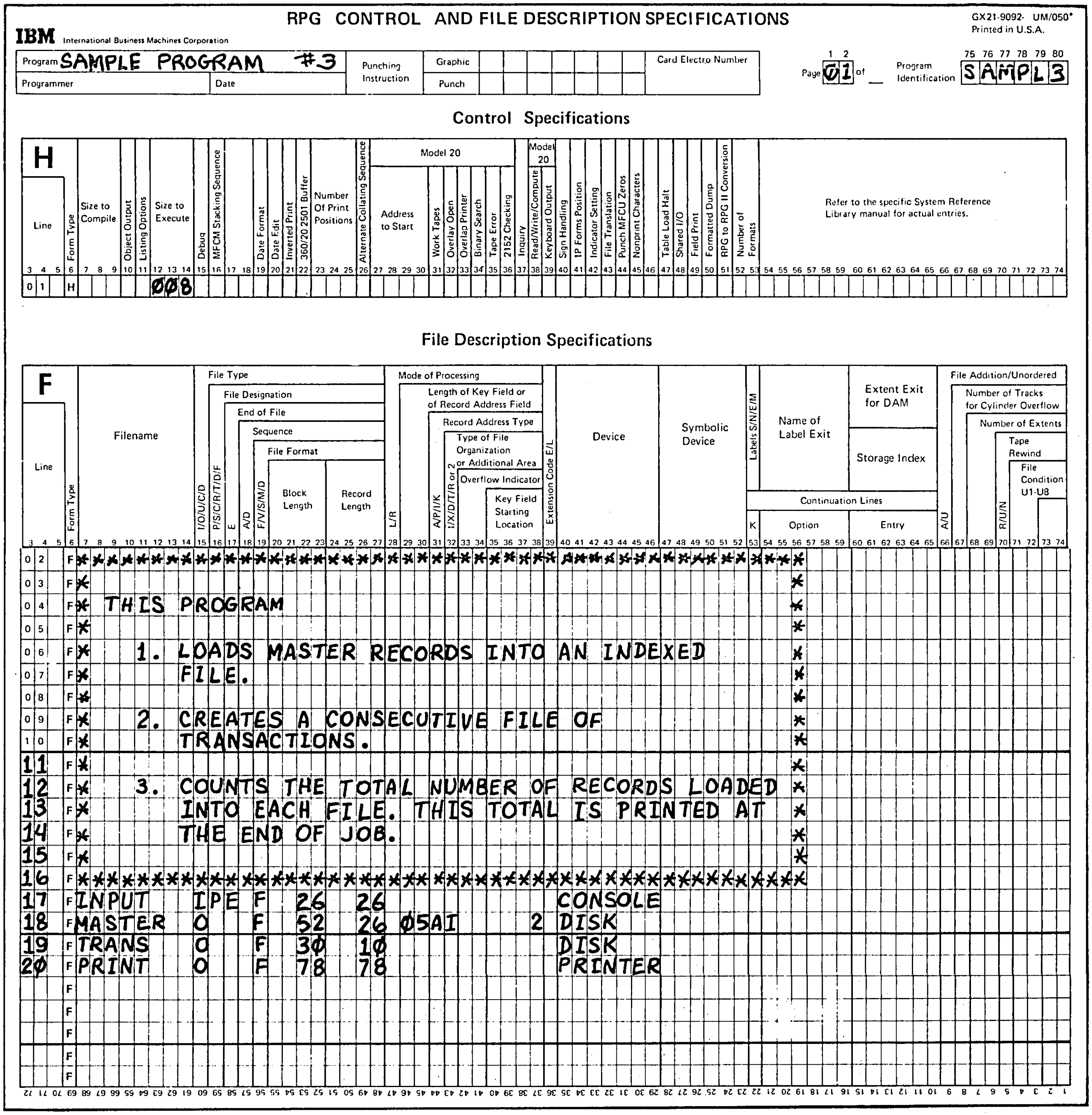
Program SAMPLE PROGRAM #2
Programmer
Date
Punching Instruction: Graphic, Punch
Card Electro Number
Page 04 of
Program Identification SAMPL2

Commas	Zero Balances to Print	No Sign	CR	-
Yes	Yes	1	A	J
Yes	No	2	B	K
No	Yes	3	C	L
No	No	4	D	M

X = Remove Plus Sign
Y = Date Field Edit
Z = Zero Suppress

```
  01 OOUTPUT  H  204   1P
  02 O       OR        OF
  03 O                                    5 'KEY'
  04 O                                   22 'DESCRIPTION'
  05 O                                   30 'PAGE'
  06 O                         PAGE  Z   35
  07 O        D  1     01
  08 O                         KEY        6
  09 O                         DESC      21
  10 O                         RECNBRZ   25
  11 O        T  3  01 LR
  12 O                         COUNT Z    3
  13 O                                   26 'RECORDS WERE READ FROM'
  14 O                                   44 'THE INDEXED FILE'
  15 O
  16 O
  17 O
```

Figure 19-2 (Part 2 of 2). Sample Program 2 (SAMPL2)



IBM International Business Machines Corporation

RPG CONTROL AND FILE DESCRIPTION SPECIFICATIONS

GX21-9092- UM/050*
Printed in U.S.A.

Program SAMPLE PROGRAM #3 | Punching Instruction | Graphic | Punch | Card Electro Number
Programmer | Date

Page 01 of ___ Program Identification (75–80) SAMPL3

Control Specifications

H: Line | Form Type | Size to Compile | Object Output | Listing Options | Size to Execute | Debug | MFCM Stacking Sequence | Date Format | Date Edit | Inverted Print | 360/20 2501 Buffer | Number Of Print Positions | Alternate Collating Sequence | Model 20: Address to Start, Work Tapes, Overlay Open, Overlap Printer, Binary Search, Tape Error, 2152 Checking | Inquiry | Model 20: Read/Write/Compute, Keyboard Output | Sign Handling | 1P Forms Position | Indicator Setting | File Translation | Punch MFCU Zeros | Nonprint Characters | Table Load Halt | Shared I/O | Field Print | Formatted Dump | RPG to RPG II Conversion | Number of Formats | Refer to the specific System Reference Library manual for actual entries.

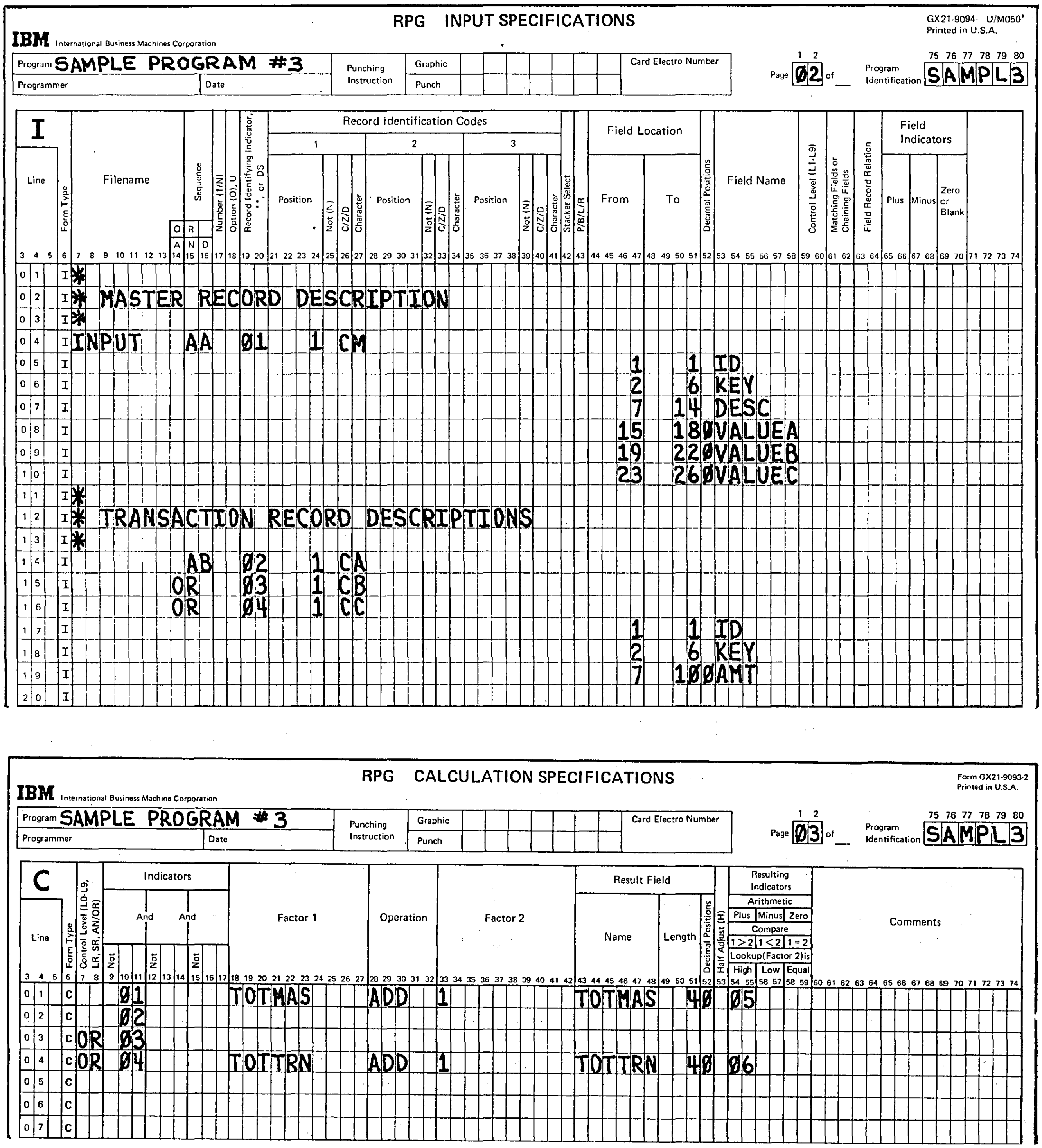
```
  01 H     ØØ8
```

File Description Specifications

F: Line | Form Type | Filename | File Type (I/O/U/C/D) | File Designation (P/S/C/R/T/D/F) | End of File (E) | Sequence (A/D) | File Format (F/V/S/M/D) | Block Length | Record Length | Mode of Processing (L/R) | Length of Key Field or of Record Address Field | Record Address Type (A/P/I/K) | Type of File Organization or Additional Area (I/X/D/T/R or 2) | Overflow Indicator | Key Field Starting Location | Extension Code E/L | Device | Symbolic Device | Labels S/N/E/M | Name of Label Exit | Extent Exit for DAM | Storage Index | Continuation Lines: K, Option, Entry | A/U | File Addition/Unordered | Number of Tracks for Cylinder Overflow | Number of Extents | Tape Rewind (R/U/N) | File Condition U1-U8

```
  02 F**************************************************
  03 F*                                                *
  04 F* THIS PROGRAM                                   *
  05 F*                                                *
  06 F*   1. LOADS MASTER RECORDS INTO AN INDEXED      *
  07 F*      FILE.                                     *
  08 F*                                                *
  09 F*   2. CREATES A CONSECUTIVE FILE OF             *
  10 F*      TRANSACTIONS.                             *
  11 F*                                                *
  12 F*   3. COUNTS THE TOTAL NUMBER OF RECORDS LOADED *
  13 F*      INTO EACH FILE. THIS TOTAL IS PRINTED AT  *
  14 F*      THE END OF JOB.                           *
  15 F*                                                *
  16 F**************************************************
  17 FINPUT   IPE F  26  26            CONSOLE
  18 FMASTER  O   F  52  26 Ø5AI     2 DISK
  19 FTRANS   O   F  3Ø  1Ø            DISK
  20 FPRINT   O   F  78  78            PRINTER
     F
     F
     F
     F
     F
```

Figure 19-3 (Part 1 of 3). Specifications for SAMPL3



IBM International Business Machines Corporation

RPG INPUT SPECIFICATIONS

GX21-9094- U/M050*
Printed in U.S.A.

Program SAMPLE PROGRAM #3 — Programmer — Date — Punching Instruction: Graphic, Punch — Card Electro Number — Page Ø2 of — Program Identification SAMPL3

```
01 I*
02 I* MASTER RECORD DESCRIPTION
03 I*
04 IINPUT   AA  Ø1   1 CM
05 I                                        1   1 ID
06 I                                        2   6 KEY
07 I                                        7  14 DESC
08 I                                       15  18ØVALUEA
09 I                                       19  22ØVALUEB
10 I                                       23  26ØVALUEC
11 I*
12 I* TRANSACTION RECORD DESCRIPTIONS
13 I*
14 I        AB  Ø2   1 CA
15 I       OR   Ø3   1 CB
16 I       OR   Ø4   1 CC
17 I                                        1   1 ID
18 I                                        2   6 KEY
19 I                                        7  1ØØAMT
20 I
```

IBM International Business Machine Corporation

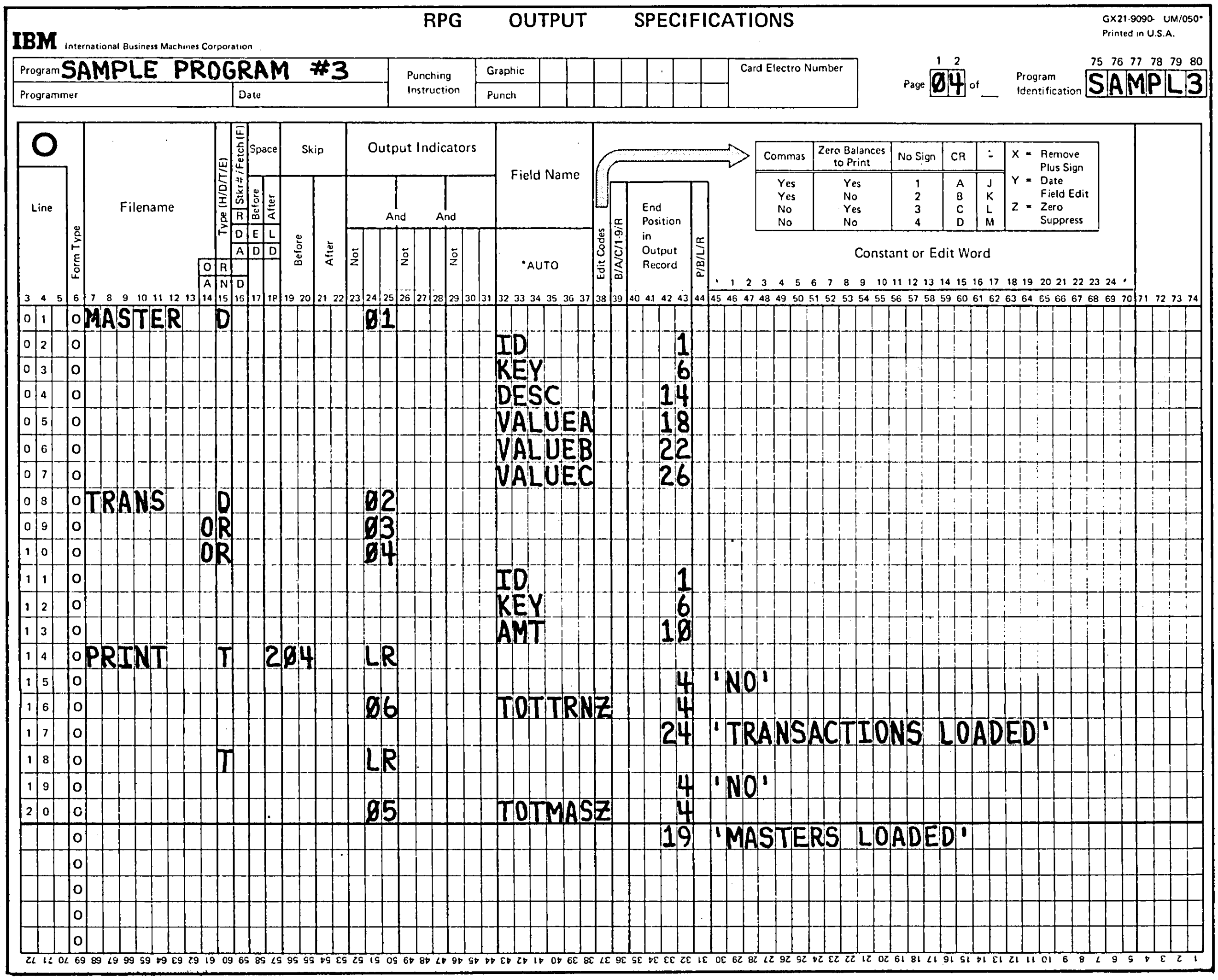
RPG CALCULATION SPECIFICATIONS

Form GX21-9093-2
Printed in U.S.A.

Program SAMPLE PROGRAM #3 — Programmer — Date — Punching Instruction: Graphic, Punch — Card Electro Number — Page Ø3 of — Program Identification SAMPL3

```
01 C   Ø1      TOTMAS    ADD  1         TOTMAS  4Ø Ø5
02 C   Ø2
03 COR Ø3
04 COR Ø4      TOTTRN    ADD  1         TOTTRN  4Ø Ø6
05 C
06 C
07 C
```

Figure 19-3 (Part 2 of 3). Specifications for SAMPL3



IBM International Business Machines Corporation

RPG OUTPUT SPECIFICATIONS

GX21-9090- UM/050* Printed in U.S.A.

Program SAMPLE PROGRAM #3

Programmer

Date

Punching Instruction: Graphic / Punch

Card Electro Number

Page Ø4 of

Program Identification SAMPL3

Commas	Zero Balances to Print	No Sign	CR	-
Yes	Yes	1	A	J
Yes	No	2	B	K
No	Yes	3	C	L
No	No	4	D	M

X = Remove Plus Sign
Y = Date Field Edit
Z = Zero Suppress

Line	Form Type	Filename	Type (H/D/T/E)	Space Before	Space After	Skip Before	Skip After	Output Indicators	Field Name	Edit Codes	End Position in Output Record	Constant or Edit Word
01	O	MASTER	D					Ø1				
02	O								ID		1	
03	O								KEY		6	
04	O								DESC		14	
05	O								VALUEA		18	
06	O								VALUEB		22	
07	O								VALUEC		26	
08	O	TRANS	D					Ø2				
09	O	OR						Ø3				
10	O	OR						Ø4				
11	O								ID		1	
12	O								KEY		6	
13	O								AMT		1Ø	
14	O	PRINT	T	2		Ø4		LR				
15	O										4	'NO'
16	O							Ø6	TOTTRN	Z	4	
17	O										24	'TRANSACTIONS LOADED'
18	O		T					LR				
19	O										4	'NO'
20	O							Ø5	TOTMAS	Z	4	
	O										19	'MASTERS LOADED'
	O											
	O											
	O											
	O											
	O											

Figure 19-3 (Part 3 of 3). Specifications for SAMPL3

Input Specifications

The CONSOLE input file, INPUT, contains two types of records: master and transaction. An M in position 1 of the input record turns on record identifying indicator 01, indicating a master record. An A, B, or C in position 1 of the input record turns on record identifying indicator 02, 03, or 04, respectively, indicating transaction record type A, B, or C. No sequence checking occurs for either type of record (AA and AB in columns 15 and 16).

Figure 19-4 shows the display screen for record type M before any data is entered. The top line of the display screen shows the record identification code and the record identifying indicator for the record being prompted. Also shown are the record identifying indicators for the record types that can be selected before any data is entered for this record and for the record types that can be selected after data is entered for this record. You can select any of these record types by pressing the Cmd key and the digit key corresponding to the record type you want.

The fields in the record are displayed in the order they are specified on the input specifications. (Notice that the entire record is displayed.) The type of field (alphameric or numeric) and the field length as well as the field name are displayed. The cursor indicates the position to be entered next.

Figure 19-5 shows the display screen for record type M after the first four fields have been entered.

When you want to stop entering input data, press command key 12 (the Cmd and = keys).

Calculation Specifications

The field name TOTMAS is incremented by one when record identifying indicator 01 is on. This maintains a running total of the master records that have been read from INPUT and transferred to disk. The field TOTTRN is incremented by one when record identifying indicator 02, 03, or 04 is on, maintaining a running total of the transaction records that are read from INPUT and transferred to disk.

Output Specifications

Four different output records are described in these specifications: one detail record for the master file, MASTER; one detail record for the transaction file, TRANS; and two total records for the printer file, PRINT.

The detail records for MASTER are conditioned by record identifying indicator 01. The detail records for TRANS are conditioned by record identifying indicators 02, 03, and 04. Both total lines for PRINT are printed when the last record identifying indicator is turned on (LR in columns 23 through 25). The first total line is for total transactions loaded. The printer skips to line 4 before the printing of the first total line, and double spacing occurs before the printing of the second total line. The second total line is for total masters loaded.

SAMPLE PROGRAM 4 (SAMPL4)

SAMPL4 must be preceded by SAMPL3. SAMPL4 reads from the transaction file, TRANS, created by SAMPL3 and accumulates totals for A, B, and C records. SAMPL4 also retrieves corresponding master records for transaction records and prints an error message if a corresponding master record is not found. Figure 19-6 shows the completed specifications for SAMPL4.

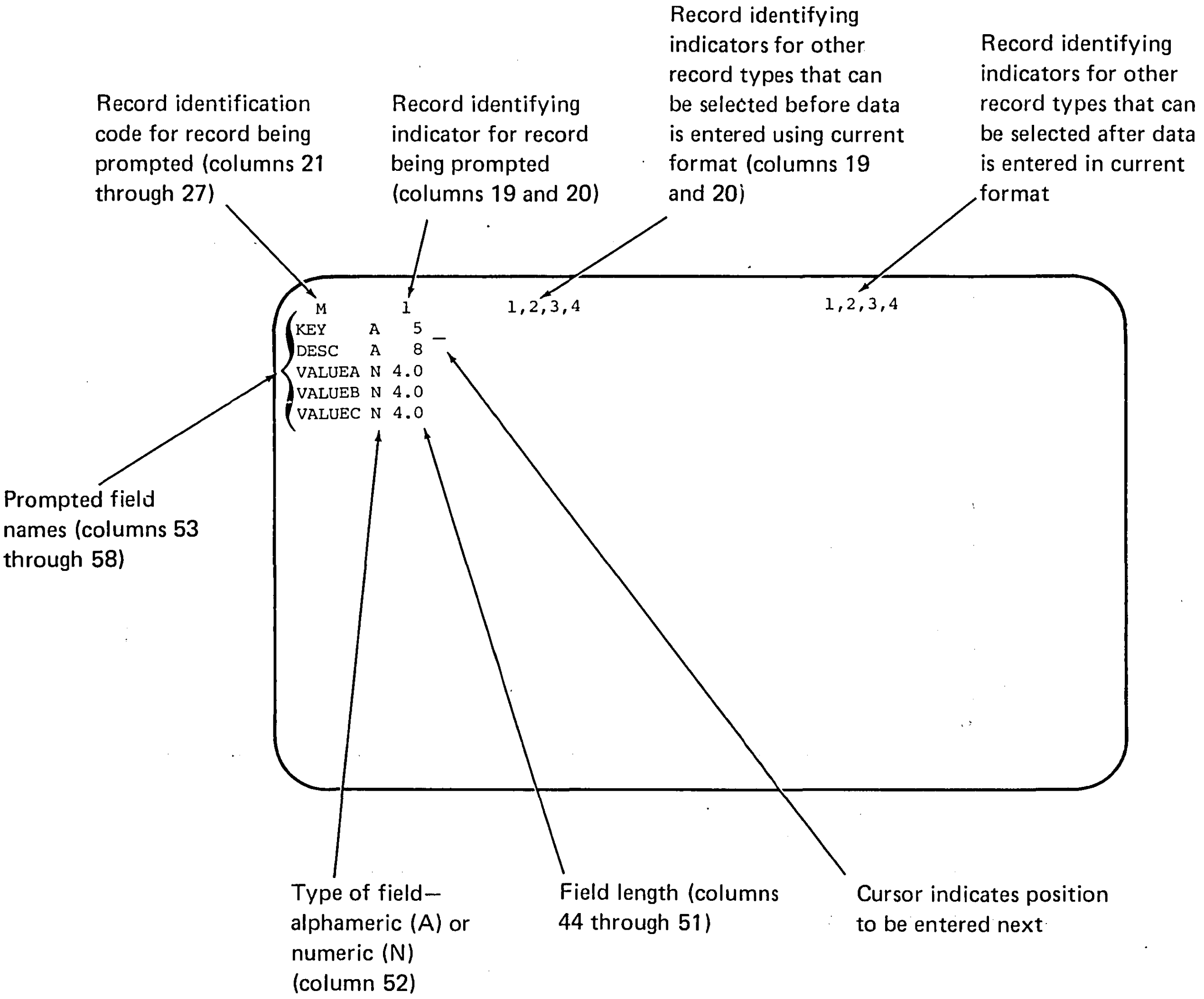
Control Specifications

Control specifications should be present in every program. They are the first record in the program and identify the program.

File Description Specifications

The input file for SAMPL4, TRANS (the output transaction file for SAMPL3), is read from disk. An E in column 17 indicates that the program ends when the last data record in the input file is processed. The output file, PRINT, consists of 72-position records.

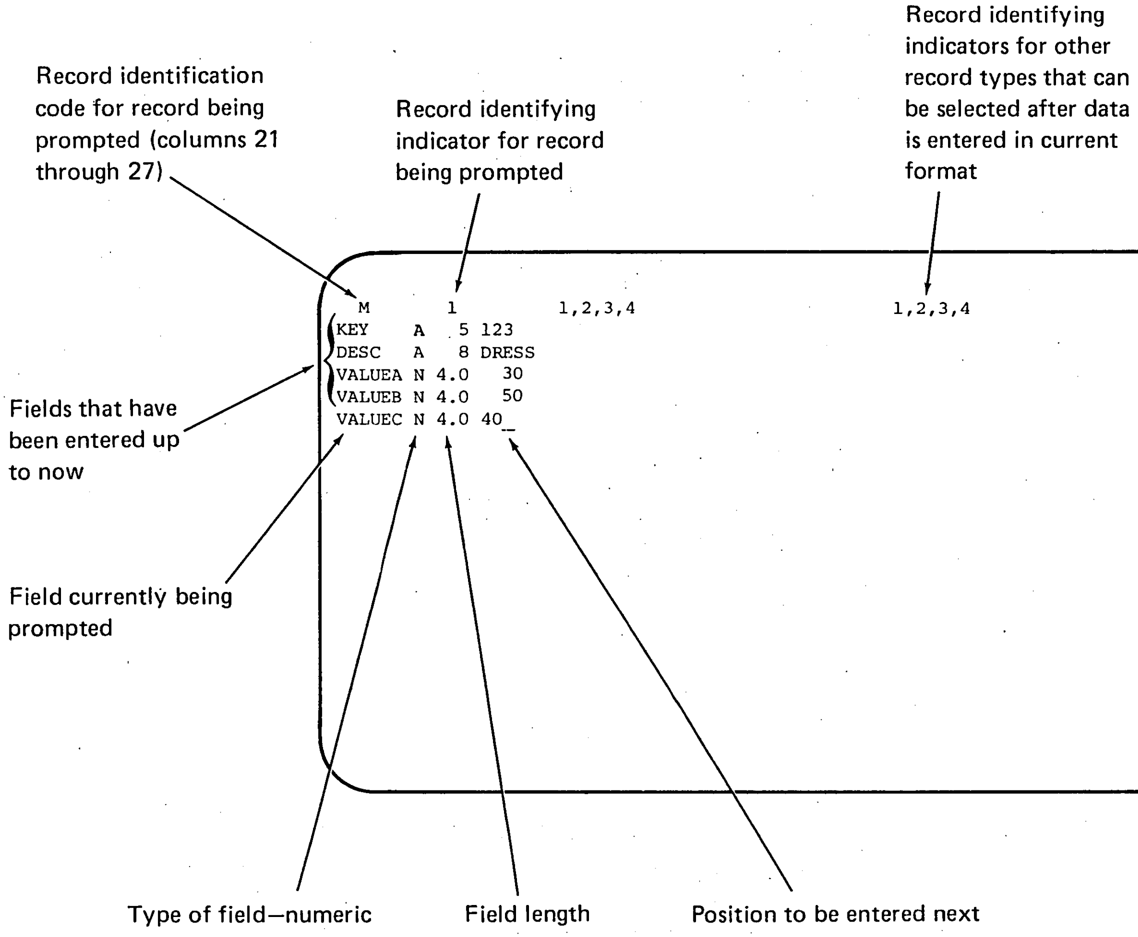
An overflow indicator (OF in columns 33 and 34) conditions printing of records in the file. The indexed file, MASTER, is a chained update file to be processed by keys. It consists of 26-position records with a 5-position key field starting in the second record position.



```
 M         1          1,2,3,4                          1,2,3,4
KEY    A   5 _
DESC   A   8
VALUEA N 4.0
VALUEB N 4.0
VALUEC N 4.0
```

Record identification code for record being prompted (columns 21 through 27)
Record identifying indicator for record being prompted (columns 19 and 20)
Record identifying indicators for other record types that can be selected before data is entered using current format (columns 19 and 20)
Record identifying indicators for other record types that can be selected after data is entered in current format
Prompted field names (columns 53 through 58)
Type of field—alphameric (A) or numeric (N) (column 52)
Field length (columns 44 through 51)
Cursor indicates position to be entered next

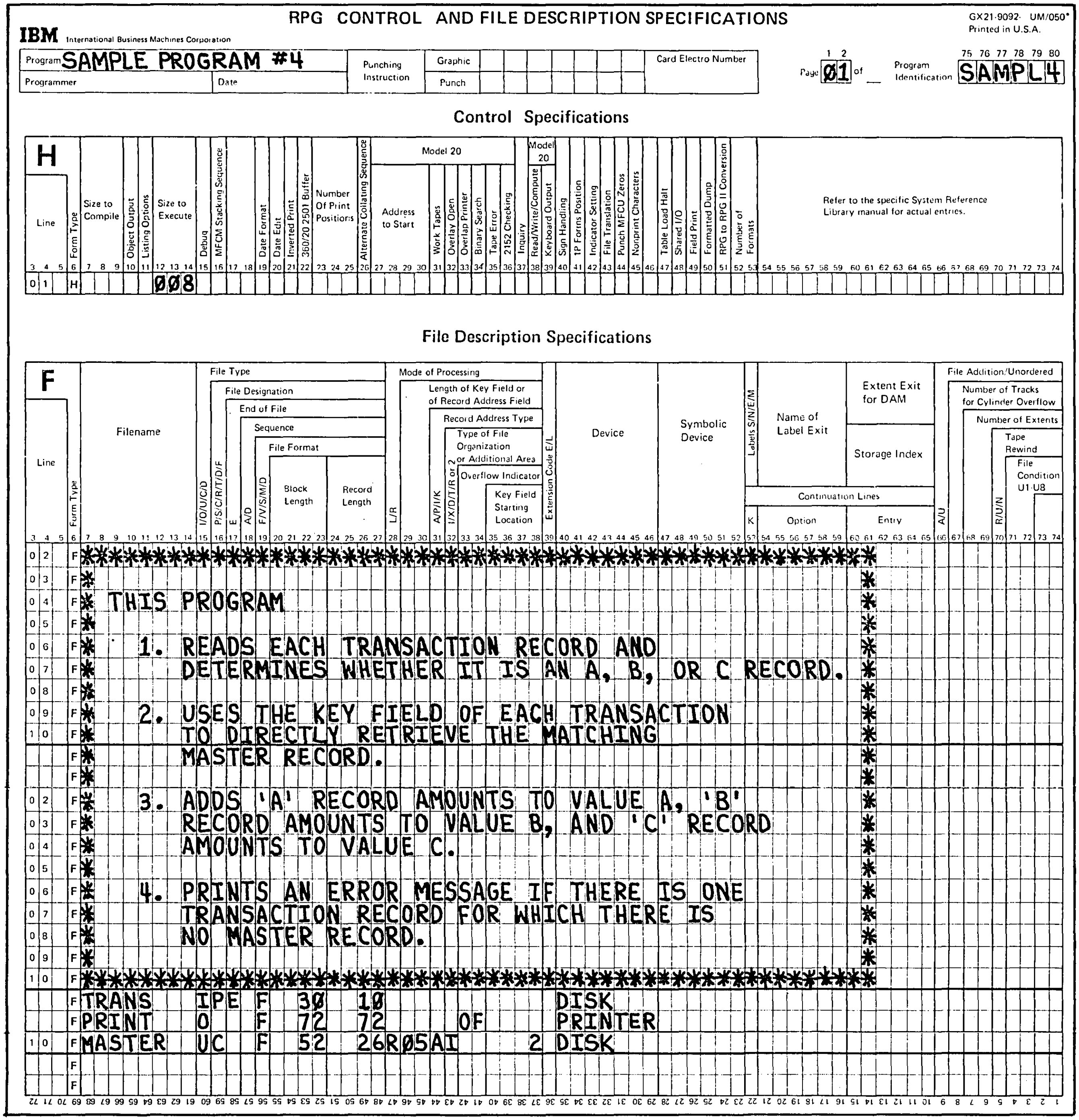
Figure 19-4. Display Screen Before Record Type M is Entered



```
  M       1        1,2,3,4                        1,2,3,4
KEY    A   5 123
DESC   A   8 DRESS
VALUEA N 4.0   30
VALUEB N 4.0   50
VALUEC N 4.0 40_
```

Record identification code for record being prompted (columns 21 through 27)
Record identifying indicator for record being prompted
Record identifying indicators for other record types that can be selected after data is entered in current format
Fields that have been entered up to now
Field currently being prompted
Type of field—numeric
Field length
Position to be entered next

Figure 19-5. Display Screen When Record Type M is Partially Entered



IBM International Business Machines Corporation

RPG CONTROL AND FILE DESCRIPTION SPECIFICATIONS

GX21-9092- UM/050*
Printed in U.S.A.

Program SAMPLE PROGRAM #4

Programmer

Date

Page Ø1 of

Program Identification SAMPL4

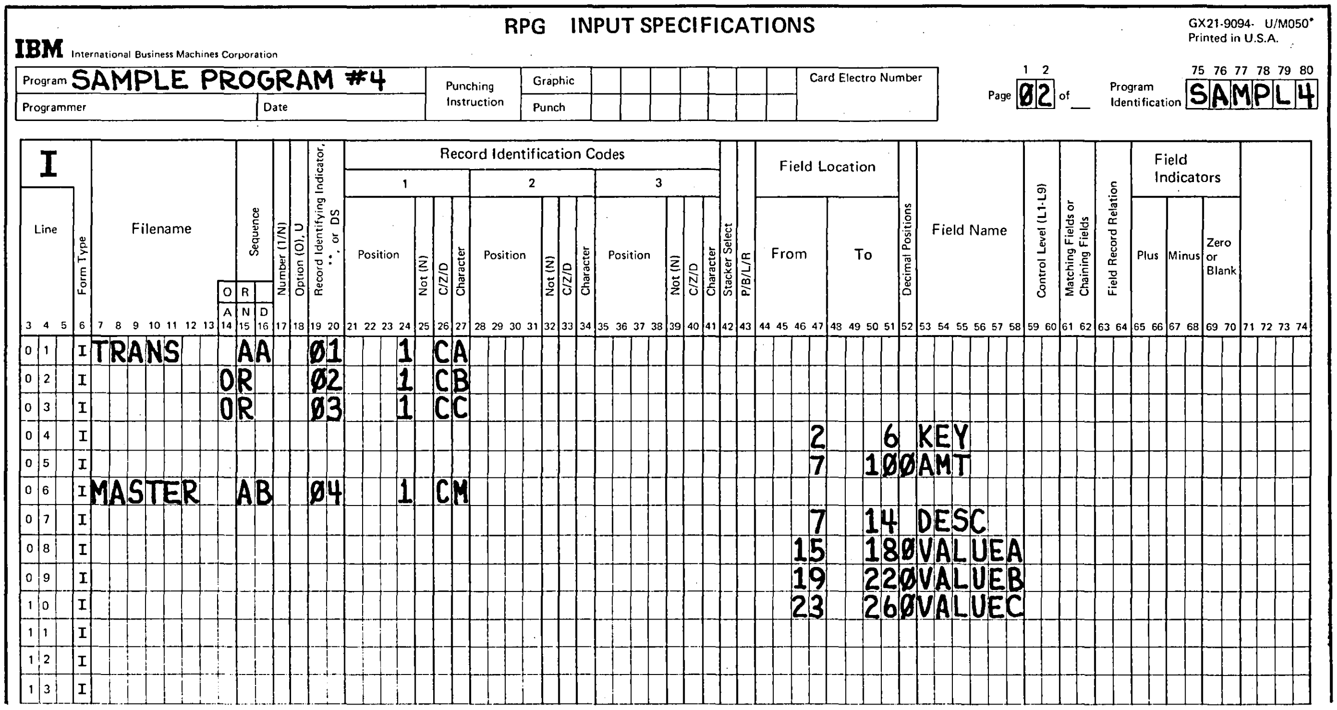
Control Specifications

```
  01 H     ØØ8
```

File Description Specifications

```
  02 F******************************************************
  03 F*                                                    *
  04 F* THIS PROGRAM                                       *
  05 F*                                                    *
  06 F*   1. READS EACH TRANSACTION RECORD AND             *
  07 F*       DETERMINES WHETHER IT IS AN A, B, OR C RECORD.*
  08 F*                                                    *
  09 F*   2. USES THE KEY FIELD OF EACH TRANSACTION        *
  10 F*       TO DIRECTLY RETRIEVE THE MATCHING            *
     F*       MASTER RECORD.                               *
     F*                                                    *
  02 F*   3. ADDS 'A' RECORD AMOUNTS TO VALUE A, 'B'       *
  03 F*       RECORD AMOUNTS TO VALUE B, AND 'C' RECORD    *
  04 F*       AMOUNTS TO VALUE C.                          *
  05 F*                                                    *
  06 F*   4. PRINTS AN ERROR MESSAGE IF THERE IS ONE       *
  07 F*       TRANSACTION RECORD FOR WHICH THERE IS        *
  08 F*       NO MASTER RECORD.                            *
  09 F*                                                    *
  10 F******************************************************
     FTRANS   IPE F  3Ø  1Ø                DISK
     FPRINT   O   F  72  72      OF        PRINTER
  10 FMASTER  UC  F  52  26RØ5AI      2    DISK
     F
     F
```

Figure 19-6 (Part 1 of 4). Specifications for SAMPL4



IBM International Business Machines Corporation

RPG INPUT SPECIFICATIONS

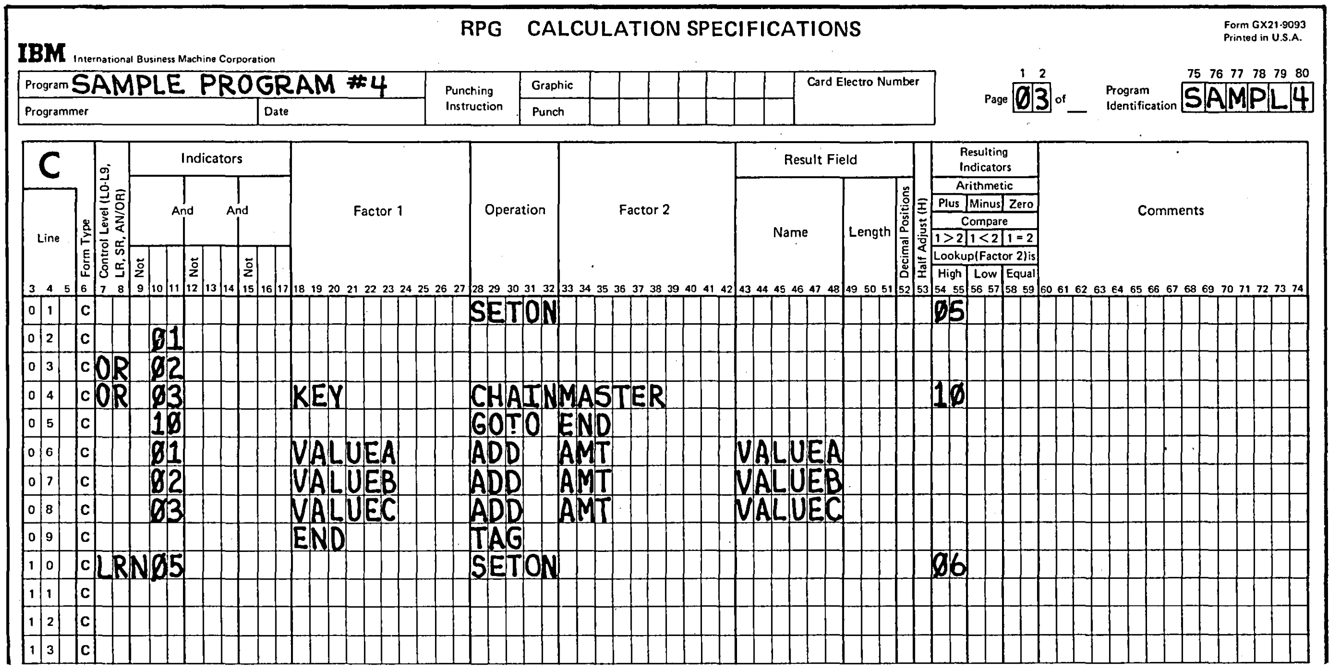
GX21-9094- U/M050*
Printed in U.S.A.

Program SAMPLE PROGRAM #4
Programmer
Date

Page Ø2 of

Program Identification SAMPL4

```
01 ITRANS   AA  Ø1   1 CA
02 I       OR   Ø2   1 CB
03 I       OR   Ø3   1 CC
04 I                                        2   6 KEY
05 I                                        7  1ØØAMT
06 IMASTER  AB  Ø4   1 CM
07 I                                        7  14 DESC
08 I                                       15  18ØVALUEA
09 I                                       19  22ØVALUEB
10 I                                       23  26ØVALUEC
11 I
12 I
13 I
```



IBM International Business Machine Corporation

RPG CALCULATION SPECIFICATIONS

Form GX21-9093
Printed in U.S.A.

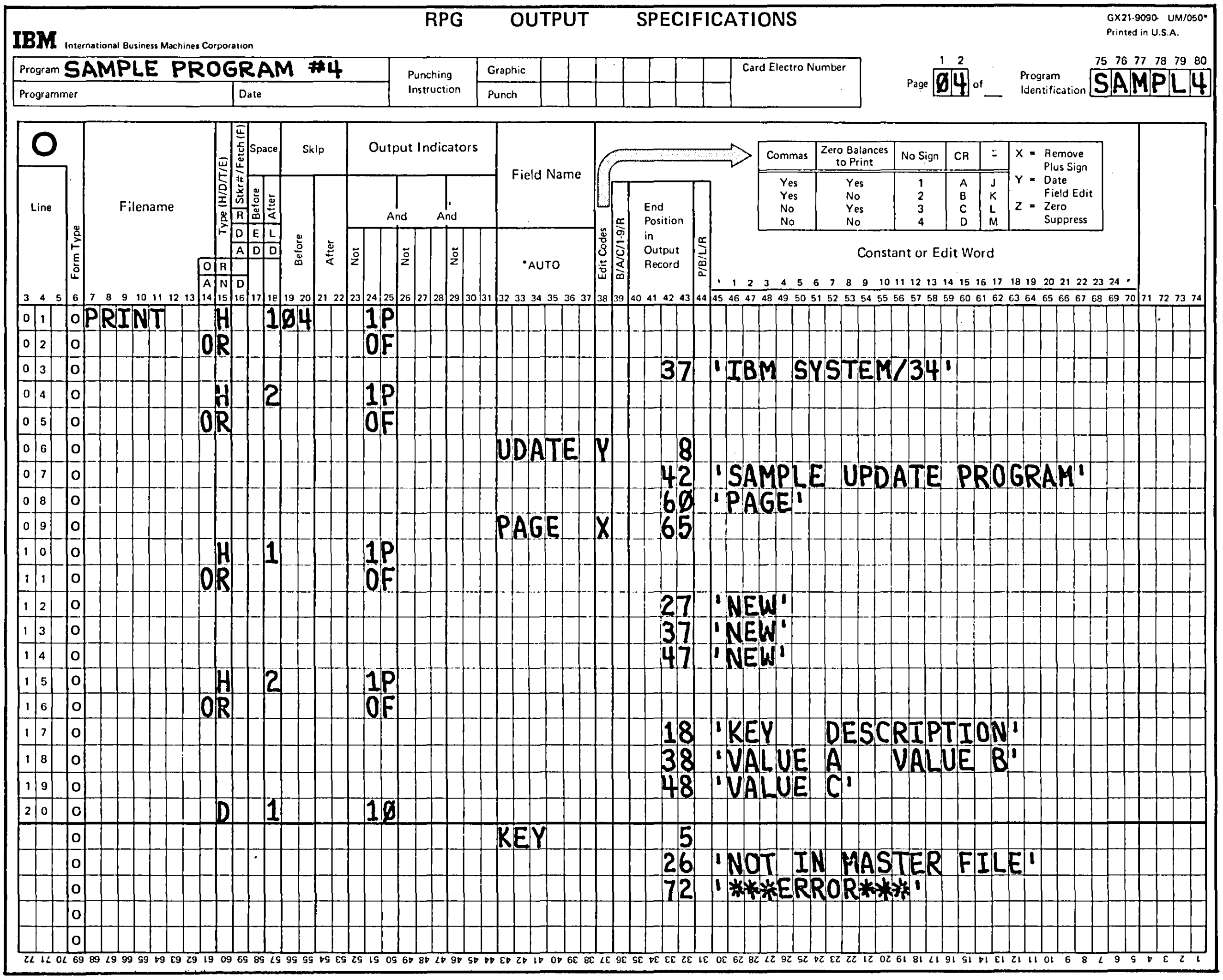
Program SAMPLE PROGRAM #4
Programmer
Date

Page Ø3 of

Program Identification SAMPL4

```
01 C                     SETON                     Ø5
02 C   Ø1
03 COR Ø2
04 COR Ø3      KEY       CHAINMASTER               1Ø
05 C   1Ø                GOTO END
06 C   Ø1      VALUEA    ADD  AMT       VALUEA
07 C   Ø2      VALUEB    ADD  AMT       VALUEB
08 C   Ø3      VALUEC    ADD  AMT       VALUEC
09 C           END       TAG
10 CLRNØ5                SETON                     Ø6
11 C
12 C
13 C
```

Figure 19-6 (Part 2 of 4). Specifications for SAMPL4



IBM International Business Machines Corporation

RPG OUTPUT SPECIFICATIONS

GX21-9090- UM/050*
Printed in U.S.A.

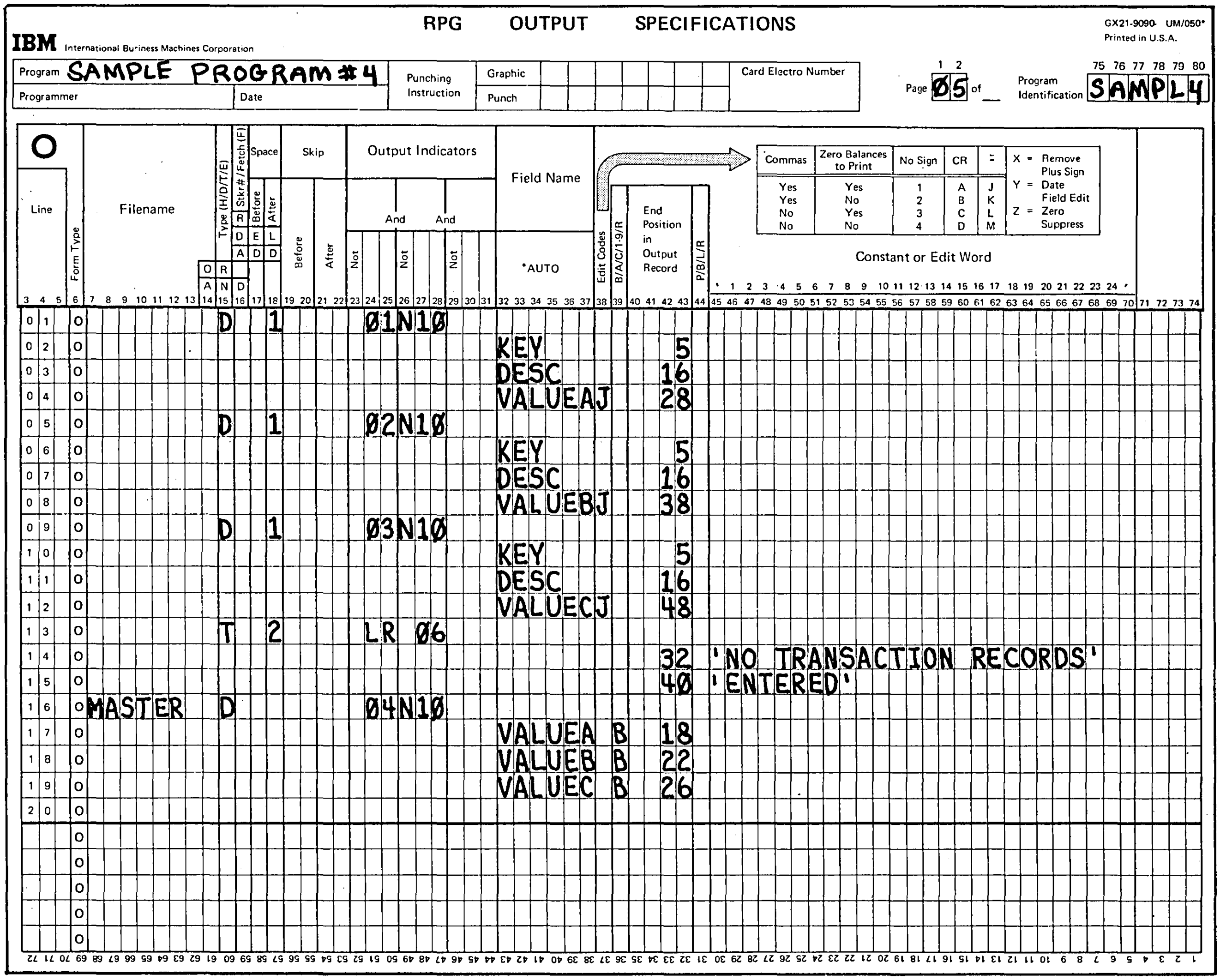
Program SAMPLE PROGRAM #4
Programmer
Date
Punching Instruction
Graphic
Punch
Card Electro Number
Page Ø4 of
Program Identification SAMPL4

Commas	Zero Balances to Print	No Sign	CR	-
Yes	Yes	1	A	J
Yes	No	2	B	K
No	Yes	3	C	L
No	No	4	D	M

X = Remove Plus Sign
Y = Date Field Edit
Z = Zero Suppress

Line	Form Type	Filename	Type (H/D/T/E)	Space Before	Space After	Skip Before	Skip After	Output Indicators	Field Name	Edit Codes	End Position in Output Record	Constant or Edit Word
01	O	PRINT	H		1	Ø4		1P				
02	O		OR					OF				
03	O										37	'IBM SYSTEM/34'
04	O		H		2			1P				
05	O		OR					OF				
06	O								UDATE	Y	8	
07	O										42	'SAMPLE UPDATE PROGRAM'
08	O										6Ø	'PAGE'
09	O								PAGE	X	65	
10	O		H		1			1P				
11	O		OR					OF				
12	O										27	'NEW'
13	O										37	'NEW'
14	O										47	'NEW'
15	O		H		2			1P				
16	O		OR					OF				
17	O										18	'KEY DESCRIPTION'
18	O										38	'VALUE A VALUE B'
19	O										48	'VALUE C'
20	O		D		1			1Ø				
	O								KEY		5	
	O										26	'NOT IN MASTER FILE'
	O										72	'***ERROR***'
	O											
	O											

Figure 19-6 (Part 3 of 4). Specifications for SAMPL4



IBM International Business Machines Corporation

RPG OUTPUT SPECIFICATIONS

GX21-9090- UM/050*
Printed in U.S.A.

Program SAMPLE PROGRAM #4

Programmer

Date

Punching Instruction: Graphic / Punch

Card Electro Number

Page Ø5 of

Program Identification SAMPL4

Commas	Zero Balances to Print	No Sign	CR	-
Yes	Yes	1	A	J
Yes	No	2	B	K
No	Yes	3	C	L
No	No	4	D	M

X = Remove Plus Sign
Y = Date Field Edit
Z = Zero Suppress

```
  01 O        D  1     Ø1N1Ø
  02 O                         KEY        5
  03 O                         DESC      16
  04 O                         VALUEAJ   28
  05 O        D  1     Ø2N1Ø
  06 O                         KEY        5
  07 O                         DESC      16
  08 O                         VALUEBJ   38
  09 O        D  1     Ø3N1Ø
  10 O                         KEY        5
  11 O                         DESC      16
  12 O                         VALUECJ   48
  13 O        T  2     LR Ø6
  14 O                                   32 'NO TRANSACTION RECORDS'
  15 O                                   4Ø 'ENTERED'
  16 OMASTER  D        Ø4N1Ø
  17 O                         VALUEA B  18
  18 O                         VALUEB B  22
  19 O                         VALUEC B  26
  20 O
     O
     O
     O
     O
     O
```

Figure 19-6 (Part 4 of 4). Specifications for SAMPL4

Input Specifications

Two types of files are specified by the input specifications: transaction and master. An A, B, or C in position 1 of the input record turns on record identifying indicator 01, 02, or 03, respectively. An M in position 1 of the update record turns on record identifying indicator 04, indicating an update record. No sequence checking occurs for either type (AA and AB in columns 15 and 16).

Calculation Specifications

When indicator 01, 02, or 03 is on, two operations occur:

1. A matching master record is retrieved for a transaction record (lines 02, 03, and 04 of the calculation specifications).

2. The AMT field of the transaction record is added to the appropriate value (VALUEA, VALUEB, or VALUEC) on the master record depending on the type of record (record identifying indicator 01, 02, or 03).

If no matching record is found, indicator 10 turns on.

Output Specifications

Nine printer output lines are described in these specifications. Four header lines conditioned by the first page indicator (1P in columns 23 through 25) or an overflow indicator (OF in columns 23 through 25) are printed. They are printed at the top of each page of the listing.

Four detail lines are also printed. A detail line is printed for each transaction record with no matching master record. For each type of transaction record, A, B, or C, the accumulative value is printed (detail lines conditioned by indicators 01, 02, or 03, and not 10). These detail lines are single spaced.

A total line is printed if no transaction records were entered.

A detail record is written on disk for the indexed update file, MASTER, when indicator 04 is on and indicator 10 is off. Indicator 04 turns on for an update record, and indicator 10 turns on if no matching record is found.

SAMPLE PROGRAM 5 (SAMPL5)

SAMPL5 must be preceded by SAMPL4. SAMPL5 reads from the indexed file, MASTER, and performs the following calculation: value A + value B - value C. If the result is negative, a message is printed. Figure 19-7 shows the completed specification sheets for SAMPL5.

Control Specifications

Control specifications should be present in every program. They are the first record in the source program and identify the program.

File Description Specifications

The input file for SAMPL5, MASTER, is an indexed file (I in column 32). An E in column 17 indicates that the program ends when the last data record in the input file is processed. The file consists of 26-position records with a 5-position key field starting in the second record position. A printer output file, PRINT, with a record length of 78 is also defined by the file description specifications.

Input Specifications

An M in position 1 of the input record turns on record identifying indicator 01.

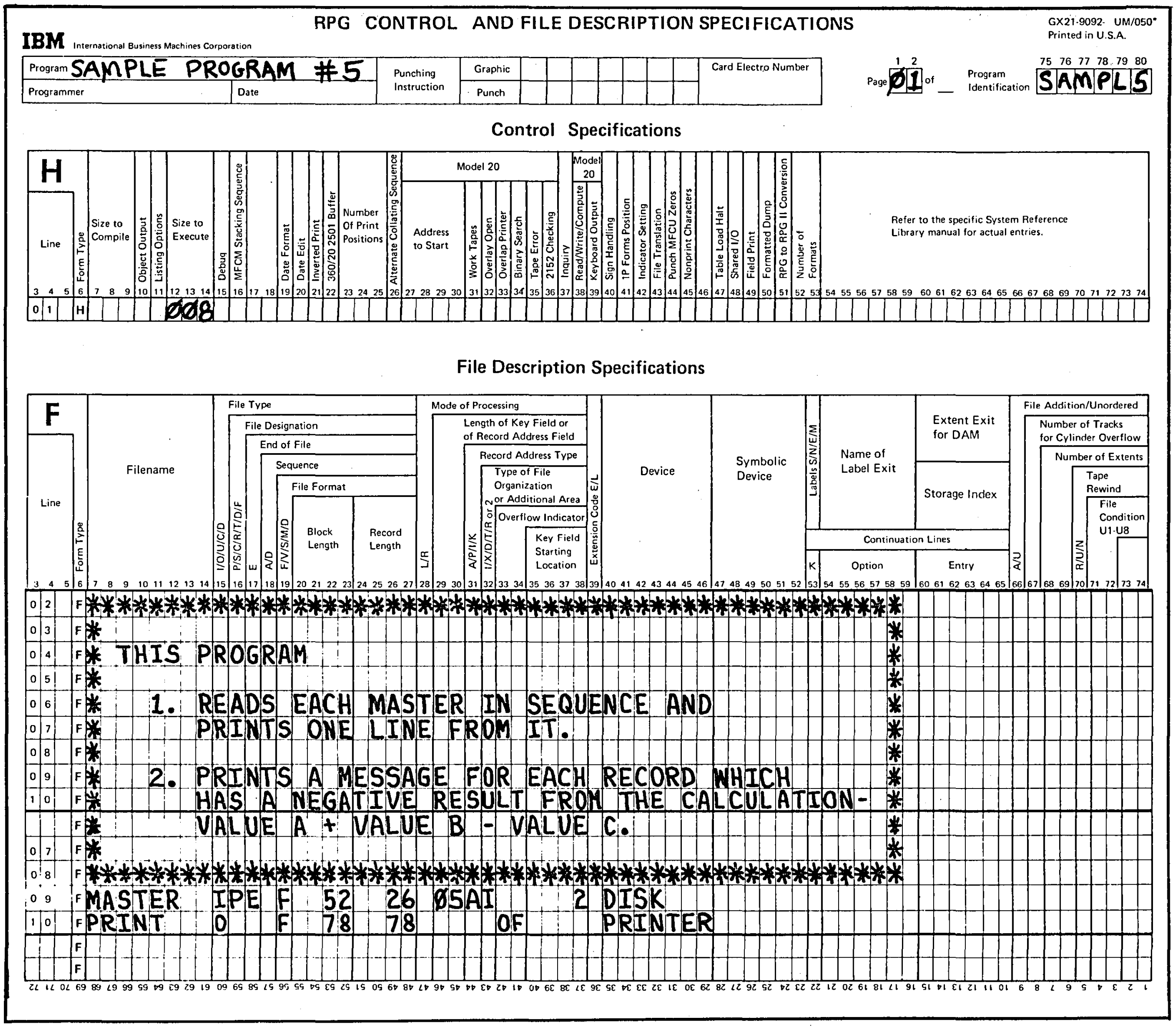
Calculation Specifications

Record identifying indicator 01 conditions all calculations. Values A, B, and C are accumulated (lines 03 through 05). The calculation, value A + value B - value C, is performed and accumulated (lines 01, 02, and 06). If the calculation is negative, resulting indicator 22 is set on to condition the printing of a message.

Output Specifications

These specifications print four header lines, each conditioned by the first page (1P) indicator or an overflow indicator (OF).

One detail line is printed for each program cycle. One total line is also printed when the last record indicator, LR, is on.



IBM International Business Machines Corporation

RPG CONTROL AND FILE DESCRIPTION SPECIFICATIONS

GX21-9092- UM/050* Printed in U.S.A.

Program SAMPLE PROGRAM #5

Programmer

Date

Punching Instruction: Graphic / Punch

Card Electro Number

Page Ø1 of

Program Identification SAMPL5

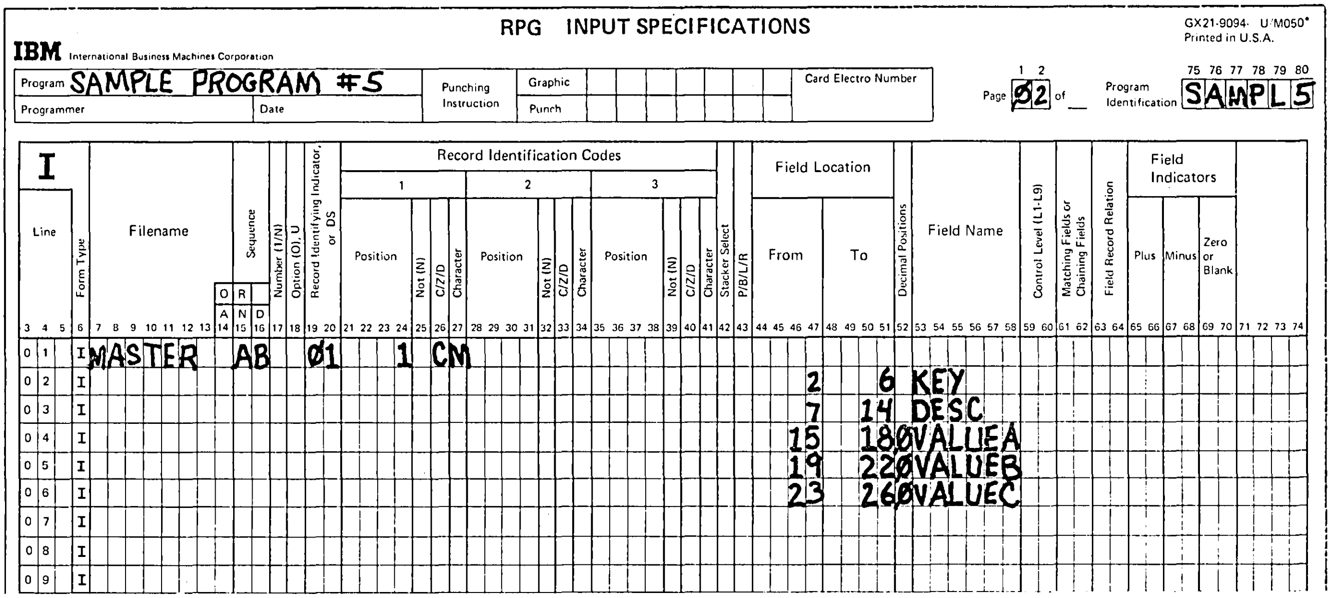
Control Specifications

```
01 H     ØØ8
```

File Description Specifications

```
02 F*****************************************************
03 F*                                                   *
04 F* THIS PROGRAM                                      *
05 F*                                                   *
06 F*   1. READS EACH MASTER IN SEQUENCE AND            *
07 F*      PRINTS ONE LINE FROM IT.                     *
08 F*                                                   *
09 F*   2. PRINTS A MESSAGE FOR EACH RECORD WHICH       *
10 F*      HAS A NEGATIVE RESULT FROM THE CALCULATION-  *
   F*      VALUE A + VALUE B - VALUE C.                 *
07 F*                                                   *
08 F*****************************************************
09 FMASTER  IPE F  52  26 Ø5AI     2 DISK
10 FPRINT   O   F  78  78     OF     PRINTER
```

Figure 19-7 (Part 1 of 4). Specifications for SAMPL5



IBM International Business Machines Corporation

RPG INPUT SPECIFICATIONS

GX21-9094- U/M050* Printed in U.S.A.

Program SAMPLE PROGRAM #5

Programmer

Date

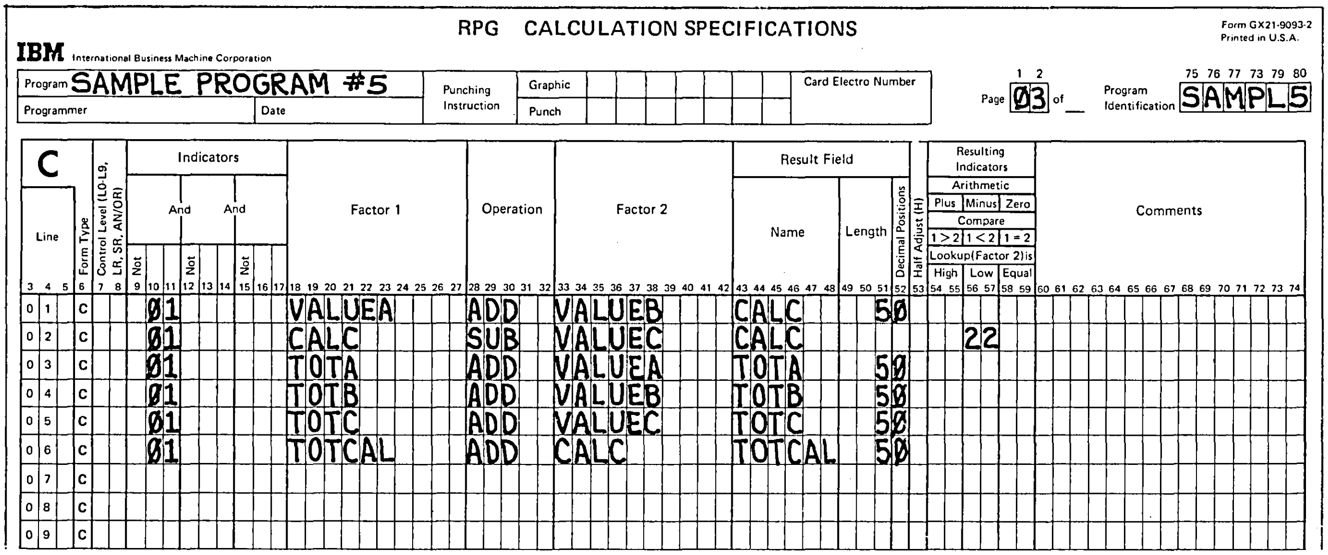
Punching Instruction: Graphic / Punch

Card Electro Number

Page Ø2 of

Program Identification SAMPL5

Line	Form Type	Filename	Sequence	Record Identifying Indicator	Position	C/Z/D	Character	From	To	Decimal Positions	Field Name
01	I	MASTER	AB	Ø1	1	C	M				
02	I							2	6		KEY
03	I							7	14		DESC
04	I							15	18	Ø	VALUEA
05	I							19	22	Ø	VALUEB
06	I							23	26	Ø	VALUEC
07	I										
08	I										
09	I										



IBM International Business Machine Corporation

RPG CALCULATION SPECIFICATIONS

Form GX21-9093-2 Printed in U.S.A.

Program SAMPLE PROGRAM #5

Programmer

Date

Punching Instruction: Graphic / Punch

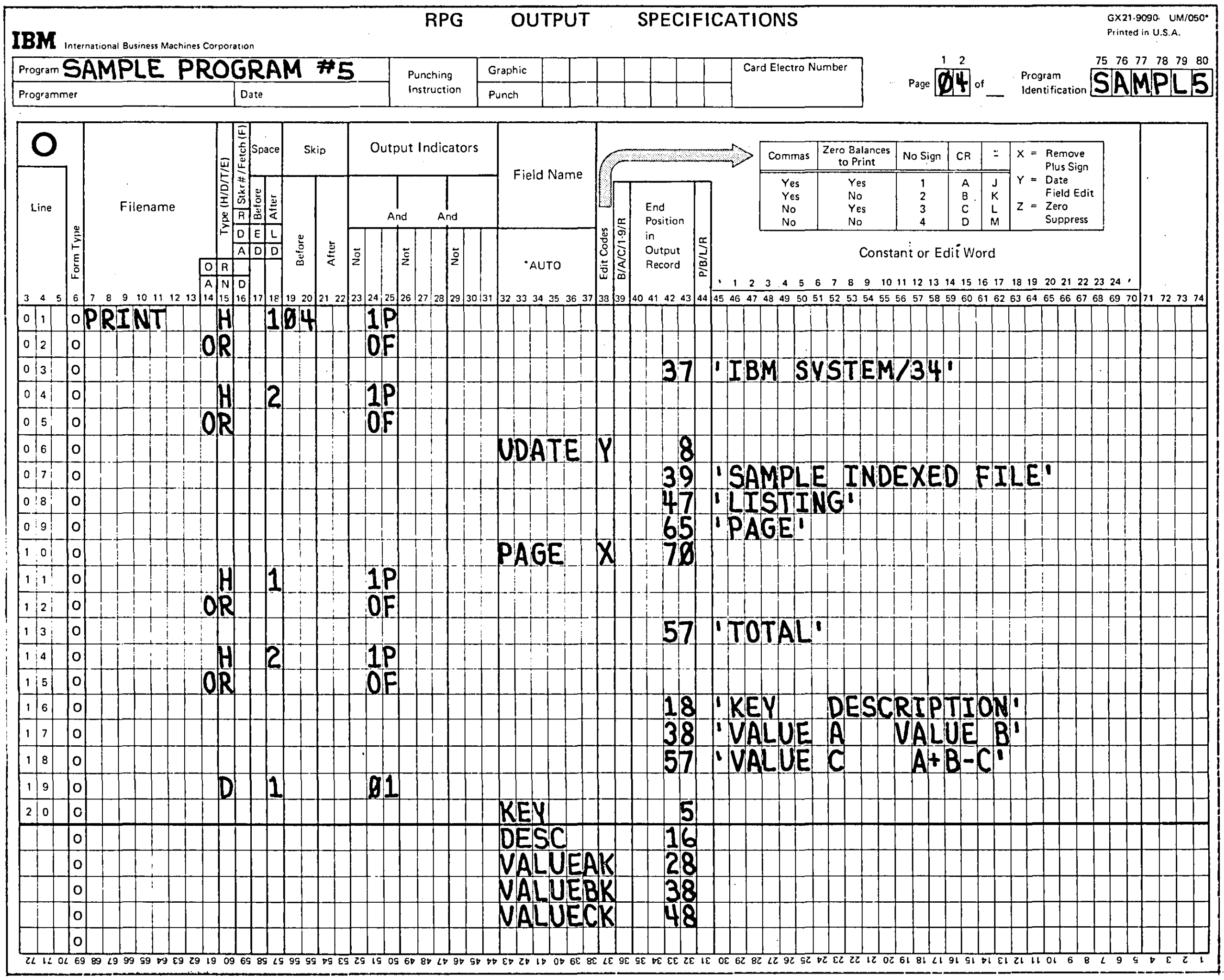
Card Electro Number

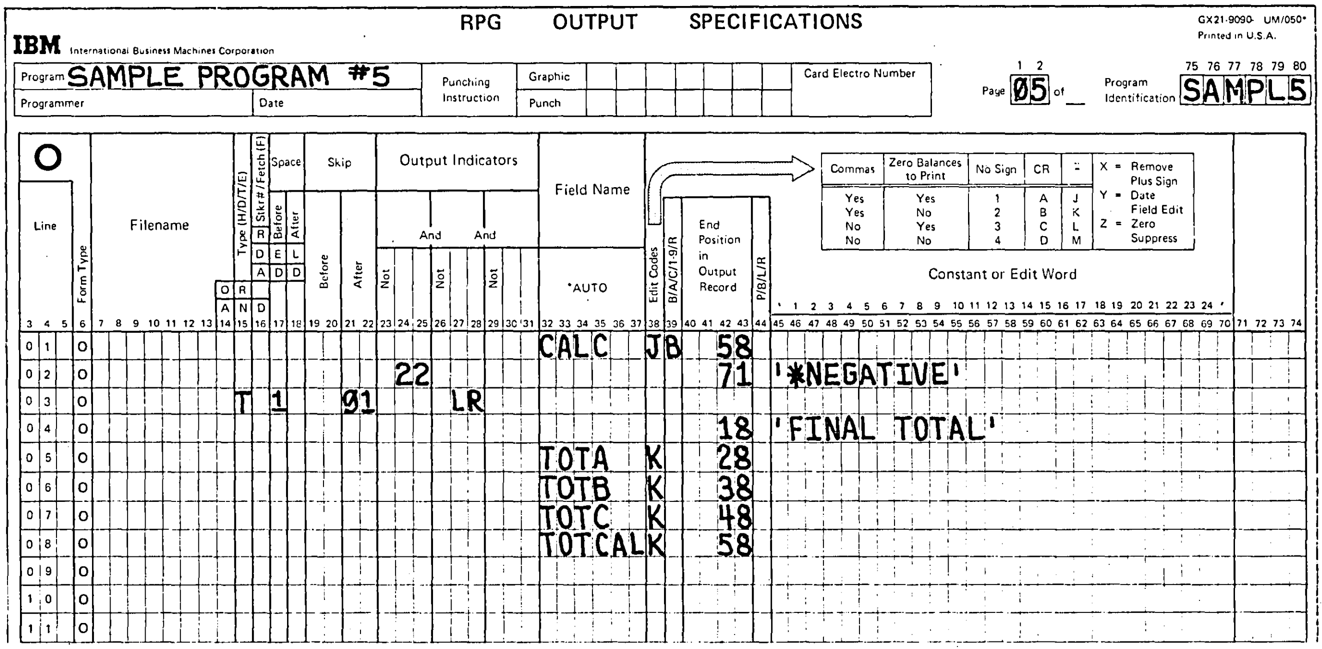
Page Ø3 of

Program Identification SAMPL5

Line	Form Type	Indicators	Factor 1	Operation	Factor 2	Result Field Name	Length	Decimal Positions	Resulting Indicators Minus (Low)
01	C	Ø1	VALUEA	ADD	VALUEB	CALC	5	Ø	
02	C	Ø1	CALC	SUB	VALUEC	CALC			22
03	C	Ø1	TOTA	ADD	VALUEA	TOTA	5	Ø	
04	C	Ø1	TOTB	ADD	VALUEB	TOTB	5	Ø	
05	C	Ø1	TOTC	ADD	VALUEC	TOTC	5	Ø	
06	C	Ø1	TOTCAL	ADD	CALC	TOTCAL	5	Ø	
07	C								
08	C								
09	C								

Figure 19-7 (Part 2 of 4). Specifications for SAMPL5





IBM International Business Machines Corporation

RPG OUTPUT SPECIFICATIONS

GX21-9090- UM/050* Printed in U.S.A.

Program SAMPLE PROGRAM #5

Programmer

Date

Punching Instruction: Graphic / Punch

Card Electro Number

Page Ø5 of __

Program Identification (75–80): SAMPL5

Commas	Zero Balances to Print	No Sign	CR	–
Yes	Yes	1	A	J
Yes	No	2	B	K
No	Yes	3	C	L
No	No	4	D	M

X = Remove Plus Sign
Y = Date Field Edit
Z = Zero Suppress

```
  01 O                         CALC  JB  58
  02 O                 22              71 '*NEGATIVE'
  03 O        T 1   Ø1    LR
  04 O                                 18 'FINAL TOTAL'
  05 O                         TOTA  K   28
  06 O                         TOTB  K   38
  07 O                         TOTC  K   48
  08 O                         TOTCALK   58
  09 O
  10 O
  11 O
```

Figure 19-7 (Part 4 of 4). Specifications for SAMPL5

SAMPLE AUTO REPORT PROGRAM (EXAUT2)

This job prepares a cash receipts register using RPG II with the auto report function. The *AUTO page heading function and the *AUTO output function generate the RPG II output specifications for the report and the calculation specifications to accumulate final totals for several fields on the report. RPG II calculation specifications that cannot be generated by auto report are included in the auto report program to verify the discount taken by each customer and to calculate the balance due.

The file description specifications for the cash receipts register printer file, CSHRECRG, and the file description and input specifications for the input file, CASHRC, are cataloged as separate members in the library (see Figure 19-8). The cataloged specifications are included in the program by the auto report copy function.

The input data for the file CASHRC in EXAUT2 is generated by the program EXAUT1 (see Figure 19-9). Figure 19-10 shows the input data.

Control Specifications

The RPG II control specifications shown in Figure 19-11 should be included in the auto report program because they are not present in the cataloged specifications (see Figure 19-8). None of the control specification options are required in this program, so the specification need contain only an H in column 6 and the program identification, EXAUT2, in columns 75 through 80. This program identification is placed in columns 75 through 80 of all specifications in the generated RPG II source program.

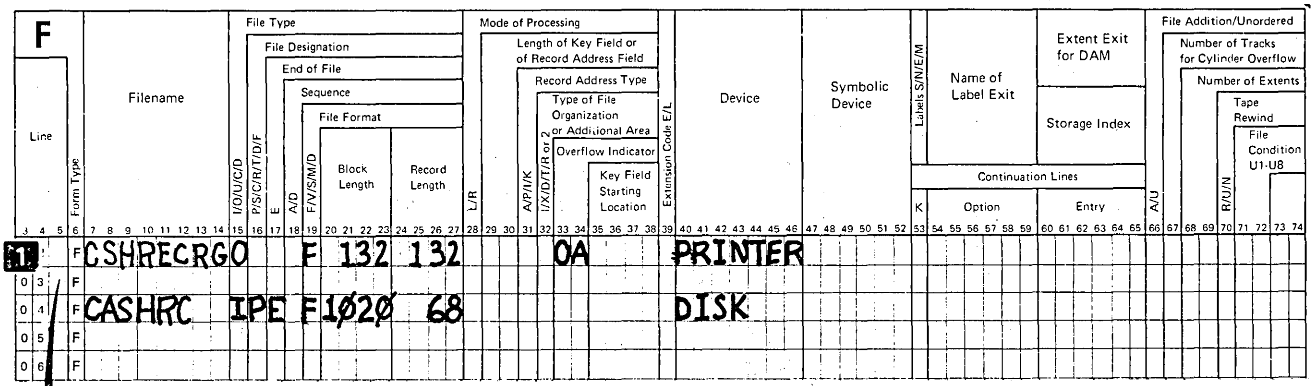
/COPY Statements

The /COPY statements shown in Figure 19-11 copy the file description and input specifications for the job from the system library. The first statement copies the file description specifications for the printer file from the library member named EXAUT3. The second statement copies the file description and input specifications for the disk file, CASHRC, from the library member named EXAUT4. A modifier statement adds an input field definition for the REGION field. As a result of these /COPY statements, the file description and input specifications shown in Figure 19-8 are included in the RPG II source program generated by auto report.

1 The file description for the printer file is in the library member, EXAUT3.

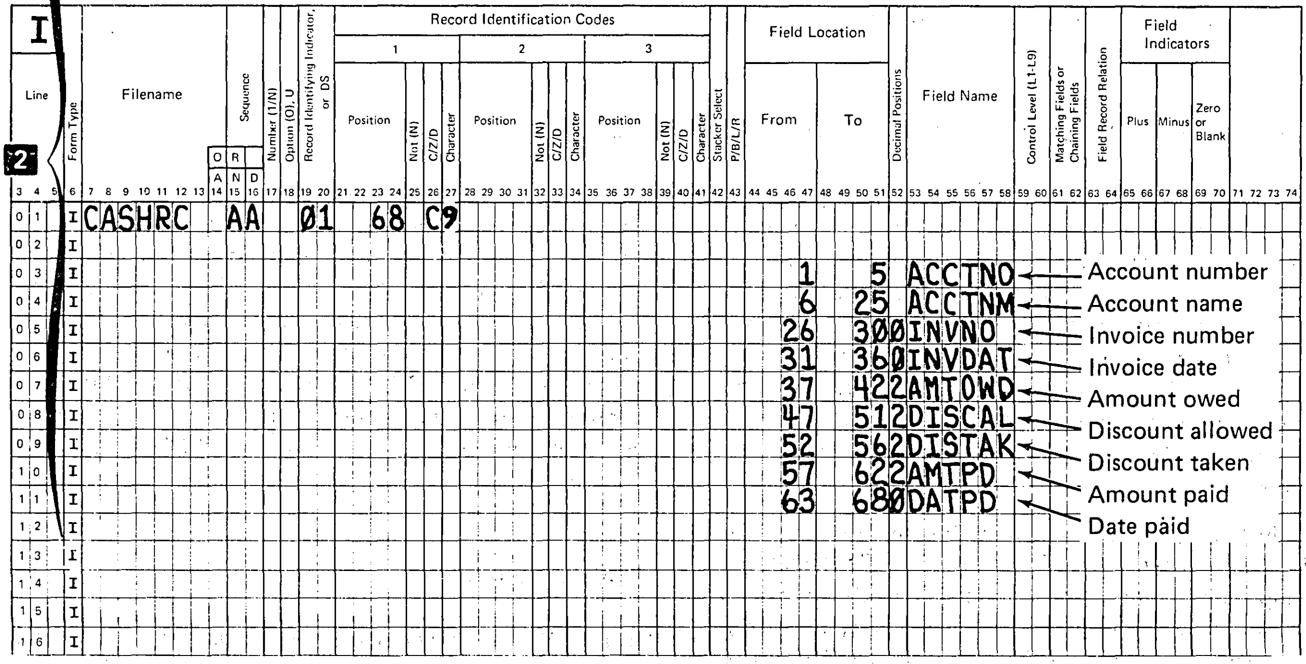
2 The file description and input specifications for the disk file, CASHRC, are in the library member named EXAUT4.

File Description Specifications



```
     FCSHRECRGO   F 132 132     OA     PRINTER
     F
     FCASHRC  IPE F1Ø2Ø  68            DISK
     F
     F
```

Input Specifications



```
     ICASHRC  AA  Ø1  68 C9
     I
     I                                      1   5 ACCTNO
     I                                      6  25 ACCTNM
     I                                     26  3ØØINVNO
     I                                     31  36ØINVDAT
     I                                     37  422AMTOWD
     I                                     47  512DISCAL
     I                                     52  562DISTAK
     I                                     57  622AMTPD
     I                                     63  68ØDATPD
     I
     I
     I
     I
     I
```

Account number
Account name
Invoice number
Invoice date
Amount owed
Discount allowed
Discount taken
Amount paid
Date paid

Figure 19-8. File Description and Input Specifications that are Cataloged in the Library Members EXAUT3 and EXAUT4

```
IBM SYSTEM/34 RPGII AUTO REPORT
 0001  01010H      008
 0002  0102 FKEYIN   IP  F 100 100            KEYBORD
 0003  0103 FCASHRC  O   F1020  68            DISK
 0004  0104 FPRINTER O   F 120 120            PRINTER
 0005  0203 C                    SETOF                      02
 0006  0204 C                    SETON                      LR
 0007  0205 C    02              KEY01           DUMMY    1
 0008  0301 OPRINTER T   301  LR
 0009  0302 O                                      54 'DATA FOR SAMPLE PROGRAM'
 0010  030210                      02DUMMY         56
 0011  0303 O        T   2    LR
 0012  0304 O                                      24 '11243JONES HARDWARE     '
 0013  0305 O                                      48 ' 27541123199  2375CASH  '
 0014  0306 O                                      68 ' 47    47  2328123199'
 0015  0307 O        T   2    LR
 0016  0308 O                                      24 '11352NU-STYLE CLOTHIERS '
 0017  0309 O                                      48 ' 27987123199  8707CASH  '
 0018  0310 O                                      68 '174         4000123199'
 0019  0311 O        T   2    LR
 0020  0312 O                                      24 '11886MIDI FASHIONS INC  '
 0021  0313 O                                      48 ' 15771123199 10722CASH  '
 0022  0314 O                                      68 '214   214 10508123199'
 0023  0401 O        T   2    LR
 0024  0402 O                                      24 '12874ULOOK INTERIORS    '
 0025  0403 O                                      48 ' 25622123199  6795CASH  '
 0026  0404 O                                      68 '136         6795123199'
 0027  0405 O        T   2    LR
 0028  0406 O                                      24 '18274STREAMLINE PAPER IN'
 0029  0407 O                                      48 'C29703123199 27403      '
 0030  0408 O                                      68 '548   238 17055123199'
 0031  0409 O        T   2    LR
 0032  0410 O                                      24 '23347RITE-BEST PENS CO  '
 0033  0411 O                                      48 ' 20842123199  1580      '
 0034  0412 O                                      68 '31          1000123199'
 0035  0413 O        T   2    LR
 0036  0414 O                                      24 '25521IMPORTS OF NM      '
 0037  0415 O                                      48 ' 29273123199 79740     1'
 0038  0416 O                                      68 '593 1193 58547123199'
 0039  0501 O        T   2    LR
 0040  0502 O                                      24 '26723ALRIGHT CLEANERS   '
 0041  0503 O                                      48 ' 19473123199 46200CASH  '
 0042  0504 O                                      68 '924        46200123199'
 0043  0505 O        T   2    LR
 0044  0506 O                                      24 '28622NORTH CENTRAL SUPPL'
 0045  0507 O                                      48 'Y17816123199  7597CASH  '
 0046  0508 O                                      68 '152         7597123199'
 0047  0509 O        T   2    LR
 0048  0510 O                                      24 '29871FERGUSON DEALERS   '
 0049  0511 O                                      48 ' 27229123199  6191CASH  '
 0050  0512 O                                      68 '124         6191123199'
 0051  0513 O        T   2    LR
 0052  0514 O                                      24 '30755FASTWAY AIRLINES   '
 0053  0515 O                                      48 ' 26158123199 74272CASH 1'
 0054  0516 O                                      68 '495 1685 72587123199'
 0055  0517 O        T   2    LR
 0056  0518 O                                      24 '31275ENVIRONMENT CONCERN'
 0057  0519 O                                      48 'S20451123199  2943      '
 0058  0520 O                                      68 ' 59         1500123199'
```

Figure 19-9 (Part 1 of 3). EXAUT1 Program

```
0059  0601 0         T  2     LR
0060  0602 0                                     24 '32457B SOLE SILOS       '
0061  0603 0                                     48 ' 27425123199 11005CASH  '
0062  0604 0                                     68 '220      11005123199'
0063  0605 0         T  2     LR
0064  0606 0                                     24 '37945HOFFTA BREAKS INC  '
0065  0607 0                                     48 ' 18276123199  4723CASH  '
0066  0608 0                                     68 ' 94       4723123199'
0067  0609 0         T  2     LR
0068  0610 0                                     24 '42622EASTLAKE GRAVEL CO '
0069  0611 0                                     48 ' 16429123199  2937CASH  '
0070  0612 0                                     68 ' 58       2937123199'
0071  0613 0CASHRC   T        LR
0072  0704 0                                     24 '11243JONES HARDWARE     '
0073  0705 0                                     48 ' 27541123199  2375CASH  '
0074  0706 0                                     68 ' 47   47  2328123199'
0075  0707 0         T        LR
0076  0708 0                                     24 '11352NU-STYLE CLOTHIERS '
0077  0709 0                                     48 ' 27987123199  8707CASH  '
0078  0710 0                                     68 '174       4000123199'
0079  0711 0         T        LR
0080  0712 0                                     24 '11886MIDI FASHIONS INC  '
0081  0713 0                                     48 ' 15771123199 10722CASH  '
0082  0714 0                                     68 '214  214 10508123199'
0083  0801 0         T        LR
0084  0802 0                                     24 '12874ULOOK INTERIORS    '
0085  0803 0                                     48 ' 25622123199  6795CASH  '
0086  0804 0                                     68 '136       6795123199'
0087  0805 0         T        LR
0088  0806 0                                     24 '18274STREAMLINE PAPER IN'
0089  0807 0                                     48 'C29703123199 27403      '
0090  0808 0                                     68 '548  238 17055123199'
0091  0809 0         T        LR
0092  0810 0                                     24 '23347RITE-BEST PENS CO  '
0093  0811 0                                     48 ' 20842123199  1580      '
0094  0812 0                                     68 '31        1000123199'
0095  0813 0         T        LR
0096  0814 0                                     24 '25521IMPORTS OF NM      '
0097  0815 0                                     48 ' 29273123199 79740     1'
0098  0816 0                                     68 '593 1193 58547123199'
0099  0901 0         T        LR
0100  0902 0                                     24 '26723ALRIGHT CLEANERS   '
0101  0903 0                                     48 ' 19473123199 46200CASH  '
0102  0904 0                                     68 '924      46200123199'
0103  0905 0         T        LR
0104  0906 0                                     24 '28622NORTH CENTRAL SUPPL'
0105  0907 0                                     48 'Y17816123199  7597CASH  '
0106  0908 0                                     68 '152       7597123199'
0107  0909 0         T        LR
0108  0910 0                                     24 '29871FERGUSON DEALERS   '
0109  0911 0                                     48 ' 27229123199  6191CASH  '
0110  0912 0                                     68 '124       6191123199'
0111  0913 0         T        LR
0112  0914 0                                     24 '30755FASTWAY AIRLINES   '
0113  0915 0                                     48 ' 26158123199 74272CASH 1'
0114  0916 0                                     68 '495 1685 72587123199'
0115  0917 0         T        LR
0116  0918 0                                     24 '31275ENVIRONMENT CONCERN'
0117  0919 0                                     48 'S20451123199  2943      '
```

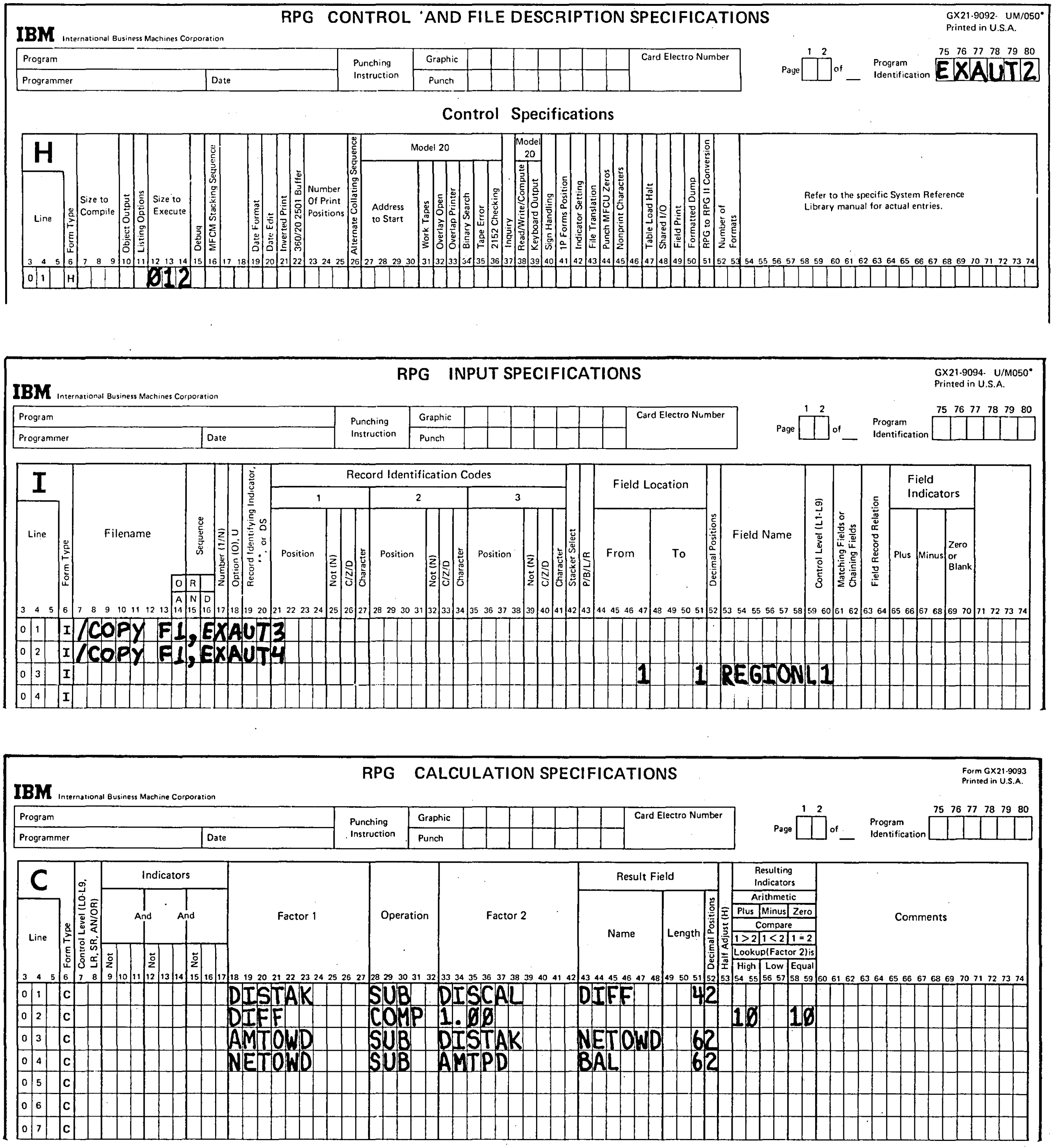
Figure 19-9 (Part 2 of 3). EXAUT1 Program

```
0118  0920 0                              68 ' 59          1500123199'
0119  1001 0        T         LR
0120  1002 0                              24 '32457B SOLE SILOS        '
0121  1003 0                              48 ' 27425123199 11005CASH   '
0122  1004 0                              68 '220         11005123199'
0123  1005 0        T         LR
0124  1006 0                              24 '37945HOFFTA BREAKS INC   '
0125  1007 0                              48 ' 18276123199  4723CASH   '
0126  1008 0                              68 ' 94          4723123199'
0127  1009 0        T         LR
0128  1010 0                              24 '42622EASTLAKE GRAVEL CO '
0129  1011 0                              48 ' 16429123199  2937CASH   '
0130  1012 0                              68 ' 58          2937123199'
```

Figure 19-9 (Part 3 of 3). EXAUT1 Program

```
                             DATA FOR SAMPLE PROGRAM

11243JONES HARDWARE       27541123199  2375CASH   47   47  2328123199

11352NU-STYLE CLOTHIERS   27987123199  8707CASH  174       4000123199

11886MIDI FASHIONS INC    15771123199 10722CASH  214  214 10508123199

12874ULOOK INTERIORS      25622123199  6795CASH  136       6795123199

18274STREAMLINE PAPER INC29703123199 27403       548  238 17055123199

23347RITE-BEST PENS CO    20842123199  1580      31        1000123199

25521IMPORTS OF NM        29273123199 79740     1593 1193 58547123199

26723ALRIGHT CLEANERS     19473123199 46200CASH  924      46200123199

28622NORTH CENTRAL SUPPLY17816123199  7597CASH  152       7597123199

29871FERGUSON DEALERS     27229123199  6191CASH  124       6191123199

30755FASTWAY AIRLINES     26158123199 74272CASH 1495 1685 72587123199

31275ENVIRONMENT CONCERNS20451123199  2943        59       1500123199

32457B SOLE SILOS         27425123199 11005CASH  220      11005123199

37945HOFFTA BREAKS INC    18276123199  4723CASH   94       4723123199

42622EASTLAKE GRAVEL CO   16429123199  2937CASH   58       2937123199
```

Figure 19-10. Input Data Generated by EXAUT1 for Auto Report Sample Program EXAUT2



IBM International Business Machines Corporation

RPG CONTROL AND FILE DESCRIPTION SPECIFICATIONS

GX21-9092- UM/050* Printed in U.S.A.

Program Identification EXAUT2

Control Specifications

Line	Form Type	Size to Execute (12-14)
01	H	Ø12

Refer to the specific System Reference Library manual for actual entries.

IBM International Business Machines Corporation

RPG INPUT SPECIFICATIONS

GX21-9094- U/M050* Printed in U.S.A.

Line	Form Type	Filename / Entry	Field Location From	Field Location To	Field Name	Control Level (L1-L9)
01	I	/COPY F1,EXAUT3				
02	I	/COPY F1,EXAUT4				
03	I		1	1	REGION	L1
04	I					

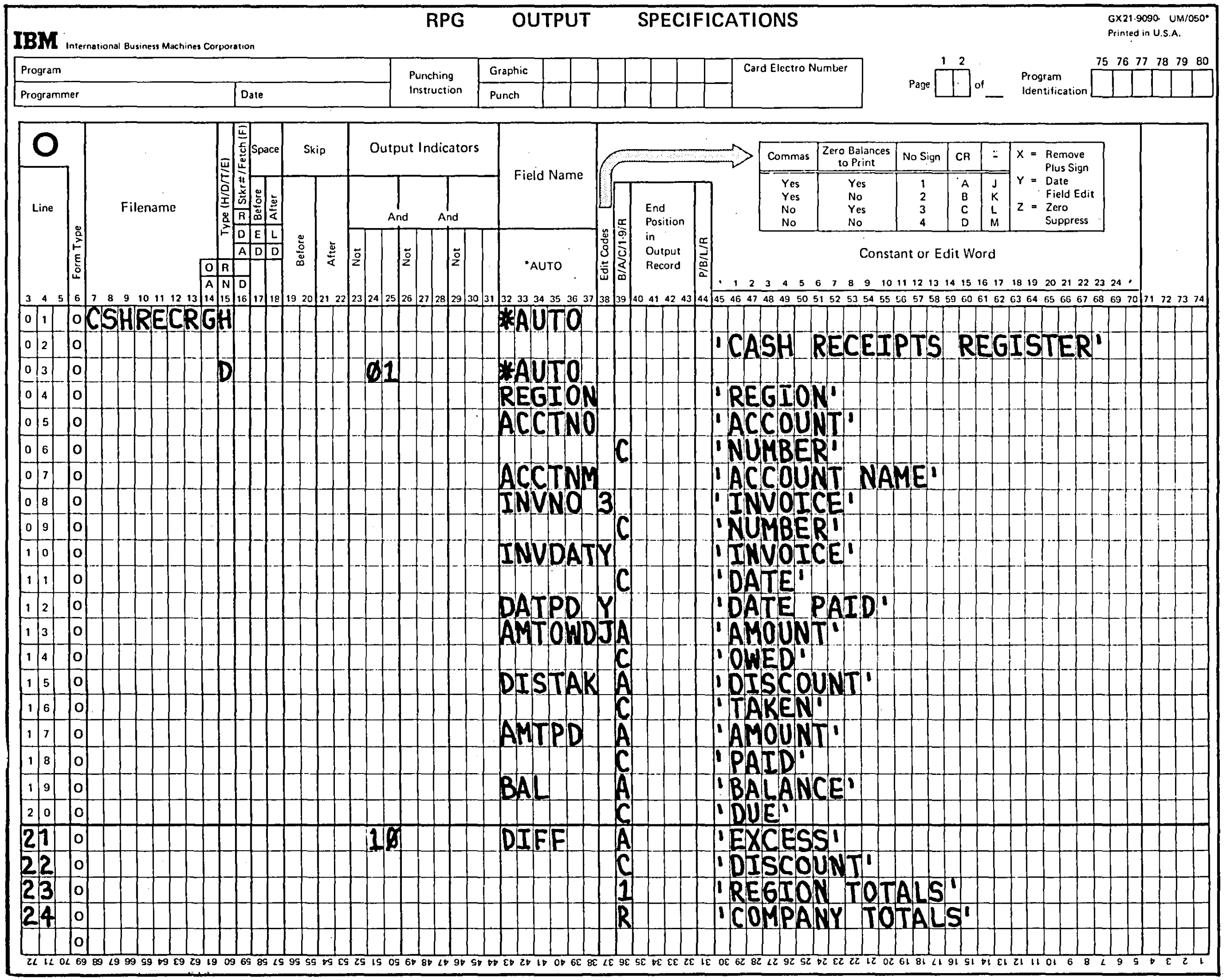
IBM International Business Machine Corporation

RPG CALCULATION SPECIFICATIONS

Form GX21-9093 Printed in U.S.A.

Line	Form Type	Factor 1	Operation	Factor 2	Result Field Name	Length	Decimal Positions	High (54-55)	Low (56-57)	Equal (58-59)
01	C	DISTAK	SUB	DISCAL	DIFF	4	2			
02	C	DIFF	COMP	1.ØØ				1Ø		1Ø
03	C	AMTOWD	SUB	DISTAK	NETOWD	6	2			
04	C	NETOWD	SUB	AMTPD	BAL	6	2			
05	C									
06	C									
07	C									

Figure 19-11 (Part 1 of 2). RPG II and Auto Report Specifications to Produce the Cash Receipts Register



IBM International Business Machines Corporation

RPG OUTPUT SPECIFICATIONS

GX21-9090- UM/050*
Printed in U.S.A.

Program | Punching Instruction | Graphic | Punch | Card Electro Number | Page of | Program Identification

Programmer | Date

Commas	Zero Balances to Print	No Sign	CR	-	
Yes	Yes	1	A	J	X = Remove Plus Sign
Yes	No	2	B	K	Y = Date Field Edit
No	Yes	3	C	L	Z = Zero Suppress
No	No	4	D	M	

Line	Form Type	Filename	Type (H/D/T/E)	Output Indicators	Field Name	Edit Codes	B/A/C/1/9/R	Constant or Edit Word
01	O	CSHRECRG	H		*AUTO			
02	O							'CASH RECEIPTS REGISTER'
03	O		D	Ø1	*AUTO			
04	O				REGION			'REGION'
05	O				ACCTNO			'ACCOUNT'
06	O						C	'NUMBER'
07	O				ACCTNM			'ACCOUNT NAME'
08	O				INVNO	3		'INVOICE'
09	O						C	'NUMBER'
10	O				INVDAT	Y		'INVOICE'
11	O						C	'DATE'
12	O				DATPD	Y		'DATE PAID'
13	O				AMTOWD	J	A	'AMOUNT'
14	O						C	'OWED'
15	O				DISTAK		A	'DISCOUNT'
16	O						C	'TAKEN'
17	O				AMTPD		A	'AMOUNT'
18	O						C	'PAID'
19	O				BAL		A	'BALANCE'
20	O						C	'DUE'
21	O			1Ø	DIFF		A	'EXCESS'
22	O						C	'DISCOUNT'
23	O						1	'REGION TOTALS'
24	O						R	'COMPANY TOTALS'
	O							

Figure 19-11 (Part 2 of 2). RPG II and Auto Report Specifications to Produce the Cash Receipts Register

Calculation Specifications

The calculation specifications shown in Figure 19-11 are included in the auto report program to perform special operations that cannot be generated by auto report. First, the discount allowed for each customer is subtracted from the discount taken by each customer. Indicator 10 turns off if the difference is greater than or equal to $1.00. The remaining calculations subtract the discount taken and the amount paid from the amount owed.

The order in which these calculations are placed in relation to the calculations generated by auto report is shown in the auto report listing of the generated RPG II source program (see Figure 19-12).

*AUTO Specifications

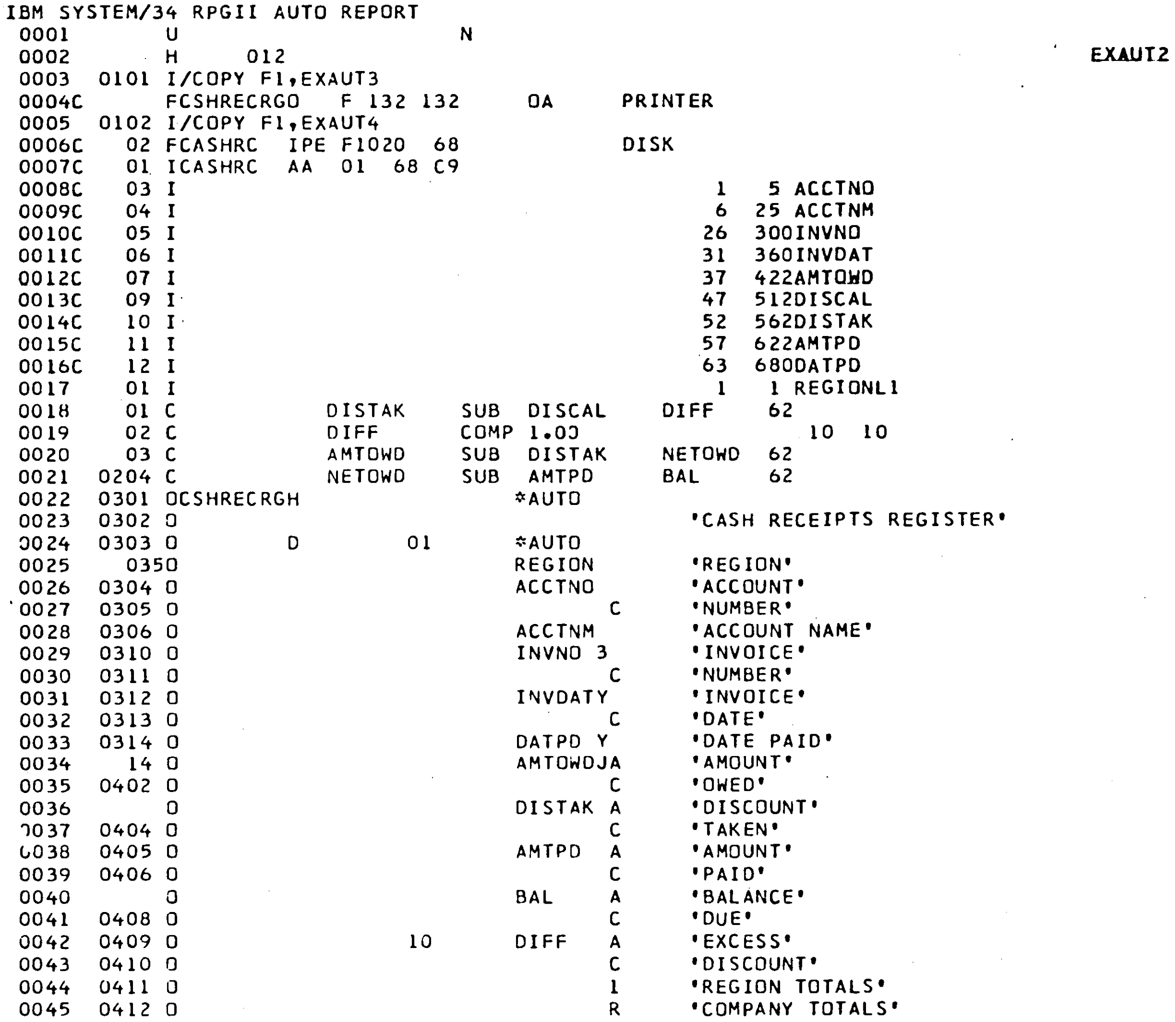
The coding for the *AUTO page heading and the *AUTO output functions is shown in Figure 19-11. Notice that the Y edit code is used for the date fields (lines 10 and 12). Auto report generates a K edit code for numeric fields when an edit code is not specified. No edit code is generated for numeric fields when they are described with a digit (1 through 9) or R in column 39. The edit code 3 is specified for the INVNO field to suppress the printing of the comma edit character.

DIFF is printed on the detail line only if it is $1.00 or more. Remember, output indicator 10 conditions only the printing of the field on the detail line; it does not affect the printing of the generated field on the total line.

The J edit code allows zero balance to print for the AMTOWD field.

Totals are accumulated and printed by auto report for five fields as indicated by A entries in column 39. Because an L1 control level is defined in the input field specifications for REGION, which is added to the input specifications for CASHRC (see Figure 19-11), regional and final totals are accumulated for each field that has an A in column 39. The total lines are identified by the literals shown in lines 23 and 24 of the *AUTO specifications (see Figure 19-11).

Figure 19-13 shows the output data produced by EXAUT2.



```
IBM SYSTEM/34 RPGII AUTO REPORT
 0001       U                     N
 0002       H     012                                                            EXAUT2
 0003  0101 I/COPY F1,EXAUT3
 0004C      FCSHRECRGO   F 132 132     OA      PRINTER
 0005  0102 I/COPY F1,EXAUT4
 0006C   02 FCASHRC  IPE F1020  68            DISK
 0007C   01 ICASHRC  AA  01   68 C9
 0008C   03 I                                        1   5 ACCTNO
 0009C   04 I                                        6  25 ACCTNM
 0010C   05 I                                       26  300INVNO
 0011C   06 I                                       31  360INVDAT
 0012C   07 I                                       37  422AMTOWD
 0013C   09 I                                       47  512DISCAL
 0014C   10 I                                       52  562DISTAK
 0015C   11 I                                       57  622AMTPD
 0016C   12 I                                       63  680DATPD
 0017    01 I                                        1   1 REGIONL1
 0018    01 C           DISTAK    SUB  DISCAL    DIFF    62
 0019    02 C           DIFF      COMP 1.00                 10  10
 0020    03 C           AMTOWD    SUB  DISTAK    NETOWD  62
 0021  0204 C           NETOWD    SUB  AMTPD     BAL     62
 0022  0301 OCSHRECRGH                *AUTO
 0023  0302 O                                     'CASH RECEIPTS REGISTER'
 0024  0303 O        D        01      *AUTO
 0025    0350                         REGION      'REGION'
 0026  0304 O                         ACCTNO      'ACCOUNT'
 0027  0305 O                                C     'NUMBER'
 0028  0306 O                         ACCTNM      'ACCOUNT NAME'
 0029  0310 O                         INVNO 3     'INVOICE'
 0030  0311 O                                C     'NUMBER'
 0031  0312 O                         INVDATY     'INVOICE'
 0032  0313 O                                C     'DATE'
 0033  0314 O                         DATPD Y     'DATE PAID'
 0034    14 O                         AMTOWDJA     'AMOUNT'
 0035  0402 O                                C     'OWED'
 0036       O                         DISTAK A     'DISCOUNT'
 0037  0404 O                                C     'TAKEN'
 0038  0405 O                         AMTPD  A     'AMOUNT'
 0039  0406 O                                C     'PAID'
 0040       O                         BAL    A     'BALANCE'
 0041  0408 O                                C     'DUE'
 0042  0409 O                 10      DIFF   A     'EXCESS'
 0043  0410 O                                C     'DISCOUNT'
 0044  0411 O                                1     'REGION TOTALS'
 0045  0412 O                                R     'COMPANY TOTALS'
```

Figure 19-12 (Part 1 of 3). Auto Report Sample Program (EXAUT2)

```
          IBM SYSTEM/34 RPGII COMPILER

      0010 H     012                                                             EXAUT2

0001  0020CFCSHRECRGO   F 132 132     OA     PRINTER                             EXAUT2
0002  0030CFCASHRC  IPE F1020   68           DISK                                EXAUT2

      0040 I*/COPY F1,EXAUT3                                                     EXAUT2
      0050 I*/COPY F1,EXAUT4                                                     EXAUT2
0003  0060CICASHRC  AA  01  68 C9                                                EXAUT2
0004  0070CI                                         1   5 ACCTNO                EXAUT2
0005  0080CI                                         6  25 ACCTNM                EXAUT2
0006  0090CI                                        26  300INVNO                 EXAUT2
0007  0100CI                                        31  360INVDAT                EXAUT2
0008  0110CI                                        37  422AMTOWD                EXAUT2
0009  0120CI                                        47  512DISCAL                EXAUT2
0010  0130CI                                        52  562DISTAK                EXAUT2
0011  0140CI                                        57  622AMTPD                 EXAUT2
0012  0150CI                                        63  680DATPD                 EXAUT2
0013  0160 I                                         1   1 REGIONL1              EXAUT2

0014  0170 C           DISTAK    SUB  DISCAL    DIFF    62                       EXAUT2
0015  0180 C           DIFF      COMP 1.00                  10  10               EXAUT2
0016  0190 C           AMTOWD    SUB  DISTAK    NETOWD  62                       EXAUT2
0017  0200 C           NETOWD    SUB  AMTPD     BAL     62                       EXAUT2
0018  0210EC   01                EXSR A$$SUM                                     EXAUT2
0019  0220ECL1         AMTOWR    ADD  AMTOW1    AMTOWR  82                       EXAUT2
0020  0230ECL1         DISTAR    ADD  DISTA1    DISTAR  72                       EXAUT2
0021  0240ECL1         AMTPDR    ADD  AMTPD1    AMTPDR  82                       EXAUT2
0022  0250ECL1         BALR      ADD  BAL1      BALR    82                       EXAUT2
0023  0260ECL1         DIFFR     ADD  DIFF1     DIFFR   82                       EXAUT2
0024  0270ECSR         A$$SUM    BEGSR                                           EXAUT2
0025  0280ECSR         AMTOW1    ADD  AMTOWD    AMTOW1  82                       EXAUT2
0026  0290ECSR         DISTA1    ADD  DISTAK    DISTA1  72                       EXAUT2
0027  0300ECSR         AMTPD1    ADD  AMTPD     AMTPD1  82                       EXAUT2
0028  0310ECSR         BAL1      ADD  BAL       BAL1    82                       EXAUT2
0029  0320ECSR 10      DIFF1     ADD  DIFF      DIFF1   82                       EXAUT2
0030  0330ECSR                   ENDSR                                           EXAUT2

0031  0340EOCSHRECRGH  206   1P                                                  EXAUT2
0032  0350EO       OR        OA                                                  EXAUT2
0033  0360EO                                   76 'CASH RECEIPTS REGISTER'       EXAUT2
0034  0370EO                         UDATE Y    8                                EXAUT2
0035  0380EO                         PAGE  Z  131                                EXAUT2
0036  0390EO                                  127 'PAGE '                        EXAUT2
0037  0400EOCSHRECRGH  1     1P                                                  EXAUT2
0038  0410EO       OR        OA                                                  EXAUT2
0039  0420EO                                    6 'REGION'                       EXAUT2
0040  0430EO                                   15 'ACCOUNT'                      EXAUT2
```

Figure 19-12 (Part 2 of 3). Auto Report Sample Program (EXAUT2)

```
0041   0440EO                                   29 'ACCOUNT NAME'                EXAUT2
0042   0450EO                                   46 'INVOICE'                     EXAUT2
0043   0460EO                                   56 'INVOICE'                     EXAUT2
0044   0470EO                                   67 'DATE PAID'                   EXAUT2
0045   0480EO                                   80 'AMOUNT'                      EXAUT2
0046   0490EO                                   92 'DISCOUNT'                    EXAUT2
0047   0500EO                                  105 'AMOUNT'                      EXAUT2
0048   0510EO                                  118 'BALANCE'                     EXAUT2
0049   0520EO                                  130 'EXCESS'                      EXAUT2
0050   0530EOCSHRECRGH  2     1P                                                 EXAUT2
0051   0540EO       OR        0A                                                 EXAUT2
0052   0550EO                                   14 'NUMBER'                      EXAUT2
0053   0560EO                                   45 'NUMBER'                      EXAUT2
0054   0570EO                                   54 'DATE'                        EXAUT2
0055   0580EO                                   79 'OWED'                        EXAUT2
0056   0590EO                                   90 'TAKEN'                       EXAUT2
0057   0600EO                                  104 'PAID'                        EXAUT2
0058   0610EO                                  116 'DUE'                         EXAUT2
0059   0620EO                                  131 'DISCOUNT'                    EXAUT2
0060   0630EOCSHRECRGD  1     01                                                 EXAUT2
0061   0640EO                         REGION      3                              EXAUT2
0062   0650EO                         ACCTNO     14                              EXAUT2
0063   0660EO                         ACCTNM     37                              EXAUT2
0064   0670EO                         INVNO 3    45                              EXAUT2
0065   0680EO                         INVDATY    56                              EXAUT2
0066   0690EO                         DATPD Y    66                              EXAUT2
0067   0700EO                         AMTOWDJB   80                              EXAUT2
0068   0710EO                         DISTAKKB   92                              EXAUT2
0069   0720EO                         AMTPD KB  105                              EXAUT2
0070   0730EO                         BAL   KB  118                              EXAUT2
0071   0740EO                 10      DIFF  KB  131                              EXAUT2
0072   0750EOCSHRECRGT 12     L1                                                 EXAUT2
0073   0760EO                         AMTOW1JB   80                              EXAUT2
0074   0770EO                         DISTA1KB   92                              EXAUT2
0075   0780EO                         AMTPD1KB  105                              EXAUT2
0076   0790EO                         BAL1  KB  118                              EXAUT2
0077   0800EO                         DIFF1 KB  131                              EXAUT2
0078   0810EO                                   67 'REGION TOTALS'               EXAUT2
0079   0820EOCSHRECRGT 12     LR                                                 EXAUT2
0080   0830EO                         AMTOWRJB   80                              EXAUT2
0081   0840EO                         DISTARKB   92                              EXAUT2
0082   0850EO                         AMTPDRKB  105                              EXAUT2
0083   0860EO                         BALR  KB  118                              EXAUT2
0084   0870EO                         DIFFR KB  131                              EXAUT2
0085   0880EO                                   67 'COMPANY TOTALS'              EXAUT2
```

Figure 19-12 (Part 3 of 3). Auto Report Sample Program (EXAUT2)

```
                                                    CASH RECEIPTS REGISTER                                                                       PAGE    1

REGION  ACCOUNT  ACCOUNT NAME            INVOICE   INVOICE   DATE PAID      AMOUNT    DISCOUNT      AMOUNT       BALANCE      EXCESS
        NUMBER                           NUMBER    DATE                      OWED      TAKEN         PAID          DUE        DISCOUNT

  1      11243   JONES HARDWARE           27541   12/31/99  12/31/99         23.75       .47         23.28
  1      11352   NU-STYLE CLOTHIERS       27987   12/31/99  12/31/99         87.07                   40.00         47.07
  1      11886   MIDI FASHIONS INC        15771   12/31/99  12/31/99        107.22      2.14        105.08
  1      12874   ULOOK INTERIORS          25622   12/31/99  12/31/99         67.95                   67.95
  1      18274   STREAMLINE PAPER INC     29703   12/31/99  12/31/99        274.03      2.38        170.55        101.10

                                                      REGION TOTALS         560.02      4.99        406.86        148.17

  2      23347   RITE-BEST PENS CO        20842   12/31/99  12/31/99         15.80                   10.00          5.80
  2      25521   IMPORTS OF NM            29273   12/31/99  12/31/99        797.40     11.93        585.47        200.00
  2      26723   ALRIGHT CLEANERS         19473   12/31/99  12/31/99        462.00                  462.00
  2      28622   NORTH CENTRAL SUPPLY     17816   12/31/99  12/31/99         75.97                   75.97
  2      29871   FERGUSON DEALERS         27229   12/31/99  12/31/99         61.91                   61.91

                                                      REGION TOTALS       1,413.08     11.93      1,195.35        205.80

  3      30755   FASTWAY AIRLINES         26158   12/31/99  12/31/99        742.72     16.85        725.87                      1.90
  3      31275   ENVIRONMENT CONCERNS     20451   12/31/99  12/31/99         29.43                   15.00         14.43
  3      32457   B SOLE SILOS             27425   12/31/99  12/31/99        110.05                  110.05
  3      37945   HOFFTA BREAKS INC        18276   12/31/99  12/31/99         47.23                   47.23

                                                      REGION TOTALS         929.43     16.85        898.15         14.43        1.90

  4      42622   EASTLAKE GRAVEL CO       16429   12/31/99  12/31/99         29.37                   29.37

                                                      REGION TOTALS          29.37                   29.37

                                                     COMPANY TOTALS       2,931.90     33.77      2,529.73        368.40        1.90
```

Figure 19-13. Output from Auto Report Sample Program (EXAUT2)

SAMPLE WORKSTN FILE PROGRAM (ORD)

The sample program ORD creates records for the disk file TRANS (see Figures 19-14 and 19-15). The display format ZCNUM prompts the operator to enter a customer number. The display format ZSHIP displays the customer name and ship-to headings and fields, and allows the operator to change the ship-to fields. The display format ZITEM prompts the operator for the item number and quantity on line 22 of the display screen. The display formats SHOWITEM and ZITEMHED display the item number, quantity, description, price, amount, and total headings and fields. The ZITEMHED format also requests the operator to enter the next item number and quantity (line 22).

A code 2 indicates that the customer name and address record is written to the TRANS file. A code 4 indicates that the ship-to name and address record is written to the TRANS file. A code 6, which is created when three items have been entered, indicates that the items record is written to the TRANS file.

The following command keys are allowed in this program:

- Cmd key 1 (which corresponds to indicator KA) completes an order and:
  - Creates a final code 6 record, if needed
  - Creates a code 8 (total) record
  - Enables input to allow the operator to enter the next customer number

- Cmd key 2 (which corresponds to indicator KB) cancels an order and:
  - Creates a code 0 (cancel order) record
  - Enables input to allow the operator to enter the next customer number

- Cmd key 3 (which corresponds to indicator KC) ends the program and updates the HEADER record to contain the group number of the last order written to the disk file TRANS.

The following error messages are displayed for the program: CUSTOMER NOT FOUND, ITEM NOT FOUND, and INVALID COMMAND KEY.

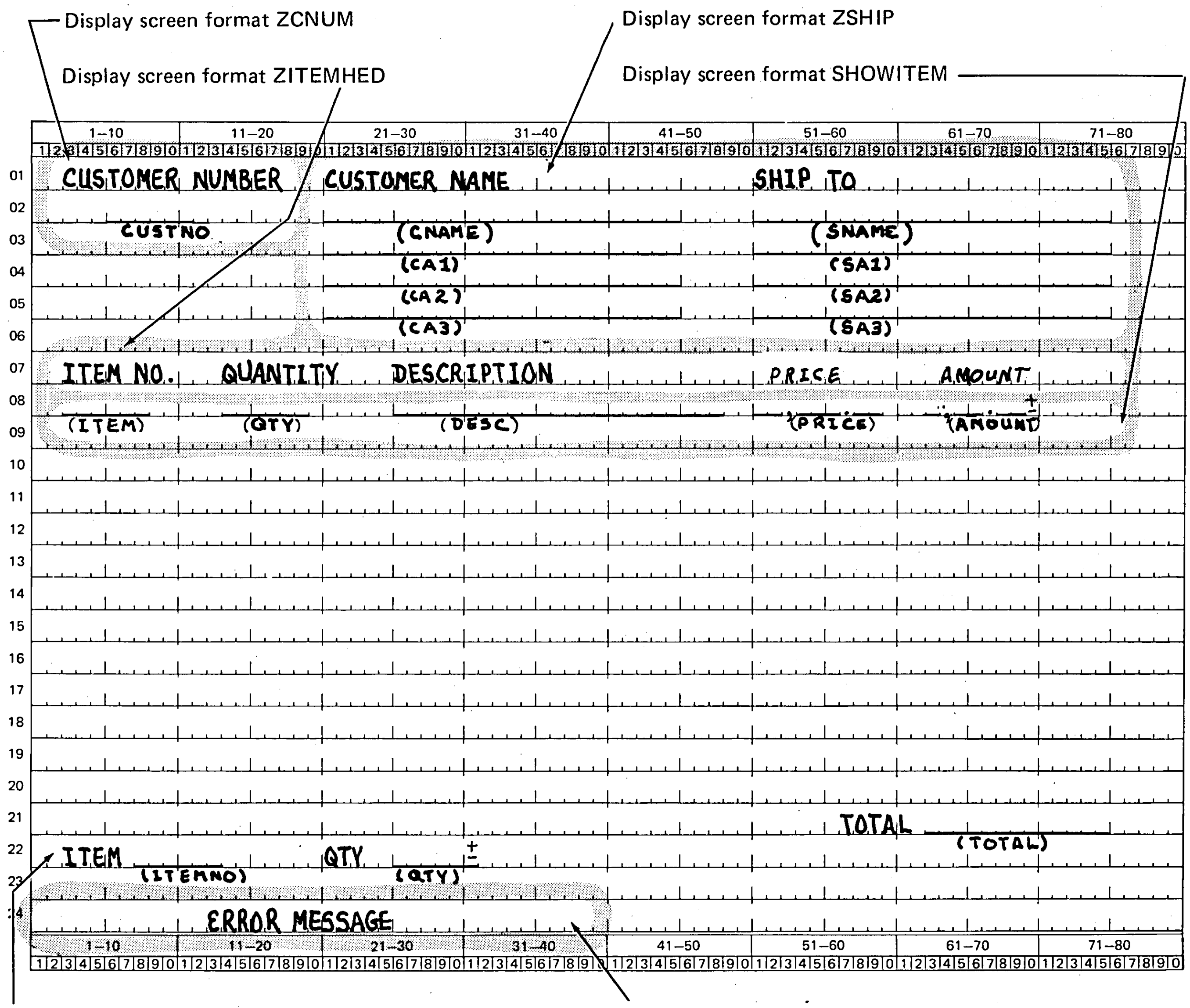
Functions Not Used in Sample Program ORD

The following functions were not used in the sample program ORD:

- ACQ operation code. The ACQ operation code is used only to cause a specific device to be allocated to tne program.

- NEXT operation code. The NEXT operation code is used to force the next input record to come from a specific device.

- READ operation code. The READ operation code is used to provide input from a WORKSTN file on demand. When you use the READ operation code, you must set on LR when the file is at end of file.



Display screen format ZCNUM

Display screen format ZITEMHED

Display screen format ZSHIP

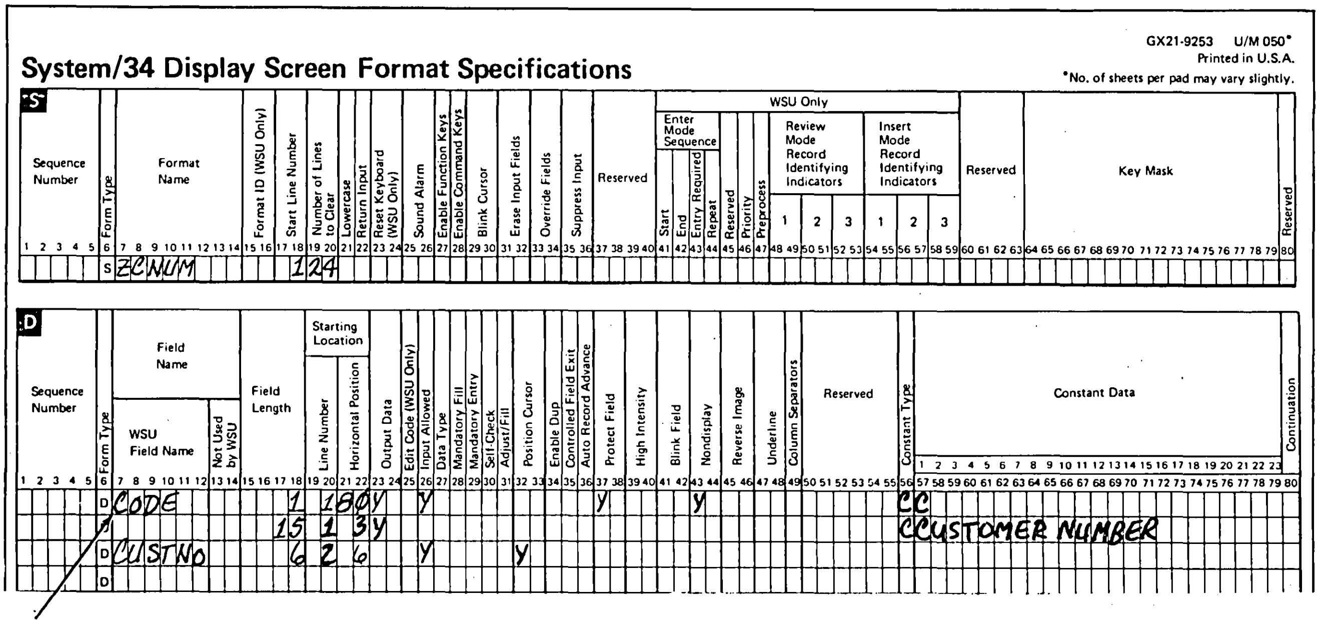
Display screen format SHOWITEM

The headings ITEM, QTY, and TOTAL are described by the display screen format ZITEMHED. The fields TOTAL, ITEMNO, and QTY are described by the display screen format ZITEM.

Display screen format ZERROR. The constant for the error message is defined on the RPG II output specifications (see Part 10 of this figure).

Display screen formats ZCNUM, ZSHIP, ZITEM, ZITEMHED, SHOWITEM, and ZERROR are contained in the format load member ORDFM.

Figure 19-14 (Part 1 of 11). Sample Program ORD



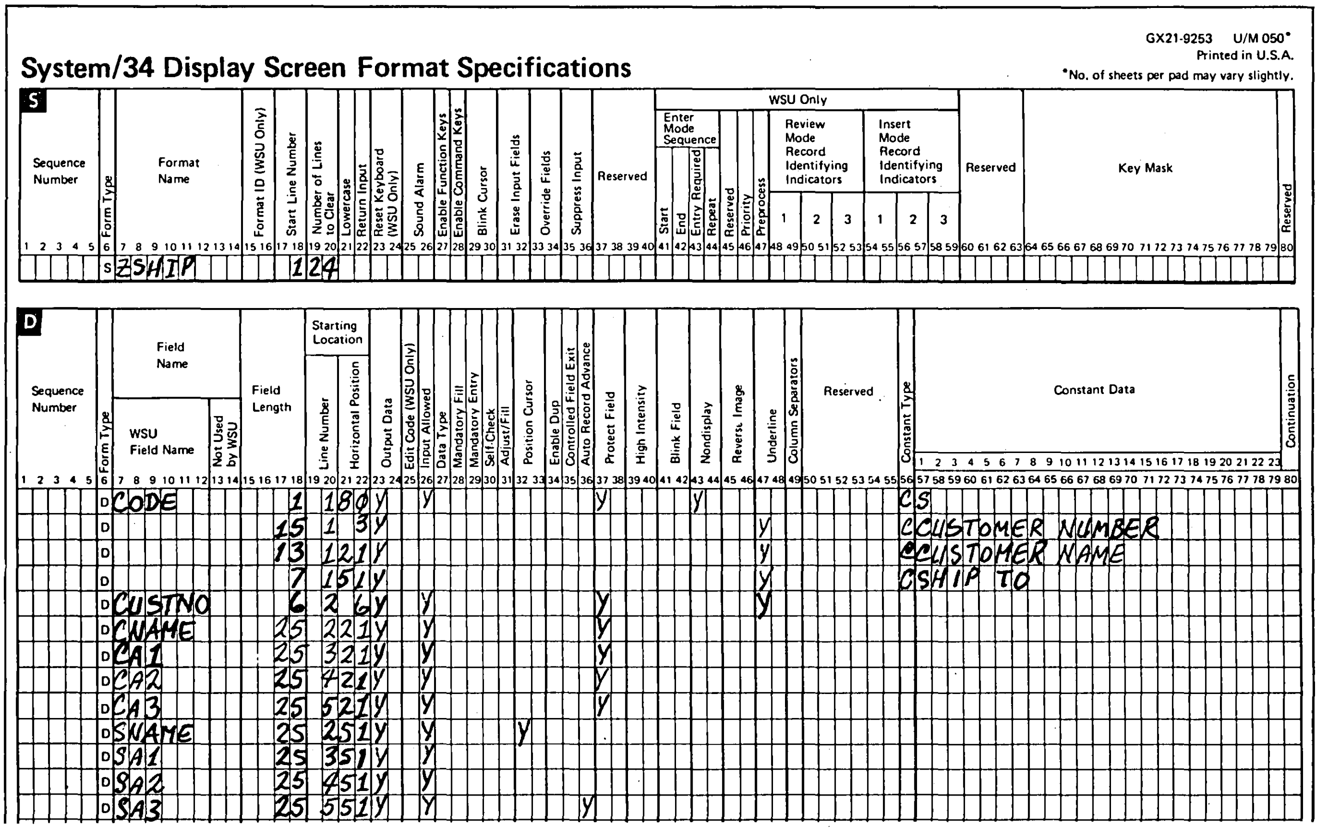
System/34 Display Screen Format Specifications

GX21-9253 U/M 050*
Printed in U.S.A.
*No. of sheets per pad may vary slightly.

Form Type (6)	Format Name (7-14)	Start Line Number (17-18)	Number of Lines to Clear (19-20)
S	ZCNUM	1	24

Form Type (6)	WSU Field Name (7-12)	Field Length (15-18)	Line Number (19-20)	Horizontal Position (21-22)	Output Data (23)	Input Allowed (26)	Position Cursor (32)	Protect Field (37)	Nondisplay (43)	Constant Type (56)	Constant Data (57-80)
D	CODE	1	1	80	Y	Y		Y	Y	C	C
D		15	1	3	Y					C	CUSTOMER NUMBER
D	CUSTNO	6	2	6		Y	Y				
D											

Code is the first input field in formats ZCNUM, ZSHIP, and ZITEM that allows input. This field is on the screen but is not displayed (Y in column 43). The field is used to determine the record type.



System/34 Display Screen Format Specifications

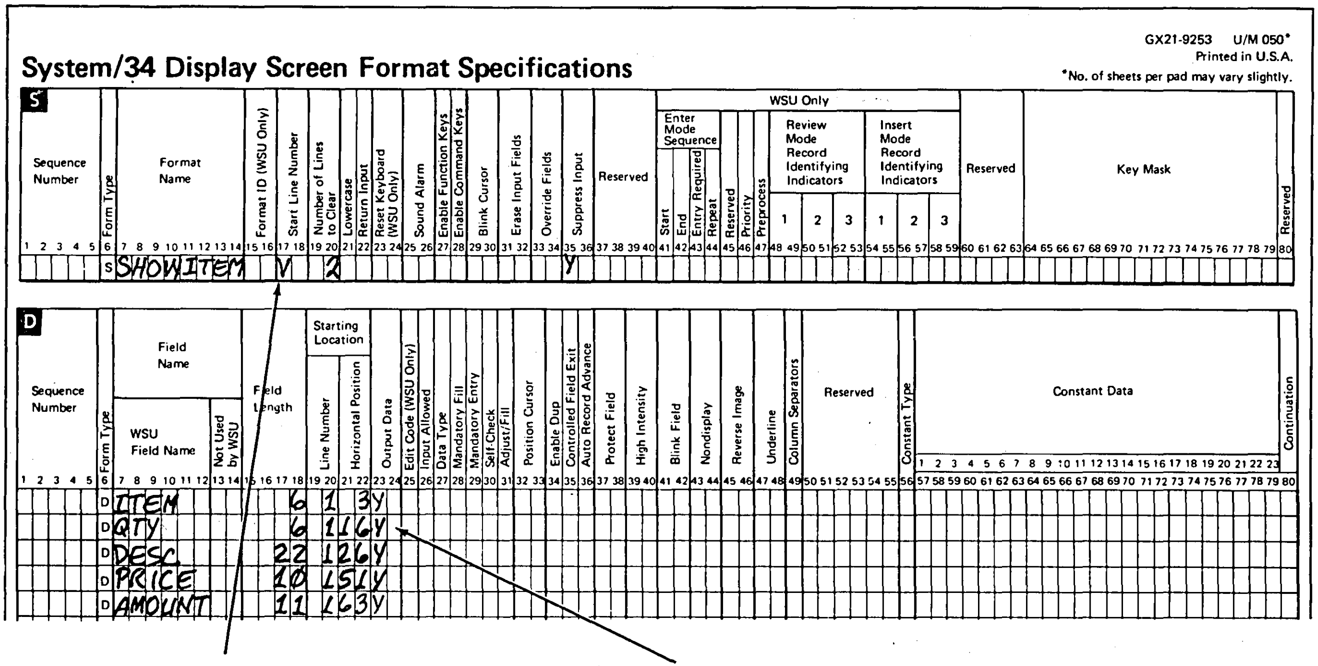
GX21-9253 U/M 050*
Printed in U.S.A.
*No. of sheets per pad may vary slightly.

Form Type (6)	Format Name (7-14)	Start Line Number (17-18)	Number of Lines to Clear (19-20)
S	ZSHIP	1	24

Form Type (6)	WSU Field Name (7-12)	Field Length (15-18)	Line Number (19-20)	Horizontal Position (21-22)	Output Data (23)	Input Allowed (26)	Position Cursor (32)	Auto Record Advance (36)	Protect Field (37)	Nondisplay (43)	Underline (47)	Constant Type (56)	Constant Data (57-80)
D	CODE	1	1	80	Y	Y			Y	Y		C	S
D		15	1	3	Y						Y	C	CUSTOMER NUMBER
D		13	12	1	Y						Y	C	CUSTOMER NAME
D		7	15	1	Y						Y	C	SHIP TO
D	CUSTNO	6	2	6	Y	Y			Y		Y		
D	CNAME	25	2	21	Y	Y			Y				
D	CA1	25	3	21	Y	Y			Y				
D	CA2	25	4	21	Y	Y			Y				
D	CA3	25	5	21	Y	Y			Y				
D	SNAME	25	2	51	Y	Y	Y						
D	SA1	25	3	51	Y	Y							
D	SA2	25	4	51	Y	Y							
D	SA3	25	5	51	Y	Y		Y					

The order in which these fields appear on the screen is not the same as the order in which they appear in the RPG II output record (see Part 11 of this figure).

Figure 19-14 (Part 2 of 11). Sample Program ORD



System/34 Display Screen Format Specifications

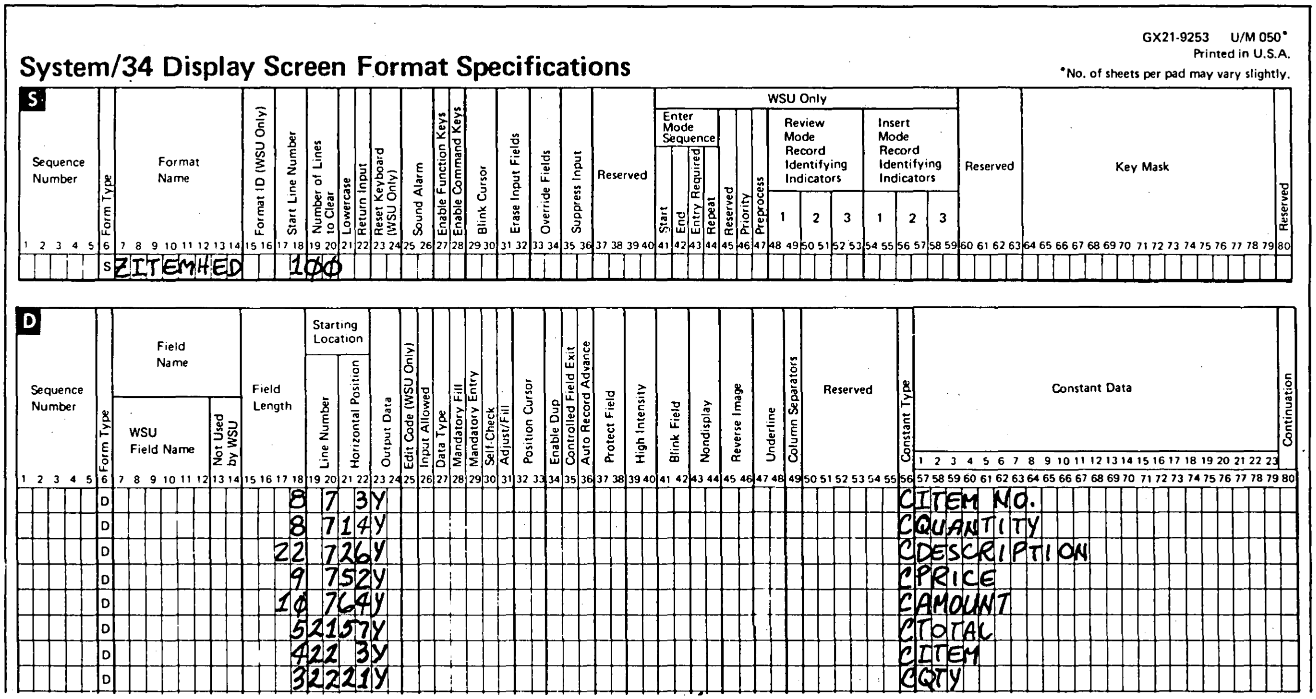
GX21-9253 U/M 050*
Printed in U.S.A.
*No. of sheets per pad may vary slightly.

Sequence Number (1-5)	Form Type (6)	Format Name (7-14)	Format ID (WSU Only) (15-16)	Start Line Number (17-18)	Number of Lines to Clear (19-20)	Suppress Input (35-36)
	S	SHOWITEM		V	2	Y

Sequence Number (1-5)	Form Type (6)	Field Name (7-14)	Field Length (15-18)	Line Number (19-20)	Horizontal Position (21-22)	Output Data (23)
	D	ITEM	6	1	3	Y
	D	QTY	6	1	16	Y
	D	DESC	22	1	26	Y
	D	PRICE	10	1	51	Y
	D	AMOUNT	11	1	63	Y

A variable starting line number is specified for SHOWITEM. The starting line number is contained in the field SLN, which is incremented in the ITEMS subroutine (see Part 9 of this figure).

The lengths of the fields QTY, PRICE, and AMOUNT on the display screen format specifications must be the lengths of the fields after they are edited by RPG II (see Part 10 of this figure).



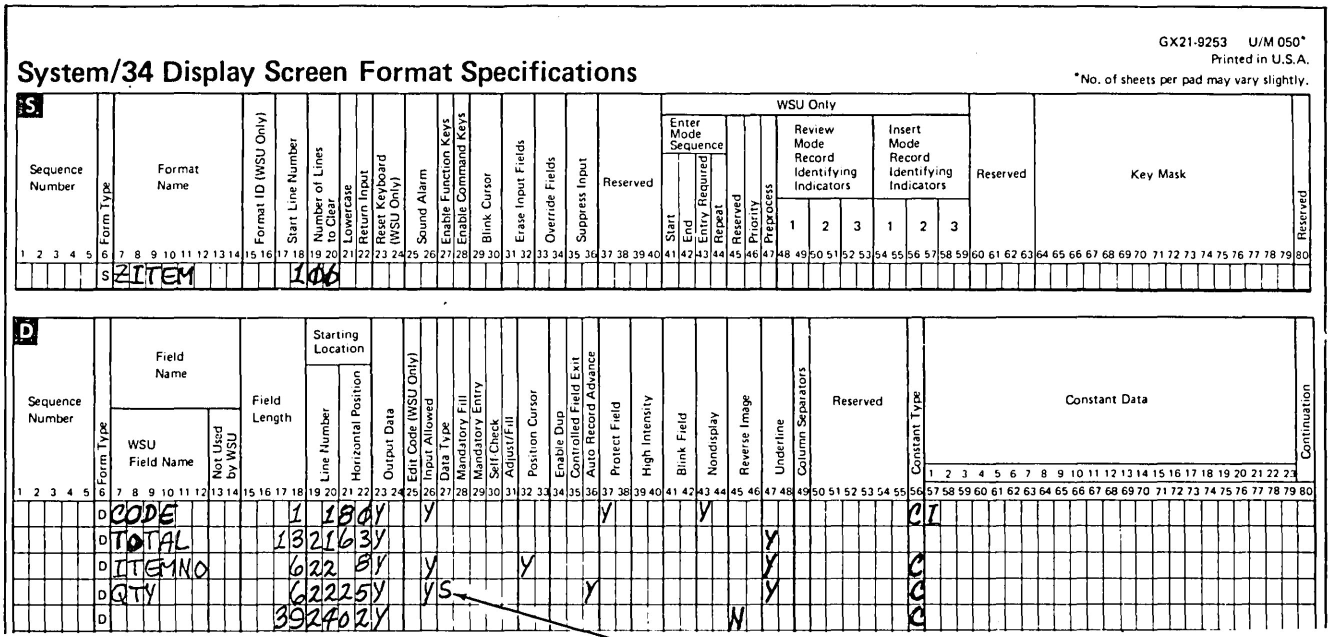
System/34 Display Screen Format Specifications

GX21-9253 U/M 050*
Printed in U.S.A.
*No. of sheets per pad may vary slightly.

Sequence Number (1-5)	Form Type (6)	Format Name (7-14)	Format ID (WSU Only) (15-16)	Start Line Number (17-18)	Number of Lines to Clear (19-20)
	S	ZITEMHED		1	ØØ

Sequence Number (1-5)	Form Type (6)	Field Name (7-14)	Field Length (15-18)	Line Number (19-20)	Horizontal Position (21-22)	Output Data (23)	Constant Type (56)	Constant Data (57-79)
	D		8	7	3	Y	C	ITEM NO.
	D		8	7	14	Y	C	QUANTITY
	D		22	7	26	Y	C	DESCRIPTION
	D		9	7	52	Y	C	PRICE
	D		1Ø	7	64	Y	C	AMOUNT
	D		5	21	57	Y	C	TOTAL
	D		4	22	3	Y	C	ITEM
	D		3	22	21	Y	C	QTY

Figure 19-14 (Part 3 of 11). Sample Program ORD



GX21-9253 U/M 050*
Printed in U.S.A.
*No. of sheets per pad may vary slightly.

System/34 Display Screen Format Specifications

S

Form Type (6)	Format Name (7–14)	Start Line Number (17–18)	Number of Lines to Clear (19–20)
S	ZITEM	1	ØØ

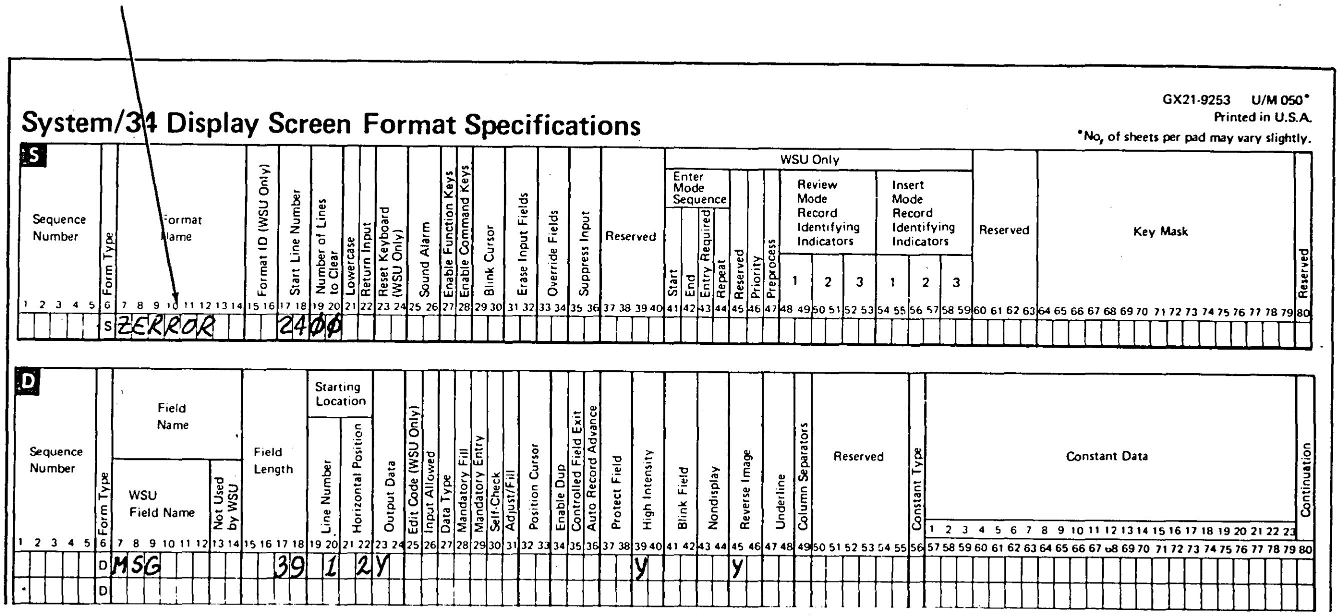
D

Form Type (6)	Field Name (7–14)	Field Length (15–18)	Line Number (19–20)	Horizontal Position (21–22)	Output Data (23)	Input Allowed (26)	Data Type (27)	Position Cursor (32)	Auto Record Advance (36)	Protect Field (37)	Nondisplay (43)	Reverse Image (45)	Underline (47)	Constant Type (56)	Constant Data (57–79)
D	CODE	1	1	8Ø	Y	Y				Y	Y			C	I
D	TOTAL	13	21	63	Y								Y		
D	ITEMNO	6	22	8	Y	Y		Y					Y	C	
D	QTY	6	22	25	Y	Y	S		Y				Y	C	
D		39	24	Ø2	Y							N		C	

Format ZITEM clears no lines (00 in columns 19 and 20 of the S specification). Because ZITEM clears no lines, it must output a blank field on line 24 to clear the field from the format ZERROR (see last line of ZITEM).

The length of the field QTY, which is defined as 5 in RPG II, for the display format is increased to 6 to allow for the sign position.

The format ZERROR does not clear any lines (00 in columns 19 and 20).



GX21-9253 U/M 050*
Printed in U.S.A.
*No. of sheets per pad may vary slightly.

System/34 Display Screen Format Specifications

S

Form Type (6)	Format Name (7–14)	Start Line Number (17–18)	Number of Lines to Clear (19–20)
S	ZERROR	24	ØØ

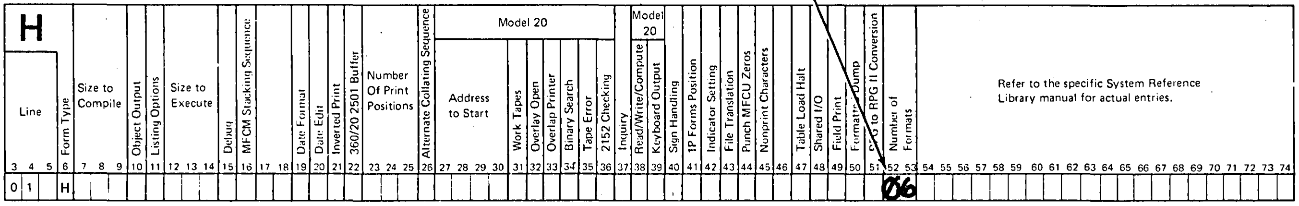
D

Form Type (6)	Field Name (7–14)	Field Length (15–18)	Line Number (19–20)	Horizontal Position (21–22)	Output Data (23)	High Intensity (39)	Reverse Image (45)
D	MSG	39	1	2	Y	Y	Y
D							

Figure 19-14 (Part 4 of 11). Sample Program ORD

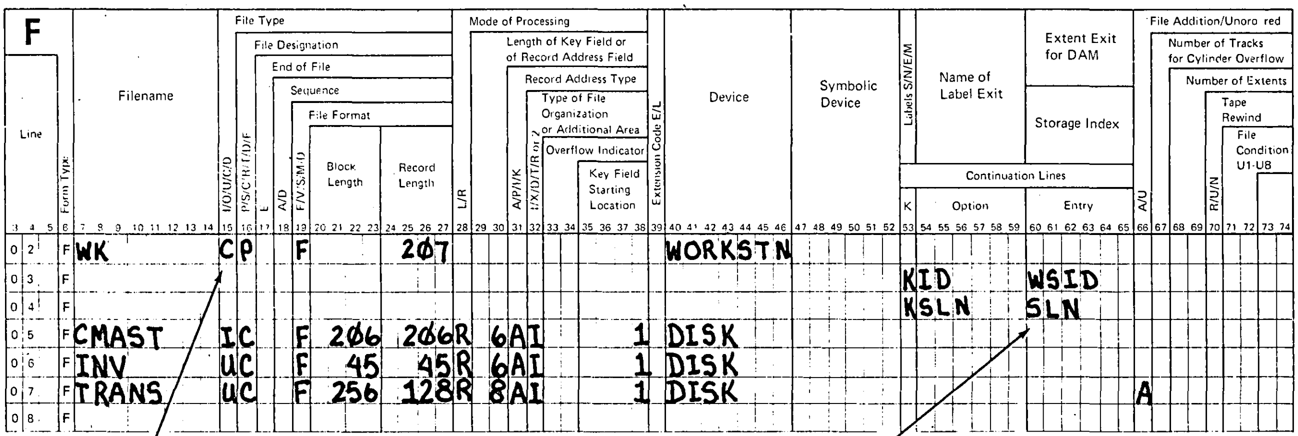
The format load member contains six display screen formats.

Control Specifications



Line	Form Type	Number of Formats (52-53)
01	H	06

File Description Specifications



Line	Form Type	Filename	File Type (15)	File Designation (16)	File Format (19)	Block Length	Record Length	L/R (28)	Length of Key Field	Record Address Type (31)	Type of File Organization (32)	Key Field Starting Location	Device	K (53)	Option	Entry	A/U (66)
02	F	WK	C	P	F		207						WORKSTN				
03	F													K	ID	WSID	
04	F													K	SLN	SLN	
05	F	CMAST	I	C	F	206	206	R	6	A	I	1	DISK				
06	F	INV	U	C	F	45	45	R	6	A	I	1	DISK				
07	F	TRANS	U	C	F	256	128	R	8	A	I	1	DISK				A
08	F																

The WORKSTN file is specified as combined (C in column 15) and primary (P in column 16).

The WSID field will contain the work station ID for the display station. The SLN field will contain the starting line number for the display screen format SHOWITEM because a variable starting line number (V in column 17 of the S specification) is specified.

Figure 19-14 (Part 5 of 11). Sample Program ORD

Indicator Summary

F ✱

Circle Indicators Used:

Note: All indicators are not valid with all systems.

General Indicators									
	[✓ 01]	02	03	04	05	06	07	08	09
[✓ 10]	[✓ 11]	[✓ 12]	[✓ 13]	14	[✓ 15]	16	17	18	19
20	21	22	23	24	25	26	27	28	29
30	31	32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47	48	49
50	51	52	53	54	55	56	57	58	59
60	61	62	63	64	65	66	67	68	69
70	71	72	73	74	75	76	77	78	79
80	81	82	83	84	85	86	87	88	89
[✓ 90]	[✓ 91]	[✓ 92]	93	94	95	[✓ 96]	[✓ 97]	[✓ 98]	[✓ 99]

Control Level Indicators	L1	L2	L3	L4	L5	L6	L7	L8	L9
Halt Indicators	H1	H2	H3	H4	H5	H6	H7	H8	H9
External Indicators	U1	U2	U3	U4	U5	U6	U7	U8	
Command Key Indicators	[✓ KA]	[✓ KB]	[✓ KC]	KD	KE	KF	KG	KH	KI
	KJ	KK	KL	KM	KN	KP	KQ	KR	KS
	KT	KU	KV	KW	KX	KY			
Overflow Indicators	OA	OB	OC	OD	OE	OF	OG	OV	
Special Purpose Indicators	1P	LO	HO	MR	LR				

Line (3-5)	Form Type (6-7)	Indicators: Record Identifying (8-10)	Input Field (11-13)	Calculation Result and Command Key (14-16)	Halt and User (17-19)	Control Level and Overflow (20-22)	(23-74)
01	F ✱	ID	F	C	H	L	FUNCTION OF INDICATORS
02	F ✱	NØ1					FIRST TIME FOR THIS WORKSTN
03	F ✱	1Ø					NEW GROUP, OUTPUT CUSTOMER NUMBER PROMPT
04	F ✱	11					CUSTOMER NUMBER INPUT, OUTPUT NAME & ADDRESS SCREEN
05	F ✱	12					NAME & ADDRESS IN, OUTPUT ITEM-QTY PROMPT
06	F ✱	13					ITEM-QTY INPUT, SHOW THE ITEM AND PROMPT FOR NEXT
07	F ✱						
08	F ✱	99					ERROR HAS OCCURRED
09	F ✱	96					INVALID COMMAND KEY
10	F ✱	97					ITEM NOT FOUND
11	F ✱	98					CUSTOMER NOT FOUND
12	F ✱						
13	F ✱	15					NEED TO OUTPUT IQPD ARRAY
14	F ✱						A) ARRAY IS FULL
15	F ✱						B) ORDER COMPLETE AND ARRAY IS NOT EMPTY
	F ✱	9Ø	91	92			WORK INDICATORS
	F ✱	KA					ORDER IS COMPLETE (INVALID IF STARTING NEW GROUP)
	F ✱	KB					CANCEL ORDER (INVALID IF STARTING NEW GROUP)
	F ✱	KC					END OF JOB (INVALID IF NOT STARTING GROUP)

Extension Specifications

E

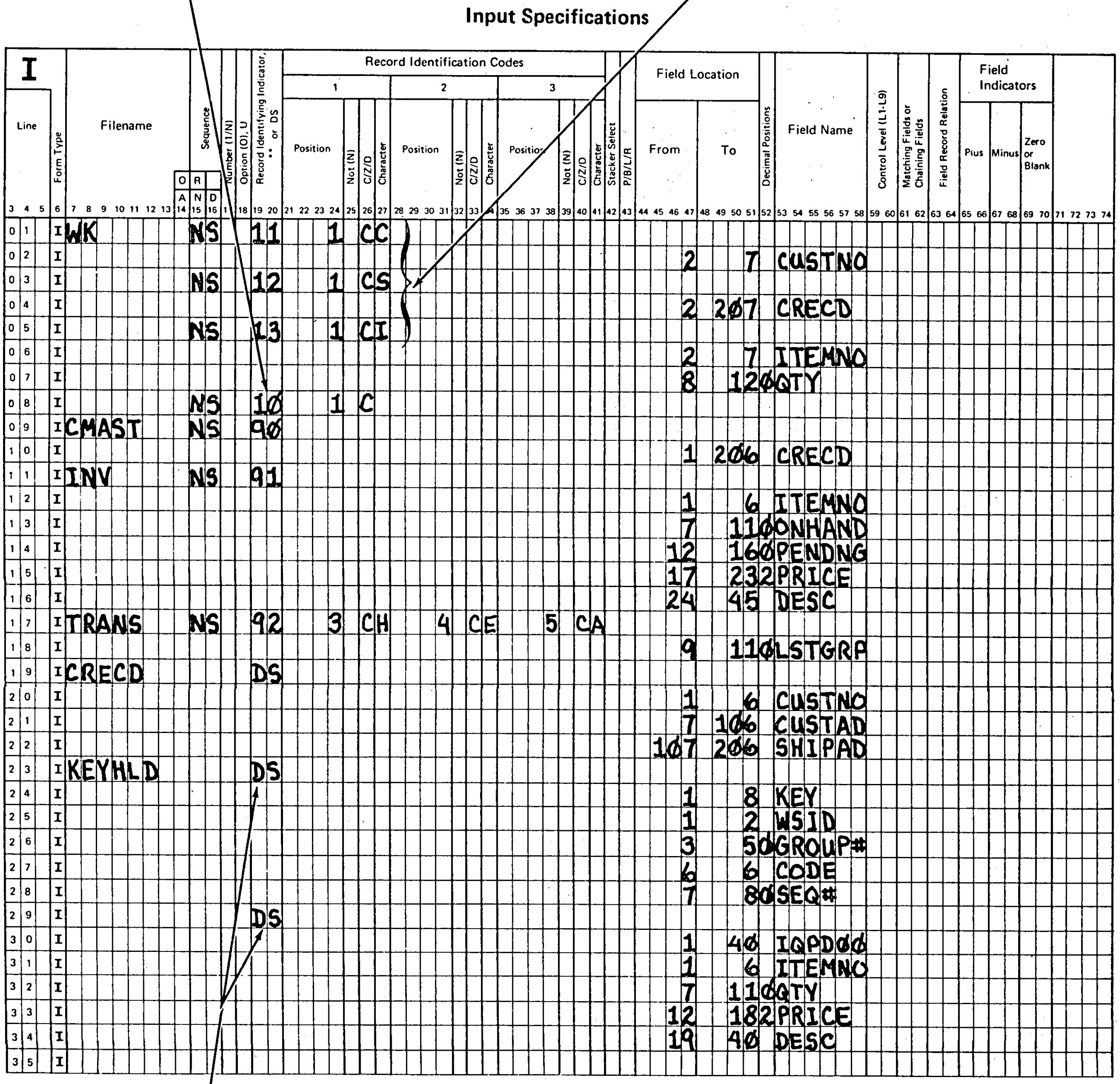
Line (3-5)	Form Type (6)	Record Sequence of the Chaining File (7-8)	Number of the Chaining Field (9-10)	From Filename (11-18)	To Filename (19-26)	Table or Array Name (27-32)	Number of Entries Per Record (33-35)	Number of Entries Per Table or Array (36-39)	Length of Entry (40-42)	P/B/L/R (43)	Decimal Positions (44)	Sequence (A/D) (45)	Table or Array Name (Alternating Format) (46-51)	Length of Entry (52-54)	P/B/L/R (55)	Decimal Positions (56)	Sequence (A/D) (57)	Comments (58-74)
01	E					IQPD		3	4Ø									
02	E																	

This array is used to output three ITEM, QTY, PRICE, and DESC entries.

Figure 19-14 (Part 6 of 11). Sample Program ORD

Indicator 10 identifies a blank record.

The record identification codes are output as constants in the formats and are used to control which set of calculations is executed.



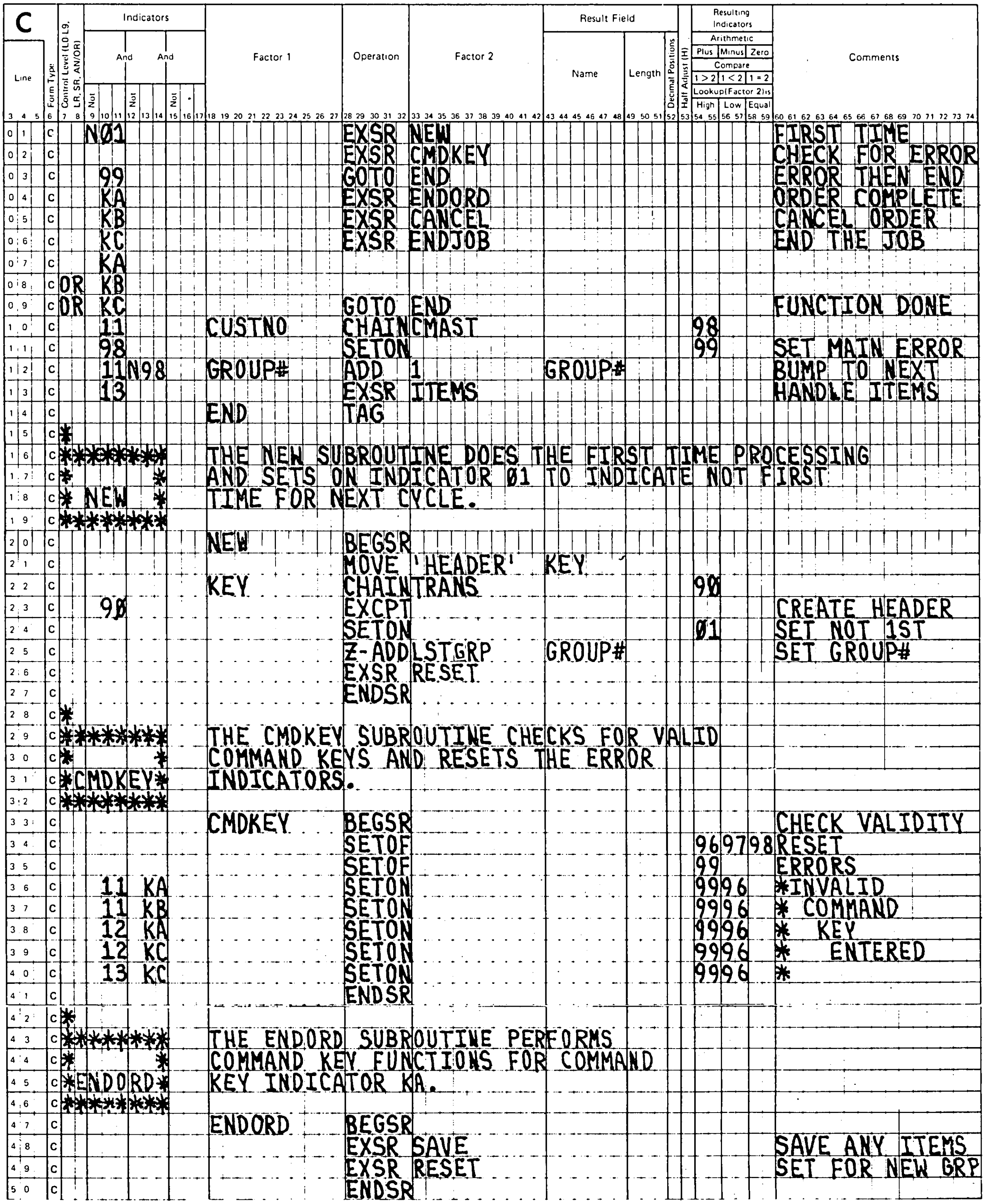
Input Specifications

I							Record Identification Codes														Field Location							Field Indicators			
							1				2				3																
Line	Form Type	Filename	Sequence OR AND	Number (1/N)	Option (O), U	Record Identifying Indicator, ** or DS	Position	Not (N)	C/Z/D	Character	Position	Not (N)	C/Z/D	Character	Position	Not (N)	C/Z/D	Character	Stacker Select	P/B/L/R	From	To	Decimal Positions	Field Name	Control Level (L1-L9)	Matching Fields or Chaining Fields	Field Record Relation	Plus	Minus	Zero or Blank	
3 4 5	6	7 8 9 10 11 12 13	14 15 16	17	18	19 20	21 22 23 24	25	26	27	28 29 30 31	32	33	34	35 36 37 38	39	40	41	42	43	44 45 46 47	48 49 50 51	52	53 54 55 56 57 58	59 60	61 62	63 64	65 66	67 68	69 70	71 72 73 74
01	I	WK	NS			11	1		C	C																					
02	I																				2	7		CUSTNO							
03	I		NS			12	1		C	S																					
04	I																				2	2Ø7		CRECD							
05	I		NS			13	1		C	I																					
06	I																				2	7		ITEMNO							
07	I																				8	12	Ø	QTY							
08	I		NS			1Ø	1		C																						
09	I	CMAST	NS			9Ø																									
10	I																				1	2Ø6		CRECD							
11	I	INV	NS			91																									
12	I																				1	6		ITEMNO							
13	I																				7	11	Ø	ONHAND							
14	I																				12	16	Ø	PENDNG							
15	I																				17	23	2	PRICE							
16	I																				24	45		DESC							
17	I	TRANS	NS			92	3		C	H	4		C	E	5		C	A													
18	I																				9	11	Ø	LSTGRP							
19	I	CRECD				DS																									
20	I																				1	6		CUSTNO							
21	I																				7	1Ø6		CUSTAD							
22	I																				1Ø7	2Ø6		SHIPAD							
23	I	KEYHLD				DS																									
24	I																				1	8		KEY							
25	I																				1	2		WSID							
26	I																				3	5	Ø	GROUP#							
27	I																				6	6		CODE							
28	I																				7	8	Ø	SEQ#							
29	I					DS																									
30	I																				1	4Ø		IQPDØØ							
31	I																				1	6		ITEMNO							
32	I																				7	11	Ø	QTY							
33	I																				12	18	2	PRICE							
34	I																				19	4Ø		DESC							
35	I																														

The data structure KEYHLD allows manipulation of the fields within the key for the TRANS file. The data structure containing IQPD00 is used to build an element for the array IQPD.

Figure 19-14 (Part 7 of 11). Sample Program ORD

Calculation Specifications

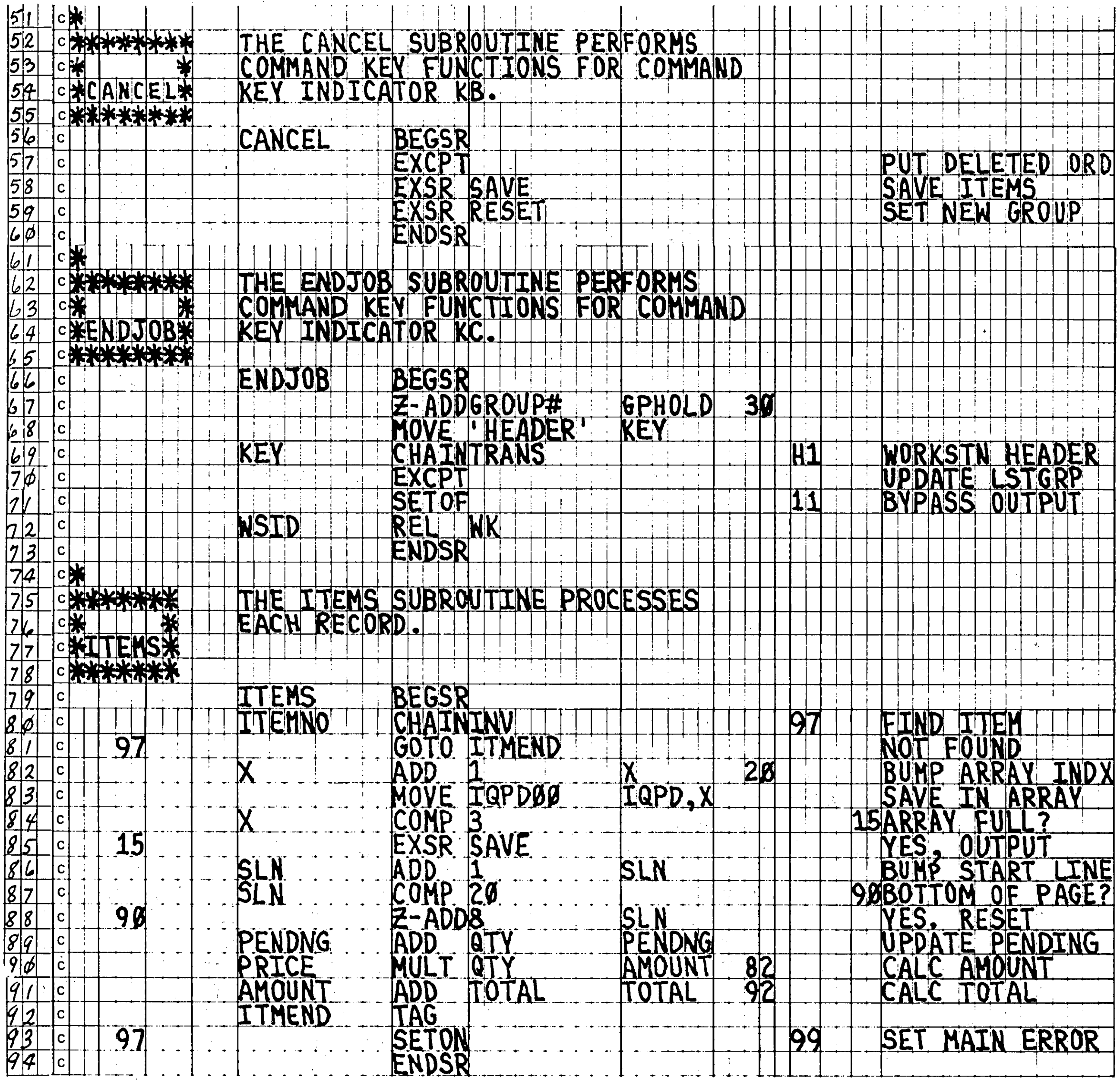


```
01 C  NØ1                EXSR NEW                        FIRST TIME
02 C                     EXSR CMDKEY                     CHECK FOR ERROR
03 C   99                GOTO END                        ERROR THEN END
04 C   KA                EXSR ENDORD                     ORDER COMPLETE
05 C   KB                EXSR CANCEL                     CANCEL ORDER
06 C   KC                EXSR ENDJOB                     END THE JOB
07 C   KA
08 COR KB
09 COR KC                GOTO END                        FUNCTION DONE
10 C   11      CUSTNO    CHAINCMAST                98
11 C   98                SETON                     99    SET MAIN ERROR
12 C   11N98   GROUP#    ADD  1         GROUP#           BUMP TO NEXT
13 C   13                EXSR ITEMS                      HANDLE ITEMS
14 C           END       TAG
15 C*
16 C***********THE NEW SUBROUTINE DOES THE FIRST TIME PROCESSING
17 C*         *AND SETS ON INDICATOR Ø1 TO INDICATE NOT FIRST
18 C*  NEW    *TIME FOR NEXT CYCLE.
19 C***********
20 C           NEW       BEGSR
21 C                     MOVE 'HEADER'  KEY
22 C           KEY       CHAINTRANS                9Ø
23 C   9Ø                EXCPT                           CREATE HEADER
24 C                     SETON                     Ø1    SET NOT 1ST
25 C                     Z-ADDLSTGRP    GROUP#           SET GROUP#
26 C                     EXSR RESET
27 C                     ENDSR
28 C*
29 C***********THE CMDKEY SUBROUTINE CHECKS FOR VALID
30 C*         *COMMAND KEYS AND RESETS THE ERROR
31 C*CMDKEY*   INDICATORS.
32 C***********
33 C           CMDKEY    BEGSR                           CHECK VALIDITY
34 C                     SETOF                     969798RESET
35 C                     SETOF                     99    ERRORS
36 C   11 KA             SETON                     9996  *INVALID
37 C   11 KB             SETON                     9996  * COMMAND
38 C   12 KA             SETON                     9996  *  KEY
39 C   12 KC             SETON                     9996  *    ENTERED
40 C   13 KC             SETON                     9996  *
41 C                     ENDSR
42 C*
43 C***********THE ENDORD SUBROUTINE PERFORMS
44 C*         *COMMAND KEY FUNCTIONS FOR COMMAND
45 C*ENDORD*   KEY INDICATOR KA.
46 C***********
47 C           ENDORD    BEGSR
48 C                     EXSR SAVE                       SAVE ANY ITEMS
49 C                     EXSR RESET                      SET FOR NEW GRP
50 C                     ENDSR
```

(Calculation specifications are continued on the next page.)

Figure 19-14 (Part 8 of 11). Sample Program ORD

(Calculation specifications continued from the previous page.)

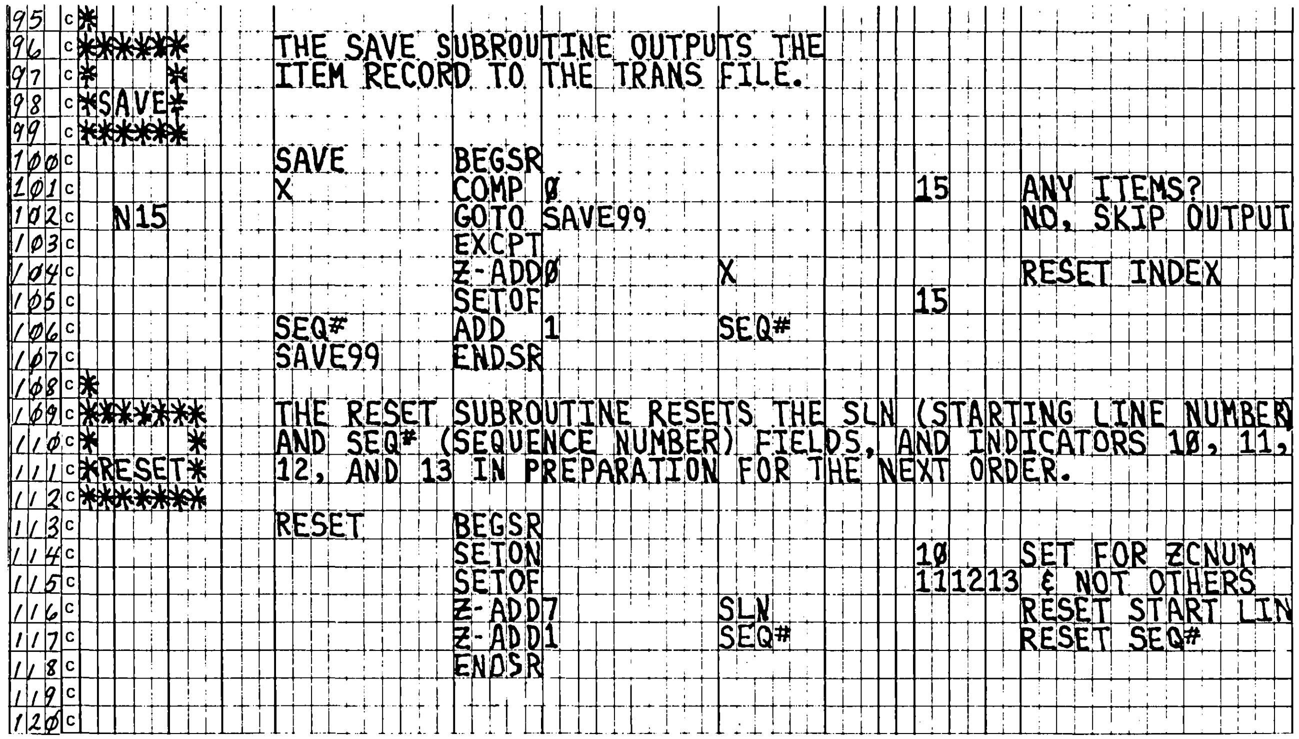


```
51 C*
52 C********   THE CANCEL SUBROUTINE PERFORMS
53 C*      *   COMMAND KEY FUNCTIONS FOR COMMAND
54 C*CANCEL*   KEY INDICATOR KB.
55 C********
56 C           CANCEL    BEGSR
57 C                     EXCPT                           PUT DELETED ORD
58 C                     EXSR SAVE                       SAVE ITEMS
59 C                     EXSR RESET                      SET NEW GROUP
6Ø C                     ENDSR
61 C*
62 C********   THE ENDJOB SUBROUTINE PERFORMS
63 C*      *   COMMAND KEY FUNCTIONS FOR COMMAND
64 C*ENDJOB*   KEY INDICATOR KC.
65 C********
66 C           ENDJOB    BEGSR
67 C                     Z-ADDGROUP#    GPHOLD  3Ø
68 C                     MOVE 'HEADER'  KEY
69 C           KEY       CHAINTRANS                H1    WORKSTN HEADER
7Ø C                     EXCPT                           UPDATE LSTGRP
71 C                     SETOF                     11    BYPASS OUTPUT
72 C           WSID      REL  WK
73 C                     ENDSR
74 C*
75 C*******    THE ITEMS SUBROUTINE PROCESSES
76 C*     *    EACH RECORD.
77 C*ITEMS*
78 C*******
79 C           ITEMS     BEGSR
8Ø C           ITEMNO    CHAININV                  97    FIND ITEM
81 C   97                GOTO ITMEND                     NOT FOUND
82 C           X         ADD  1         X       2Ø       BUMP ARRAY INDX
83 C                     MOVE IQPDØØ    IQPD,X           SAVE IN ARRAY
84 C           X         COMP 3                        15ARRAY FULL?
85 C   15                EXSR SAVE                       YES, OUTPUT
86 C           SLN       ADD  1         SLN              BUMP START LINE
87 C           SLN       COMP 2Ø                       9ØBOTTOM OF PAGE?
88 C   9Ø                Z-ADD8         SLN              YES, RESET
89 C           PENDNG    ADD  QTY       PENDNG           UPDATE PENDING
9Ø C           PRICE     MULT QTY       AMOUNT  82       CALC AMOUNT
91 C           AMOUNT    ADD  TOTAL     TOTAL   92       CALC TOTAL
92 C           ITMEND    TAG
93 C   97                SETON                     99    SET MAIN ERROR
94 C                     ENDSR
```

(Calculation specifications are continued on the next page.)

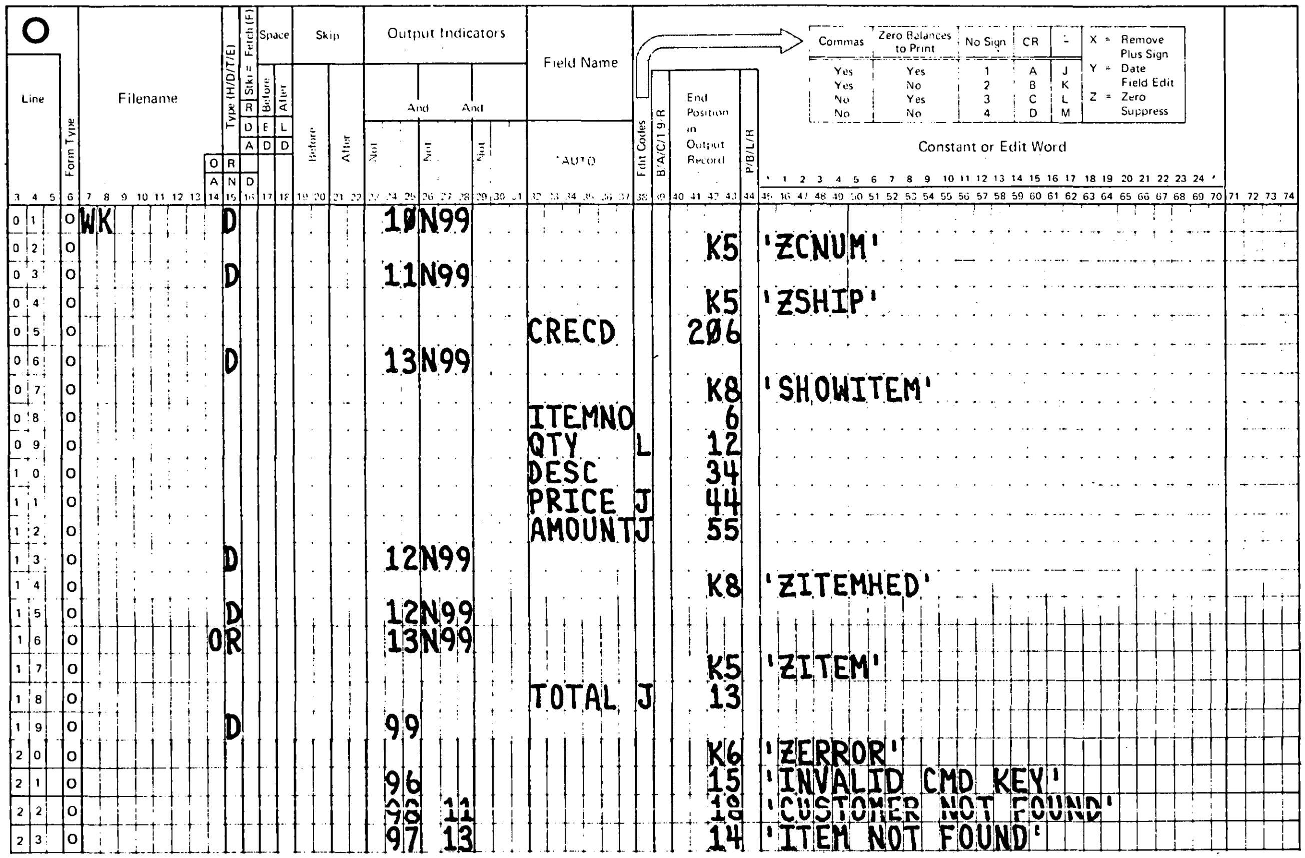
Figure 19-14 (Part 9 of 11). Sample Program ORD

(Calculation specifications continued from the previous page.)



```
   95C*
   96C******     THE SAVE SUBROUTINE OUTPUTS THE
   97C*    *     ITEM RECORD TO THE TRANS FILE.
   98C*SAVE*
   99C******
  100C           SAVE      BEGSR
  101C           X         COMP Ø                    15    ANY ITEMS?
  102C  N15                GOTO SAVE99                     NO, SKIP OUTPUT
  103C                     EXCPT
  104C                     Z-ADDØ         X                RESET INDEX
  105C                     SETOF                         15
  106C           SEQ#      ADD  1         SEQ#
  107C           SAVE99    ENDSR
  108C*
  109C*******    THE RESET SUBROUTINE RESETS THE SLN (STARTING LINE NUMBER)
  110C*     *    AND SEQ# (SEQUENCE NUMBER) FIELDS, AND INDICATORS 1Ø, 11,
  111C*RESET*    12, AND 13 IN PREPARATION FOR THE NEXT ORDER.
  112C*******
  113C           RESET     BEGSR
  114C                     SETON                         1Ø    SET FOR ZCNUM
  115C                     SETOF                         111213 & NOT OTHERS
  116C                     Z-ADD7         SLN              RESET START LIN
  117C                     Z-ADD1         SEQ#             RESET SEQ#
  118C                     ENDSR
  119C
  120C
```

Output Specifications



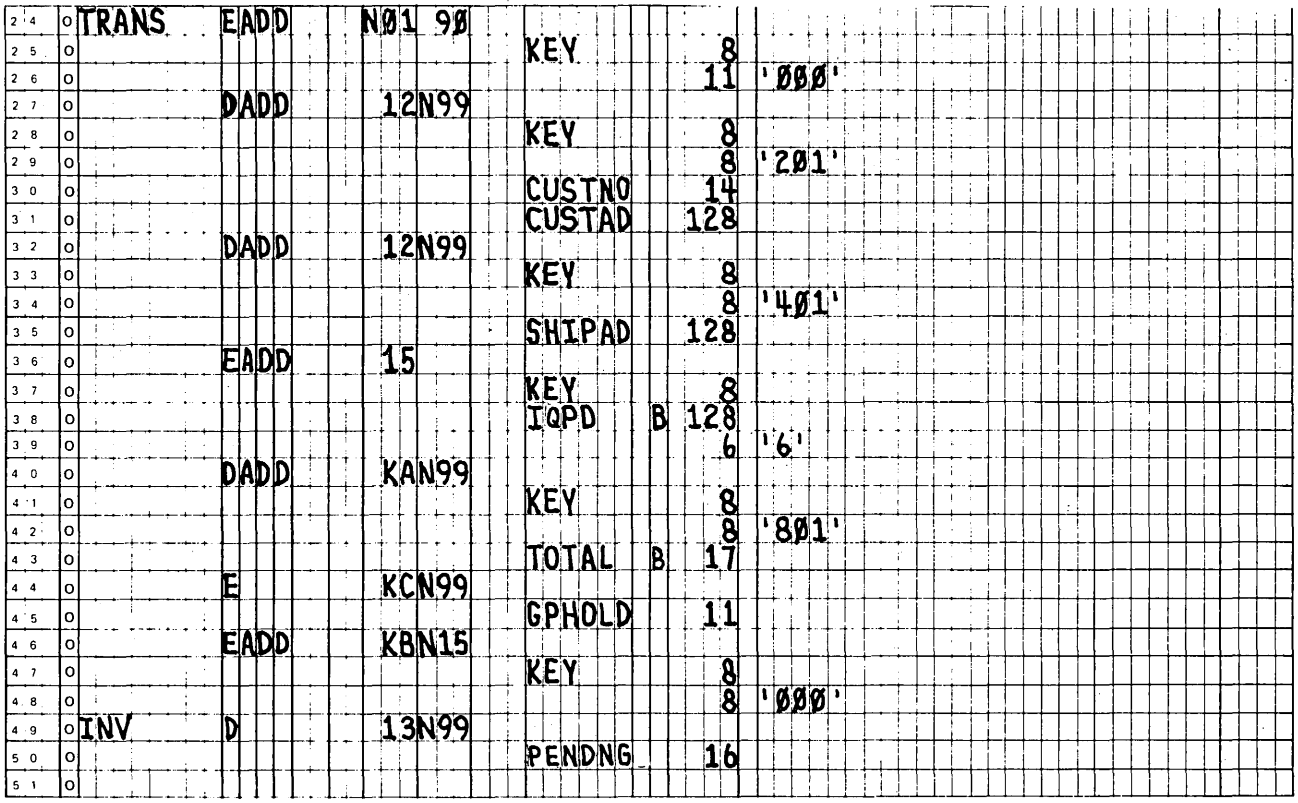
O
Line
Form Type
Filename
Type (H/D/T/E)
Fetch (F)
Stkr #
Space
Before
After
Skip
Before
After
Output Indicators
And
And
Not
Not
Not
O R
A N D
A D D
D E L
Field Name
*AUTO
Edit Codes
B/A/C/1-9-R
End Position in Output Record
P/B/L/R
Constant or Edit Word
Commas
Zero Balances to Print
No Sign
CR
-
Yes Yes 1 A J
Yes No 2 B K
No Yes 3 C L
No No 4 D M
X = Remove Plus Sign
Y = Date Field Edit
Z = Zero Suppress

```
  01 OWK      D        1ØN99
  02 O                                   K5 'ZCNUM'
  03 O        D        11N99
  04 O                                   K5 'ZSHIP'
  05 O                         CRECD    2Ø6
  06 O        D        13N99
  07 O                                   K8 'SHOWITEM'
  08 O                         ITEMNO     6
  09 O                         QTY   L   12
  10 O                         DESC      34
  11 O                         PRICE J   44
  12 O                         AMOUNTJ   55
  13 O        D        12N99
  14 O                                   K8 'ZITEMHED'
  15 O        D        12N99
  16 O       OR        13N99
  17 O                                   K5 'ZITEM'
  18 O                         TOTAL J   13
  19 O        D        99
  20 O                                   K6 'ZERROR'
  21 O                 96                15 'INVALID CMD KEY'
  22 O                 98 11             18 'CUSTOMER NOT FOUND'
  23 O                 97 13             14 'ITEM NOT FOUND'
```

(Output specifications are continued on the next page.)

Figure 19-14 (Part 10 of 11). Sample Program ORD

(Output specifications continued from the previous page.)



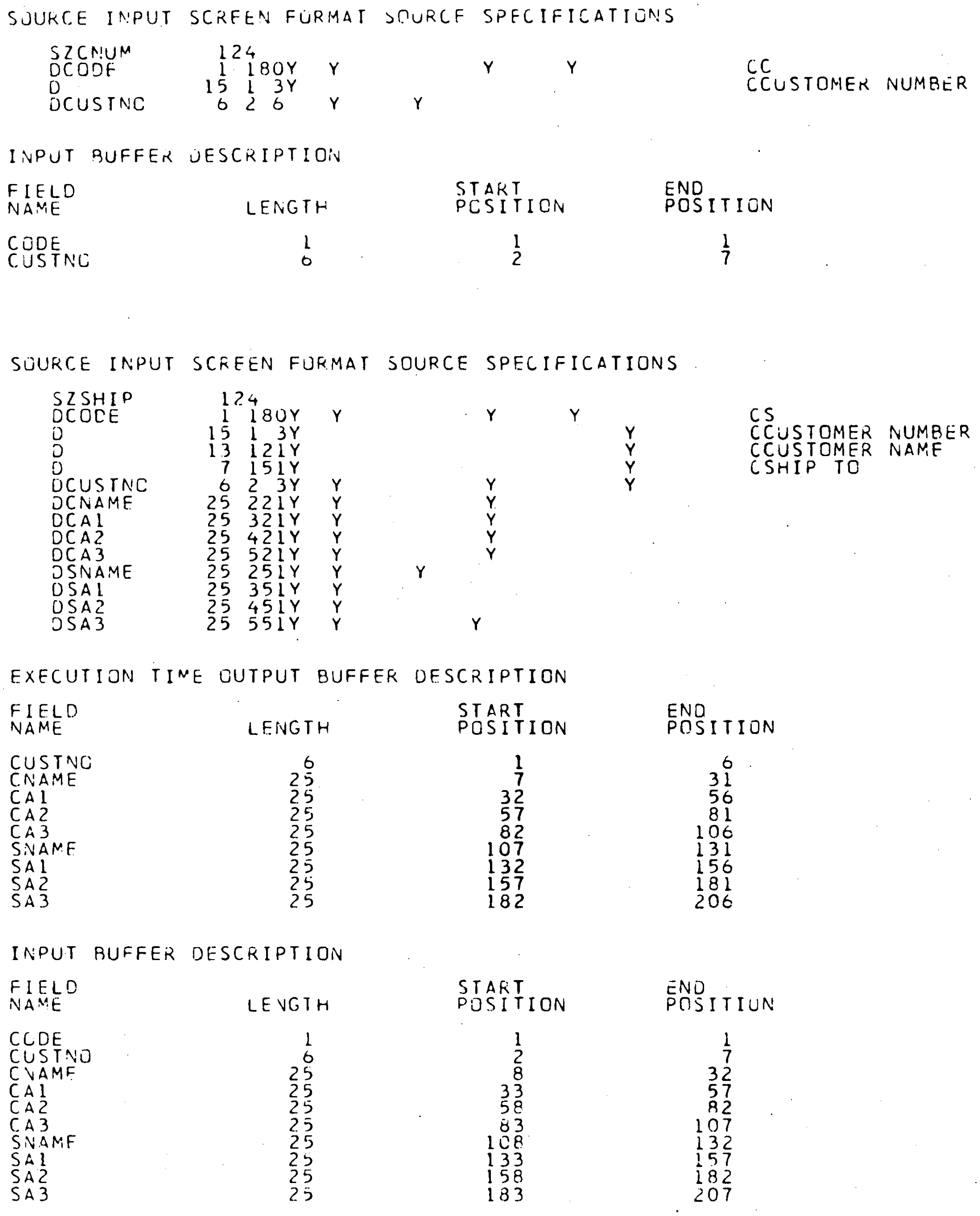
```
24 OTRANS   EADD    NØ1 9Ø
25 O                         KEY        8
26 O                                   11 'ØØØ'
27 O        DADD     12N99
28 O                         KEY        8
29 O                                    8 '2Ø1'
30 O                         CUSTNO    14
31 O                         CUSTAD   128
32 O        DADD     12N99
33 O                         KEY        8
34 O                                    8 '4Ø1'
35 O                         SHIPAD   128
36 O        EADD     15
37 O                         KEY        8
38 O                         IQPD   B 128
39 O                                    6 '6'
40 O        DADD     KAN99
41 O                         KEY        8
42 O                                    8 '8Ø1'
43 O                         TOTAL  B  17
44 O        E        KCN99
45 O                         GPHOLD    11
46 O        EADD     KBN15
47 O                         KEY        8
48 O                                    8 'ØØØ'
49 OINV     D        13N99
50 O                         PENDNG    16
51 O
```

Records output to the WORKSTN file are conditioned on a successful (N99) input of the previous record type:

- Customer number as first screen
- Customer number precedes ship to
- Ship to or item precedes an item
- Item precedes displaying the item

The WORKSTN error record (ITEM NOT FOUND) is separate from the preceding order. However, because the ZERROR format allows input, the correction can be made as if no error had occurred.

Figure 19-14 (Part 11 of 11). Sample Program ORD

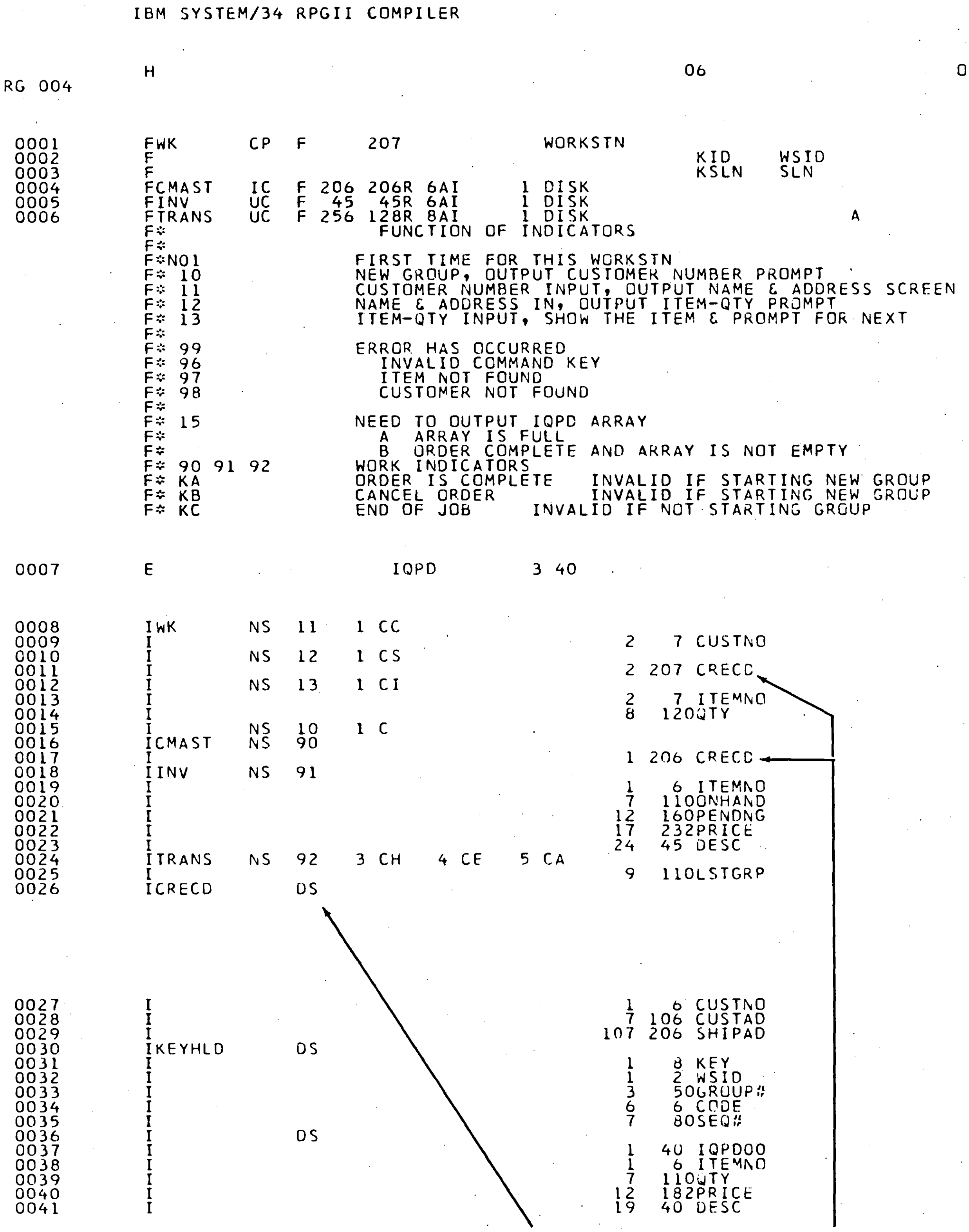


```
SOURCE INPUT SCREEN FORMAT SOURCE SPECIFICATIONS

   SZCNUM         124
   DCODF           1 180Y  Y          Y     Y            CC
   D              15 1 3Y                                CCUSTOMER NUMBER
   DCUSTNC         6 2 6   Y     Y

INPUT BUFFER DESCRIPTION

FIELD                           START          END
NAME             LENGTH         POSITION       POSITION

CODE                 1              1              1
CUSTNO               6              2              7



SOURCE INPUT SCREEN FORMAT SOURCE SPECIFICATIONS

   SZSHIP         124
   DCODE           1 180Y  Y          Y     Y            CS
   D              15 1 3Y                       Y        CCUSTOMER NUMBER
   D              13 121Y                       Y        CCUSTOMER NAME
   D               7 151Y                       Y        CSHIP TO
   DCUSTNO         6 2 3Y  Y          Y         Y
   DCNAME         25 221Y  Y          Y
   DCA1           25 321Y  Y          Y
   DCA2           25 421Y  Y          Y
   DCA3           25 521Y  Y          Y
   DSNAME         25 251Y  Y     Y
   DSA1           25 351Y  Y
   DSA2           25 451Y  Y
   DSA3           25 551Y  Y         Y

EXECUTION TIME OUTPUT BUFFER DESCRIPTION

FIELD                           START          END
NAME             LENGTH         POSITION       POSITION

CUSTNO               6              1              6
CNAME               25              7             31
CA1                 25             32             56
CA2                 25             57             81
CA3                 25             82            106
SNAMF               25            107            131
SA1                 25            132            156
SA2                 25            157            181
SA3                 25            182            206

INPUT BUFFER DESCRIPTION

FIELD                           START          END
NAME             LENGTH         POSITION       POSITION

CODE                 1              1              1
CUSTNO               6              2              7
CNAMF               25              8             32
CA1                 25             33             57
CA2                 25             58             82
CA3                 25             83            107
SNAMF               25            108            132
SA1                 25            133            157
SA2                 25            158            182
SA3                 25            183            207
```

Figure 19-15 (Part 1 of 6). Compiler Listing for Sample Program ORD

```
SOURCE INPUT SCREEN FORMAT SOURCE SPECIFICATIONS

   SSHOwITEM  V  2
   DITEM       6 1 3Y
   DQTY        6 116Y
   DDESC      22 126Y
   DPRICE     10 151Y
   DAMOUNT    11 163Y

EXECUTION TIME OUTPUT BUFFER DESCRIPTION

FIELD                            START          END
NAME             LENGTH          POSITION       POSITION

ITEM                 6               1              6
QTY                  6               7             12
DESC                22              13             34
PRICE               10              35             44
AMOUNT              11              45             55


SOURCE INPUT SCREEN FORMAT SOURCE SPECIFICATIONS

   SZITEMHED   100
   D           8 7 3Y                                  CITEM NO.
   D           8 714Y                                  CQUANTITY
   D          22 726Y                                  CDESCRIPTION
   D           9 752Y                                  CPRICE
   D          10 764Y                                  CAMOUNT
   D           52157Y                                  CTOTAL
   D           422 3Y                                  CITEM
   D           32221Y                                  CQTY


SOURCE INPUT SCREEN FORMAT SOURCE SPECIFICATIONS

   SZITEM     100
   DCODE       1 180Y   Y          Y     Y             CI
   DTOTAL     132163Y                        Y
   DITEMNO     622 8Y   Y     Y              Y         C
   DQTY        62225Y   YS        Y          Y         C
   D          392402Y                      N           C

EXECUTION TIME OUTPUT BUFFER DESCRIPTION

FIELD                            START          END
NAME             LENGTH          POSITION       POSITION

TOTAL               13               1             13

INPUT BUFFER DESCRIPTION

FIELD                            START          END
NAME             LENGTH          POSITION       POSITION

CODE                 1               1              1
ITEMNO               6               2              7
QTY                  5               8             12


SOURCE INPUT SCREEN FORMAT SOURCE SPECIFICATIONS

   SZERROR    2400
   DMSG       39 1 2Y                   Y     Y

EXECUTION TIME OUTPUT BUFFER DESCRIPTION

FIELD                            START          END
NAME             LENGTH          POSITION       POSITION

MSG                 39               1             39


 ORDFM      SCREEN FORMAT LOAD MEMBER

FORMAT ZCNUM     REQUIRES    512 BYTES OF STORAGE
FORMAT ZSHIP     REQUIRES    768 BYTES OF STORAGE
FORMAT SHOWITEM  REQUIRES    512 BYTES OF STORAGE
FORMAT ZITEMHED  REQUIRES    512 BYTES OF STORAGE
FORMAT ZITEM     REQUIRES    512 BYTES OF STORAGE
FORMAT ZERROR    REQUIRES    512 BYTES OF STORAGE
```

Figure 19-15 (Part 2 of 6). Compiler Listing for Sample Program ORD



```
           IBM SYSTEM/34 RPGII COMPILER

           H                                             06                      ORD
RG 004

0001       FWK      CP  F     207           WORKSTN
0002       F                                             KID    WSID
0003       F                                             KSLN   SLN
0004       FCMAST   IC  F 206 206R 6AI      1 DISK
0005       FINV     UC  F  45  45R 6AI      1 DISK
0006       FTRANS   UC  F 256 128R 8AI      1 DISK                   A
           F*                 FUNCTION OF INDICATORS
           F*
           F*NO1            FIRST TIME FOR THIS WORKSTN
           F* 10            NEW GROUP, OUTPUT CUSTOMER NUMBER PROMPT
           F* 11            CUSTOMER NUMBER INPUT, OUTPUT NAME & ADDRESS SCREEN
           F* 12            NAME & ADDRESS IN, OUTPUT ITEM-QTY PROMPT
           F* 13            ITEM-QTY INPUT, SHOW THE ITEM & PROMPT FOR NEXT
           F*
           F* 99            ERROR HAS OCCURRED
           F* 96              INVALID COMMAND KEY
           F* 97              ITEM NOT FOUND
           F* 98              CUSTOMER NOT FOUND
           F*
           F* 15            NEED TO OUTPUT IQPD ARRAY
           F*                 A  ARRAY IS FULL
           F*                 B  ORDER COMPLETE AND ARRAY IS NOT EMPTY
           F* 90 91 92      WORK INDICATORS
           F* KA            ORDER IS COMPLETE    INVALID IF STARTING NEW GROUP
           F* KB            CANCEL ORDER         INVALID IF STARTING NEW GROUP
           F* KC            END OF JOB      INVALID IF NOT STARTING GROUP


0007       E                     IQPD        3 40


0008       IWK      NS  11   1 CC
0009       I                                        2    7 CUSTNO
0010       I        NS  12   1 CS
0011       I                                        2  207 CRECD
0012       I        NS  13   1 CI
0013       I                                        2    7 ITEMNO
0014       I                                        8   120QTY
0015       I        NS  10   1 C
0016       ICMAST   NS  90
0017       I                                        1  206 CRECD
0018       IINV     NS  91
0019       I                                        1    6 ITEMNO
0020       I                                        7   110ONHAND
0021       I                                       12   160PENDNG
0022       I                                       17   232PRICE
0023       I                                       24   45 DESC
0024       ITRANS   NS  92   3 CH    4 CE    5 CA
0025       I                                        9   110LSTGRP
0026       ICRECD       DS



0027       I                                        1    6 CUSTNO
0028       I                                        7  106 CUSTAD
0029       I                                      107  206 SHIPAD
0030       IKEYHLD      DS
0031       I                                        1    8 KEY
0032       I                                        1    2 WSID
0033       I                                        3   50GROUP#
0034       I                                        6    6 CODE
0035       I                                        7   80SEQ#
0036       I            DS
0037       I                                        1   40 IQPD00
0038       I                                        1    6 ITEMNO
0039       I                                        7   110QTY
0040       I                                       12   182PRICE
0041       I                                       19   40 DESC
```

The data structure CRECD is used to define input records and an output record (see Part 5 of this figure). The data structure for the output record is defined further by the ZSHIP display screen format. See Part 1 of this figure for the relationship between the display screen format and the input and output records.

Figure 19-15 (Part 3 of 6). Compiler Listing for Sample Program ORD

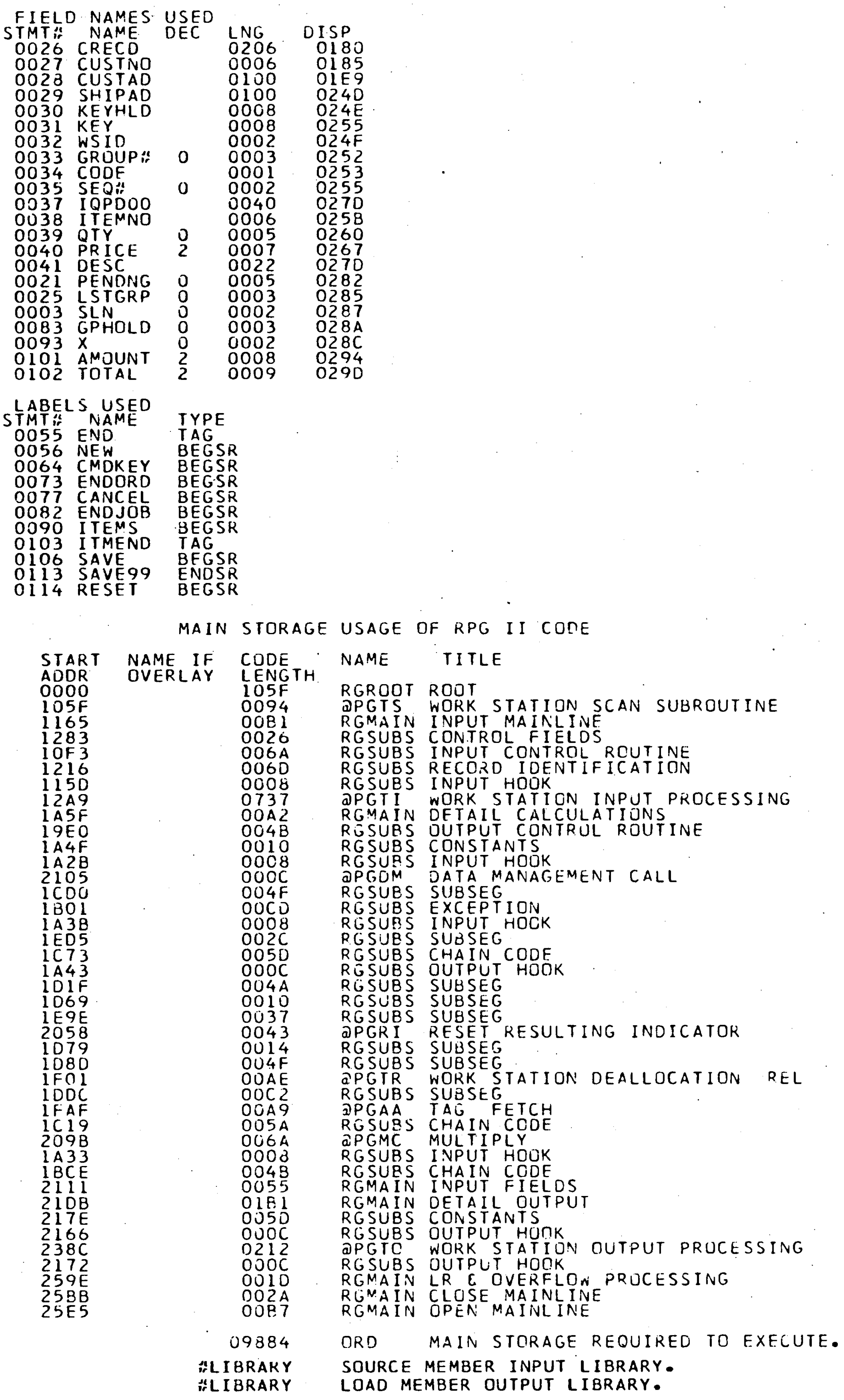
```
0042       C  N01                EXSR NEW
0043       C                     EXSR CMDKEY
0044       C   99                GOTO END
0045       C   KA                EXSR ENDORD
0046       C   KB                EXSR CANCEL
0047       C   KC                EXSR ENDJOB
0048       C   KA
0049       COR KB
0050       COR KC                GOTO END
0051       C   11      CUSTNO    CHAINCMAST                98
0052       C   98                SETON                     99
0053       C   11N98   GROUP#    ADD  1         GROUP#
0054       C   13                EXSR ITEMS
0055       C           END       TAG
           C*
0056       C           NEW       BEGSR
0057       C                     MOVE 'HEADER'  KEY
0058       C           KEY       CHAINTRANS                90
0059       C   90                EXCPT                           CREATE HEADER
0060       C                     SETON                     01    SET NCT 1ST
0061       C                     Z-ADDLSTGRP    GROUP#           SET GROUP#
0062       C                     EXSR RESET                      START NEW GROUP
0063       C                     ENDSR
0064       C           CMDKEY    PEGSR                           CHECK VALIDITY
0065       C                     SETOF                     969798RESET
0066       C                     SETOF                     99    ERRORS
0067       C   11 KA             SETON                     9996  *INVALID
0068       C   11 KB             SETON                     9996  * COMMAND
0069       C   12 KA             SETON                     9996  *  KEY
0070       C   12 KC             SETON                     9996  *   ENTERED
0071       C   13 KC             SETON                     9996  *
0072       C                     ENDSR
           C*
0073       C           ENDORD    BEGSR
0074       C                     EXSR SAVE
0075       C                     EXSR RESET
0076       C                     ENDSR
           C*
0077       C           CANCEL    PEGSR
0078       C                     EXCPT
0079       C                     EXSR SAVE
0080       C                     EXSR RESET
0081       C                     ENDSR
```

```
           C*
0082       C           ENDJOB    BEGSR
0083       C                     Z-ADDGROUP#    GPHOLD  30
0084       C                     MOVE 'HEADER'  KEY
0085       C           KEY       CHAINTRANS                H1
0086       C                     EXCPT
0087       C                     SETOF                     11
0088       C           WSID      REL  wK
0089       C                     ENDSR
           C*
0090       C           ITEMS     BEGSR
0091       C           ITEMNO    CHAININV                  97
0092       C   97                GOTO ITMEND
0093       C           X         ADD  1         X       20
0094       C                     MOVE IQPD00    IQPD,X
0095       C           X         COMP 3                        15
0096       C   15                EXSR SAVE
0097       C           SLN       ADD  1         SLN
0098       C           SLN       COMP 20                       90
0099       C   90                Z-ADD8         SLN
0100       C           PENDNG    ADD  QTY       PENDNG
0101       C           PRICE     MULT QTY       AMOUNT  82
0102       C           AMOUNT    ADD  TOTAL     TOTAL   92
0103       C           ITMEND    TAG
0104       C   97                SETON                     99
0105       C                     ENDSR
           C*
0106       C           SAVE      BEGSR
0107       C           X         COMP 0                    15
0108       C  N15                GOTO SAVE99
0109       C                     EXCPT
0110       C                     Z-ADD0         X
0111       C                     SETOF                     15
0112       C           SEQ#      ADD  1         SEQ#
0113       C           SAVE99    ENDSR
           C*
           C*
0114       C           RESET     BEGSR
0115       C                     SETON                     10
0116       C                     SETOF                     111213
0117       C                     Z-ADD7         SLN
0118       C                     Z-ADD1         SEQ#
0119       C                     ENDSR
```

Figure 19-15 (Part 4 of 6). Compiler Listing for Sample Program ORD

```
0121       OWK      D        10N99
0122       O                                   K5 'ZCNUM'
0123       O        D        11N99
0124       O                                   K5 'ZSHIP'
0125       O                         CRECD    206
0126       O        D        13N99
0127       O                                   K8 'SHOWITEM'
0128       O                         ITEMNO     6
0129       O                         QTY   L   12
0130       O                         DESC      34
0131       O                         PRICE J   44
0132       O                         AMOUNTJ   55
0133       O        D        12N99
0134       O                                   K8 'ZITEMHED'




0135       O        D        12N99
0136       O       OR        13N99
0137       O                                   K5 'ZITEM'
0138       O                         TOTAL J   13
0139       O        D        99
0140       O                                   K6 'ZERROR'
0141       O                 96                15 'INVALID CMD KEY'
0142       O                 98 11             18 'CUSTOMER NOT FOUND'
0143       O                 97 13             14 'ITEM NOT FOUND'
0144       OTRANS   EADD    N01 90
0145       O                         KEY        8
0146       O                                   11 '000'
0147       O        DADD     12N99
0148       O                         KEY        8
0149       O                                    8 '201'
0150       O                         CUSTNO    14
0151       O                         CUSTAD   128
           O*
0152       O        DADD     12N99
0153       O                         KEY        8
0154       O                                    8 '401'
0155       O                         SHIPAD   128
0156       O        EADD     15
0157       O                         KEY        8
0158       O                         IQPD   B 128
0159       O                                    6 '6'
0160       O        DADD     KAN99
0161       O                         KEY        8
0162       O                                    8 '801'
0163       O                         TOTAL  B  17
0164       O        E        KCN99
0165       O                         GPHOLD    11
0166       OTRANS   EADD     KBN15
0167       O                         KEY        8
0168       O                                    8 '000'
0169       OINV     D        13N99
0170       O                         PENDNG    16


INDICATORS USED
        H1 KA KB KC 01 10 11 12 13 15 90 91 92 96 97 98 99


           EXECUTION TIME TABLES AND ARRAYS

 STMT#   TABLE/  DEC  ENTRY  NUMBER OF    DTT      T/A
 DEFINED ARRAY   POS  LENGTH ENTRIES      DISP     DISP
  0007   IQPD           040   00003       0100     C12F

RPG-0314 UNREFERENCED FIELD NAMES
 STMT#  NAME
  0030 KEYHLD
  0020 ONHAND
```

Figure 19-15 (Part 5 of 6). Compiler Listing for Sample Program ORD



```
 FIELD NAMES USED
STMT#   NAME   DEC    LNG     DISP
 0026 CRECD           0206     0180
 0027 CUSTNO          0006     0185
 0028 CUSTAD          0100     01E9
 0029 SHIPAD          0100     024D
 0030 KEYHLD          0008     024E
 0031 KEY             0008     0255
 0032 WSID            0002     024F
 0033 GROUP#   0      0003     0252
 0034 CODF            0001     0253
 0035 SEQ#     0      0002     0255
 0037 IQPDOO          0040     027D
 0038 ITEMNO          0006     025B
 0039 QTY      0      0005     0260
 0040 PRICE    2      0007     0267
 0041 DESC            0022     027D
 0021 PENDNG   0      0005     0282
 0025 LSTGRP   0      0003     0285
 0003 SLN      0      0002     0287
 0083 GPHOLD   0      0003     028A
 0093 X        0      0002     028C
 0101 AMOUNT   2      0008     0294
 0102 TOTAL    2      0009     029D

 LABELS USED
STMT#   NAME     TYPE
 0055 END        TAG
 0056 NEW        BEGSR
 0064 CMDKEY     BEGSR
 0073 ENDORD     BEGSR
 0077 CANCEL     BEGSR
 0082 ENDJOB     BEGSR
 0090 ITEMS      BEGSR
 0103 ITMEND     TAG
 0106 SAVE       BEGSR
 0113 SAVE99     ENDSR
 0114 RESET      BEGSR

                 MAIN STORAGE USAGE OF RPG II CODE

    START    NAME IF    CODE       NAME     TITLE
    ADDR     OVERLAY    LENGTH
    0000                105F       RGROOT ROOT
    105F                0094       @PGTS  WORK STATION SCAN SUBROUTINE
    1165                00B1       RGMAIN INPUT MAINLINE
    1283                0026       RGSUBS CONTROL FIELDS
    10F3                006A       RGSUBS INPUT CONTROL ROUTINE
    1216                006D       RGSUBS RECORD IDENTIFICATION
    115D                0008       RGSUBS INPUT HOOK
    12A9                0737       @PGTI  WORK STATION INPUT PROCESSING
    1A5F                00A2       RGMAIN DETAIL CALCULATIONS
    19E0                004B       RGSUBS OUTPUT CONTROL ROUTINE
    1A4F                0010       RGSUBS CONSTANTS
    1A2B                0008       RGSUBS INPUT HOOK
    2105                000C       @PGDM  DATA MANAGEMENT CALL
    1CD0                004F       RGSUBS SUBSEG
    1B01                00CD       RGSUBS EXCEPTION
    1A3B                0008       RGSUBS INPUT HOOK
    1ED5                002C       RGSUBS SUBSEG
    1C73                005D       RGSUBS CHAIN CODE
    1A43                000C       RGSUBS OUTPUT HOOK
    1D1F                004A       RGSUBS SUBSEG
    1D69                0010       RGSUBS SUBSEG
    1E9E                0037       RGSUBS SUBSEG
    2058                0043       @PGRI  RESET RESULTING INDICATOR
    1D79                0014       RGSUBS SUBSEG
    1D8D                004F       RGSUBS SUBSEG
    1F01                00AE       @PGTR  WORK STATION DEALLOCATION  REL
    1DDC                00C2       RGSUBS SUBSEG
    1FAF                00A9       @PGAA  TAG  FETCH
    1C19                005A       RGSUBS CHAIN CODE
    209B                006A       @PGMC  MULTIPLY
    1A33                0008       RGSUBS INPUT HOOK
    1BCE                004B       RGSUBS CHAIN CODE
    2111                0055       RGMAIN INPUT FIELDS
    21DB                01B1       RGMAIN DETAIL OUTPUT
    217E                005D       RGSUBS CONSTANTS
    2166                000C       RGSUBS OUTPUT HOOK
    238C                0212       @PGTO  WORK STATION OUTPUT PROCESSING
    2172                000C       RGSUBS OUTPUT HOOK
    259E                001D       RGMAIN LR & OVERFLOW PROCESSING
    25BB                002A       RGMAIN CLOSE MAINLINE
    25E5                00B7       RGMAIN OPEN MAINLINE

                       09884       ORD      MAIN STORAGE REQUIRED TO EXECUTE.
                    #LIBRARY       SOURCE MEMBER INPUT LIBRARY.
                    #LIBRARY       LOAD MEMBER OUTPUT LIBRARY.
```

Figure 19-15 (Part 6 of 6). Compiler Listing for Sample Program ORD

Expanding Sample WORKSTN File Program ORD

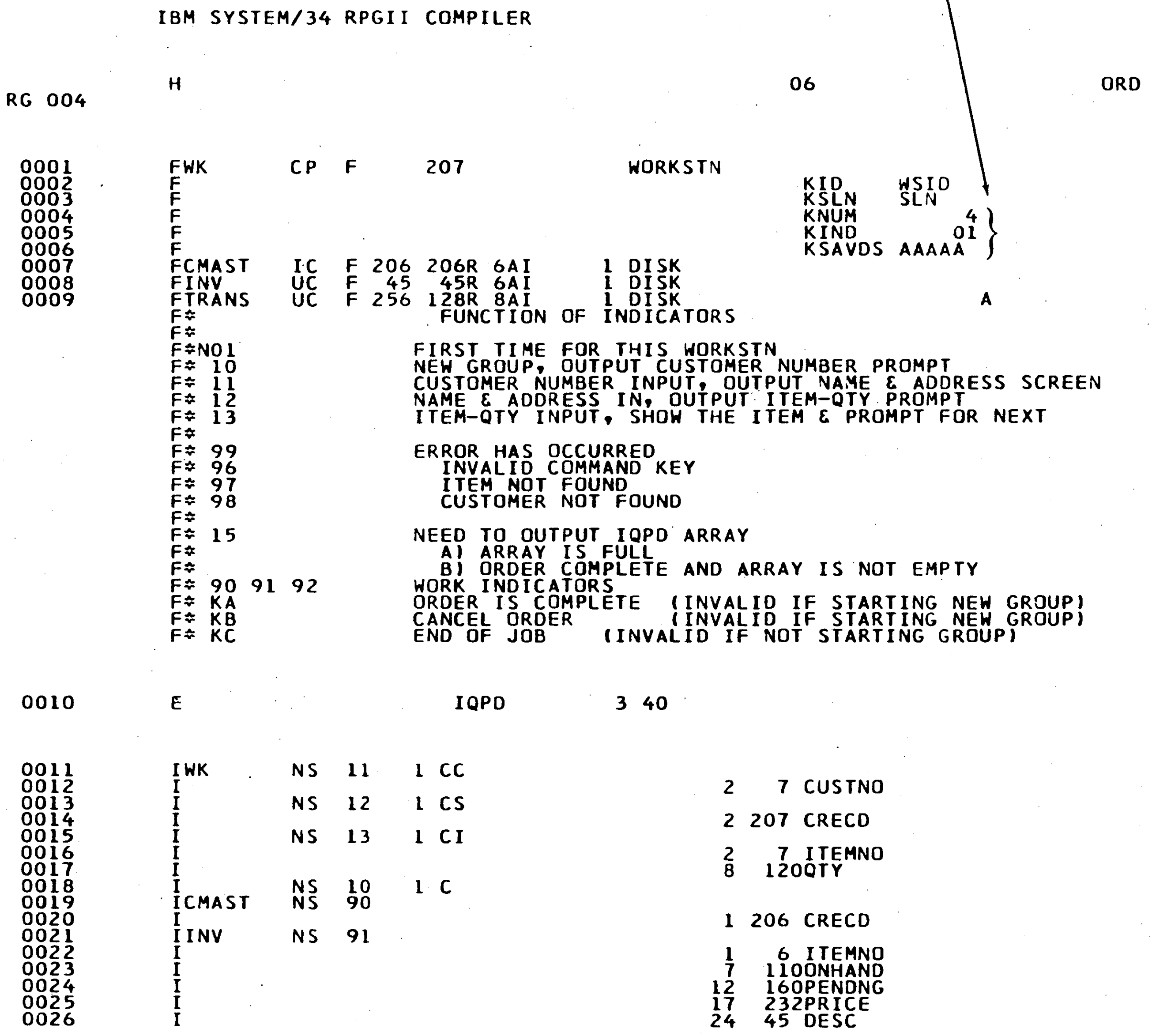
To expand ORD to support four display stations, the following must be done to the sample program shown in Figure 19-14.

- Add three continuation lines to the files description specifications using the keyword NUM with a value of 4 (which allows up to four display stations to call the program), the keyword IND with a value of 01 (which saves indicator 01 for each display station), and the keyword SAVDS with a value of AAAAA (which saves the fields in AAAAA).

- Define AAAAA as a data structure that contains the KEYHLD field, IQPD array and its index (field X), SLN field, and TOTAL field.

The compiler listing for the expanded sample program is shown in Figure 19-16.

To expand the program ORD to support four display stations, three continuation lines are added to the file description specifications. NUM with a value of 4 allows four display stations to call the program. IND with a value of 01 saves indicator 01 for each display station. SAVDS saves the fields in AAAAA for each display station.



```
          IBM SYSTEM/34 RPGII COMPILER

           H                                              06                     ORD
RG 004

0001       FWK      CP  F     207            WORKSTN
0002       F                                               KID    WSID
0003       F                                               KSLN   SLN
0004       F                                               KNUM       4
0005       F                                               KIND      01
0006       F                                               KSAVDS AAAAA
0007       FCMAST   IC  F 206 206R 6AI      1 DISK
0008       FINV     UC  F  45  45R 6AI      1 DISK
0009       FTRANS   UC  F 256 128R 8AI      1 DISK                      A
           F*                   FUNCTION OF INDICATORS
           F*
           F*N01              FIRST TIME FOR THIS WORKSTN
           F* 10              NEW GROUP, OUTPUT CUSTOMER NUMBER PROMPT
           F* 11              CUSTOMER NUMBER INPUT, OUTPUT NAME & ADDRESS SCREEN
           F* 12              NAME & ADDRESS IN, OUTPUT ITEM-QTY PROMPT
           F* 13              ITEM-QTY INPUT, SHOW THE ITEM & PROMPT FOR NEXT
           F*
           F* 99              ERROR HAS OCCURRED
           F* 96                INVALID COMMAND KEY
           F* 97                ITEM NOT FOUND
           F* 98                CUSTOMER NOT FOUND
           F*
           F* 15              NEED TO OUTPUT IQPD ARRAY
           F*                   A) ARRAY IS FULL
           F*                   B) ORDER COMPLETE AND ARRAY IS NOT EMPTY
           F* 90 91 92        WORK INDICATORS
           F* KA              ORDER IS COMPLETE  (INVALID IF STARTING NEW GROUP)
           F* KB              CANCEL ORDER       (INVALID IF STARTING NEW GROUP)
           F* KC              END OF JOB    (INVALID IF NOT STARTING GROUP)


0010       E                     IQPD        3 40


0011       IWK      NS  11    1 CC
0012       I                                         2   7 CUSTNO
0013       I        NS  12    1 CS
0014       I                                         2 207 CRECD
0015       I        NS  13    1 CI
0016       I                                         2   7 ITEMNO
0017       I                                         8  120QTY
0018       I        NS  10    1 C
0019       ICMAST   NS  90
0020       I                                         1 206 CRECD
0021       IINV     NS  91
0022       I                                         1   6 ITEMNO
0023       I                                         7  110ONHAND
0024       I                                        12  160PENDNG
0025       I                                        17  232PRICE
0026       I                                        24  45 DESC
```

Figure 19-16 (Part 1 of 5). Compiler Listing for Expanded Sample Program ORD

```
0027        ITRANS   NS   92   3 CH   4 CE   5 CA
0028        I                                          9  110LSTGRP
0029        ICRECD        DS
0030        I                                          1   6 CUSTNO
0031        I                                          7 106 CUSTAD
0032        I                                        107 206 SHIPAD
0033        IAAAAA        DS
0034        I                                          1   8 KEY
0035        I                                          1   2 WSID
0036        I                                          3   50GROUP#
0037        I                                          6   6 CODE
0038        I                                          7   80SEQ#
0039        I                                          9 128 IQPD
0040        I                                        129 1300X
0041        I                                        131 1320SLN
0042        I                                        133 1412TOTAL
0043        I             DS
0044        I                                          1  40 IQPD00
0045        I                                          1   6 ITEMNO
0046        I                                          7  110QTY
```

The data structures KEYHLD and IQPD00 from sample program ORD are now included in the data structure AAAAA, which is saved for each display station. The index (field X) for the array IQPD, the starting line number (field SLN) and total fields are also included in the data structure AAAAA.

```
0047        I                                         12  182PRICE
0048        I                                         19  40 DESC

0049        C N01                 EXSR NEW
0050        C                     EXSR CMDKEY
0051        C  99                 GOTO END
0052        C  KA                 EXSR ENDORD
0053        C  KB                 EXSR CANCEL
0054        C  KC                 EXSR ENDJOB
0055        C  KA
0056        COR KB
0057        COR KC                GOTO END
0058        C  11      CUSTNO     CHAINCMAST                98
0059        C  98                 SETON                     99
0060        C  11N98   GROUP#     ADD  1         GROUP#
0061        C  13                 EXSR ITEMS
0062        C          END        TAG
            C*
0063        C          NEW        BEGSR
0064        C                     MOVE 'HEADER'  KEY
0065        C          KEY        CHAINTRANS                90
0066        C  90                 EXCPT                           CREATE HEADER
0067        C                     SETON                     01    SET NOT 1ST
0068        C                     Z-ADDLSTGRP    GROUP#           SET GROUP#
0069        C                     EXSR RESET                      START NEW GROUP
0070        C                     ENDSR
0071        C          CMDKEY     BEGSR                           CHECK VALIDITY
0072        C                     SETOF                     969798RESET
0073        C                     SETOF                     99    ERRORS
0074        C  11 KA              SETON                     9996  *INVALID
0075        C  11 KB              SETON                     9996  * COMMAND
0076        C  12 KA              SETON                     9996  *  KEY
0077        C  12 KC              SETON                     9996  *    ENTERED
0078        C  13 KC              SETON                     9996  *
0079        C                     ENDSR
            C*
0080        C          ENDORD     BEGSR
0081        C                     EXSR SAVE
0082        C                     EXSR RESET
0083        C                     ENDSR
```

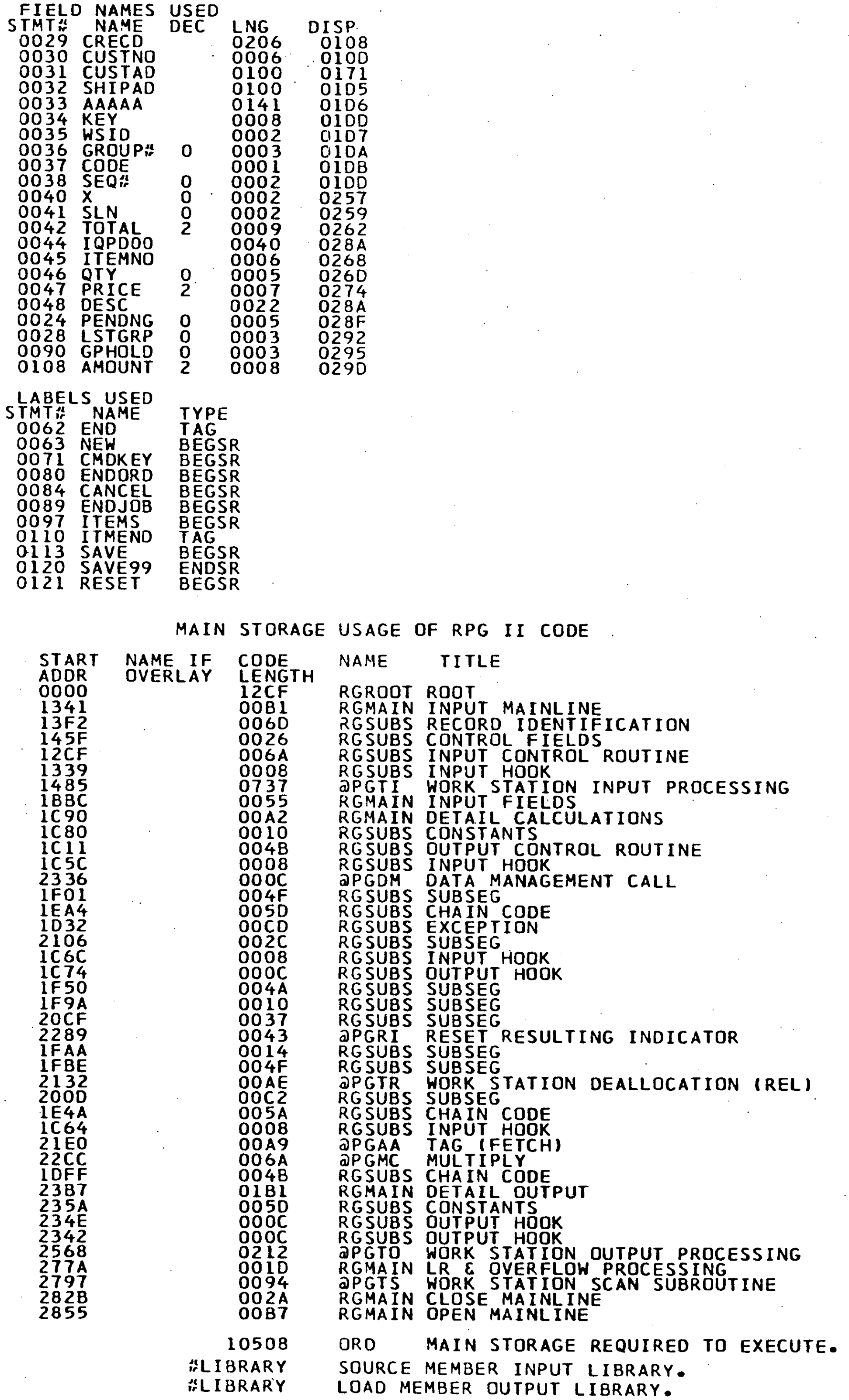
Figure 19-16 (Part 2 of 5). Compiler Listing for Sample Program ORD

```
           C*
0084       C           CANCEL   BEGSR
0085       C                    EXCPT
0086       C                    EXSR SAVE
0087       C                    EXSR RESET
0088       C                    ENDSR
           C*
0089       C           ENDJOB   BEGSR
0090       C                    Z-ADDGROUP#    GPHOLD  30
0091       C                    MOVE 'HEADER'  KEY
0092       C           KEY      CHAINTRANS                H1
0093       C                    EXCPT
0094       C                    SETOF                     11
0095       C           WSID     REL  WK
0096       C                    ENDSR
           C*
0097       C           ITEMS    BEGSR
0098       C           ITEMNO   CHAININV                  97
0099       C   97               GOTO ITMEND
0100       C           X        ADD  1         X       20
0101       C                    MOVE IQPD00    IQPD,X
0102       C           X        COMP 3                        15
0103       C   15               EXSR SAVE
0104       C           SLN      ADD  1         SLN
0105       C           SLN      COMP 20                       90
0106       C   90               Z-ADD8         SLN
0107       C           PENDNG   ADD  QTY       PENDNG
0108       C           PRICE    MULT QTY       AMOUNT  82
0109       C           AMOUNT   ADD  TOTAL     TOTAL   92
0110       C           ITMEND   TAG
0111       C   97               SETON                     99
0112       C                    ENDSR
           C*
0113       C           SAVE     BEGSR
0114       C           X        COMP 0                    15
0115       C  N15               GOTO SAVE99
0116       C                    EXCPT
0117       C                    Z-ADD0         X
0118       C                    SETOF                     15
0119       C           SEQ#     ADD  1         SEQ#
0120       C           SAVE99   ENDSR
           C*
           C*
0121       C           RESET    BEGSR
0122       C                    SETON                     10
0123       C                    SETOF                     111213
0124       C                    Z-ADD7         SLN
0125       C                    Z-ADD1         SEQ#
0126       C                    ENDSR


0128       OWK      D        10N99
0129       O                                  K5 'ZCNUM'
0130       O        D        11N99
0131       O                                  K5 'ZSHIP'
0132       O                        CRECD    206
0133       O        D        13N99
0134       O                                  K8 'SHOWITEM'
```

Figure 19-16 (Part 3 of 5). Compiler Listing for Sample Program ORD

```
0135       O                          ITEMNO     6
0136       O                          QTY   L   12
0137       O                          DESC      34
0138       O                          PRICE J   44
0139       O                          AMOUNTJ   55
0140       O        D       12N99
0141       O                                    K8 'ZITEMHED'
0142       O        D       12N99
0143       O       OR       13N99
0144       O                                    K5 'ZITEM'
0145       O                          TOTAL J   13
0146       O        D        99
0147       O                                    K6 'ZERROR'
0148       O                 96                 15 'INVALID CMD KEY'
0149       O                 98 11              18 'CUSTOMER NOT FOUND'
0150       O                 97 13              14 'ITEM NOT FOUND'
0151       OTRANS   EADD   NO1 90
0152       O                          KEY        8
0153       O                                    11 '000'
0154       O        DADD    12N99
0155       O                          KEY        8
0156       O                                     8 '201'
0157       O                          CUSTNO    14
0158       O                          CUSTAD   128
           O≑
0159       O        DADD    12N99
0160       O                          KEY        8
0161       O                                     8 '401'
0162       O                          SHIPAD   128
0163       O        EADD     15
0164       O                          KEY        8
0165       O                          IQPD   B 128
0166       O                                     6 '6'
0167       O        DADD    KAN99
0168       O                          KEY        8
0169       O                                     8 '801'
0170       O                          TOTAL  B  17
0171       O        E       KCN99
0172       O                          GPHOLD    11
0173       OTRANS   EADD    KBN15
0174       O                          KEY        8
0175       O                                     8 '000'
0176       OINV     D       13N99
0177       O                          PENDNG    16


INDICATORS USED
        H1 KA KB KC 01 10 11 12 13 15 90 91 92 96 97 98 99
RPG-0305 INDICATORS UNREFERENCED
        H1 91 92
            EXECUTION TIME TABLES AND ARRAYS

 STMT#   TABLE/  DEC   ENTRY  NUMBER OF   DTT      T/A
 DEFINED ARRAY   POS   LENGTH ENTRIES     DISP     DISP
  0010   IQPD            040   00003      0100     0205
```

Figure 19-16 (Part 4 of 5). Compiler Listing for Sample Program ORD



```
 FIELD NAMES USED
STMT#   NAME   DEC   LNG    DISP
 0029 CRECD          0206    0108
 0030 CUSTNO         0006    010D
 0031 CUSTAD         0100    0171
 0032 SHIPAD         0100    01D5
 0033 AAAAA          0141    01D6
 0034 KEY            0008    01DD
 0035 WSID           0002    01D7
 0036 GROUP#   0     0003    01DA
 0037 CODE           0001    01DB
 0038 SEQ#     0     0002    01DD
 0040 X        0     0002    0257
 0041 SLN      0     0002    0259
 0042 TOTAL    2     0009    0262
 0044 IQPD00         0040    028A
 0045 ITEMNO         0006    0268
 0046 QTY      0     0005    026D
 0047 PRICE    2     0007    0274
 0048 DESC           0022    028A
 0024 PENDNG   0     0005    028F
 0028 LSTGRP   0     0003    0292
 0090 GPHOLD   0     0003    0295
 0108 AMOUNT   2     0008    029D

 LABELS USED
STMT#   NAME    TYPE
 0062 END       TAG
 0063 NEW       BEGSR
 0071 CMDKEY    BEGSR
 0080 ENDORD    BEGSR
 0084 CANCEL    BEGSR
 0089 ENDJOB    BEGSR
 0097 ITEMS     BEGSR
 0110 ITMEND    TAG
 0113 SAVE      BEGSR
 0120 SAVE99    ENDSR
 0121 RESET     BEGSR

                MAIN STORAGE USAGE OF RPG II CODE

  START  NAME IF  CODE     NAME    TITLE
  ADDR   OVERLAY  LENGTH
  0000            12CF     RGROOT ROOT
  1341            00B1     RGMAIN INPUT MAINLINE
  13F2            006D     RGSUBS RECORD IDENTIFICATION
  145F            0026     RGSUBS CONTROL FIELDS
  12CF            006A     RGSUBS INPUT CONTROL ROUTINE
  1339            0008     RGSUBS INPUT HOOK
  1485            0737     @PGTI  WORK STATION INPUT PROCESSING
  1BBC            0055     RGMAIN INPUT FIELDS
  1C90            00A2     RGMAIN DETAIL CALCULATIONS
  1C80            0010     RGSUBS CONSTANTS
  1C11            004B     RGSUBS OUTPUT CONTROL ROUTINE
  1C5C            0008     RGSUBS INPUT HOOK
  2336            000C     @PGDM  DATA MANAGEMENT CALL
  1F01            004F     RGSUBS SUBSEG
  1EA4            005D     RGSUBS CHAIN CODE
  1D32            00CD     RGSUBS EXCEPTION
  2106            002C     RGSUBS SUBSEG
  1C6C            0008     RGSUBS INPUT HOOK
  1C74            000C     RGSUBS OUTPUT HOOK
  1F50            004A     RGSUBS SUBSEG
  1F9A            0010     RGSUBS SUBSEG
  20CF            0037     RGSUBS SUBSEG
  2289            0043     @PGRI  RESET RESULTING INDICATOR
  1FAA            0014     RGSUBS SUBSEG
  1FBE            004F     RGSUBS SUBSEG
  2132            00AE     @PGTR  WORK STATION DEALLOCATION (REL)
  200D            00C2     RGSUBS SUBSEG
  1E4A            005A     RGSUBS CHAIN CODE
  1C64            0008     RGSUBS INPUT HOOK
  21E0            00A9     @PGAA  TAG (FETCH)
  22CC            006A     @PGMC  MULTIPLY
  1DFF            004B     RGSUBS CHAIN CODE
  23B7            01B1     RGMAIN DETAIL OUTPUT
  235A            005D     RGSUBS CONSTANTS
  234E            000C     RGSUBS OUTPUT HOOK
  2342            000C     RGSUBS OUTPUT HOOK
  2568            0212     @PGTO  WORK STATION OUTPUT PROCESSING
  277A            001D     RGMAIN LR & OVERFLOW PROCESSING
  2797            0094     @PGTS  WORK STATION SCAN SUBROUTINE
  282B            002A     RGMAIN CLOSE MAINLINE
  2855            00B7     RGMAIN OPEN MAINLINE

                 10508     ORD    MAIN STORAGE REQUIRED TO EXECUTE.
              #LIBRARY     SOURCE MEMBER INPUT LIBRARY.
              #LIBRARY     LOAD MEMBER OUTPUT LIBRARY.
```

Figure 19-16 (Part 5 of 5). Compiler Listing for Sample Program ORD

Chapter 20. Ideographic Support

RPG II provides ideographic support when used with the ideographic version of the SSP and the ideographic hardware devices that version supports. Display stations with ideographic capability are supported by the WORKSTN file only.

Ideographic support allows the RPG II compiler to process IBM-supplied or user-defined ideographic character sets. Very little error checking is performed on ideographic data. Basically, the fact that data is ideographic is transparent to the RPG II compiler. You must ensure that the ideographic data is processed properly by your program.

Ideographic characters can be present in literals, constants, fields, tables, and arrays. The transparent literal option has been added to the control specifications (column 57) to signal to the compiler that transparent literals or constants can be present in the program. (For more information on the transparent literal option, see *Column 57 (Transparent Literal)* in Chapter 3.) A field, table, or array containing ideographic data is considered to contain alphameric data by the RPG II compiler. No error checking has been added for ideographic data in fields, tables, or arrays.

Ideographic data has a 2-byte representation, rather than a 1-byte representation as in the EBCDIC character set. This can cause the RPG operation codes that process data 1 byte at a time (COMP, MOVE, and so on) to produce incorrect results. In addition, ideographic data is enclosed by the shift-out (S/O) control character (hex 0E) and the shift-in (S/I) control character (hex 0F). These control characters must be taken into consideration when an operation that processes ideographic data is performed. (For more information on considerations that apply to processing ideographic data, see *Processing Considerations* in this chapter.)

SPECIFYING IDEOGRAPHIC DATA

Ideographic Constants

Ideographic characters can be specified in the constant or edit word section of the output specifications (columns 45 through 70). Ideographic constants must begin with an apostrophe immediately followed by the S/O control character. Ideographic constants must end with the S/I control character immediately followed by the terminating apostrophe.

Note: When ideographic constants are processed by RPG, the S/O and S/I control characters are considered to be part of the constant data. When the constant is displayed or printed on an ideographic device, these control characters appear as blanks.

When an ideographic constant is used, the transparent literal option must be specified on the control specifications. When this option is specified, the compiler checks for constants (and literals) that begin with an apostrophe followed by the S/O control character. If a constant is found that begins with an apostrophe followed by the S/O control character, the compiler checks to see if the constant is a valid transparent constant. A constant is not a valid transparent constant if:

- A second S/O control character is found before the S/I control character.
- An odd number of 1-byte characters are found between the S/O and S/I control characters.
- The S/I control character is not immediately followed by the terminating apostrophe.

If a constant is determined to be an invalid transparent constant, it is rechecked to see if it is a valid alphameric constant. If the constant is a valid transparent constant, it is not checked for embedded apostrophes.

Any ideographic character can be entered in an ideographic constant. Each ideographic character has a 2-byte hex representation. (An ideographic blank also occupies 2 bytes.) Because each character occupies 2 bytes in storage, ideographic constants can only be from 1 to 11 characters long (this also allows for the control characters).

Note: An ideographic constant can be composed only of ideographic data. Mixing ideographic and EBCDIC data in the same constant causes the constant to be checked as alphameric.

Ideographic Literals

Ideographic characters can be specified as a literal in factor 1 or factor 2 of the calculation specifications. Ideographic literals must begin with an apostrophe immediately followed by the S/O control character. Ideographic literals must end with the S/I control character immediately followed by the terminating apostrophe.

Note: When ideographic data is processed by RPG, the S/O and S/I control characters are considered to be part of the literal data. When the literal is used by an RPG operation, these control characters are processed along with the rest of the literal. If you do not allow for the control characters, the operation can produce incorrect results. For example, if you are moving an ideographic literal to a field, you must allow two positions in the field for the control characters.

When an ideographic literal is used, the transparent literal option must be specified on the control specifications. When this option is specified, the compiler checks for literals (and constants) that begin with an apostrophe followed by the S/O control character. If a literal is found that does start with an apostrophe followed by the S/O control character, the compiler checks to see if the literal is a valid transparent literal. A literal is not a valid transparent literal if:

- A second S/O control character is found before the S/I control character.

- An odd number of 1-byte characters are found between the S/O and S/I control characters.

- The S/I control character is not immediately followed by the terminating apostrophe.

If a literal is determined to be an invalid transparent literal, it is rechecked to see if it is a valid alphameric literal. If the literal is a valid transparent literal, it is not checked for embedded apostrophes.

Any ideographic character can be entered in an ideographic literal. Each ideographic character has a 2-byte hex representation. (An ideographic blank also occupies 2 bytes.) Because each character occupies 2 bytes in storage, an ideographic literal can only be from 1 to 3 characters long (this also allows for the control characters).

Note: An ideographic literal can be composed only of ideographic data. Mixing ideographic and EBCDIC data in the same literal causes the literal to be checked as alphameric.

Ideographic Fields, Tables, and Arrays

Ideographic characters can be present in fields, tables, and arrays. The RPG compiler does not recognize these characters as ideographic. The compiler treats ideographic characters as alphameric. Ideographic fields, tables, and arrays must therefore conform to the rules for alphameric fields, tables, and arrays.

When ideographic data is present in a field, table, or array, the data must be enclosed in the S/O and S/I control characters. These control characters are considered to be part of the field, table element, or array element. Therefore, when the length of the field, table element, or array element is defined, space must be left for the control characters. For example, if you want to define a field so that it can contain four ideographic characters, you must specify a field length of 10 (two positions for each ideographic character, and one position for each control character). If you do not specify a large enough length, the field, table element, or array element is truncated, causing one of the control characters to be lost.

You must also consider the control characters when the field, table element, or array element is processed. For example, if a field is being printed or displayed on an ideographic device, the control characters appear as blanks. If blank after (column 39 of the output specifications contains a B) is specified for a field, the control characters are also blanked out and must be reconstructed if the field is to still contain ideographic data.

Note: When a field, table, or array contains ideographic data, it should contain only ideographic data. Mixing ideographic and EBCDIC data in the same field, table, or array can cause incorrect results.

Ideographic Comments

Ideographic characters can be entered as comments in source statements. The source statements that allow comments are the extension specifications (columns 58 through 74) and the calculation specifications (columns 60 through 74). Ideographic characters can also be specified on a comment line (column 7 contains an asterisk).

PROCESSING CONSIDERATIONS

Ideographic data can produce incorrect results when used with certain RPG operation codes. Since ideographic data has a 2-byte hex representation, operations that compare data byte by byte are not meaningful unless they check for an equal condition. Care must also be taken when ideographic data is moved. If the lengths of the data being moved and the area that the data is being moved to are not correctly specified, the S/O or S/I control characters can be lost.

A number of RPG operations and functions operate by comparing data 1 byte at a time. The COMP and LOKUP operations compare for high, low, and equal conditions. These operations compare the 1-byte EBCDIC values that correspond to the data that is present and produce a result based on the standard 1-byte collating sequence. Because of this, the only valid test when ideographic data is being processed is for an equal condition. If all the bytes in a field are equal to all the bytes in another field, the fields are equal whether they contain ideographic or EBCDIC data.

Match fields and sequence checking are also invalid for ideographic data. Match fields cause data from different records to be compared, 1 byte at a time. This produces incorrect results for ideographic data. Sequence checking compares data in different fields to see if the fields are in ascending or descending order. This comparison is done 1 byte at a time and therefore produces incorrect results for ideographic data.

The SETLL operation is another 1-byte comparison operation that cannot be used with ideographic data. This operation causes the key of each record to be compared with a lower limit value. If the key of the record is higher than the lower limit, the record is selected for processing. As this comparison is carried out using 1-byte EBCDIC values, the SETLL operation can produce incorrect results when used with ideographic data.

RPG allows you to define an alternate collating sequence for EBCDIC data. In other words, you can redefine the order in which 1-byte segments of data will be sorted. This is meaningless for ideographic data.

Care must be taken when the various move operations (MOVE, MOVEA, MOVEL) are used with ideographic data. The length of the field, table element, or array element that the ideographic data is being moved to must be defined as being exactly the same length as the literal, field, table element, or array element being moved. If the lengths are not the same, the data will not be recognized as ideographic. For example, if the field that the data is being moved to is shorter than the length of the ideographic data, the data is truncated, causing one of the control characters to be lost. If the field that the data is being moved to is longer than the ideographic data, one of the control characters will be embedded in the field. This causes the control character to be considered part of the data.

MOVE IDEOGRAPHIC DATA WITH DELETION OF CONTROL CHARACTERS (SUBR40)

SUBR40 is a move and edit routine that moves the contents of one field to another field. If the S/O and S/I control characters are found as the first and last characters in the field, SUBR40 deletes them.

SUBR40 is called as shown in Figure 20-1.

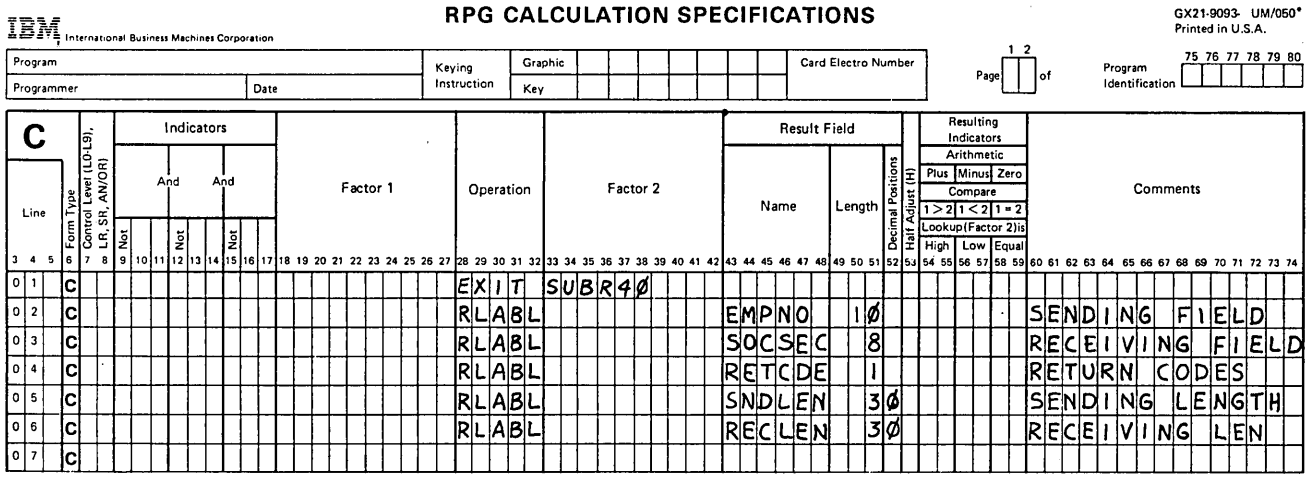
If you want the receiving field to contain all the data that was present in the sending field, you must specify a length for the receiving field that is two positions less than the length of the sending field. This allows two positions for each ideographic character (or one for each EBCDIC character) while deleting the S/O and S/I control characters (and the two positions they occupied). If you specify a receiving field longer than the sending field minus two positions, all the data from the sending field is moved and the receiving field is padded on the right with blanks (1-byte EBCDIC blanks). If the receiving field is shorter than the sending field minus two positions, the data being moved is truncated on the right.

Five RLABL fields must be specified when SUBR40 is called. The first two specify the sending and receiving fields for the move. The third field is where the return codes are written to indicate the status of the move operation. The fourth and fifth fields must be loaded with the lengths of the sending and receiving fields. These are the lengths of the fields specified on the first two RLABLs for the call to SUBR40 (in Figure 20-1, you would need to load the lengths of EMPNO and SOCSEC). The return code field must be defined as a one-position alphameric field; the length fields must be defined as three-position numeric fields with zero decimal positions.

SUBR40 produces return codes to indicate the status of the move operation. The following list contains these return codes and their meanings:

Return Code	Explanation
0	Move executed; no errors.
1	Move executed; padding occurred.
2	Move executed; truncation occurred.
3	Move executed; S/O and S/I control characters were not found.
4	Move not executed. Either an odd field length was found, a length of zero was found, the length was greater than 256, or an invalid character was found in the field length.

If more than one return code can be issued, only the highest return code is returned.



IBM International Business Machines Corporation

RPG CALCULATION SPECIFICATIONS

GX21-9093- UM/050* Printed in U.S.A.

Program | Keying Instruction | Graphic | Key | Card Electro Number | Page of | Program Identification

Programmer | Date

C

Line	Form Type	Control Level (L0-L9, LR, SR, AN/OR)	Indicators	Factor 1	Operation	Factor 2	Result Field Name	Length	Decimal Positions	Half Adjust (H)	Resulting Indicators	Comments
01	C				EXIT	SUBR4Ø						
02	C				RLABL		EMPNO	1Ø				SENDING FIELD
03	C				RLABL		SOCSEC	8				RECEIVING FIELD
04	C				RLABL		RETCDE	1				RETURN CODES
05	C				RLABL		SNDLEN	3	Ø			SENDING LENGTH
06	C				RLABL		RECLEN	3	Ø			RECEIVING LEN
07	C											

Figure 20-1. Calling SUBR40

MOVE IDEOGRAPHIC DATA WITH ADDITION OF CONTROL CHARACTERS (SUBR41)

SUBR41 is a move and edit routine that moves the contents of one field into another field. If the S/O and S/I control characters are not found in the first and last positions of the field, SUBR41 adds them to the field when it is moved.

SUBR41 is called as shown in Figure 20-2.

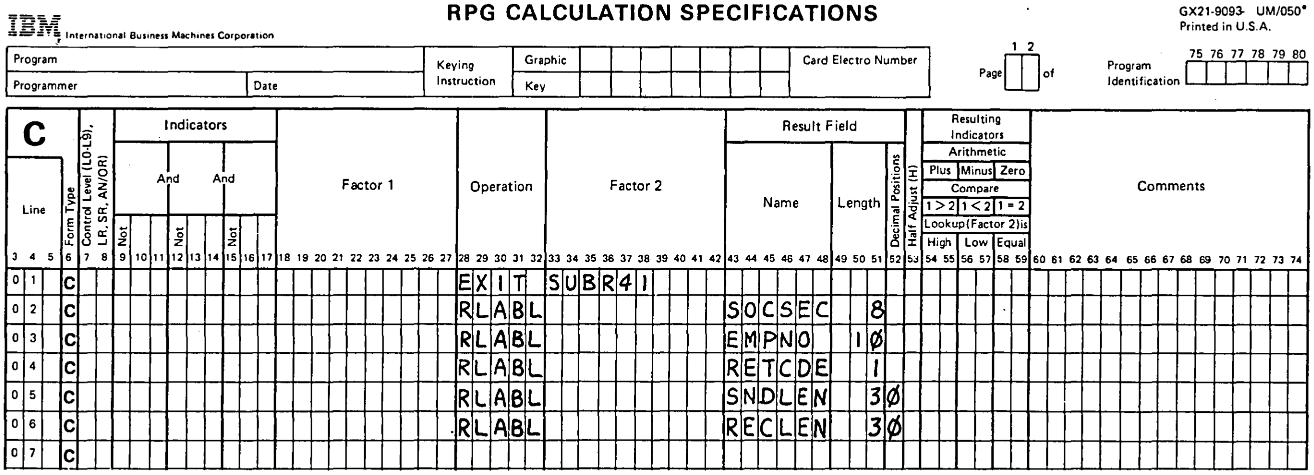
If you want the receiving field to contain all the data that is in the sending field, you must specify the length of the receiving field to be two positions longer than the length of the sending field (to hold the S/O and S/I control characters). If you specify a receiving field that is longer than the sending field plus two, the data is padded on the right when it is moved into the receiving field. If the receiving field is shorter than the sending field plus two, the data is truncated on the right when it is moved. If the receiving field is specified either longer or shorter than the sending field plus two positions, the S/I control character is still placed in the correct position (the rightmost position).

Five RLABL fields must be specified when SUBR41 is called. The first two specify the sending and receiving fields for the move. The third field is where the return codes are written to indicate the status of the move operation. The fourth and fifth fields must be loaded with the lengths of the sending and receiving fields. These are the lengths of the fields specified on the first two RLABLs for the call to SUBR41 (in Figure 20-2, you would need to load the lengths of SOCSEC and EMPNO). The return code field must be defined as a one-position alphameric field; the length fields must be defined as three-position numeric fields with zero decimal positions.

SUBR41 produces return codes to indicate the status of the move. The following list contains these return codes and their meanings:

Return Code	Explanation
0	Move executed; no errors.
1	Move executed; padding occurred to left of S/I control character.
2	Move executed; data truncated to left of S/I control character.
3	Move executed; S/O and S/I already present.
4	Move not executed. Either odd field length found, length of zero found, length greater than 256, or invalid character found in field length.

If more than one return code can be issued, only the highest return code is issued.



IBM International Business Machines Corporation

RPG CALCULATION SPECIFICATIONS

GX21-9093- UM/050*
Printed in U.S.A.

Program | Keying Instruction | Graphic | Key | Card Electro Number | Page of | Program Identification

Programmer | Date

Line	Form Type	Control Level	Indicators	Factor 1	Operation	Factor 2	Result Field Name	Length	Decimal Positions	Half Adjust	Resulting Indicators	Comments
01	C				EXIT	SUBR41						
02	C				RLABL		SOCSEC	8				
03	C				RLABL		EMPNO	1Ø				
04	C				RLABL		RETCDE	1				
05	C				RLABL		SNDLEN	3	Ø			
06	C				RLABL		RECLEN	3	Ø			
07	C											

Figure 20-2. Calling SUBR41

IDEOGRAPHIC DEVICE SUPPORT

Three new keywords have been added to the INFDS for ideographic devices. For more information on these keywords, see *Specifications for the INFDS Data Structure* in Chapter 13.

MESSAGES

The RPG displayed messages (both compile time and execution time) are displayed in either the standard character set or an ideographic character set. The messages are displayed in an ideographic character set if ideographic support was requested when the user signed on.

The RPG compiler messages are printed in either the standard character set or an ideographic character set. The messages are printed in an ideographic character set if ideographic support was requested when the user signed on.

Appendix A. Summary Charts

CONTROL SPECIFICATIONS

Columns	Name	Entry	Explanation
1-2	Page	Page number	Entry used to assign a page number to each specification sheet.
3-5	Line	Line number	Entry used to number the specification line.
6	Form type	H	Identification for the control (or header) specification.
7		*	Asterisk in this column identifies this line as a comment line.
7-9	Size to compile	Blank	
10	Object output	Blank	Blank entry causes the system to halt for terminal errors only.
		D	D entry causes the system to halt for both warning messages and severe errors.
11	Listing options	Blank	Program listing is produced.
		B	No program listing is produced.
		P	Partial program listing is produced.
12-14	Size to execute		
12		Blank or 0	Entry in columns 13 and 14 determines the size to execute.
		Q, H, T	Entry in columns 13 and 14 is rounded up to next even number.
13-14		Blank	Main storage available for object program execution defaults to region size specified.
		02-64	Enter the main storage available in a multiple of 2K bytes (K = 1,024). If entry is odd number, it is rounded up to next even number.
15	Debug	Blank	DEBUG operation is not used.
		1	DEBUG operation is used.
16-17		Blank	
18	Currency Symbol	Any character except *, 0, &, ., –, C, R, or ,.	The currency symbol used in edit words and with edit codes. Blank entry defaults to $.
19	Date format (UDATE)	Blank or M	Month/day/year format. If column 19 is blank and column 21 contains a D, I, or J, the day/month/year (ddmmyy) format is used instead of the month/day/year (mmddyy) format.
		D	Day/month/year (ddmmyy).
		Y	Year/month/day (yymmdd).

CONTROL SPECIFICATIONS (continued)

Columns	Name	Entry	Explanation
20	Date edit (Y edit code)	Blank	A blank entry defaults to slash (/) if column 19 contains an M, or if column 21 contains a D or blank and column 19 is blank. A blank entry defaults to period (.) if column 19 contains D or Y, or if column 21 contains I or J and column 19 is blank.
		&	A blank separates the date field.
		Any other character	Character entered separates the edited date field.
21	Inverted print	Blank	Numeric fields use decimal point (.). Date format is mmddyy if column 19 is blank.
		D	Numeric fields use decimal point (.). Date format is ddmmyy if column 19 is blank.
		I	Numeric fields use decimal comma (,). Date format is ddmmyy if column 19 is blank.
		J	Numeric fields use decimal comma (,) and leading zero remains for zero balance. Date format is ddmmyy if column 19 is blank.
22-25		Blank	
26	Alternate collating sequence	Blank	Normal collating sequence is used.
		S	Alternate collating sequence is used.
27-36		Blank	
37	Inquiry	Blank or I	Program, when interrupted, will not allow the operator to enter new procedures or commands.
		B	Program, when interrupted, will allow the operator to enter new procedures or commands. (For an explanation of inquiry, see *Column 37 (Inquiry)* in Chapter 2.)
38-40		Blank	
41	1P forms positions	Blank	First line is printed only once.
		1	First line can be printed repeatedly to allow operator to position forms.
42		Blank	
43	File translation	Blank	No file translation is needed.
		F	Input, output, update, or combined files are translated.
44		Blank	
45	Nonprint characters	Blank	Program halts if unprintable character was in last line printed.
		1	Program does not halt for unprintable characters.

CONTROL SPECIFICATIONS (continued)

Columns	Name	Entry	Explanation
46-47		Blank	
48	Shared I/O	Blank 1	All disk files use a separate input/output area. All disk files share a single input/output area.
49-51		Blank	
52-53	Number of formats	Blank 0-32	An entry of 32 is assumed. Entry specifies number of formats in display screen format load member for WORKSTN file.
54-56	Blank		
57	Transparent literal	Blank 1	No transparent literals or constants are present in this program. Transparent literals or constants can be present in this program.
58-74	Blank		
75-80	Program identification		Entry used to assign a unique name to the program and to name the display screen format load member used by the CONSOLE or WORKSTN file.

FILE DESCRIPTION SPECIFICATIONS

Columns	Name	Entry	Explanation
1-2	Page	Page numbers	Entry used to assign a page number to each specification sheet.
3-5	Line	Line number	Entry used to number the specification lines.
6	Form type	F	Identification for a file description specification.
7		*	Asterisk in this column identifies this line as a comment line.
7-14	Filename	Filename	Each file requires a unique name. The filename can be from 1 to 8 characters long, must begin in column 7, and must be a valid RPG II name.
15	File type	I	Input
		O	Output
		U	Update
		C	Combined (SPECIAL or WORKSTN device only)
16	File	Blank	Blank for all output files except chained output files
		P	Primary
		S	Secondary
		C	Chained
		R	Record address
		T	Table or array
		D	Demand
17	End of file	Blank	The program can end whether or not all records from this file are processed. This column must be blank for a WORKSTN or
		E	KEYBORD file. All records from the file must be processed before the program ends. An E can be specified here only if column 15 contains I or U, and column 16 contains a P, S, or R. *Note:* If column 17 is blank or contains E for all files, all records from every file must be processed before the program can end.
18	Sequence	Blank	No sequence checking is to be done. This column must be blank for a WORKSTN file.
		A	Sequence checking is done. Records are in ascending sequence.
		D	Sequence checking is done. Records are in descending sequence. *Note:* Sequence checking is required when match fields are used. Column 18 applies only to primary and secondary files.
19	File format	F or blank	Fixed-length record format.

FILE DESCRIPTION SPECIFICATIONS (continued)

Columns	Name	Entry	Explanation
20-23	Block length	Blank	These columns must be blank for a WORKSTN file and can be left blank for any other file.
		1-9999	Disk (record length or multiple of record length)
		2-1518	CONSOLE (if specified, must equal record length)
		1-79	KEYBORD
		1-132	Printer
		1-79	CRT
		1-9999	SPECIAL (record length or greater than record length)
		1-4075	BSCA (record length or multiple of record length)
24-27	Record length	1-4096	Disk
		2-1518	CONSOLE
		1-79	KEYBORD
		1-132	Printer
		1-79	CRT
		1-4096	SPECIAL
		1-4075	BSCA
		1-9999	WORKSTN
28	Mode of processing	Blank	Sequential by key
			Consecutive
			Note: Columns 28 through 32 must be blank for nondisk files.
		L	Sequential within limits
		R	Random by relative record number
			Random by key
			By addrout file
			Direct file load (random load)
29-30	Length of key field	1-8	Length of record keys in packed format (for indexed files)
	or record address field	1-29	Length of record keys in unpacked format (for indexed files and record address files)
		3	Length of relative record number in addrout file
			Note: These columns must be blank for nondisk files.
31	Record address type	Blank	Sequential or direct file
		P	Indexed file with packed keys
		A	Indexed file with alphameric keys
		I	Addrout file or processed by addrout file
			Note: Column 31 applies to disk files specified as input, update, or chained output files

Columns	Name	Entry	Explanation
32	File organization or additional I/O area	Blank	Sequential or direct file; one input/output area for the file
		I	Indexed file
		T	Addrout file
		1-9	Sequential or direct file; two input/output areas for the file
33-34	Overflow indicator	Blank	No overflow indicator used
		OA-OG, OV	Overflow indicator used to condition records in printer files
35-38	Key field starting location	1-4096	For indexed files, enter the beginning position of the key field in the record. This entry must end in column 38.
39	Extension code	Blank	No file-related line counter or extension specifications are used.
		E	Table file, array file, or record address file is further described by extension specifications.
		L	Printer file is further described by line counter specifications.
40-46	Device	DISK	Disk
		KEYBORD	Display screen — keyboard (display station)
		PRINTER	132-position printer
		CONSOLE	Display screen — keyboard (display station)
		CRT	Display screen
		SPECIAL	Used for devices not supported directly by RPG II
		BSCA	Binary synchronous communications adapter
		WORKSTN	Display screen — keyboard (display station)
47-52	Blank		
53		K	Continuation line specified
54-59	Name of label exit	Blank	No SPECIAL device used.
		SUBRxx	Name of the user-written subroutine that performs the input/output operation for a SPECIAL device (x = any alphabetic character).
		SRyzzz	Name of the IBM-written subroutine (5-character name in library is @yzzz) that performs the input/output operation for a device supported by SPECIAL (y = any of the following: B, C, D, F, G, H, I, L, M, O, P, R, S, T, or U; z = any of the following: A, B, C, D, F, G, H, I, L, M, O, P, R, S, T, or U).
	Continuation line option for SPECIAL device	Array name	Name of array to be used by user-written subroutine.

Columns	Name	Entry	Explanation
54-59 (continued)	Continuation line options for WORKSTN device		For WORKSTN options if multiple display stations are attached to a WORKSTN file or if the file information data structure (INFDS) or the exception/error processing subroutine (INFSR) is specified.
		NUM	Specify the maximum number of display stations that can be on this file in columns 60 through 65. If not specified, 1 is assumed.
		SAVDS	Enter the name of a data structure in columns 60 through 65 that is to be saved and restored for each display station in this file. If not specified, no swapping is done.
		IND	Specify the number of indicators (beginning with 01) to be swapped by display station. If not specified, no swapping is done.
		SLN	In columns 60 through 65, specify the name of a two-digit numeric field whose value determines the first line on the display screen where the display format is to begin if variable starting line number was specified in the format.
		ID	In columns 60 through 65, enter the name of a 2-character alphameric field that contains the ID of the display station currently being processed in this field.
		INFSR	Enter the name of the user-written subroutine that may receive control when WORKSTN exception/error conditions occur.
		INFDS	Enter the name of the data structure that contains file related information when exception/error conditions occur during WORKSTN operations or when the display screen size is to be posted.
		FMTS	Enter *NONE in conjunction with FMTS if there are only SSP-ICF formats present in the program. Otherwise, enter a name to be used as the display screen format load member name. If a name is not entered on the FMTS continuation line option, the compiler assumes the display screen format load member name is the program name (from columns 75 through 80 of the control specifications) with FM added to the end of the name.
	Continuation line options for DISK device	RECNO	In columns 60 through 65, enter the name of a seven-position, numeric field with zero decimal positions. This field contains the relative record number of a record to be added to a sequential or direct file processed randomly.

FILE DESCRIPTION SPECIFICATIONS (continued)

Columns	Name	Entry	Explanation
60-65	Storage index	Blank 6-9999	No storage index is kept in storage. Number of bytes reserved for storage index.
66	File addition/ unordered load	A U	New records will be added to the file. Records are to be loaded into an indexed file in unordered sequence. *Note:* This column applies to sequential and indexed disk files.
67-70		Blank	
71-72	File condition	Blank U1-U8	An external indicator does not condition the file. Specified external indicator conditions the file. *Note:* These columns apply to output files, primary and secondary input files, and update files. A record address file can be conditioned by an external indicator if its associated primary or secondary file is conditioned either by the same indicator or by no indicator.
73-74		Blank	
75-80	Program identification		This space is available for comments.

EXTENSION SPECIFICATIONS

Columns	Name	Entry	Explanation
1-2	Page	Page number	Entry used to assign a page number to each specification sheet.
3-5	Line	Line number	Entry used to number the specification lines.
6	Form type	E	Identification for an extension specification.
7		*	Asterisk in this column identifies this line as a comment line.
7-10		Blank	
11-18	From filename	Blank	Table or array load at compilation time if columns 33 through 35 contain an entry. Array is loaded at execution time if columns 33 through 35 are blank.
		Filename	Name of the table or array input file loaded at preexecution time or name of the record address file defined on the file description specifications sheet. Entry must be left-justified.
19-26	To filename	Blank	Blank if the table or array is not written at end of job.
		Filename	Name of the primary or secondary input or update file containing the data records to be processed if the file named in columns 11 through 18 is a record address file. Name of the output file to which the table or array is written at end of job if the file named in columns 11 through 18 is a table or array file.
27-32	Table or array name	Table or array name	Name of a table or array used in the program. If alternating tables or arrays are described, enter the name of the table or array whose entry is first on the input record. Entries must be left-justified and must be valid RPG II names. Table names must begin with TAB; array names must not begin with TAB.
33-35	Number of entries per record	Blank	These columns must be blank for execution-time arrays.
		1-999	Number of entries on each table or array input record. These columns must contain an entry for compile- and preexecution-time tables and arrays. Entry must be right-justified.
36-39	Number of entries per table or array	1-9999	Maximum number of entries in the table or arrays; corresponding items are considered one entry. Entry must be right-justified.
40-42	Length of entry	1-15	Length of numeric entry. For packed or binary numeric data, enter the number of digits required to represent the data in zoned decimal format. Entry must be right-justified.
		1-256	Length of alphameric entry.

EXTENSION SPECIFICATIONS (continued)

Columns	Name	Entry	Explanation
43	Packed or binary field	Blank	Alphameric or zoned decimal numeric data
		P	Packed numeric data
		B	Binary numeric data
44	Decimal positions	Blank	Alphameric table or array
		0-9	Number of positions to the right of the decimal
45	Sequence	Blank	No particular sequence
		A	Ascending sequence
		D	Descending sequence
			Note: This column describes the sequence of data in a table or array. Column 45 must contain an entry if high or low lookup is used.
46-57			Description of a second table or array entered in alternating format with the table or array named in columns 27 through 32. These entries have the same significance as the corresponding entries in columns 27 through 45.
58-74	Comments		Any helpful information about the specification line.
75-80	Program identification		This space is available for comments.

LINE COUNTER SPECIFICATIONS

Columns	Name	Entry	Explanation
1-2	Page	Page number	Entry used to assign a page number to each specification sheet.
3-5	Line	Line number	Entry used to number the specification lines.
6	Form type	L	Identification for the line counter specification.
7		*	Asterisk in this column identifies this line as a comment line.
7-14	Filename	Filename	Name of a printer file for which form size and overflow line are specified.
15-17	Line number—number of lines per page	1-112	Number of lines available for printing on the printer form.
18-19	Form length	FL	Identification that the previous entry is the form length.
20-22	Line number—overflow line	1-112	Number of the overflow line.
23-24	Overflow line	OL	Identification that the previous entry is the overflow line.
25-74		Blank	
75-80	Program identification		This space is available for comments.

TELECOMMUNICATIONS SPECIFICATIONS

Columns	Name	Entry	Explanation
1-2	Page	Page number	Entry used to assign a page number to each specifications sheet.
3-5	Line	Line number	Entry used to number the specification lines.
6	Form type	T	Identification for a telecommunications specification.
7		*	Asterisk in this column identifies this line as a comment line.
7-14	Filename	Filename	Every BSCA file in a program requires a valid filename. The same filename must appear on the file description specifications.
15	Configuration	P or blank	Point-to-point, nonswitched network.
		M	Multipoint network, where the control station selects the tributary station through polling or addressing. System/34 cannot be the control station.
		S	Point-to-point switched network.
16	Type of station	T	This station transmits messages from the file named in columns 7 through 14. The file must be designated as an output file by file description specifications and must appear on the output specifications sheet.
		R	This station receives messages into the file named in columns 7 through 14. The file must be designated as an input file by file description specifications and must appear on the input specifications sheet.
17	Type of control	T	Tributary station on a multipoint network. System/34 cannot be the control station and transmit the polling supervisory sequence. Column 17 must contain a T if column 15 contains an M (multipoint network).
		Blank	Polling is not used.

Columns	Name	Entry	Explanation
18	Type of code	A or U	ASCII transmission control characters are used. When ASCII is used, the necessary file translation is done for System/34.
		E or blank	EBCDIC transmission control characters are used.
19	Transparency	Y	EBCDIC transparency is used. The data being transferred may contain transmission control characters. Column 18 must be E or blank.
		N or blank	EBCDIC transparency is not used. Zoned decimal numeric or alphameric data is transmitted and received. The data being transferred cannot contain transmission control characters.
20	Switched	Blank	Not a switched network.
		M	Operator using this program makes the connection by dialing the number (manual dial).
		A	This program uses autoanswer.
		B	This program uses manual answer.
21-31	Blank		
32	Location of identification – this station	Blank	Nonswitched network or a switched network where no ID is desired for this station.
		S	Switched network. This station's identification is at the position specified by the symbolic name in columns 33 through 39.
		E	Switched network. The entry in columns 33 through 39 is this station's identification.
33-39	Identification – this station	Alphameric characters	When column 32 contains an E, this entry is the actual identification sequence of this station (from 2 to 15 characters). The station identification must not contain a control character sequence. When column 32 contains an S, this entry is the symbolic name of the location of this station's identification. The symbolic name must not be an array name. If the BSCA file is primary or secondary, this symbolic name must refer to the first element of a table.

Columns	Name	Entry	Explanation
40	Location of identification — remote station	Blank	Nonswitched network or a switched network where no ID is desired for the remote station.
		S	Switched network. The remote station's identification is at the position specified by the symbolic name in columns 41 through 47.
		E	Switched network. The entry in columns 41 through 47 is the remote station's identification.
41-47	Identification — remote station	Alphameric characters	When column 40 contains an E, this entry is the actual identification sequence of the remote station (from 2 to 15 characters). A station identification must not contain a control character sequence. When column 32 contains an S, this entry is the symbolic name of the location or the remote station's identification. This symbolic name must not be an array name. If the BSCA file is a primary or secondary file, this symbolic name must refer to the first element of a table.
48-51		Blank	
52	ITB	Blank	ITB is not used.
		I	Intermediate block check (ITB) is used. ITB can be used only if records are blocked. *Note:* Both ITB and EBCDIC transparency cannot be specified for a BSCA output file.
53-54	Permanent error indicator	Blank	No permanent error indicator is specified. If a permanent error occurs, a system halt occurs. The program cannot be restarted.
		01-99, L1-L9, LR, H1-H9	This indicator can be specified for every BSCA file. If you are using more than one BSCA file, each file can have a permanent error indicator. The indicator does not have to be unique for each file, however.

Columns	Name	Entry	Explanation
55-57	Wait time	Blank	System convention for timeout, 180 seconds, is used.
		1-999	Length of time in seconds (1 through 999) that BSC waits with no messages being sent or received before a permanent error occurs.
58-59	Record available indicator	Blank	No record available indicator is specified. The file cannot be used again.
		01-99, L1-L9, LR, H1-H9	This indicator must be assigned to every BSCA file that is to be reopened. (If a file is used again after end of file has been reached, the file is reopened.)
60	Last file	Blank	This BSCA file cannot be the last input file processed.
		L	This BSCA file is processed after all other input files are processed.
61-62	Polling characters	Blank	This station is not transmitting on a multipoint network.
		Alphameric characters	The polling identification of this station is needed if this station is part of a multipoint network and the BSCA file is a transmit (output) file.
63-64	Addressing characters	Blank	This station is not receiving on a multipoint network.
		Alphameric characters	Addressing identification of this station is needed if this station is a part of a multipoint network and the BSCA file is a receive (input) file. *Note:* Enter polling and addressing characters in System/34 code; the compiler converts the characters to the form required by the code specified in column 18. (Enter uppercase addressing characters, and they are converted to lowercase ASCII characters.)
65-74		Blank	
75-80	Program identification		This space is available for comments.

INPUT SPECIFICATIONS

Columns	Name	Entry	Explanation
1-2	Page	Page number	Entry used to assign a page number to each specification sheet.
3-5	Line	Line number	Entry used to number the specification lines.
6	Form type	I	Identification for an input specification.
7		*	Asterisk in this column identifies this line as a comment line.
7-14	Filename	Filename	Enter a valid RPG II filename for every input, update, and combined file your program uses.
		DS name	Name of a data structure (maximum of 6 characters).
14-16	AND/OR	AND or OR	Enter AND in columns 14 through 16 on the next line of the input specifications sheet if more than three record identification code subfields are needed to identify the record. Enter OR in columns 14 and 15 if either of the codes can be present to identify the record. A maximum of 20 AND or OR lines in any combination can describe the record identifying code. *Note:* AND lines are not allowed with CONSOLE files.
15-16	Sequence	Alphabetic	These two alphabetic characters indicate that record type sequence is not being checked. Alphabetic characters must be used for chained files, demand files (except CONSOLE), WORKSTN file, and look-ahead records. Within a file, record types with an alphabetic sequence entry must be described before record types with a numeric sequence entry.
		Numeric	This two-digit number assigns a special sequence to record types in a file and requests that the record type sequence be checked by the program.
17	Number	Blank	Columns 15 and 16 contain alphabetic characters (record type sequence is not being checked).
		1	Columns 15 and 16 contain numeric characters; only one record of this type is present in each sequenced group.
		N	Columns 15 and 16 contain numeric characters; one or more records of this type can be present in the sequenced group.
18	Option	Blank	Record type must be present.
		O	Optional; record type may or may not be present.
		U	Data structure defined in columns 19 and 20 is a display station local data area. *Notes:* 1. Column 18 is used when record types are being sequence checked (columns 15 and 16 contain a numeric entry). 2. Columns 15 through 18 are not used for a data structure except to define a display station local data area (U in column 18).

Columns	Name	Entry	Explanation
19 and 20	Record identifying indicator	01-99	Record identifying indicator. (CONSOLE files can use indicators 01 through 10 only.)
		L1-L9	Control level indicator used as a record identifying indicator when a record type rather than a control field signals the start of a new control group.
		LR	Last record indicator.
		H1-H9	Halt indicator used as a record identifying indicator when the system checks for a record type that causes an error condition.
		**	Look-ahead fields (not valid with CONSOLE or WORKSTN files.)
		DS	Data structure, which must be the last entries on the input specifications.
21-41	Record identification codes		*Note:* Columns 21 through 41 are divided into three identical subfields that are described separately: (1) columns 21 through 27, (2) columns 28 through 34, and (3) columns 35 through 41. An AND relationship exists between these three fields. These columns are not used for a data structure.
21-24, 28-31, or 35-38	Position	Blank	No record identification code is needed.
		1-4096	Record position of the record identification code.
25, 32, or 39	Not (N)	Blank	Either the record identification code is present in the specified record position, or no record identification code is needed.
		N	Record identification is being used, but this identification code must not be present in the specified record position.
26, 33, or 40	C/Z/D	C	Entire character
		Z	Zone portion of character
		D	Digit portion of character
27, 34, or 41	Character		Any alphabetic character, special character, or digit identifying the character used in the record as the record identifying code.
42		Blank	
43	Packed or binary field	Blank	Input field in zoned decimal format
		P	Input field in packed decimal format
		B	Input field in binary format
			Note: Column 43 is used for disk files only and is invalid for a field in a data structure.
44-47 and 48-51	Field location	Numeric field	Two 1- to 4-digit numbers to identify the beginning of a field (From) and the end of a field (To) in the input record or data structure. The entries are identical for a one-position field.
44-50	Field location	Reserved keyword	For the WORKSTN file information data structure (INFDS), specify keywords (*OPCODE, *RECORD, *SIZE, *STATUS, *MODE, *INP, or *OUT) to define the subfields that are to contain the file related information.

INPUT SPECIFICATIONS (continued)

Columns	Name	Entry	Explanation
52	Decimal position	Blank 0-9	Alphameric field. This column must be blank for a data structure. The number of decimal positions in the numeric field named in columns 53 through 58. This column must contain an entry for numeric fields.
53-58	Field name	Field name	Valid RPG II field name, array name, or array element for each field defined in columns 44 through 51. If an array name is entered, columns 59 through 64 must be blank. PAGE and PAGE1 through PAGE7 are special words.
59-60	Control level	Blank	Field described is not a control field. These columns must be blank for chained files, demand files, WORKSTN files, and data structures.
		L1-L9	Field described on this line is a control field.
61-62	Matching fields	M1-M9	Enter a match value (M1 through M9) to indicate match fields and sequence checking on primary and secondary files with match fields. When you have just one input, update, or combined file with match fields, this entry causes only sequence checking. Match fields are not allowed for demand files, chained files, WORKSTN files, or data structures.
63-64	Field record relation	Blank	Must be blank CONSOLE files.
		01-99	Record identifying indicator assigned to a record type
		L1-L9	Control level indicator
		MR	Matching record indicator
		U1-U8	External indicator
		H1-H9	Halt indicator
65-70	Field indicators	01-99	Field indicator.
		H1-H9	Halt indicator for an error condition in the data. *Note:* An indicator used in these columns is turned on if the condition tested for is true. For numeric fields, more than one condition may be tested at a time, but only the indicator that reflects the result of the test is turned on; the others are turned off. If a field is alphameric, an indicator can be specified only in columns 69 and 70. Field indicators are not valid for a data structure.
71-74		Blank	
75-80	Program identification		This space is available for comments.

CALCULATION SPECIFICATIONS

Columns	Name	Entry	Explanation
1-2	Page	Page number	Entry used to assign a page number to each specification sheet.
3-5	Line´	Line number	Entry used to number the snecification lines.
6	Form type	C	Identification for a calculation specification.
7		*	Asterisk in this column identifies this line as a comment line.
7-8	Control level	Blank	Calculation operation is done at detail time or is part of a subroutine.
		L0	Calculation operation is done at total time (always on).
		L1-L9	Calculation operation is done when the appropriate control break occurs or an indicator is set on.
		LR	Calculation operation is done after the last record is processed or after LR has been set on.
		SR	Calculation operation is part of a subroutine. Blank entry is also valid.
		AN, OR	Indicators specified on this list are either in an AND relationship or in an OR relationship with indicators on the preceding line. A maximum of seven AN, OR, or mixed AN and OR lines is allowed to condition an operation. *Note:* Control level entries must be in the order listed.
9-17	Indicators	Indicators	One to three indicators. Any indicators except 1P and L0 can be used. Columns 9, 12, and 15 may contain blank or N. An AND relationship exists between indicators on a line. Additional lines may be used for entering indicators in columns 9-17. These are in an AND or OR relationship with those on the first line. Enter AN or OR in columns 7 and 8.
18-27	Factor 1		Name of any field that is defined. Alphameric or numeric literal. Subroutine, table or array name, or array element. Date field name (UDATE, UMONTH, UDAY, UYEAR). Special name (PAGE, PAGE1 through PAGE7). Label for a TAG, BEGSR, or ENDSR operation. Field containing work station ID for a display station or two-position literal that is the work station ID for a display station. Figurative constant (*BLANK, *BLANKS, *ZERO, *ZEROS)
28-32	Operation	Operation code	Must be left-justified.
31-32		01-99	Message identification code (MIC) to be displayed from the user message member during SET or KEY operations. (Entries are ignored by the compiler when factor 1 is also present on the same SET or KEY operation.)

Columns	Name	Entry	Explanation
33-42	Factor 2		Name of any field that is defined. Alphameric or numeric literal. Subroutine, table or array name, or array element. Date field name (UDATE, UMONTH, UDAY, UYEAR). Special name (PAGE, PAGE1 through PAGE7). Label for a GOTO or EXSR operation. Filename for a CHAIN, DEBUG, READ, FORCE, SET, ACQ, REL, or NEXT operation. Subroutine name for EXIT operation. Figurative constant (*BLANK, *BLANKS, *ZERO, *ZEROS)
43-48	Result field	ERASE	Entry used to erase buffer for CONSOLE file.
		Field name, table names, array element	These entries hold the results of, or are the object of, the specified in column 28 through 32.
		INxx (xx = any RPG II indicator)	Indicator to be transferred to an external subroutine in an RLABL operation.
		Data structure	Data structure name can be specified as a result field only if operation code in columns 28 through 32 is RLABL or POST.
49-51	Field length	Blank	Field defined elsewhere.
		1-15	Length of a numeric result field.
		1-256	Length of an alphameric result field. The entry must be right-justified.
52	Decimal position	Blank	Alphameric field or numeric field described elsewhere.
		0-9	Number of decimal places in a numeric result field.
53	Half adjust	Blank	Do not half adjust (round) the result field.
		H	Half adjust (round) the result field. Half adjust is allowed only with arithmetic operations.

CALCULATION SPECIFICATIONS (continued)

Columns	Name	Entry	Explanation
54-59	Resulting indicator	01-99 H1-H9 L1-L9 LR OA-OG, OV KA-KN KP-KY U1-U8	Columns 54 through 59 are used to: ● Test the value of the result field after an arithmetic operation. ● Check the outcome of a CHAIN, LOKUP, COMP, TESTB, TESTZ, ACQ, REL, POST, or NEXT operation. Only columns 56 and 57 are valid for ACQ, REL, POST, and NEXT. ● Specify which indicators to set on or set off. ● Indicate end of file (columns 58 and 59) or an exception/error condition (columns 56 and 57) for the READ operation code. ● Allow command keys to be pressed using the SET operation code. ● Test the value of the result field after a KEY operation. ● Condition the files that are to be used by a specific job. ● Test whether the system operator has requested shutdown. *Note:* To enter command key KA press the Cmd key, the 1 key, and then an entry function key. To enter command key KB press the Cmd key, the 2 key, and an entry function key.
60-74	Comments		Any helpful information about this specification line.
75-80	Program identification		This space is available for comments.

OUTPUT SPECIFICATIONS

Columns	Name	Entry	Explanation
1-2	Page	Page number	Entry used to assign a page number to each specification sheet.
3-5	Line	Line number	Entry used to number the specification lines.
6	Form type	O	Identification for an output specification.
7		*	Asterisk in this column identifies this line as a comment line.
7-14	Filename	Filename	Valid RPG II filename for each output, update, and WORKSTN file used by a program. Each filename need be specified only once on the first line describing that file.
14-16	AND/OR relationship	AND OR	AND is entered if output records are in an AND relationship. OR (columns 14 and 15) is entered if output records are in an OR relationship. *Note:* A maximum of 20 AND, OR, or mixed AND and OR lines is allowed to condition an output record.
15	Type	H D T E	Heading records. Detail records. Total records. Exception records (records to be written during calculation time).
16-18	Add a record	ADD	ADD is entered in these columns if records are added to an input, update, or output disk file. An A must also be entered in column 66 of the file description specifications sheet for the file to which a record is added.
	Delete a record	DEL	DEL is entered in these columns if records are deleted from a delete-capable file. The file must be an update file (U in column 15 of the file description specifications).
16	Fetch overflow	F R	Fetch overflow. The overflow routine is fetched when overflow occurs, before the usual time in the cycle. Release a display station from the WORKSTN file.
17-22	Space/skip	See columns 17-18 and 19-22	If these columns are blank, single spacing occurs after each line is printed.
17-18	Space	0-3	Entry made in the appropriate column indicates the number of lines spaced before or after a line is printed.
19-22	Skip	Blank 01-99 A0-A9 B0-B2	No skipping. One of the two-digit numbers listed is entered to indicate the position of the next line printed. All line numbers between are bypassed. Enter the number in the Before or After columns, depending on whether skipping is to occur before or after the line is printed.

Columns	Name	Entry	Explanation
23-31	Output indicators	1 to 3 indicators	Any indicators may be used. Columns 23, 26, 29 may contain blank or N. The letter N preceding an indicator means the output operation is done only if the indicator is not on. An AND relationship exists between indicators on a line. To use additional lines of indicators in an AND or OR relationship, enter AND in columns 14 through 16 or OR in columns 14 and 15 of each additional line (up to 20). Indicators cannot be specified on a field line containing the format name for a WORKSTN file.
32-37	Field name	Field name	One of the following is entered to name every field written out: ● Any field or data structure name defined in this program. ● The special words, PAGE, PAGE1-PAGE7, *PLACE, UDATE, UDAY, UMONTH, and UYEAR. ● A defined table name, array name, or array element. These columns must be blank if a constant is entered on columns 45 through 70 of the line. If an entry is made in columns 32 through 37, columns 7 through 22 must be blank.
38	Edit codes	Edit codes	An edit code is entered in column 38 if needed to: ● Suppress leading zeros for a numeric field. ● Omit a sign from the lower order position of a numeric field. ● Punctuate a numeric field without setting up a special edit word. A table summarizing the edit codes that can be used is printed above columns 45 through 70 on the output specifications sheet.
39	Blank after	Blank	Field is not reset after writing. This column must be blank for look-ahead and UDATE fields.
		B	Alphameric field is reset to blank or numeric field is reset to zero after writing. *Note:* If the field name specified with Blank After is a table name, the element of the table looked up last is blanked or zeroed.

Columns	Name	Entry	Explanation
40-43	End position in output record	Number	Location on the output record of the field or constant written. Enter the number of the position occupied by the rightmost character of the output field. The end position entry must not be greater than the record length.
		K1-K8	End position of the format name for a WORKSTN file.
44	Packed or binary field	Blank	Field is zoned decimal numeric, or alphameric. This column must be blank for *PLACE fields.
		P	Field is packed decimal numeric data.
		B	Field is in binary format.
			Note: Packed and binary fields can be written only on disk; they cannot be printed or displayed.
45-70	Constant or edit word	Constant	Constant must be enclosed in apostrophes.
		Format name	Name of the display screen format used for the WORKSTN file.
		Edit word	Enter an edit word, enclosed in apostrophes, to specify editing of numeric fields. Edit words are not used with edit codes.
71-74		Blank	
75-80	Program identification		This space is available for comments.

OPTION SPECIFICATIONS (AUTO REPORT)

Columns	Name	Entry	Explanation
1-2	Page	Page number	Entry used to assign a page number to each specification sheet.
3-5	Line	Line number	Entry used to number the specification lines.
6	Form Type	U	Identification for an option specification.
7	Source	Blank	The generated source program is not cataloged.
		C	The generated source program is cataloged in a library on disk.
8-24	Source Member Reference	library, member	Identifies the library member to be cataloged. Specify the library name, which can be up to eight characters long, beginning in column 8. Use a comma to separate the library name and the member name, which can also be up to eight characters long.
25-26		Blank	
27	Date Suppress	Blank	Page number and date are included on the first *AUTO page heading line.
		N	Date and page number are not printed on the first *AUTO page heading line.
28	*Suppress	Blank	Asterisks are generated for total output lines.
		N	Asterisks are not generated for total output lines.
29		Blank	
30	List Options	Blank	Source program listing, headings, and diagnostics are printed, and a source program is produced if no severe errors are found.
		B	The program listing is not printed; however, a source program is produced.
		P	A partial program listing is printed. This listing includes appropriate headings and diagnostics.
31-74		Blank	
75-80	Program Identification		This space is available for comments.

OPERATION CODES

Operation Code	Control Level Indicators	Conditioning Indicators	Factor 1	Factor 2	Result Field	Resulting Indicators		
	Columns					Columns		
	7-8	9-17				54-55	56-57	58-59
ACQ	O	O	R	R			O	
ADD	O	O	O	R	R	O	O	O
BEGSR	SR or blank		R					
BITOF	O	O		R	R			
BITON	O	O		R	R			
CHAIN	O	O	R	R		O		
COMP	O	O	R	R		O[3]	O[3]	O[3]
DEBUG	O	O	O	R	O			
DIV	O	O	O	R	R	O	O	O
ENDSR	SR or blank		O	O	O			
EXCPT	O	O						
EXIT	O	O		R				
EXSR	O	O		R				
FORCE		O		R				
GOTO	O	O		R				
KEYnn	O	O	O		R	O	O	O
LOKUP (Array)	O	O	R	R		O[4]	O[4]	O[4]
LOKUP (Table)	O	O	R	R	O	O[4]	O[4]	O[4]
MHHZO	O	O		R	R			
MHLZO	O	O		R	R			
MLHZO	O	O		R	R			
MLLZO	O	O		R	R			

OPERATION CODES (continued)

Operation Code	Control Level Indicators	Conditioning Indicators	Factor 1	Factor 2	Result Field	Resulting Indicators		
	Columns					Columns		
	7-8	9-17				54-55	56-57	58-59
MOVE	O	O		R	R			
MOVEA	O	O		R	R			
MOVEL	O	O		R	R			
MULT	O	O	O	R	R	O	O	O
MVR	O	O			R	O	O	O
NEXT	O	O	R	R			O	
POST	O	O	R		R		O	
READ	O	O		R			O[2]	O
REL	O	O	R	R			O	
RLABL					R			
SETnn[1]	O	O	O	O		O	O	O
SETOF	O	O				O[3]	O[3]	O[3]
SETON	O	O				O[3]	O[3]	O[3]
SETLL	O	O	R	R				
SHTDN	O	O				R		
SORTA	O	O		R				
SQRT	O	O		R	R			
SUB	O	O	O	R	R	O	O	O
TAG	O		R					
TESTB	O	O		R	R	O[3]	O[4]	O[4]
TESTZ	O	O			R	O[3]	O[4]	O[4]
TIME	O	O			R			
XFOOT	O	O		R	R	O	O	O
Z-ADD	O	O		R	R	O	O	O
Z-SUB	O	O		R	R	O	O	O

[1] The nn entries in columns 31 and 32 are for message indicator numbers. If the result field of a SET operation contains the keyword ERASE, factor 2 must contain the name of the CONSOLE file. Otherwise, factor 2 and the result field must be blank.

[2] Columns 56 and 57 can contain an indicator when the READ operation is used with a WORKSTN device.

[3] At least one resulting indicator must be specified in columns 54 through 59.

[4] At least one resulting indicator must be specified in columns 54 through 59, but no more than two can be used.

Fields without entries must be blank.

O	=	Optional
R	=	Required
SR	=	The only allowable nonblank characters in columns 7 and 8 for the BEGSR and ENDSR operation codes.

VALID INDICATORS

Indicators	File Description Specifications		Input Specifications				Calculation Specifications			Output Specifications
	Overflow (33-34)	File Conditioning (71-72)	Record Identifying[1] (19-20)	Control Level (59-60)	Field Record Relation[1] (63-64)	Field (65-70)	Control Level (7-8)	Conditioning (9-17)	Resulting (54-59)	Conditioning (23-31)
01-99			X		X	X		X	X	X
H1-H9			X		X	X		X	X	X
1P										X[3]
MR					X[2]			X		X
OA-OG, OV	X							X	X	X[4]
L0							X			X
L1-L9			X	X	X[2]		X	X	X	X
LR			X				X	X	X	X
U1-U8		X[5]			X			X	X	X
KA-KN, KP-KY								X	X[6]	X

Note: X denotes the indicators that can be used.

[1] Not valid on look-ahead fields.
[2] When field named is not a match field or a control field.
[3] Only for detail or heading lines.
[4] Cannot condition an exception line, but can condition fields within the exception record.
[5] Not valid for table input files.
[6] Valid for SET, KEY, and SETOF operations only.

SUMMARY OF INDICATORS

	Indicators	Where Located	Where Normally Used as Conditioning Indicators	Normally Turned On		Normally Turned Off	
				By	When	By	When
Location on Specification Sheets	Record identifying indicator	Input sheet cols 19-20	Input: field record relation (cols 63-64) Calculation: indicators (cols 9-17) Output: output indicators (cols 23-31)	Record identification	Before total-time calculations	Different record type	Before total-time calculations
	Field indicators: Plus/minus	Input sheet Cols 65-68: Numeric data only	Calculation: indicators (cols 9-17)	Data field with plus or minus balance	Before detail-time calculations	Data field without a plus or minus balance	
	Zero/blank	Cols 69-70	Output: output indicators (cols 23-31)	Data field with zero or blank balance	Before detail-time calculations when field is zero or blank	Data field without a zero or blank balance	Before detail-time calculations
	Control level (L1-L9)	Input sheet cols 59-60	Calculation: control level (cols 7-8) Calculation: indicators (cols 9-17) Output: output indicators (cols 23-31)	Control break of that or higher level	Before total-time calculations	A control field with the same contents as the control field of previous record	After detail-time output
	Matching records (MR)—based on matching fields	Input sheet cols 61-62: M1-M9 control MR	Calculation: indicators (cols 9-17) Output: output indicators (cols 23-31)	Matching of primary with any secondary record	Before detail-time calculations	Nonmatch between primary and other records	Before detail-time calculations
	Calculation resulting indicators	Calculation sheet cols 54-59	Calculation: indicators (cols 9-17) Output: output indicators (cols 23-31)			Specified resulting indicators are set off prior to the execution of the calculation	Immediately
	Arith ops: Plus Minus Zero			Plus result Minus result Field contents Zero	Immediately when the specified condition is met upon execution of the operation		
	COMP: High Low Equal			Factor 1 > Factor 2 Factor 1 < Factor 2 Factor 1 = Factor 2			

SUMMARY OF INDICATORS (continued)

Indicators			Where Located	Where Normally Used as Conditioning Indicators	Normally Turned On		Normally Turned Off	
					By	When	By	When
Location on Specification Sheets	TESTZ	Plus			Presence of a C zone			
		Minus			Presence of a D zone			
		Blank			Any zone other than a C or D zone			
	TESTB	Plus			Each bit specified by factor 2 is off in the result field			
		Minus			Bits specified by factor 2 are of a mixed status in the result field			
		Equal			Each bit specified by factor 2 is on in the result field			
	LOKUP	High			Factor 1 < Factor 2			
		Low			Factor 1 > Factor 2			
		Equal			Factor 1 = Factor 2			
	KEY	Plus			Plus result	Immediately when the specified condition is met after the field is keyed	Failure to satisfy the assigned condition when the field is keyed	Immediately
		Minus			Minus result			
		Zero			Field contents zero or blank			
		Blank						
	ACQ (exception/ error, cols 56-57)				Exception/ error condition	Immediately upon execution of operation if exception/ error condition occurred		Immediately preceding execution of operation
	CHAIN (no record found, cols 54-55)				Record specified in factor 1 not found in file	Immediately upon execution of operation if specified condition exists	CHAIN operation code	Immediately preceding execution of operation
	NEXT (exception/ error, cols 56-57)				Exception/ error condition	Immediately upon execution of operation if exception/ error condition occurred		Immediately preceding execution of operation

SUMMARY OF INDICATORS (continued)

Indicators		Where Located	Where Normally Used as Conditioning Indicators	Normally Turned On		Normally Turned Off	
				By	When	By	When
Location on Specification Sheets	READ: End of file			End of file	Immediately upon execution of operation if end of file occurred	Programmer	By execution of the SETOF operation or through use as resulting indicator of another operation
	READ: Except/ error, cols 56-57			Exception/ error condition	Immediately upon execution of operation if exception/ error occurred		Immediately preceding execution of operation
	REL (exception/ error, cols 56-57)			Exception/ error condition	Immediately upon execution of operation if exception/ error occurred		Immediately preceding execution of operation
	SETOF (cols 54-59)						Immediately upon execution of operation
	SETON (cols 54-59)				Immediately upon execution of operation		
	SETnn			Operator pressing specified command key			Immediately preceding execution of operation
	SHTDN (cols 54-55)			System operator's request to shut down system	Immediately upon execution of operation if specified condition exists		Immediately preceding execution of operation

SUMMARY OF INDICATORS (continued)

Indicators		Where Normally Specified to be Turned On or Off	When Turned On by Program Itself	When Turned Off by Program Itself
Name/Number	L1-L9 (control level)	Control level: cols 59-60—input sheet	Before total time upon control break	After each detail-time output
	L0 (level zero)	Nowhere	Always on	Never
	LR (last record total)	Nowhere	Before total time following last data record (after / * b̸)	At the start of the program
	1P (first page)	Nowhere	At beginning of program execution	After first detail-time output
	OA-OG, OV (overflow)	Nowhere	When end of page is reached	After next detail-time output unless fetch overflow is specified
	H1-H9 (halt)	Field and resulting indicators	Never, but if on at detail-time output, halts system thereafter	When system is restarted after halt
	01-99 (general)	Field and resulting indicators	During calculation	During calculation
	KA-KN, KP-KY	Resulting indicators	During calculation or by display station operator at input	During calculation or by display station operator at input
	U1-U8	External or resulting indicators	During calculation	During calculation

DISPLAY SCREEN FORMAT SPECIFICATIONS

S Specifications

Columns	Name	Entry	Explanation
1-5	Sequence number	Line number	Entry used to number the specification lines.
6	Form type	S	Identification for an S specification.
7		*	Asterisk in this column identifies this line as a comment line.
7-14	Format name	Display screen format name	Name of the display screen format that the $SFGR utility program creates from the S and D specifications.
15-16		Blank	
17-18	Start line number	01-24	Number of the line at which the display begins.
		V (column 17)	Start line number is determined by the user program.
19-20	Number of lines to clear	00-24	Number of lines to clear, including and following the starting line. The specified number of lines are cleared, beginning with the start line specified in columns 17 and 18.
21	Lowercase	Y	With the Shift key, operators key uppercase characters. Without the Shift key, operators key lowercase characters.
		N or blank	Operators key uppercase characters only.
22	Return input	Y or blank	Input fields on this display are returned to the user program, even if the operator enters no data.
		N	Input fields on this display are not returned to the user program unless the operator enters data in one or more of the fields. Then all input fields are returned to the program.
23-24		Blank	
25-26	Sound alarm	Y (column 25)	The alarm sounds when this display appears.
		N (column 25) or blank	The alarm does not sound when this display appears.
		01-99	The alarm sounds when this display appears only if the specified indicator is on.

S Specifications (continued)

Columns	Name	Entry	Explanation
27	Enable function keys	Y	The function control keys identified by numbers in the key mask entry (columns 64 through 79) are enabled (allowed). If the key mask entry contains no numbers, all function control keys are disabled.
		N	The function control keys identified by numbers in the key mask entry are disabled (not allowed). If the key mask entry contains no numbers, all function control keys are enabled. If the operator presses a disabled function control key, an error message is displayed. The operator can then press the Error Reset key, followed by the correct function key.
		R	The function control key mask that is active for the display station is retained when this format is displayed.
		Blank	All function control keys are enabled. In this case, the key mask entry must not contain any numbers. *Note:* Function control keys that are not masked off and that are not supported by the program cause an error message to be displayed, which indicates that an invalid key was pressed.
28	Enable command keys	Y	The command keys identified by alphabetic characters in the key mask entry (columns 64 through 79) are enabled (allowed). If the key mask entry contains no alphabetic characters, all command keys are disabled.
		N	The command keys identified by alphabetic characters in the key mask entry are disabled (not allowed). If the key mask entry contains no alphabetic characters, all command keys are enabled. If the operator presses a disabled command key, an error message is displayed. The operator can then press the Error Reset key, followed by the correct command key.
		R	The command key mask that is active for the display station is retained when this format is displayed.
		Blank	All command keys are.enabled. In this case, the key mask entry must not contain any alphabetic characters. If a command key is pressed, the corresponding indicator (KA through KN, KP through KY) is set on in the RPG II program.

S Specifications (continued)

Columns	Name	Entry	Explanation
29-30	Blink cursor	Y (column 29)	The cursor blinks when this display appears.
		N (column 29) or blank	The cursor does not blink.
		01-99	The cursor blinks only if the specified indicator is on.
31-32	Erase input fields	Y (column 31)	All unprotected input fields on the screen are erased, the keyboard is unlocked, and no output occurs. All D specifications are ignored. The use of Y is not recommended.
		N (column 31) or blank	The input fields are not erased.
		01-99	All unprotected input fields on the screen are erased and the keyboard is unlocked if the specified indicator is on.
33-34	Override fields	Y (column 33)	An override operation is performed. The use of Y is not recommended.
		N (column 33) or blank	The operation is not an override operation.
		01-99	An override operation is performed if the specified indicator is on. An override operation allows the screen to remain unchanged except for those fields that have indicators specified for them in columns 23 and 24 of the D specification, and those indicators are on. See *Special Display Format Considerations* in Chapter 13 for a more detailed description of an override operation. The record displayed by RPG II is exactly the same whether or not override is specified when the indicator in columns 23 and 24 of the D specification is on.
35-36	Suppress input	Y (column 35)	No input is returned to the user program until a format is displayed with suppress input specified as N or with the specified indicator off.
		N (column 35) or blank	Input is returned to the user program.
		01-99	Input to the user program is suppressed if the specified indicator is on.
37-63		Blank	

S Specifications (continued)

Columns	Name	Entry	Explanation
64-79	Key mask		The key mask is a string of numbers and/or alphabetic characters that identify keys to be enabled or disabled when this format is displayed. The key mask must begin in column 64 and cannot contain embedded blanks. The numbers and alphabetic characters can be intermixed.

Numbers in the key mask identify function control keys:

Number	Function Control Key
1	Print
2	ROLL↑
3	ROLL↓
4	Clear
5	Help
6	Record Backspace

Alphabetic characters in the key mask identify command keys:

Alphabetic Character	Command Keys
A	1
B	2
C	3
D	4
E	5
F	6
G	7
H	8
I	9
J	10
K	11
L	12
M	13
N	14
P	15
Q	16
R	17
S	18
T	19
U	20
V	21
W	22
X	23
Y	24

D Specifications

Columns	Name	Entry	Explanation
1-5	Sequence number	Line number	Entry used to number the specification line.
6	Form type	D	Identification for a D specification.
7		*	Asterisk in this column identifies this line as a comment line.
7-12	Field name	Field name	Name of an input field, output field, or output/input field.
		Blank	This D specification line specifies only constant data.
13-14		Blank	
15-18	Field length	1-1919	The entry must be right-justified, but leading zeros are not required.
19-20	Line number	01-nn	Relative line number on which data appears. The actual line number is start line number (column 17 and 18 on the S specification) plus this line number, minus one. nn (maximum) = 24 - starting line number
21-22	Horizontal position	01-80	Column number of the first position of the field. Columns 19 through 22 cannot be 0101.
23-24	Output data	Y (column 23)	If constant data or a message identification code is also specified in columns 57 through 79, that constant data or the specified message is displayed in the field. If no constant data or message identification code is specified in columns 57 through 79, data from the user program output record is displayed.
		N (column 23) or blank	The field is not an output field.
		01-99	If the specified indicator is on when the format is displayed, data supplied by the user program is displayed in the field. If the specified indicator is off when the format is displayed, data specified in columns 57 through 79 is displayed. If no data is specified in columns 57 through 79, blanks are displayed. If the user program performs an override operation and the specified indicator is on, data supplied by the user program is displayed in the field. See *Special Display Format Considerations* in Chapter 13 for a description of an override operation. If the user program performs an override operation and the specified indicator is off, the field is unchanged.
25		Blank	

Columns	Name	Entry	Explanation
26	Input allowed	Y	The operator can enter information into the field from the keyboard.
		N or blank	The operator cannot enter information into the field from the keyboard.
27	Data type	A	The field can contain only alphabetic data.
		B or blank	The field can contain only alphameric data.
		K	The field can contain Katakana characters.
		N	The field can contain only numeric data. Commas, a period, a plus sign, or a minus sign can also be entered in this field. *Note:* If special characters are entered in an N-type field, the data read by the RPG program may not be as expected. The program uses only the digit portion of characters entered in an N-type field. The zone portion is forced to hex F, except for the sign position (see *Zoned Decimal Format* in Chapter 7).
		S	The field can contain only signed numeric data; the last position of the field is reserved for a sign. Only decimal digits (0 through 9) can be entered in the field. The field can be from 2 to 16 characters long.
28	Mandatory fill	Y	Operators must key all or key none of the field.
		N or blank	Operators can key all, none, or part of the input field. *Note:* Mandatory fill and adjust/fill (column 31) cannot be specified for the same field.
29	Mandatory entry	Y	Operators must enter at least one character or a blank in the input field.
		N or blank	Operators can bypass the input field.
30	Self check	T	The input field is a modulus 10 self-check field.
		E	The input field is a modulus 11 self-check field.
		Blank	The input field is not a self-check field.

D Specifications (continued)

Columns	Name	Entry	Explanation
31	Adjust/fill	Z	Information entered into the field is right-justified, and unused positions are filled with zeros.
		B	Information entered into the field is right-justified, and unused positions are filled with blanks.
		Blank	For signed numeric fields, the information entered in the field is right-justified and blank fill is assumed. For alphameric fields, the information entered in the field is unchanged.
			Note: Mandatory fill (column 28) and adjust/fill cannot be specified for the same field.

D Specifications (continued)

Columns	Name	Entry	Explanation
32-33	Position cursor	Y (column 32)	Cursor appears at the first position of the input field when this format is displayed.
		N (column 32) or blank	Cursor does not appear at the first position of the input field.
		01-99	Cursor appears at the first position of the input field only if the specified indicator is on.
34	Enable Dup	Y	When the Dup key is pressed, the position of the cursor and the remainder of the field are filled with the duplicate character value (hex 1C), which is displayed as an asterisk (*). The duplicate characters must be processed by the user porgram.
		N or blank	The Dup key has no effect in the field.
35	Controlled field exit	Y	Cursor does not leave the input field until the operator presses a field exit key (Field Adv, Enter/Rec Adv, Field Exit, Field +, Field - [if the field is a signed-numeric field], Field Backspace, Home, Erase Input, or Dup).
		N or blank	Cursor automatically skips to the next unprotected field when the operator keys the last position of the field.
36	Auto record advance	Y	The input fields on the screen automatically return to the user program when one of the following occurs: ● The operator enters the last character in the field. ● The cursor is in the input field and the operator presses the Field Exit, Field +, or Field - key (if the field is a signed-numeric field).
		N or blank	Automatic record advance does not occur for this field.
37-38	Protect field	Y (column 37)	The cursor skips the field.
		N (column 37) or blank	The cursor does not skip the field.
		01-99	The cursor skips the field if the specified indicator is on. *Note:* If an override operation is used, this indicator is ignored.

D Specifications (continued)

Columns	Name	Entry	Explanation
39-40	High intensity	Y (column 39)	The field is displayed with high intensity.
		N (column 39) or blank	The field is displayed with normal intensity.
		01-99	The field is displayed with high intensity if the specified indicator is on. *Note:* High intensity, reverse image (columns 45-46), and underline (columns 47-48) cannot all be specified for the same field at the same time.
41-42	Blink field	Y (column 41)	The field blinks.
		N (column 41) or blank	The field does not blink.
		01-99	The field blinks if the specified indicator is on when the format is displayed.
43-44	Nondisplay	Y (column 43)	The field is nondisplay; that is, information in the field when the format is displayed or information entered into the field by the operator is not visible on the screen.
		N (column 43) or blank	The information in the field is displayed.
		01-99	The field is a nondisplay field if the specified indicator is on when the format is displayed.
45-46	Reverse image	Y (column 45)	The characters in the field appear as dark characters on a light background.
		N (column 45) or blank	The characters in the field appear as light characters on a dark background.
		01-99	The characters in the field appear as dark characters on a light background if the specified indicator is on when the format is displayed.
47-48	Underline	Y (column 47)	The field is underlined.
		N (column 47) or blank	The field is not underlined.
		01-99	The field is underlined if the specified indicator is on.

D Specifications (continued)

Columns	Name	Entry	Explanation
49	Column separators	Y	Each character position in the field is preceded by a column separator (a vertical line). The column separator does not require an additional character position.
		N or blank	Column separators are not used.
50-55		Blank	
56	Constant type	C	The constant information in columns 57 through 79 is to be displayed in the output field. C is required only if columns 57 through 79 are blank and you want to display all blanks in the field. C is invalid if an indicator is specified in columns 23 and 24.
		M	A message identification code and a message member identifier are entered in columns 57 through 79.
		Blank	If columns 57 through 79 contain constant information, that information is displayed. If columns 57 through 79 are blank, then information from the program output record area is displayed.
57-79	Constant data		This field specifies the information to be placed in an output or output/input field when the format is generated. If information is to be placed in the field, columns 57 through 79 should contain one of the following: ● The actual information to be displayed. ● A four-digit message identification code in columns 57 through 60 and a 2-character message member identifier in columns 61 and 62. *Notes:* 1. If columns 57 through 79 are blank and the field is an output field (Y in column 23), then information from the program output record is displayed. 2. If a message identification code is specified in columns 57 through 79, then only 6 bytes need be reserved for the field in the program output record area.
80	Continuation	X	If more than 23 characters of data are required, an X in column 80 indicates that the record is continued. Use columns 7 through 79 of the following record for the continued constant data. *Note:* A comment cannot follow a record with X in column 80.

Appendix B. Printed Messages

This appendix describes the printed messages generated by the RPG II compiler. The compiler prints a message when an error is detected during compilation of RPG II source specifications. No operator action is required when these errors occur. The messages are for use by the programmer.

Each message includes:

- Program identification code (RPG). Auto report messages are printed with the identification NOTE instead of RPG.
- A four-digit message identification code.
- Message text.
- Severity code:
  - W (Warning) Warning that an abnormal condition exists. Corrective action is required only if the condition is unintentional. The compilation is completed, and the program can be executed with warning errors.
  - T (Terminal) An error condition exists that requires corrective action before the system can compile the program. The program cannot be executed with terminal errors.
- The specification type of the error causing the message to be issued.

An explanation of the message, when included, describes the message in more detail, describes any action taken by the system, and suggests a response to correct the error condition.

RPG II MESSAGES

RPG–0002 INVALID ENTRY IN COLUMN 10, ASSUME BLANK.

Severity: Warning

Specification Type: H

Explanation: The object output entry in column 10 is not D or blank. Blank is assumed.

RPG–0003 INVALID LISTING OPTION IN COLUMN 11, ASSUME BLANK.

Severity: Warning

Specification Type: H

Explanation: The listing options entry in column 11 is not B, P, or blank. Blank is assumed. Therefore, a source program and the object program are produced.

RPG–0004 INVALID OR BLANK STORAGE SIZE TO EXECUTE ENTRY IN COLUMN 12-14, ROUNDED UP TO 2K MULTIPLE OR REGION SIZE ASSUMED.

Severity: Warning

Specification Type: H

Explanation: (1) Columns 12 through 14 are blank, or (2) they contain an entry greater than 64K, or (3) the entry is not a multiple of 2K. The size of the region used for compiling is assumed, or, if item (3) is the error, the entry is rounded up.

RPG–0005 INVALID DEBUG CODE IN COLUMN 15, ASSUME BLANK:

Severity: Warning

Specification Type: H

Explanation: The debug code in column 15 is neither 1 nor blank.

RPG–0006 CONTROL SPECIFICATION WAS PREVIOUSLY DEFINED. CURRENT SPECIFICATION IS IGNORED.

Severity: Warning

Specification Type: H

Explanation: Only one control specification per program is allowed. The specification is ignored and the job continues.

RPG–0007 INVALID ENTRY IN COLUMN 57, ASSUME TRANSPARENT LITERAL SUPPORT.

Severity: Warning

Specification Type: H

Explanation: A character other than a 1 or a blank was found in column 57. Column 57 is assumed to contain 1.

RPG–0008 INVALID ENTRY IN COLUMN 37, ASSUME BLANK.

Severity: Warning

Specification Type: H

Explanation: The inquiry entry in column 37 of the control specification is not I, B, or blank.

RPG–0010 INVALID ENTRY IN COLUMN 18, ASSUME BLANK.

Severity: Warning

Specification Type: H

Explanation: Valid entries for column 18 are any characters except those having special significance in edit words or edit codes. The following are not valid:

0 (zero)	& (ampersand)
* (asterisk)	. (decimal point)
, (comma)	– (minus)
C (letter C)	R (letter R)

RPG–0011 INVALID ENTRY IN COLUMNS 16-17, ASSUME BLANKS.

Severity: Warning

Specification Type: H

Explanation: These columns must be left blank.

RPG–0012 INVALID INVERTED PRINT ENTRY IN COLUMN 21, ASSUME BLANK.

Severity: Warning

Specification Type: H

Explanation: The inverted print entry in column 21 of the control specification is not I, D, J, or blank.

RPG–0013 INVALID ENTRIES IN COLUMNS 22-25, ASSUME BLANKS.

Severity: Warning

Specification Type: H

Explanation: These columns must be left blank.

RPG–0014 INVALID ALTERNATE COLLATING SEQUENCE ENTRY IN COLUMN 26, ASSUME S.

Severity: Warning

Specification Type: H

Explanation: The alternate collating sequence entry in column 26 of the control specification is neither blank nor S. The S entry alters the normal collating sequence.

RPG–0015 INVALID ENTRIES IN COLUMNS 27-36 AND/OR 38-40, ASSUME BLANK.

Severity: Warning

Specification Type: H

Explanation: These columns must be left blank.

RPG–0016 INVALID 1P OUTPUT REPEAT ENTRY IN COLUMN 41, ASSUME 1.

Severity: Warning

Specification Type: H

Explanation: Column 41 (1P forms position) of the control specification is neither 1 nor blank.

RPG–0017 INVALID ENTRY IN COLUMN 42, ASSUME BLANK.

Severity: Warning

Specification Type: H

Explanation: This column must be left blank.

RPG–0018 INVALID FILE TRANSLATION ENTRY IN COLUMN 43, ASSUME F.

Severity: Warning

Specification Type: H

Explanation: The file translation entry in column 43 is neither F nor blank.

RPG–0019 INVALID ENTRY IN COLUMN 44, ASSUME BLANK.

Severity: Warning

Specification Type: H

Explanation: Column 44 must be left blank.

RPG–0020 INVALID NON-PRINTABLE CHARACTER ENTRY IN COLUMN 45, ASSUME 1.

Severity: Warning

Specification Type: H

Explanation: This column must contain a 1 or blank.

RPG–0021 INVALID ENTRIES IN COLUMNS 46-47, 49-51, 54-56, OR 58-74, ASSUME BLANKS.

Severity: Warning

Specification Type: H

Explanation: These columns must be blank.

RPG–0022 INVALID ENTRY IN COLUMN 6 OR SPECIFICATION TYPE OUT OF SEQUENCE.

Severity: Terminal

Specification Type: H, F, E, L, T, I, C, or O

Explanation: Valid entries for column 6 are H, F, E, L, T, I, C, or O, in the order listed. The job is terminated and the entire specification is ignored.

RPG–0023 INVALID OR BLANK FILENAME IN COLUMNS 7-14.

Severity: Terminal

Specification Type: F, I, L, T, or O

Explanation: Filename specified in columns 7 through 14 is invalid. The job is terminated and the entire specification is ignored.

RPG–0024 FILENAME PREVIOUSLY DEFINED IN COLUMNS 7-14.

Severity: Terminal

Specification Type: F

Explanation: The filename is not unique. The job is terminated and the entire specification line is ignored.

RPG–0025 INVALID DEVICE NAME IN COLUMNS 40-46, ASSUME DISK.

Severity: Terminal

Specification Type: F

Explanation: The entry in columns 40 through 46 is not a valid device name. DISK is assumed, but the job is terminated.

RPG–0026 INVALID OR BLANK FILE TYPE ENTRY IN COLUMN 15, ASSUME DEFAULT FOR DEVICE.

Severity: Terminal

Specification Type: F

Explanation: The file type entry in column 15 is not I, O, U, or C or a record address file (R in column 16) is specified and the file type is not I. O is assumed for files assigned to CRT or PRINTER; I is assumed for files assigned to KEYBORD, BSCA, or CONSOLE; U is assumed for files assigned to DISK; C is assumed for files assigned to SPECIAL or WORKSTN. The job is terminated.

RPG–0027 POSITION 19 IN CONTROL SPECIFICATION NOT BLANK, M, D, OR Y, ASSUME M IF POSITION 21 BLANK.

Severity: Warning

Specification Type: H

Explanation: Position 19 should be M, D, Y, or blank. Assume M if position 21 is blank, or assume D if position 21 is I or J.

RPG–0028 FILE DESIGNATION IN COLUMN 16 IS INVALID FOR EITHER FILE TYPE OR DEVICE, ASSUME SECONDARY.

Severity: Warning

Specification Type: F

Explanation: The entry in column 16 is not valid for an input or an update file. S is assumed and the job continues.

RPG–0029 INVALID ENTRY IN COLUMNS 52-53, ASSUME 32.

Severity: Warning

Specification Type: H

Explanation: An entry other than blank or 01-32 was entered in columns 52 and 53, or 00 was entered in columns 52 and 53 and *NONE was not specified on an FMTS continuation line.

RPG–0030 FILE DESIGNATION ENTRY IN COLUMN 16 INVALID FOR OUTPUT FILE, ASSUME BLANK.

Severity: Warning

Specification Type: F

Explanation: Column 16 must be blank for output files (O in column 15).

RPG–0032 NO PRIMARY FILE SPECIFIED IN COLUMN 16, ASSUME FIRST SECONDARY AS PRIMARY.

Severity: Warning

Specification Type: F

Explanation: If a primary file is not specified (P in column 16) in the file description specifications and one or more secondary files are specified, the first secondary file is assigned as the primary file. When no primary or secondary files are assigned, you must provide an exit for your program by turning on the LR indicator.

RPG–0034 MULTIPLE PRIMARY FILES DEFINED IN COLUMN 16, ASSUME SECONDARY.

Severity: Warning

Specification Type: F

Explanation: More than one primary file (P in column 16) was defined in your file description specifications. All primary files except the first one are assumed to be secondary.

RPG–0036 INVALID END OF FILE ENTRY IN COLUMN 17, ASSUME E FOR INPUT FILE TYPE WITHOUT RANDOM PROCESSING.

Severity: Warning

Specification Type: F

Explanation: The entry in column 17 of the file description specifications is neither E nor blank. E is assumed for input files not processed randomly; blank is assumed for all other files.

RPG–0037 INVALID FILE FORMAT ENTRY IN COLUMN 19.

Severity: Warning

Specification Type: F

Explanation: The file format entry in column 19 of your file description specification is not F. F is assumed.

RPG–0038 END OF FILE ENTRY IN COLUMN 17 INVALID FOR FILE TYPE.

Severity: Warning

Specification Type: F

Explanation: Column 18 must be blank for output, demand, and table files. Blank is assumed.

RPG–0039 INVALID SEQUENCE ENTRY IN COLUMN 18, ASSUME PREVIOUS ENTRY.

Severity: Warning

Specification Type: F

Explanation: The sequence entry in column 18 is not A, D, or blank. The entry in column 18 from the previous line is assumed.

RPG–0040 SEQUENCE ENTRY IN COLUMN 18 INVALID FOR TYPE OF FILE OR MODE OF PROCESSING, ASSUME BLANK.

Severity: Warning

Specification Type: F

Explanation: Column 18 must be blank for demand files, output files, record address files, and for any files processed randomly.

RPG–0041 INVALID RECORD LENGTH ENTRY IN COLUMNS 24-27, ASSUME DEFAULT FOR DEVICE.

Severity: Warning

Specification Type: F

Explanation: Incorrect record length was specified in columns 24 through 27. The maximum record length for the device is assumed, except DISK which is assumed to be 256.

RPG–0042 INVALID BLOCK LENGTH ENTRIES IN COLUMNS 20-23, ASSUME RECORD LENGTH.

Severity: Warning

Specification Type: F

Explanation: The block length entry in columns 20 through 23 is neither equal to nor a multiple of the record length specified in columns 24 through 27.

RPG–0043 DUAL I/O ENTRY IN COLUMN 32 INVALID FOR TYPE OF FILE OR MODE OF PROCESSING, ASSUME BLANK.

Severity: Warning

Specification Type: F

Explanation: Dual I/O (1-9 in column 32) cannot be specified for demand, table, and update files, or for any file processed randomly. Dual I/O also cannot be specified if shared I/O (column 48 of the control specifications) has been specified.

RPG–0044 INVALID ENTRY IN COLUMN 32, ASSUME BLANK.

Severity: Warning

Specification Type: F

Explanation: The entry in column 32 was not 1-9, I, T, or blank.

RPG–0045 OVERFLOW INDR IN COLS 33-34 PREVIOUSLY DEFINED.

Severity: Terminal

Specification Type: F

RPG–0046 INVALID OVERFLOW INDICATOR IN COLUMNS 33-34, ASSUME BLANK.

Severity: Terminal

Specification Type: F

Explanation: The overflow indicator entry in columns 33 and 34 was not OA-OG, or OV. Blank is assumed, but the job is terminated.

RPG–0047 OVERFLOW INDICATOR IN COLUMNS 33-34 INVALID FOR DEVICE, ASSUME BLANK.

Severity: Warning

Specification Type: F

Explanation: The overflow indicator in columns 33 and 34 was not assigned to a printer file.

RPG–0048 INVALID OR BLANK EXTENSION CODE ENTRY IN COLUMN 39 FOR TABLE FILE OR RECORD ADDRESS FILE, ASSUME E.

Severity: Warning

Specification Type: F

RPG–0049 INVALID EXTENSION CODE ENTRY IN COLUMN 39, ASSUME L.

Severity: Warning

Specification Type: F

Explanation: The entry in column 39 is neither L nor blank for output files assigned to the printer. L is assumed and the job continues.

RPG–0051 EXTENSION CODE ENTRY IN COLUMN 39 INVALID WITH DEVICE, OR WITH P, S, C, OR D IN COLUMN 16, ASSUME BLANK.

Severity: Warning

Specification Type: F

RPG–0052 DEVICE IN COLUMNS 40-46 PREVIOUSLY ASSIGNED TO OUTPUT OR, NON-TABLE INPUT FILE OR MORE THAN EIGHT PRINTER FILES DEFINED.

Severity: Terminal

Specification Type: F

Explanation: The device name in columns 40 through 46 was assigned to more than one output or non-table input file. The job is terminated and the entire specification may cause other errors to be generated.

RPG–0053 INVALID ENTRIES IN COLUMNS 47-52, ASSUME BLANKS.

Severity: Warning

Specification Type: F

Explanation: These columns must be left blank.

RPG–0055 FILE CONDITIONING ENTRIES IN COLUMNS 71-72 INVALID FOR TABLE FILES OR KEYBORD, ASSUME BLANK.

Severity: Warning

Specification Type: F

Explanation: Columns 71 and 72 must be left blank for table files, because table files cannot be conditioned by U1-U8.

RPG–0057 INVALID FILE CONDITIONING ENTRIES IN COLUMNS 71-72.

Severity: Terminal

Specification Type: F

Explanation: Columns 71 and 72 of your file description specification are not blank nor do they contain one of the external indicators (U1-U8).

RPG–0058 INVALID ENTRIES IN COLUMNS 67, AND/OR 73-74, ASSUME BLANK.

Severity: Warning

Specification Type: F

Explanation: Columns 67, and 73 and 74 must be left blank.

RPG–0060 INVALID ENTRY IN COLUMN 48, ASSUME BLANK.

Severity: Warning

Specification Type: H

Explanation: To indicate shared input/output buffer areas for disk files, enter a 1 in column 48; otherwise, leave column 48 blank.

RPG–0061 INVALID ENTRIES IN COLUMNS 7-10, ASSUME BLANK.

Severity: Warning

Specification Type: E

Explanation: Columns 7 through 10 must be left blank.

RPG–0062 INVALID OR UNDEFINED FROM FILENAME ENTRY IN COLUMNS 11-18.

Severity: Terminal

Specification Type: E

Explanation: The from filename in columns 11 through 18 of your extension specifications is invalid or has not been previously defined in file description specifications. (The from filename must start in column 11.)

RPG–0063 TYPE OF FILE INVALID FOR FROM FILENAME ENTRY IN COLUMNS 11-18.

Severity: Terminal

Specification Type: E

Explanation: The from filename does not refer to a table or record address input file.

RPG–0064 INVALID OR UNDEFINED TO FILENAME IN COLUMNS 19-26.

Severity: Terminal

Specification Type: E

Explanation: The to filename in columns 19 through 26 of your extension specifications is invalid or has not been defined in file description specifications. (The to filename must start in column 19.)

RPG–0065 TYPE OF FILE INVALID OR INCORRECT FOR TO FILENAME ENTRY IN COLUMNS 19-26.

Severity: Terminal

Specification Type: E

Explanation: The to filename entry does not refer to an output file or to a file processed by a record address file.

RPG–0067 INVALID TABLE OR ARRAY NAME IN COLUMNS 27-32.

Severity: Terminal

Specification Type: E

Explanation: The table or array name in columns 27 through 32 was not specified properly. A table or array name must start in column 27. A table name must begin with TAB; an array name must not begin with TAB.

RPG–0068 INVALID OR MISSING NUMBER OF ENTRIES PER RECORD ENTRY IN COLUMNS 33-35, ASSUME 08.

Severity: Terminal

Specification Type: E

Explanation: The entry in columns 33 through 35 is missing on a specification line that has a from filename in columns 11 through 18, or it is not a one- to three-digit number (1-999). 08 is assumed, but the job is terminated.

RPG–0070 INVALID OR MISSING NUMBER OF ENTRIES PER TABLE OR ARRAY IN COLUMNS 36-39, ASSUME 05.

Severity: Terminal

Specification Type: E

Explanation: The entry in columns 36 through 39 is missing or it is not a one- to four-digit number (1-9999). 05 is assumed, but the job is terminated.

RPG–0071 NUMBER OF ENTRIES PER RECORD IN COLUMNS 33-35 EXCEEDS NUMBER OF ENTRIES PER TABLE/ARRAY IN COLUMNS 36-39.

Severity: Terminal

Specification Type: E

RPG–0072 INVALID OR MISSING LENGTH OF ENTRY IN COLUMNS 40-42 OR 52-54, ASSUME 05.

Severity: Terminal

Specification Type: E

Explanation: The length of entry specified is missing or is not a one- to three-digit number (1-15 for numeric entries; 1-256 for alphabetic entries). 05 is assumed, but the job is terminated.

RPG–0073 LENGTH SPECIFIED FOR EACH TABLE/ARRAY RECORD, IN COLUMNS 33-35 AND 40-42 OR 52-54, EXCEEDS RECORD LENGTH.

Severity: Terminal

Specification Type: E

Explanation: The table record length specified (length of entry times number of entries per record) is greater than the record length you specified for the table file in the file description specifications.

RPG–0074 INVALID PACKED OR BINARY ENTRY IN COLUMN 43 OR 55, ASSUME BLANK.

Severity: Warning

Specification Type: E

Explanation: The entry in column 43 or column 55 of the extension specifications is not P, B, or blank.

RPG–0075 PACKED OR BINARY VALID ONLY FOR PRE-EXECUTION TIME TABLES OR ARRAYS.

Severity: Terminal

Specification Type: E

Explanation: Packed or binary format can be specified (column 43 or column 55) only for preexecution-time tables or arrays. Blank is assumed, but the job is terminated.

RPG–0076 INVALID DECIMAL POSITION ENTRY IN COLUMN 44 OR 56, ASSUME 0.

Severity: Terminal

Specification Type: E

Explanation: The decimal position entry in column 44 or column 56 is not 0-9 or blank. Zero is assumed, but the job is terminated.

RPG–0077 INVALID SEQUENCE ENTRY IN COLUMN 45 OR 57, ASSUME BLANK.

Severity: Terminal

Specification Type: E

Explanation: The sequence entry in column 45 or column 57 is not A, D, or blank. Blank is assumed, but the job is terminated.

RPG–0079 INVALID ALTERNATE TABLE/ARRAY NAME IN COLUMNS 46-51.

Severity: Terminal

Specification Type: E

Explanation: The table or array name in columns 46 through 51 was not specified properly. The table or array name must start in column 46. A table name must begin with TAB.

RPG–0080 ALTERNATE TABLE/ARRAY NAME IN COLUMNS 46-51 AND/OR 27-32 MISSING FOR ENTRIES IN COLUMNS 33-45 AND/OR 52-57, ASSUME COLUMNS 33-57 AND/OR 46-57 BLANK.

Severity: Terminal

Specification Type: E

Explanation: Columns 52 through 57 contain entries describing an alternating table or array, but no alternating table or array name was specified in columns 46 through 51 or no table or array name was specified in columns 27 through 32.

RPG—0082 LENGTH OF TABLE/ARRAY IN COLUMNS 40-42 OR 52-54 FOR ALPHAMERIC FIELD EXCEEDS MAXIMUM.

Severity: Terminal

Specification Type: E

Explanation: The length of table or array entry specified in columns 40 through 42 or 52 through 54 is too large. 256 is assumed for noncompile time tables or arrays; a record length of 96 is assumed for compile time tables or arrays.

RPG—0083 LENGTH OF TABLE/ARRAY ENTRY IN COLUMNS 40-42 OR 52-54 FOR NUMERIC FIELD EXCEEDS 15, ASSUME 15.

Severity: Terminal

Specification Type: E

RPG—0084 FILE AND RECORD TYPE ENTRIES COLUMNS 7-42 AND FIELD TYPE ENTRIES COLUMNS 43-74 ON THE SAME LINE, ASSUME 7-42 BLANK.

Severity: Terminal

Specification Type: I

Explanation: Field type entries (columns 43 through 74) are not specified one line lower than file and record type entries (columns 7 through 42).

RPG—0085 INVALID, MISSING OR UNDEFINED FILENAME OR DATA STRUCTURE.

Severity: Terminal

Specification Type: L, I, C

Explanation: Either (1) the filename was missing, (2) the filename was not specified properly, or (3) the filename was not previously defined in the file description specifications.

This message is issued when the entries in columns 7 through 14 of the input specifications are not in the filename table or the name table (for data structure), or the name is invalid (for example, more than 6 characters used for data structure).

RPG—0086 FILENAME IN COLUMNS 7-14 DOES NOT REFER TO PRINTER FILE.

Severity: Terminal

Specification Type: L

RPG—0087 FORM LENGTH ENTRY IN COLUMNS 15-17 INVALID OR GREATER THAN 255.

Severity: Terminal

Specification Type: L

RPG—0088 INVALID OR MISSING FORMS LENGTH ENTRY IN COLUMNS 18-19, ASSUME FL.

Severity: Warning

Specification Type: L

RPG—0089 OVERFLOW LINE ENTRY IN COLUMNS 20-22 INVALID OR GREATER THAN 255.

Severity: Terminal

Specification Type: L

RPG—0090 INVALID OR MISSING OVERFLOW LINE ENTRY IN COLUMNS 23-24, ASSUME OL.

Severity: Warning

Specification Type: L

RPG—0091 OVERFLOW LINE IN COLUMNS 20-22 EXCEEDS FORM LENGTH IN COLUMNS 15-17, ASSUME FORM LENGTH.

Severity: Terminal

Specification Type: L

RPG—0092 INVALID OR UNDEFINED FILENAME IN COLUMNS 7-14.

Severity: Terminal

Specification Type: L, I, C, O

Explanation: The filename entry is not specified properly, or it was not previously defined in the file description specifications.

RPG–0093 FILE AND RECORD TYPE ENTRIES IN COLUMNS 7-42 AND FIELD TYPE ENTRIES IN COLUMNS 43-74.

Severity: Terminal

Specification Type: I

Explanation: Field description entries (columns 43 through 74) are not specified one line lower than file and record identification entries (columns 7 through 42). Field type entries (columns 43 through 74) are assumed to be blank and the job is terminated.

RPG–0094 FILE AND RECORD TYPE DESCRIPTION MUST PRECEDE THIS SPECIFICATION.

Severity: Terminal

Specification Type: I

Explanation: File and record type entries in columns 7 through 42 do not precede the related field description entries in columns 43 through 74. Enter the file and record type entries in columns 7 through 42 of the specifications line immediately preceding the related field description entries in columns 43 through 74.

RPG–0095 AND OR OR LINE OUT OF ORDER.

Severity: Terminal

Specification Type: I, C

Explanation: The AND or OR line does not follow the proper file or record type entries or is on the first line of the calculation specifications. (The system may have dropped your file and record type specifications because of other errors in your program.)

RPG–0096 AND LINE FOLLOWS LINE WITH NO RECORD IDENTIFICATION CODES.

Severity: Terminal

Specification Type: I

RPG–0097 NO FIELD DESCRIBED FOR THIS OR PREVIOUS RECORD OR DATA STRUCTURE. IF DATA STRUCTURE, LENGTH DEFAULTS TO ONE.

Severity: Warning

Specification Type: I

RPG–0098 INVALID SEQUENCE ENTRY IN COLUMNS 15-16, ASSUME ALPHABETIC SEQUENCE ENTRY.

Severity: Warning

Specification Type: I

Explanation: The sequence entry in columns 15 and 16 is neither a two-digit number nor a 2-character alphabetic entry; or numeric entry is invalid for device type or file type.

RPG–0101 NUMERIC SEQUENCE ENTRY IN COLUMNS 15-16 NOT IN ASCENDING ORDER OR THE FIRST IS NOT 01, ASSUME PREVIOUS NUMERIC SEQUENCE OR 01 IF FIRST NUMERIC RECORD.

Severity: Warning

Specification Type: I

Explanation: Either the first numeric sequence entry is not 01 or the numeric sequence entries are not in ascending order. If this is the first numeric sequence entry, 01 is assumed; otherwise, the numeric sequence entry from the previous specification line is assumed.

RPG–0102 INVALID NUMBER ENTRY IN COLUMN 17 FOR NUMERIC SEQUENCE, ASSUME N.

Severity: Warning

Specification Type: I

Explanation: The number entry in column 17 is neither 1 nor N.

RPG–0103 INVALID OPTION ENTRY IN COLUMN 18 FOR NUMERIC SEQUENCE, ASSUME 0.

Severity: Warning

Specification Type: I

Explanation: The option entry (column 18) must be blank, O, or U (if DS has been specified in columns 19 and 20 of the input specifications).

RPG–0104 NUMBER/OPTION ENTRIES IN COLUMNS 17-18 INVALID WITH ALPHAMERIC SEQUENCE ENTRIES. ENTRIES NOT SPECIFIED CORRECTLY FOR DATA STRUCTURES.

Severity: Terminal

Specification Type: I

Explanation: Columns 17 and 18 must be blank when columns 15 and 16 contain an alphabetic sequence entry.

Column 17 must be blank for a data structure. Column 18 must be blank or contain a U for a data structure specified as the display station local data area.

RPG–0105 NUMBER/OPTION ENTRIES IN COLUMNS 17-18 INVALID FOR AND OR OR LINE, ASSUME BLANK.

Severity: Warning

Specification Type: I

Explanation: Columns 17 and 18 must be blank in an AND or OR line.

RPG–0106 INVALID POSITION ENTRY FOR RECORD IDENTIFICATION CODES IN COLUMNS 21-24, 28-31, OR 35-38, OR TO POSITION COLUMNS 48-51, ASSUME 1.

Severity: Terminal

Specification Type: I

Explanation: The position entry for record identification codes or the to position for a field exceeds the record length.

RPG–0107 INVALID NOT ENTRY IN COLUMN 25, 32, OR 39, ASSUME N.

Severity: Warning

Specification Type: I

Explanation: The entry in column 25, 32, or 39 is not N or blank.

RPG–0108 INVALID C/Z/D ENTRY IN COLUMN 26, 33, OR 40, ASSUME C.

Severity: Warning

Specification Type: I

RPG–0111 INVALID ENTRY IN COLUMN 43, ASSUME BLANK.

Severity: Warning

Specification Type: I

Explanation: The entry in column 43 is not P, B, or blank.

RPG–0112 INVALID OR BLANK FROM AND/OR TO ENTRY OR INVALID USE OF A KEYWORD, COLUMNS 44-51 AND 52, ASSUME 1 FOR FROM AND TO POSITIONS.

Severity: Terminal

Specification Type: I

Explanation: Columns 44 through 47 and/or 48 through 51 do not contain an entry from 1 to 4096; or for the WORKSTN file information data structure (INFDS), columns 44 through 50 do not contain *RECORD, *OPCODE, *SIZE, *MODE, *INP, *OUT, or *STATUS or the keywords are specified incorrectly. Columns 51 and 52 must be blank if a keyword is specified.

RPG–0113 FROM ENTRY IN COLUMNS 44-47 EXCEEDS TO ENTRY IN COLUMNS 48-51, ASSUME TO ENTRY EQUAL TO FROM ENTRY.

Severity: Terminal

Specification Type: I

RPG–0114 LENGTH OF NUMERIC FIELDS IN COLUMNS 44-51 EXCEEDS 15, ASSUME 15.

Severity: Terminal

Specification Type: I

RPG–0115 ALPHAMERIC FIELD SPECIFIED AS PACKED OR BINARY, ASSUME NUMERIC FIELD.

Severity: Terminal

Specification Type: I

Explanation: Leave column 43 blank for alphameric fields, or make an entry (0-9) in column 52 for numeric fields.

RPG–0116 INVALID DECIMAL POSITION ENTRY IN COLUMN 52, ASSUME 0.

Severity: Terminal

Specification Type: I

Explanation: The decimal position entry in column 52 is not 0-9 or blank.

RPG–0117 DECIMAL POSITION IN COLUMN 52 INVALID FOR ARRAY, ASSUME BLANK.

Severity: Warning

Specification Type: I

Explanation: No decimal position entry can be specified in column 52 for an array. Decimal positions for arrays must be specified in your extension specifications.

RPG–0118 FIELD NAME IN COLUMNS 53-58 MISSING OR INVALID.

Severity: Terminal

Specification Type: I

Explanation: The field name entry in columns 53 through 58 is missing or is not specified properly. An array element or a table name cannot be specified in a data structure.

RPG–0119 INVALID CONTROL LEVEL INDICATOR IN COLUMNS 59-60, ASSUME BLANK.

Severity: Terminal

Specification Type: I

Explanation: The control level entry in columns 59 and 60 is neither L1-L9 nor blank. Blank is assumed, but the job is terminated.

RPG–0120 INVALID MATCHING FIELD ENTRY IN COLUMNS 61-62, ASSUME M1.

Severity: Terminal

Specification Type: I

Explanation: The matching field entry in columns 61 and 62 is not M1-M9 or blank. M1 is assumed, but the job is terminated.

RPG–0121 FROM FILE CANNOT HAVE AN E IN COLUMN 17 OF FILE DESCRIPTION SPECIFICATION WHEN TO FILE IS A DEMAND FILE.

Severity: Terminal

Specification Type: E

Explanation: End of file, E in column 17 of the file description specifications, cannot be used for a record address file that is used to process a demand file. Leave column 17 blank.

RPG–0122 FIELD WAS PREVIOUSLY DEFINED WITH DIFFERENT LENGTH OR DECIMAL POSITIONS, OR FIELD WAS ALREADY DEFINED IN ONE DATA STRUCTURE. FIRST DEFINITION IS ASSUMED, OR FIELD IS NOW DEFINED AS A LOOK AHEAD FIELD.

Severity: Warning

Specification Type: I, C

RPG–0123 INVALID CONTROL LEVEL ENTRY IN COLUMNS 7-8.

Severity: Terminal

Specification Type: C

Explanation: The control level entry in columns 7 and 8 is not AN, OR, L0-L9, LR, SR, or blank.

RPG–0124 INVALID NOT ENTRY IN COLUMN 9, 12, OR 15, ASSUME N.

Severity: Warning

Specification Type: C

Explanation: The entry in column 9, 12, or 15 is not N or blank.

RPG–0125 INVALID FIELD NAME OR CONSTANT FOR FACTOR 1 IN COLUMNS 18-27.

Severity: Terminal

Specification Type: C, H

Explanation: The field name or constant in columns 18 through 27 of the calculation specification is not specified properly. Both must begin in column 18. If a constant contains ideographic data, you may have forgotten to code the transparent literal option in column 57 of the control specification.

RPG–0126 LENGTH OF TABLE/ARRAY EXCEEDS MAXIMUM STORAGE.

Severity: Terminal

Specification Type: E

Explanation: The number of entries per table/array (columns 36 through 39) multiplied by the length of entry (columns 40 through 42) exceeds maximum storage. Reduce the number of entries or the length of the entries.

RPG–0127 FIELD LENGTH ENTRY IN COLUMNS 49-51 INVALID WITH NO RESULT FIELD, ASSUME 49-51 BLANK.

Severity: Warning

Specification Type: C

RPG–0128 INVALID OPERATION CODE IN COLUMNS 28-32.

Severity: Terminal

Specification Type: C

RPG–0129 FACTOR 2 FIELD NAME IN COLUMNS 33-42 EXCEEDS SIX CHARACTERS.

Severity: Terminal

Specification Type: C

RPG–0130 TO FILE MUST BE A LIMITS FILE IF FROM FILE IS A RECORD ADDRESS FILE, OR TO FILE MUST BE A RANDOM ACCESS FILE IF FROM FILE IS AN ADDROUT FILE.

Severity: Terminal

Specification Type: E

RPG–0131 FACTOR 2 IN COLUMNS 33-42 INVALID.

Severity: Terminal

Specification Type: C

Explanation: The field name or constant in columns 33 through 42 of the calculation specification is not specified properly. Entry must start in column 33. If a constant contains ideographic data, you may have forgotton to code the transparent literal option in column 57 of the control specification.

RPG–0132 FACTOR 2 MUST BE A FILENAME.

Severity: Terminal

Specification Type: C

RPG–0133 NUMERIC FIELD LENGTH EXCEEDS 15, ASSUME 15.

Severity: Terminal

Specification Type: C

Explanation: Length specified in columns 49 through 51 for numeric field is too large.

RPG–0134 ALPHAMERIC FIELD LENGTH EXCEEDS 256, ASSUME 256.

Severity: Terminal

Specification Type: I, C

Explanation: Length specified in columns 49 through 51 of the calculation specifications or in columns 44 through 51 of the input specifications for an alphameric field is too large.

RPG–0135 INVALID RESULT FIELD ENTRY IN COLUMNS 43-53.

Severity: Terminal

Specification Type: C

RPG–0136 INVALID IDEOGRAPHIC LITERAL, ASSUME LITERAL IS ALPHAMERIC.

Severity: W

Specification Type: C, O

Explanation: A literal or a constant beginning with an apostrophe and the S/O control character was found, but either no S/I control character was found, a S/I control character was found but was not immediately followed by an apostrophe, or an odd number of 1-byte characters were found between the S/O and S/I control characters.

RPG–0137 INVALID RESULT FIELD LENGTH IN COLUMNS 49-51, ASSUME 15 FOR NUMERIC OR 256 FOR ALPHAMERIC FIELD.

Severity: Terminal

Specification Type: C

RPG–0138 DECIMAL POSITION ENTRY IN COLUMN 52 INVALID WITH NO FIELD LENGTH ENTRY IN COLUMNS 49-51, ASSUME BLANK.

Severity: Terminal

Specification Type: C

RPG–0139 INVALID DECIMAL POSITION ENTRY IN COLUMN 52, ASSUME 0.

Severity: Terminal

Specification Type: C

Explanation: The decimal position entry in column 52 is not 0-9 or blank.

RPG–0140 INVALID HALF ADJUST ENTRY IN COLUMN 53, ASSUME H.

Severity: Warning

Specification Type: C

Explanation: The half-adjust entry in column 53 is neither H nor blank.

RPG–0141 DEBUG CALCULATION OPERATION USED BUT DEBUG OPTION NOT SPECIFIED IN THE CONTROL SPECIFICATION.

Severity: Warning

Specification Type: C

Explanation: You used the DEBUG operation code in your calculation specifications, but you did not specify the DEBUG option (1 in column 15) in your control specifications. DEBUG operations are not executed.

RPG–0142 FILE AND RECORD IDENTIFICATION ENTRIES IN COLUMNS 7-31 AND FIELD DESCRIPTION ENTRIES IN COLUMNS 32-74 ON SAME LINE.

Severity: Terminal

Specification Type: O

Explanation: Your field description entries in columns 23 through 74 are not specified one line lower than the file and record identification entries in columns 7 through 31. Blanks are assumed for columns 7 through 31 and the job is terminated.

RPG–0143 INVALID LINE TYPE ENTRY IN COLUMN 15.

Severity: Terminal

Specification Type: O

Explanation: The line type entry in column 15 is not H, D, T, or E. An E can be used only if an EXCPT operation is used in the calculation specifications. H is assumed; the job is terminated.

RPG–0144 AND OR OR LINE NOT PRECEDED BY RECORD IDENTIFICATION.

Severity: Terminal

Specification Type: O

Explanation: An AND or OR line is not preceded by record identification entries in columns 15 through 31.

RPG–0145 INVALID SKIP/SPACE ENTRIES IN COLUMNS 17-22 FOR AND LINE, ASSUME BLANK.

Severity: Warning

Specification Type: O

Explanation: Columns 17 through 22 of an AND line contain space/skip entries; they should be blank.

RPG–0146 INVALID FILENAME OR ENTRY IN COLUMN 15 MISSING ON FIRST OUTPUT SPECIFICATION.

Severity: Terminal

Specification Type: O

Explanation: Either columns 7 through 14 contain an invalid filename or no line type entry was specified in column 15 of the specification line.

RPG–0147 INVALID NOT ENTRY IN COLUMN 23, 26, OR 29, ASSUME N.

Severity: Warning

Specification Type: O

Explanation: The entry in column 23, 26, or 29 is neither N nor blank.

RPG–0148 INVALID FIELD NAME IN COLUMNS 32-37.

Severity: Terminal

Specification Type: O

Explanation: The field name entry in columns 32 through 37 is not specified properly or was not defined previously in input or calculation specifications.

RPG–0149 INVALID OR MISSING CONSTANT.

Severity: Terminal

Specification Type: O, H

Explanation: The constant in columns 45 through 70 of the output specification is not specified properly. If the constant contains ideographic data, you may have forgotten to code the transparent literal option in column 57 of the control specification.

RPG–0150 INVALID BLANK AFTER ENTRY IN COLUMN 39, ASSUME BLANK.

Severity: Terminal

Specification Type: O

Explanation: The blank after entry in column 39 is neither B nor blank.

RPG–0151 MISSING OR INCORRECTLY SPECIFIED END POSITION IN COLUMNS 40-43, ASSUME END POSITION IS BLANK.

Severity: Warning

Specification Type: O

RPG–0152 INVALID PACKED OR BINARY ENTRY IN COLUMN 44, ASSUME BLANK.

Severity: Warning

Specification Type: O

Explanation: The entry in column 44 is not P, B, or blank.

RPG–0153 BLANK END POSITION WAS SPECIFIED; END POSITION WAS CALCULATED.

Severity: Warning

Specification Type: O

Explanation: The end position was calculated from the last end position specified in this record or from position 1 if no end positions were specified.

RPG–0154 ENTRIES IN COLUMNS 7-22 INVALID FOR A FIELD DESCRIPTION SPECIFICATION, ASSUME BLANK.

Severity: Terminal

Specification Type: O

Explanation: The file and record identification entries in columns 7 through 22 are not specified one line above the first related field description entries. Place your file and record identification entries (columns 7 through 22) one line above the field description entries (columns 32 through 74).

RPG–0155 INVALID ENTRY IN COLUMNS 71-74, ASSUME BLANK.

Severity: Terminal

Specification Type: I, O

Explanation: Columns 71 through 74 must be blank.

RPG–0158 TABLE NAME INVALID FOR FIELD NAME ENTRY IN COLUMNS 53-58.

Severity: Terminal

Specification Type: I

RPG–0159 MISSING RECORD IDENTIFYING INDICATOR IN COLUMNS 19-20.

Severity: Warning

Specification Type: I

Explanation: No record identifying indicator is specified in columns 19 and 20. Check your input specifications to determine whether or not a record identifying indicator should be entered in columns 19 and 20.

RPG–0160 FILE NAME IN COLUMNS 7-14 NOT SPECIFIED AS AN INPUT OR UPDATE-SECONDARY, DEMAND, PRIMARY OR CHAINED FILE.

Severity: Terminal

Specification Type: I

Explanation: The file named in columns 7 through 14 was not previously defined in file description specifications as an input or update file with a designation of primary, secondary, demand, or chained.

RPG–0161 AND OR OR LINE INVALID WITH LOOK AHEAD RECORDS, DATA STRUCTURES, OR RLABL.

Severity: Terminal

Specification Type: I, C

Explanation: An AND or OR line was used with look-ahead fields or RLABL. Make sure that AND or OR lines are not specified for look-ahead fields (** in columns 19 and 20) or for RLABL. AND or OR lines are not valid with a data structure on input specifications.

RPG–0162 RECORD IDENTIFYING INDICATOR IN COLUMNS 19-20 INVALID FOR AN AND LINE.

Severity: Warning

Specification Type: I

Explanation: A record identifying indicator is in columns 19 and 20 of an AND line. Blanks are assumed.

RPG–0163 ENTRIES IN COLUMNS 17-18 AND 21-42 INVALID FOR LOOK AHEAD RECORD. ENTRIES IN COLUMNS 59-74 INVALID FOR LOOK AHEAD FIELD, OR FOR FIELD IN A DATA STRUCTURE.

Severity: Terminal

Specification Type: I

Explanation: Columns 17 and 18 and 21 through 42 must be blank for look-ahead records; columns 59 through 74 must be blank for look-ahead fields. Entries in columns 59 through 74 are not valid for a field in a data structure.

RPG–0165 INDICATORS IN COLUMNS 65-70 INVALID FOR TABLE/ARRAY OR FOR FIELD DEFINED AS DATA STRUCTURE.

Severity: Terminal

Specification Type: I

Explanation: Field indicators cannot be used if columns 53 through 58 contain a table/array name. Use the field indicators to test numeric fields. Field indicators in columns 65 through 70 are invalid with a field defined as a data structure.

RPG–0166 PLUS OR MINUS INDICATOR IN COLUMNS 65-68 INVALID FOR ALPHAMERIC FIELD.

Severity: Terminal

Specification Type: I

Explanation: A plus or minus indicator in columns 65 through 68 cannot be used to test an alphameric field. Use plus or minus indicators only to test numeric fields. An alphameric field can be tested only for a blank condition (entry in columns 69 and 70). Blank is assumed.

RPG–0167 RECORD IDENTIFICATION POSITION COLUMNS 21-38 OR TO ENTRY IN COLUMNS 48-51 EXCEEDS RECORD LENGTH, ASSUME RECORD LENGTH.

Severity: Terminal

Specification Type: I

Explanation: Field location entries (columns 21 through 38 and 48 through 51) exceed record length specified in the file description specifications.

RPG–0168 FIELD NAME IN COLUMNS 53-58 IS A RESERVED WORD OTHER THAN PAGE, PAGE1 - PAGE7.

Severity: Terminal

Specification Type: I

Explanation: PAGE1-PAGE7 are the only RPG II reserved words that can be entered in these columns.

RPG–0169 CONTROL OR MATCHING FIELDS INVALID FOR AN ARRAY OR DATA STRUCTURE.

Severity: Terminal

Specification Type: I

Explanation: Control or matching fields must not be specified for arrays. Columns 59 and 60 control levels and columns 61 and 62 matching fields are not valid for a data structure.

RPG–0170 MATCHING OR CONTROL FIELDS INVALID WITH DEMAND OR CHAIN FILES OR WORKSTN DEVICE.

Severity: Terminal

Specification Type: I

Explanation: Matching or control fields cannot be specified for demand or chain files or WORKSTN device.

RPG−0171 LOOK AHEAD RECORDS INVALID WITH FILE TYPE, OR WITH THIS DEVICE.

Severity: Terminal

Specification Type: I

Explanation: Look-ahead records cannot be specified for demand files, chained files, CONSOLE files, or WORKSTN files.

RPG−0172 INCORRECT SEQUENCE OF INPUT SPECIFICATIONS. IF DATA STRUCTURE SPECIFIED IT MUST BE LAST INPUT SPECIFIED.

Severity: Terminal

Specification Type: I

Explanation: All records from one input or update file are not specified consecutively. A data structure must be specified last on the input specifications.

RPG−0173 NO FIELDS SPECIFIED FOR LOOK AHEAD RECORD.

Severity: Terminal

Specification Type: I

Explanation: A look-ahead record is specified (** in columns 19 and 20), but no look-ahead fields are defined (columns 53 through 58).

RPG−0174 LIMITS FILE NOT PROCESSED BY RECORD ADDRESS FILE OR SETLL OPERATION CODE.

Severity: Terminal

Specification Type: F, C

Explanation: A file is designated to be processed sequentially within limits, but does not have a record address file or a SETLL operation code associated with it.

RPG−0175 INVALID FILE TYPE FOR SETLL OPERATION.

Severity: Terminal

Specification Type: C

Explanation: The file to be processed by a SETLL operation code must be a limits file that has not already been specified to be processed via a record address file.

RPG−0178 BINARY INVALID WITH CONTROL OR MATCHING FIELDS.

Severity: Terminal

Specification Type: I

Explanation: Binary fields have been used as control or matching fields.

RPG−0179 ARRAY LENGTH SPECIFIED IN DATA STRUCTURE NOT LARGE ENOUGH TO CONTAIN ARRAY AS SPECIFIED IN EXTENSION SPECS.

Severity: Terminal

Specification Type: I

RPG−0180 ARRAY LENGTH EXCEEDS LENGTH SPECIFIED IN COLUMNS 36-42 OF EXTENSION SPECIFICATIONS OR NOT A MULTIPLE OF THE ENTRY LENGTH IN COLUMNS 40-42 OF THE EXTENSION SPECIFICATIONS.

Severity: Terminal

Specification Type: I

RPG−0181 INCONSISTENT FIELD LENGTHS FOR CONTROL OR MATCHING FIELDS OF ONE LEVEL. ASSUME FIRST VALID LENGTH.

Severity: Terminal

Specification Type: I

RPG−0182 INVALID SPLIT CONTROL FIELD SPECIFICATION. ASSUME PREVIOUS TOTAL LENGTH FOR THIS LEVEL.

Severity: Terminal

Specification Type: I

Explanation: Specifications for split control fields of the same level are not specified on successive lines.

RPG–0183 CONTROL OR MATCHING FIELDS OF A LEVEL SPECIFIED AS BOTH ALPHAMERIC AND NUMERIC. ASSUME NUMERIC.

Severity: Warning

Specification Type: I

Explanation: All control and matching fields assigned the same level are not the same type (alphameric or numeric). Numeric is assumed for all fields assigned the same control or matching level. If any field specified as alphameric is greater than 15 characters, only a portion of the field will be used.

RPG–0184 ALL OF THE VALID MATCH LEVELS WERE NOT REFERENCED IN THE LAST RECORD GROUP.

Severity: Terminal

Specification Type: I

Explanation: The same number of match levels are not specified to all record types in a file.

RPG–0186 MATCH OR CONTROL FIELDS WITHOUT FIELD RECORD RELATION ENTRIES MUST PRECEDE MATCH OR CONTROL FIELDS WITH FIELD RECORD RELATION ENTRIES. ASSUME PART OF A NEW GROUP OF MATCH FIELDS.

Severity: Terminal

Specification Type: I

RPG–0187 MATCH AND CONTROL FIELDS WITH FIELD RECORD RELATION ENTRIES MUST BE GROUPED ACCORDING TO THE FIELD RECORD RELATION INDICATOR. ASSUME NEW GROUP OF MATCH FIELDS.

Severity: Terminal

Specification Type: I

Explanation: When field record relation is used, all match and control fields assigned the same indicator (columns 63 and 64) must be grouped together.

RPG–0188 FIELD RECORD RELATION INDICATOR USED IMPROPERLY WITH MATCH OR CONTROL FIELD.

Severity: Terminal

Specification Type: I

Explanation: When used with match or control fields, the field record relation indicator in columns 63 and 64 does not match a record identifying indicator used for this record.

RPG–0189 INVALID SEQUENCE FOR CALCULATION SPECIFICATIONS.

Severity: Terminal

Specification Type: C

Explanation: Calculation specifications must be specified in the following order: detail, total, subroutine.

RPG–0190 INVALID SEQUENCE FOR BEGSR AND ENDSR OPERATION CODES.

Severity: Terminal

Specification Type: C

Explanation: BEGSR operation code does not precede ENDSR operation code.

RPG–0191 A SUBROUTINE MUST NOT CALL ITSELF.

Severity: Terminal

Specification Type: C

Explanation: An EXSR specification within a subroutine must not call the subroutine it is in. If you wish to branch to another point within the same subroutine, use a GOTO and TAG operation.

RPG–0192 BRANCHING BETWEEN SUBROUTINE AND OTHER CALCULATIONS INVALID.

Severity: Terminal

Specification Type: C

Explanation: Branching (GOTO and TAG) can only occur within a subroutine. You cannot branch into a subroutine or out of a subroutine.

RPG–0193 BRANCHING BETWEEN DETAIL, TOTAL AND LR CALCULATIONS INVALID.

Severity: Terminal

Specification Type: C

Explanation: Branching must be from detail operation to detail operation or from total operation to total operation. It cannot be from detail to total operation or vice versa.

RPG–0194 SETOF OPERATION INVALID FOR LR INDICATOR.

Severity: Terminal

Specification Type: C

Explanation: The LR indicator cannot be turned off by the SETOF operation code.

RPG–0195 LENGTH OF SEARCH WORD NOT EQUAL TO LENGTH OF ELEMENT IN TABLE OR ARRAY.

Severity: Terminal

Specification Type: C

Explanation: The length of the search word (factor 1) is not equal to the length of the element in the table or array being searched.

RPG–0196 FACTOR 2 OR RESULT FIELD INVALID FOR LOKUP OPERATION CODE.

Severity: Terminal

Specification Type: C

RPG–0197 SEARCH TABLE HAS MORE ENTRIES THAN ITS RELATED TABLE.

Severity: Warning

Specification Type: C

Explanation: The search table (factor 2) contains more entries than its related table, which is specified in the result field.

RPG–0198 INDICATOR ENTERED IN COLUMNS 54-57 INVALID WITH LOKUP ON AN UNSEQUENCED TABLE OR ARRAY.

Severity: Warning

Specification Type: C

Explanation: Do not specify a search for high or low in a LOKUP operation on an unsequenced table or array. Unpredictable results may occur. Specify the LOKUP operation on an unsequenced table or array for an equal condition only (indicator in columns 58 and 59). The system accepts the indicator as specified.

RPG–0199 TEST FOR BOTH HIGH AND LOW INVALID FOR LOKUP OPERATION.

Severity: Terminal

Specification Type: C

RPG–0200 RESULTING INDICATORS IN COLUMNS 54-59 REQUIRED OR NOT ALLOWED FOR OPERATION SPECIFIED.

Severity: Terminal

Specification Type: C

Explanation: The resulting indicator entry in columns 54 through 59 is not specified properly. Check to determine whether resulting indicators are required for this operation. If so, make the proper entries (01-99, H1-H9, L1-L9, LR, OA-OG, OV, or KA-KN, KP-KY).

RPG–0201 HALF ADJUST ENTRY IN COLUMN 53 FOR DIVISION OPERATION FOLLOWED BY AN MVR OPERATION, ASSUME NO HALF ADJUST.

Severity: Warning

Specification Type: C

Explanation: When an MVR operation follows a DIV operation, the DIV operation must not be half-adjusted.

RPG−0202 MVR OPERATION CODE DOES NOT FOLLOW DIV OPERATION.

Severity: Terminal

Specification Type: C

Explanation: The MVR operation must immediately follow a DIV operation.

RPG−0204 HALF ADJUST ENTRY IN COLUMN 53 INVALID FOR OPERATION OR NUMBER OF DECIMAL POSITIONS SPECIFIED, ASSUME BLANK.

Severity: Warning

Specification Type: C

RPG−0205 COMP, TESTZ, OR MVR INVALID FOR AN ARRAY.

Severity: Terminal

Specification Type: C

RPG−0206 INVALID USE OF COMP OR LOKUP.

Severity: Terminal

Specification Type: C

Explanation: COMP or LOKUP operation specified improperly. Make sure that factor 1 and factor 2 of a COMP operation are both alphameric or both numeric. Make sure the search word and the table or array to be searched are both alphameric or both numeric.

RPG−0207 FIELD TYPE, ALPHAMERIC OR NUMERIC, INVALID FOR OPERATION SPECIFIED.

Severity: Terminal

Specification Type: C

RPG−0208 FORCE OPERATION INVALID AT TOTAL TIME.

Severity: Terminal

Specification Type: C

Explanation: FORCE operation must be specified at detail time only.

RPG−0209 FILE TYPE INVALID FOR USE WITH THIS OPERATION CODE.

Severity: Terminal

Specification Type: C

Explanation: DEBUG must be used with an output file; EXCPT must be used with an output file or a combined file; FORCE must be used with an input, update, or combined primary or secondary file. READ must be used with an input, update, or combined demand file.

RPG−0211 DEBUG SPECIFIED FOR MORE THAN ONE OUTPUT FILE.

Severity: Terminal

Specification Type: C

Explanation: The filename entered in factor 2 is not the same for all DEBUG operations.

RPG−0212 EXCPT OPERATION CODE SPECIFIED BUT NO EXCPT OUTPUT RECORDS SPECIFIED.

Severity: Warning

Specification Type: C

Explanation: The EXCPT operation code is used, but no EXCPT records are specified (E in column 15 of the output specifications).

RPG−0213 PROGRAM CONTAINS UNASSOCIATED OR MISSING EXSR/BEGSR LABEL.

Severity: Terminal

Specification Type: C

Explanation: The label in factor 2 of an EXSR operation is not the same as the label in factor 1 of a BEGSR operation.

RPG−0214 GOTO BRANCHES TO A BEGSR NAME.

Severity: Terminal

Specification Type: C

Explanation: The label in factor 2 of a GOTO operation must be the same as the label in factor 1 of a TAG or ENDSR operation. The EXSR operation must be used to call a subroutine.

RPG–0215 FACTOR 1 ENTRY IN COLUMNS 18-27 MISSING.

Severity: Terminal

Specification Type: C

RPG–0216 FACTOR 1 ENTRY IN COLUMNS 18-27 INVALID FOR THIS OPERATION.

Severity: Terminal

Specification Type: C

Explanation: For this operation either an entry must not be specified in factor 1, factor 1 must not be an alphameric literal, or factor 1 must not be a figurative constant.

RPG–0217 FACTOR 2 ENTRY IN COLUMNS 33-42 MISSING.

Severity: Terminal

Specification Type: C

RPG–0218 FACTOR 2 ENTRY IN COLUMNS 33-42 INVALID FOR THIS OPERATION.

Severity: Terminal

Specification Type: C

Explanation: Either an entry must not be specified in factor 2, factor 2 must not be an alphameric literal, or factor 2 must not be a figurative constant for this operation. Otherwise, the entry specified for the ENDSR operation code for the INFSR subroutine is invalid. (Valid entries are *DETC, *GETIN, and *CANCL.)

RPG–0219 RESULT FIELD ENTRY IN COLUMNS 43-48 MISSING.

Severity: Terminal

Specification Type: C

RPG–0220 RESULT FIELD ENTRY IN COLUMNS 43-48 INVALID FOR THIS OPERATION.

Severity: Terminal

Specification Type: C

Explanation: An entry must not be specified in the result field for this operation.

RPG–0221 RESULT FIELD LENGTH MAY NOT BE LARGE ENOUGH.

Severity: Warning

Specification Type: C

Explanation: The result field may not be large enough for the calculation operation specified. Depending on the contents of the field, significant digits may be lost or unpredictable results may occur.

RPG–0223 SUBROUTINE SPECIFICATIONS ARE THE ONLY CALCULATION SPECIFICATIONS SPECIFIED.

Severity: Terminal

Specification Type: C

Explanation: Subroutine specifications do not follow detail and total calculations.

RPG–0224 A ZERO CONSTANT IS INVALID AS A DIVISOR IN COLUMNS 33-42.

Severity: Terminal

Specification Type: C

Explanation: The constant entered in factor 2 of a DIV operation must not be zero.

RPG–0225 CONDITIONING INDICATORS IN COLUMNS 9-17 INVALID WITH TAG, BEGSR, ENDSR OR RLABL OPERATION.

Severity: Terminal

Specification Type: C

RPG–0226 A RESERVED WORD OTHER THAN PAGE INVALID.

Severity: Terminal

Specification Type: C, I, O

Explanation: No reserved word other than PAGE or PAGE1–or PAGE7 can be specified as a result field. CONTD is a reserved word, for compatibility with other systems. Make sure no reserved word other than PAGE or PAGE1-PAGE7 is specified in columns 43 through 48 as the result field.

RPG–0227 RESULT FIELD IN COLUMNS 43-48 IS A LOOK AHEAD FIELD OR CONSTANT.

Severity: Terminal

Specification Type: C

Explanation: The result field must not be a look ahead field or a constant.

RPG–0228 INVALID INDEX.

Severity: Terminal

Specification Type: C

Explanation: The array index is not specified properly. The index field name must contain a valid combination of characters. The index constant or field value must be a positive number that does not exceed the number of elements in the array and that has zero decimal positions.

RPG–0229 INDEXING INVALID FOR TABLES OR FIELDS.

Severity: Terminal

Specification Type: C

Explanation: Indexing can be specified for arrays only.

RPG–0231 GOTO DOES NOT BRANCH TO A TAG.

Severity: Terminal

Specification Type: C

Explanation: The label in factor 2 of this GOTO operation is not the same as the label in factor 1 of a TAG or ENDSR operation.

RPG–0232 THIS NAME WAS PREVIOUSLY USED ON A TAG, BEGSR OR ENDSR.

Severity: Terminal

Specification Type: C

Explanation: The label in factor 1 was previously specified in another TAG, BEGSR, or ENDSR operation.

RPG–0233 CONFIGURATION, COLUMN 15 CONTAINS AN ENTRY OTHER THAN P, S, M OR BLANK. IF CONTROL/TRIBUTARY, COLUMN 17 IS BLANK, ASSUME SWITCHED NETWORK. IF COLUMN 17 IS NOT BLANK, ASSUME MULTIPOINT NETWORK.

Severity: Terminal

Specification Type: T

RPG–0234 TRANSMITTER/RECEIVER, COLUMN 16, DOES NOT CONTAIN T OR R.

Severity: Terminal

Specification Type: T

Explanation: The type of station entry in column 16 is neither T nor R. Enter T (for a transmitter station) or R (for a receiver station).

RPG–0235 CONTROL/TRIBUTARY, COLUMN 17, CONTAINS A CHARACTER OTHER THAN T OR BLANK. IF THIS IS A SWITCHED OR POINT TO POINT NETWORK, ASSUME BLANK. IF MULTIPOINT ASSUME T.

Severity: Warning

Specification Type: T

Explanation: The type of control entry in column 17 is neither T nor blank. Blank is assumed if this is a switched network or a point-to-point leased line (column 15 contains an S or P); T is assumed if this is a multipoint leased line (column 15 contains an M).

RPG–0236 ASCII/EBCDIC, COLUMN 18, IS NOT U, A, E, OR BLANK. ASSUME EBCDIC.

Severity: Warning

Specification Type: T

Explanation: The type of code entry in column 18 is not U or A for ASCII, or E or blank for EBCDIC.

RPG–0237 TRANSPARENT FEATURE, COLUMN 19, IS NOT Y, N, OR BLANK. ASSUME NO TRANSPARENCY.

Severity: Warning

Specification Type: T

Explanation: The entry in column 19 is not Y for transparency or N or blank for no transparency.

RPG–0238 INVALID ENTRY IN COLUMN 20, MUST CONTAIN AN A, B, M, OR BLANK.

Severity: Terminal

Specification Type: T

Explanation: The entry in column 20 is not M, A, B, or blank. The system ignores entries in columns 21 through 31.

RPG–0239 ENTRY FOR DIAL NUMBER, COLUMNS 21-31, IS NOT ALLOWED.

Severity: Terminal

Specification Type: T

RPG–0240 IDENTIFICATION TYPE FOR THIS STATION, COLUMN 32, IS NOT S, E, OR BLANK. COLUMNS 33-39 WILL NOT BE CHECKED.

Severity: Warning

Specification Type: T

RPG–0241 IDENTIFICATION FOR THIS STATION, COLUMNS 33-39, CONTAIN AN INVALID ENTRY FOR THE IDENTIFICATION TYPE INDICATED IN COLUMN 32.

Severity: Terminal

Specification Type: T

Explanation: The identification entry in columns 33 through 39 is invalid for the identification type specified in column 32. Enter identification sequence in columns 33 through 39 if column 32 contains an E; enter symbolic name in columns 33 through 39 if column 32 contains an S.

RPG–0242 IDENTIFICATION TYPE FOR THE REMOTE STATION, COLUMN 40, IS NOT S, E, OR BLANK. COLUMNS 41-47 WILL NOT BE CHECKED.

Severity: Warning

Specification Type: T

RPG–0243 REMOTE STATION IDENTIFICATION IN COLUMNS 41-47, IS INVALID FOR THE IDENTIFICATION TYPE GIVEN IN COLUMN 40.

Severity: Terminal

Specification Type: T

RPG–0244 COLUMNS 48-51 AND/OR 65-70 MUST BE BLANK.

Severity: Terminal

Specification Type: T

RPG–0245 ITB, COLUMN 52, IS NOT I OR BLANK. ASSUME I.

Severity: Warning

Specification Type: T

RPG–0246 PERMANENT ERROR INDICATOR, COLUMNS 53-54, IS INVALID.

Severity: Terminal

Specification Type: T

Explanation: The indicator specified in columns 53 and 54 is not 01-99, L1-L9, LR, or H1-H9.

RPG–0247 WAIT TIME, COLUMNS 55-57, IS INVALID. ASSUME SYSTEM CONVENTION FOR TIMEOUT, 180 SECONDS.

Severity: Warning

Specification Type: T

Explanation: The wait time entry specified in columns 55 through 57 is not 1-999 or blank.

RPG–0248 RECORD AVAILABLE INDICATOR, COLUMNS 58-59, IS INVALID.

Severity: Terminal

Specification Type: T

Explanation: The record available indicator specified in columns 58 and 59 is not 01-99, L1-L9, LR, or H1-H9.

RPG–0249 LAST FILE PROCESSED, COLUMN 60, IS NOT L OR BLANK.

Severity: Terminal

Specification Type: T

Explanation: The last file processed entry in column 60 is not L or blank. Enter L in column 60 if the BSCA input file must be processed last; blank if not.

RPG–0250 POLLING CHARACTERS COLUMNS 61-62, CONTAIN AN INVALID CHARACTER FOR THE CODE TYPE ENTRY IN COLUMN 18.

Severity: Terminal

Specification Type: T

Explanation: The polling characters specified in columns 61 and 62 are invalid, or are missing on a line configuration that requires them.

(A list of the valid polling characters is included in the *IBM System/34 Data Communications Reference Manual*, SC21-7703.)

RPG–0251 ADDRESSING CHARACTERS, COLUMNS 63-64, ARE INVALID FOR THE CODE TYPE ENTRY IN COLUMN 18. ENTRY IS IGNORED.

Severity: Terminal

Specification Type: T

Explanation: The addressing characters in columns 63 and 64 are invalid for the code type specified in column 18, or are missing on a line configuration that requires them. (A list of the valid addressing characters is included in the *IBM SYSTEM/34 Data Communications Reference Manual*, SC21-7703.)

RPG–0258 SPACE AND/OR SKIP ENTRIES IN COLUMNS 17-22 INVALID FOR DEVICE, ASSUME BLANK.

Severity: Warning

Specification Type: O

Explanation: The space and/or skip entries in columns 17 through 22 are invalid for the device. Leave columns 17 through 22 blank for all devices except CRT and PRINTER.

RPG–0259 SKIP ENTRIES IN COLUMNS 19-22 INVALID OR GREATER THAN FORM LENGTH SPECIFIED, ASSUME BLANK.

Severity: Warning

Specification Type: O

Explanation: The skip entries in columns 19 through 22 are not specified properly or they exceed the form length in your line counter specifications.

RPG–0260 INVALID SPACE ENTRIES IN COLUMNS 17-18, ASSUME SPACE 1 AFTER OR BLANK.

Severity: Warning

Specification Type: O

Explanation: The space entries in columns 17 and 18 are not a number from 0 to 3 or blank. If the space entries in columns 17 and 18 are invalid and the skip-after entry is blank, a space-after of 1 is assumed. When skip- and space-before entries are valid but space-after is not, space-after is assumed blank.

RPG–0261 FETCH OVERFLOW ENTRY IN COLUMN 16 INVALID FOR DEVICE, ASSUME BLANK.

Severity: Warning

Specification Type: O

Explanation: The fetch overflow entry specified in column 16 is invalid for the device. Specify fetch overflow for printer files only. Blank is assumed; therefore, no fetch overflow is done.

RPG–0262 OVERFLOW INDICATOR INVALID FOR AN EXCEPT RECORD.

Severity: Terminal

Specification Type: O

Explanation: An overflow indicator must not be specified for an exception record (E in column 15).

RPG–0263 FETCH OVERFLOW INVALID WITH OVERFLOW INDICATOR ENTERED IN COLUMNS 23-31, ASSUME NO FETCH.

Severity: Warning

Specification Type: O

Explanation: An overflow indicator and fetch overflow (F in column 16) must not be specified on the same output line. Blank in column 16 is assumed; therefore, no fetch overflow is done.

RPG–0264 OVERFLOW INDICATOR USED IS NOT ASSIGNED TO THIS FILE.

Severity: Terminal

Specification Type: O

Explanation: The overflow indicator specified was not assigned to this file in your file description specifications.

RPG–0265 1P INDICATOR INVALID WITH TOTAL OR EXCPT RECORDS.

Severity: Warning

Specification Type: O

Explanation: First page (1P) indicator must not be specified for total or exception records. Specify the 1P indicator with heading and detail records only.

RPG–0266 FETCH OVERFLOW INVALID WITH 1P INDICATOR, ASSUME NO FETCH OVERFLOW.

Severity Warning

Specification Type: O

Explanation: A fetch overflow line (F in column 16) must not be conditioned by the 1P indicator.

RPG–0267 1P INDICATOR INVALID FOR COMBINED FILES.

Severity: Terminal

Specification Type O

RPG–0269 INVALID INDICATORS USED IN AN AND RELATIONSHIP WITH 1P.

Severity: Terminal

Specification Type: O

Explanation: Only external indicators (U1-U8) can be specified in an AND relationship with the 1P indicator.

RPG–0270 END POSITION ENTRY IN COLUMNS 40-43 FOR CONSTANT, DATA STRUCTURE, EDIT WORD, FIELD, TABLE, OR ARRAY EXCEEDS RECORD LENGTH.

Severity: Terminal

Specification Type: O

Explanation: The end position entry in columns 40 through 43 exceeds the record length specified in your file description specifications.

RPG–0271 LENGTH OF TABLE ELEMENT ARRAY, ARRAY ELEMENT, DATA STRUCTURE, OR FIELD EXCEEDS RECORD LENGTH.

Severity: Terminal

Specification Type: O

Explanation: The length specified for a table element, array, array element, data structure, or field exceeds the record length specified in your file description specifications.

RPG–0272 END POSITION ENTRY IN COLUMNS 40-43 FOR CONSTANT, EDIT WORD, FIELD, DATA STRUCTURE, OR ARRAY TOO LOW.

Severity: Terminal

Specification Type: O

Explanation: The end position entry in columns 40 through 43 is too small to allow data to be written, printed, or displayed in its entirety.

RPG–0273 OUTPUT INDICATORS IN COLUMNS 23-31 MISSING OR ALL NEGATIVE.

Severity: Warning

Specification Type: O

Explanation: No output indicators are specified in columns 23 through 31 or all those indicators specified are negative. Output may be written when not desired. Specify at least one positive indicator to condition output records to ensure that output is written only when desired.

RPG–0274 INDICATORS MISSING FOR AN AND OR OR LINE.

Severity: Warning

Specification Type: O

Explanation: No conditioning indicators were specified in columns 23 through 31 for an AND or OR line.

RPG–0275 CONDITIONING INDICATORS COLUMNS 23-31, MAY NOT BE SPECIFIED ON THE OUTPUT LINE CONTAINING THE WORKSTN FORMAT NAME.

Severity: Terminal

Specification Type: O

Explanation: Conditioning indicators must not be used to condition output of a WORKSTN format except by entire record.

RPG–0276 INVALID EDIT CODE IN COLUMN 38.

Severity: Terminal

Specification Type: O

Explanation: The edit code specified in column 38 is not one of the following: 1-4, A-D, J-M, X, Y, Z, or blank.

RPG–0277 INVALID EDIT WORD SIZE.

Severity: Terminal

Specification Type: O

Explanation: The number of replaceable characters in this edit word (columns 45 through 70) must equal the length of the field to be edited.

RPG–0278 EDIT CODES INVALID WITH FIELDS OTHER THAN UNPACKED NUMERIC FIELDS OR CONSTANTS OTHER THAN * OR THE CURRENCY SYMBOL.

Severity: Terminal

Specification Type: O

RPG–0279 CONSTANTS IN COLUMNS 45-70 INVALID FOR X, Y AND Z EDIT CODES.

Severity: Terminal

Specification Type: O

Explanation: Edit codes X, Y, and Z must not be specified for edit words with the currency symbol or * in columns 45 through 47.

RPG–0280 FIELD LENGTH FOR Y EDIT CODE LESS THAN 3 OR GREATER THAN 6. IF GREATER THAN 6 ONLY LOW ORDER 6 DIGITS ARE EDITED.

Severity: Warning

Specification Type: O

Explanation: The field edited by the Y edit code is not from 3 to 6 characters long. If less than 3 characters long, the field is not edited; if more than 6 characters long, only the 6 low-order digits are edited.

RPG–0281 DECIMAL POSITIONS INVALID FOR FIELD EDITED BY Y CODE.

Severity: Terminal

Specification Type: O

Explanation: Decimal positions must not be specified for a field edited by the Y code.

RPG–0282 NAME OF FIELD TO BE EDITED, BY CODE SPECIFIED IN COLUMN 38, MISSING.

Severity: Terminal

Specification Type: O

Explanation: An edit code is specified in column 38, but the name of the field to be edited is not entered in columns 32 through 37.

RPG–0283 INVALID FILE TYPE FOR OUTPUT RECORD.

Severity: Terminal

Specification Type: O

Explanation: The file specified in columns 7 through 14 of your output specifications is not an update file, combined file, output file, or a file associated with ADD.

RPG–0287 COLUMNS 32-37 NOT BLANK WHEN 40-43 IS K1-K8.

Severity: Terminal

Specification Type: O

Explanation: Format names and operation codes may not be in fields; they must be constants.

RPG–0288 BLANK AFTER ENTRY IN COLUMN 39 INVALID WITH RESERVED WORD OTHER THAN PAGE, ASSUME BLANK.

Severity: Warning

Specification Type: O

RPG–0290 *PLACE PRECEDES ALL FIELD NAMES AND CONSTANTS.

Severity: Terminal

Specification Type: O

Explanation: When *PLACE is used, it must be specified after fields that are to be placed in different locations. Specify the fields to be moved before you specify *PLACE.

RPG–0291 INVALID ENTRIES IN COLUMNS 38, 39, OR 44-74 FOR OUTPUT OPERATION, ASSUME BLANKS.

Severity: Terminal

Specification Type: O

Explanation: Columns 38, 39, and 44 through 74 must be blank for *PLACE.

RPG–0292 TOO MANY AND/OR LINES.

Severity: Terminal

Specification Type: I, O

Explanation: More than 20 AND/OR lines are specified in your input or output specifications.

RPG–0293 BLANK AFTER SPECIFIED FOR A CONSTANT.

Severity: Warning

Specification Type: O

Explanation: Blank after (B in column 39) should not be specified for a constant because constants will be blanked out whenever they are used.

RPG–0294 INVALID ENTRY IN COLUMN 42, ASSUME BLANK.

Severity: Warning

Specification Type: I

Explanation: Column 42 must be left blank.

RPG–0295 INVALID ENTRY IN COLUMN 16, ASSUME BLANK.

Severity: Warning

Specification Type: O

Explanation: Column 16 was not F, R, or blank.

RPG–0296 FORMAT NAME FOR WORKSTN FILE OUTPUT REQUIRED BUT MISSING OR MULTIPLE FORMAT NAMES SPECIFIED FOR ONE OUTPUT RECORD.

Severity: Terminal

Specification Type: O

Explanation: One format name is required, but no more than one format name is allowed for each WORKSTN output record.

RPG–0300 VALUE OF ARRAY INDEX EXCEEDS NUMBER OF ARRAY ELEMENTS.

Severity: Terminal

Specification Type: O

Explanation: The array index specified exceeds the number of elements in the array. The index must not exceed the number of array elements specified for the array in columns 36 through 39 of your extension specifications.

RPG–0302 BLANK AFTER ENTRY IN COLUMN 39 INVALID FOR LOOK AHEAD FIELD, ASSUME BLANK.

Severity: Warning

Specification Type: O

Explanation: Column 39 must be blank for a look-ahead field.

RPG–0304 INVALID INDICATOR OR IMPROPER USE OF A VALID INDICATOR.

Severity: Terminal

Specification Type: I, C, O

Explanation: The indicator specified is invalid or used improperly, or a data structure is specified incorrectly. If the indicator is invalid, make the proper indicator entry (only indicators 01-99, H1-H9, L1-L9, LR, U1-U8, OA-OG, OV, KA-KN, KP-KY can be assigned). If the indicator has been used improperly, see the restrictions concerning proper use of indicators under *Operation Codes, Setting Indicators* in Chapter 10 of this manual. A data structure (DS) is specified incorrectly in columns 19 and 20 of the input specifications.

RPG–0305 INDICATOR ASSIGNED BUT NOT USED TO CONDITION OPERATIONS.

Severity Warning

Specification Type: I, C, O

RPG–0306 INDICATOR USED TO CONDITION OPERATIONS BUT NOT ASSIGNED.

Severity: Terminal

Specification Type: I, C, O

Explanation: All indicators except LR, MR, 1P, L0, and, if a WORKSTN file is specified, KA-KN and KP-KY must be assigned before they can be used to condition operations.

RPG–0307 FILENAME DEFINED BUT NEVER USED. SPECIFICATION IS DROPPED.

Severity: Warning

Specification Type: F

Explanation: A filename was defined in columns 7 through 14, but no input or output specifications exist for this file. Remove the file description specifications for files not used in the program.

RPG–0308 SEQUENCING INVALID FOR FILE WITH NO MATCH FIELD, ASSUME COLUMN 18 ON FILE DESCRIPTION SPECIFICATION BLANK.

Severity: Warning

Specification Type: F

RPG–0309 SEQUENCE ENTRY INVALID OR BLANK FOR FILE WITH MATCH FIELD SPECIFIED, ASSUME FIRST VALID SEQUENCE OR A.

Severity: Warning

Specification Type: F

Explanation: No sequence entry (A or D) or an invalid sequence entry is specified in column 18 for a file with match fields. For a primary file, A is assumed. If no valid sequence entry is specified for a secondary file, the primary sequence value is assumed.

RPG–0310 EXTENSION CODE SPECIFIED IN COLUMN 39 ON FILE DESCRIPTION SPECIFICATION FOR THIS FILE BUT EXTENSION SPECIFICATION MISSING.

Severity: Terminal

Specification Type: F

Explanation: An extension code is specified (E in column 39) in your file description specifications, but no extension specifications were supplied. Either supply the proper extension specifications or delete the E for column 39 of your file description specifications if no extension specifications are required for this program.

RPG–0311 AN EXTENSION OR LINE COUNTER SPECIFICATION WAS PROVIDED FOR THIS FILE BUT AN EXTENSION CODE WAS NOT ENTERED IN COLUMN 39 OF THE FILE DESCRIPTION SPECIFICATIONS.

Severity: Warning

Specification Type: F

Explanation: An extension code (E in column 39 of file description) is assumed.

RPG–0314 FIELD, TABLE OR ARRAY NAME DEFINED BUT NEVER USED.

Severity: Warning

Specification Type: E, I, C

RPG–0315 FIELD NAME USED BUT NEVER DEFINED, OR TABLE NAME OR ARRAY ELEMENT USED AS AN ARRAY INDEX.

Severity: Terminal

Specification Type: C, O

RPG–0316 INVALID DEFINITION FOR RESERVED WORD, ASSUME VALID DEFINITION.

Severity: Terminal

Specification Type: I, C

Explanation: The field named by one of the RPG II reserved words is not specified according to the predefined format. Reserved words are not valid in a data structure.

RPG–0317 NUMBER OF DECIMAL POSITIONS SPECIFIED EXCEEDS FIELD LENGTH.

Severity: Terminal

Specification Type: I, C, O

RPG–0318 MISSING A RECORD CONDITIONED BY 1P AND FORMS POSITIONING SPECIFIED ON CONTROL SPECIFICATION.

Severity: Warning

Specification Type: H, O

Explanation: Repetitive 1P output for forms positioning is specified in the control specifications, but 1P is not used to condition an output record.

RPG–0319 NO DATA FOR ALTERNATE COLLATING SEQUENCE OR FILE TRANSLATION.

Severity: Terminal

Specification Type: H

Explanation: Alternate collating sequence or file translation is specified in the header line, but no alternate collating sequence table or file translation table was supplied.

RPG–0320 INVALID ALTERNATE COLLATING SEQUENCE DATA RECORD.

Severity: Terminal

Specification Type: Not applicable

Explanation: Columns 1 through 6 in the alternate collating sequence data records do not contain ALTSEQ.

RPG–0321 INVALID OR UNDEFINED OR A TABLE FILENAME ON FILE TRANSLATION DATA RECORD.

Severity: Terminal

Specification Type: Not applicable

Explanation: The entry in columns 1 through 8 of the file translation data record is invalid, was not previously defined, or is a table filename. Make the entry in columns 1 through 8 of each file translation data record a filename previously defined in file description specifications or the characters *FILESƀƀ (ƀ = blank). The entry must not be a table filename.

RPG–0322 ALTERNATE COLLATING SEQUENCE OR FILE TRANSLATION DATA INVALID.

Severity: Terminal

Specification Type: Not applicable

Explanation: The data supplied for alternate collating sequence or file translation is invalid (not 0-9 and A-F).

RPG–0323 TABLE OR ARRAY NAME SPECIFIED MORE THAN ONCE ON EXTENSION SPECIFICATIONS OR REDEFINED ELSEWHERE. DATA STRUCTURE SPECIFIED MORE THAN ONCE ON INPUT SPECS.

Severity: Terminal

Specification Type: E, C, I

Explanation: A table or array name can be specified only once in extension specifications and cannot be redefined elsewhere in the program. Make sure that the names specified in columns 27 through 32 and columns 46 through 51 of the extension specifications are specified only once, and that the table or array is defined only on the extension specifications. A data structure has been defined more than once on input specifications.

RPG–0324 TOTAL LENGTH OF ALL CONTROL OR ALL MATCHING FIELDS EXCEEDS 144 CHARACTERS.

Severity: Terminal

Specification Type: I

Explanation: The total length of all control or all matching fields is too long. Make the total length of all matching fields (M1-M9) or all control fields (L1-L9) equal to or less than 144.

RPG–0325 ALL PRIMARY AND SECONDARY FILES CONDITIONED BY EXTERNAL INDICATORS.

Severity: Warning

Specification Type: I

Explanation: When all primary and secondary files are conditioned by external indicators (U1-U8), be sure all indicators are not off. If they are all off, the job will not be done.

RPG–0326 COMPILE TIME TABLES SPECIFIED BUT NO DATA FOUND.

Severity: Terminal

Specification Type: Not applicable

Explanation: Compile-time table specified (from filename in columns 11 through 18 of extension specifications blank), but no table input records were supplied after the source program.

RPG–0327 SPLIT CONTROL FIELDS MAY NOT HAVE PARTS THAT ARE PACKED.

Severity: Terminal

Specification Type: I

Explanation: All parts of a split control field must be either packed or unpacked.

RPG–0328 PACKED OR BINARY DATA ALLOWED FOR THIS DEVICE BUT MAY CAUSE ERRORS.

Severity: Warning

Specification Type: I, O

Explanation: Packed or binary data is allowed for this device but in some cases may cause errors to the device.

RPG–0329 PACKED OR BINARY DATA NOT VALID FOR DEVICE, ASSUME ZONED DECIMAL.

Severity: Warning

Specification Type: I, O

Explanation: Packed or binary data can be specified for disk and WORKSTN files only. Data errors may occur if program is executed.

RPG–0330 ALPHAMERIC FIELD SPECIFIED AS PACKED OR BINARY.

Severity: Terminal

Specification Type: O

Explanation: Packed data cannot be specified for alphameric fields. Specify packed data for numeric fields only.

RPG–0331 NO INPUT SPECIFICATIONS FOUND.

Severity: Terminal

Specification Type: Not applicable

Explanation: No valid input specifications are supplied for this job. Input specifications are required for update or combined files and input files unless the input comes from the KEYBORD device.

RPG–0332 SEQUENCE ERROR FOUND IN COMPILE TIME TABLE/ARRAY.

Severity: Terminal

Specification Type: Not applicable

Explanation: Compile time table or array is not in the sequence specified in columns 45 or 57. Make sure the data is in the sequence specified (A or D) in column 45 or 57.

RPG–0333 TABLE/ARRAY FULL OR NO TABLES/ARRAYS FOR FOLLOWING DATA.

Severity: Warning

Specification Type: Not applicable

Explanation: Either too much data is supplied for the table or array, or no table or array is defined for the data supplied. Make sure that the data supplied does not exceed the maximum table size, or that a table or array is defined for the data you supply. The system accepts no more data for tables or arrays.

RPG–0334 SHORT TABLE.

Severity: Warning

Specification Type: Not applicable

Explanation: The number of entries supplied is less than the maximum number of entries the table can contain. The remaining entries are filled with blanks or zeros.

RPG–0335 EDIT WORD SPECIFIED WITH OTHER THAN UNPACKED NUMERIC FIELDS.

Severity: Terminal

Specification Type: O

Explanation: Edit words are allowed only with unpacked numeric fields.

RPG–0337 INVALID SEQUENCE FOR EXIT AND RLABL OPERATION CODES.

Severity: Terminal

Specification Type: C

Explanation: The RLABL operation code does not immediately follow an EXIT operation.

RPG–0338 SUBR MUST BE USED WITH EXIT OPERATION CODE.

Severity: Terminal

Specification Type: C

Explanation: The entry specified in factor 2 of an EXIT operation does not start with SUBR.

RPG–0339 AN OUTPUT REFERENCE IS REQUIRED FOR EACH UPDATE FILE, OR IF ADD IS SPECIFIED.

Severity: Terminal

Specification Type: F

Explanation: The proper output specifications have not been specified for the update file.

RPG–0340 CONTROL/TRIBUTARY, COLUMN 17, CONTAINS A BLANK FOR A MULTIPOINT LINE. ASSUME T.

Severity: Warning

Specification Type: T

Explanation: Column 17 was left blank for a multipoint line (M in column 15).

RPG–0341 CONTROL/TRIBUTARY, COLUMN 17, CONTAINS A T FOR A SWITCHED OR A POINT TO POINT NETWORK. ASSUME BLANK.

Severity: Warning

Specification Type: T

Explanation: Column 17 contains a T for a point-to-point network (P in column 15).

RPG–0342 TRANSPARENT MODE IS SPECIFIED, COLUMN 19, WHEN ASCII CONTROL CHARACTERS, COLUMN 18, ARE TO BE USED.

Severity: Terminal

Specification Type: T

Explanation: The transparent mode cannot be specified on an adapter using ASCII data link characters.

RPG–0343 AUTOANSWER, COLUMN 20, IS NOT BLANK FOR NON-SWITCHED NETWORK.

Severity: Terminal

Specification Type: T

Explanation: Column 20 contains an entry for a network that is not switched. Leave column 20 blank for a network that is not switched.

RPG–0346 COLUMN 32 IS NOT BLANK FOR A NON-SWITCHED NETWORK.

Severity: Terminal

Specification Type: T

Explanation: Column 32 must be left blank for a non-switched network.

RPG–0347 IDENTIFICATION FOR THIS STATION, COLUMNS 33-39, CONTAINS AN ARRAY.

Severity: Terminal

Specification Type: T

Explanation: An array name was used as the station identification. Enter the table element or field name to be used as the station identification in columns 33 through 39. If you want to use an array element as the station identification, you must use calculation specifications to move the contents of the array element into the field you specify in columns 33 through 39.

RPG–0348 COLUMN 40 IS NOT BLANK FOR A NON-SWITCHED NETWORK.

Severity: Terminal

Specification Type: T

Explanation: Column 40 must be left blank for a non-switched network.

RPG–0349 IDENTIFICATION FOR THE REMOTE STATION, COLUMNS 41-47, CONTAINS AN ARRAY.

Severity: Terminal

Specification Type: T

Explanation: An array name was used as the remote station identification. Enter the table element or field name to be used as the remote station identification in columns 41 through 47. If you want to use an array element as the remote station identification, you must use calculation specifications to move the contents of the array element into the field you specify in columns 41 through 47.

RPG–0350 RECORD AVAILABLE INDICATOR IS PRESENT ON TRANSMIT FILE, OR A PROGRAM WITH ONLY 1 BSCA FILE. INDICATOR IS DROPPED.

Severity: Warning

Specification Type: T

Explanation: A record-available indicator was specified for a transmit file or in a program that has only one BSCA file. Remove the record-available indicator or define the other BSCA file if a transmit interspersed with a receive program is desired.

RPG–0351 LAST FILE PROCESSED, COLUMN 60, IS NOT A BLANK ON A TRANSMIT FILE OR A PRIMARY INPUT FILE. ENTRY IS IGNORED.

Severity: Warning

Specification Type: T

Explanation: L was entered in column 60 for a transmit file or for a primary input file. Remove the L from column 60 if the file is a transmit file. If it is a primary input file, remove the L or change the file designation to secondary.

RPG–0352 POLLING CHARACTERS WERE GIVEN ON OTHER THAN A TRANSMIT FILE ON A MULTIPOINT NETWORK. ENTRY IS IGNORED.

Severity: Warning

Specification Type: T

Explanation: Polling characters are specified in columns 61 and 62 for a file other than a transmit file on a multipoint network.

RPG–0353 THERE IS AN ENTRY IN THE ADDRESSING CHARACTERS COLUMNS 63-64, ON A FILE THAT IS NOT A MULTIPOINT RECEIVER FILE. ENTRY IS IGNORED.

Severity: Warning

Specification Type: T

RPG–0354 CORRESPONDING FILE DESCRIPTION SPECIFICATION FILE IS NOT A BSC FILE.

Severity: Terminal

Specification Type: T

Explanation: A BSCA device entry was not made for this file on the file description specifications sheet.

RPG–0356 PACKED OR BINARY FIELD SPECIFIED IN A FILE WITHOUT THE TRANSPARENT FEATURE.

Severity: Terminal

Specification Type: T

RPG–0357 THE FILE CORRESPONDING TO THIS TRANSMITTER SPECIFICATION IS NOT AN OUTPUT FILE ON THE FILE DESCRIPTION SPECIFICATIONS.

Severity: Terminal

Specification Type: T

RPG–0358 CORRESPONDING FILE DESCRIPTION SPECIFICATIONS FILE IS NOT DEFINED AS AN INPUT FILE FOR THIS RECEIVE FILE.

Severity: Terminal

Specification Type: T

RPG–0360 THERE IS NO TELECOMMUNICATIONS SPECIFICATION FOR A FILE DEFINED AS A BSCA FILE ON THE FILE DESCRIPTION SPECIFICATIONS.

Severity: Terminal

Specification Type: T

RPG–0361 LOOK AHEAD FIELDS SPECIFIED FOR BSC FILE.

Severity: Terminal

Specification Type: T

Explanation: Look-ahead fields are not allowed for a BSCA file.

RPG–0363 MATCHING FIELDS DEFINED FOR A FILE DESIGNATED TO BE THE LAST FILE PROCESSED IN COLUMN 60 OF THE TELECOMMUNICATIONS SPECIFICATIONS.

Severity: Terminal

Specification Type: T

Explanation: Matching fields were defined for a file designated as the last file to be processed (L in column 60). Remove the matching fields definition if the file was the last one to be processed, or remove the L entry in column 60.

RPG–0364 FOR A TRANSMIT THEN RECEIVE BSCA PROGRAM, IF END-OF-FILE IS SPECIFIED FOR ANY INPUT FILE, E IS ASSUMED IN COLUMN 17 OF THE BSCA INPUT FILE.

Severity: Warning

Specification Type: T

Explanation: E was entered in column 17 of some input files, but not for the BSCA file, which has an L in column 60 of the telecommunications specifications sheet. End of file is assumed if end of file (E in column 17 of the file description specifications sheet) is specified for any input file the program uses.

RPG–0365 ITB IS SPECIFIED ON A FILE WITHOUT BLOCKED RECORDS. ITB IS DROPPED.

Severity: Warning

Specification Type: T

Explanation: The intermediate block check specification (I in column 52) is ignored.

RPG–0366 AUTOANSWER, COLUMN 20, IS BLANK FOR A SWITCHED NETWORK.

Severity: Terminal

Specification Type: T

Explanation: Column 20 was left blank for a switched network. Column 20 should contain M, A, or B.

RPG–0368 THE FIELD OR TABLE HOLD AREA FOR A STATION IDENTIFICATION, COLUMNS 33-39 OR 41-47, IS MORE THAN FIFTEEN CHARACTERS IN LENGTH.

Severity: Terminal

Specification Type: T

Explanation: Either the field or table hold area used for a station identification (columns 33 through 39 or 41 through 47) contains more than 15 characters, or the dial number (columns 21 through 31) contains more than 12 digits. The field or table hold area used for a station identification should be numeric and from 2 to 15 characters long.

RPG–0369 WARNING-ONLY ONE INPUT/OUTPUT AREA WAS SPECIFIED FOR A BSCA FILE.

Severity: Warning

Specification Type: T

Explanation: Because only one I/O area is specified for a BSCA file, processing time is likely to be slow.

RPG–0370 THE LINE CONFIGURATION AND LINE CONTROL ENTRIES, COLUMNS 17-47, ARE NOT THE SAME ON EACH TELECOMMUNICATIONS SPECIFICATION.

Severity: Terminal

Specification Type: T

RPG–0371 WARNING-THE STATION IDENTIFICATION, COLUMNS 33-39 OR 41-47, HAS BEEN DEFINED AS ONLY ONE CHARACTER IN LENGTH. THE CHARACTER WILL BE DUPLICATED SO A TWO CHARACTER IDENTIFICATION WILL BE USED.

Severity: Warning

Specification Type: T

RPG–0372 A B IN COLUMN 37 OF THE CONTROL SPECIFICATION IS AN INVALID ENTRY IN A BSCA PROGRAM.

Severity: Terminal

Specification Type: H

RPG–0373 THE SAME FILE NAME WAS GIVEN ON TWO TELECOMMUNICATION SPECIFICATIONS.

Severity: Terminal

Specification Type: T

RPG–0374 ENTRY IN COLUMN 16 INVALID.

Severity: Warning

Specification Type: F, O

Explanation: The entry in column 16 of the file description specifications is not P, S, C, R, T, D, or blank, or the entry in column 16 of the output specifications is not F (fetch overflow), R (release), A (if ADD is specified in columns 16 through 18), or blank. For the file description specifications, blank is assumed if the file is an output file; otherwise, S is assumed.

RPG–0376 INVALID OR BLANK NAME IN COLUMNS 75-80 OF CONTROL SPECIFICATION, ASSIGNED NAME ASSUMED.

Severity: Warning

Specification Type: H

Explanation: The entry in columns 75 through 80 of your control specification is neither a valid RPG program name nor blanks. Blanks are assumed.

RPG–0377 RECORD ADDRESS FILE, COLUMN 31, IS NOT ALLOWED ON A BSCA FILE.

Severity: Terminal

Specification Type: F

RPG–0388 FACTOR 1 MUST BE EITHER A FIELD NAME OR A LITERAL WHEN USED WITH THIS OPERATION.

Severity: Terminal

Specification Type: C

RPG–0390 SEQUENCE CHECKING IS NOT PERFORMED ON EXECUTION TIME ARRAYS.

Severity: Warning

Specification Type: E

Explanation: Sequence must be specified if high or low LOKUP is to be done; however, no sequence checking is done at input time. A sequenced array is assumed.

RPG–0391 A FIELD WITH A LENGTH GREATER THAN 8 CHARACTERS CANNOT BE USED IN FACTOR 1 WITH DEBUG OPERATION.

Severity: Terminal

Specification Type: C

RPG–0392 LAST ENTRY IN ONE OR MORE COMPILE TIME TABLE/ARRAYS WAS BLANK.

Severity: Warning

Specification Type: E

Explanation: The compile time table/array contains fewer entries than the number of entries specified in columns 36 through 39 of the extension specifications.

RPG–0394 ADD OR DEL IN COLUMNS 16-18 NOT ALLOWED ON AND/OR LINES, ASSUME BLANK.

Severity: Terminal

Specification Type: O

Explanation: ADD or DEL must be specified on the file and record identification line of the output specifications, and applies to all AND/OR lines. Columns 16 through 18 are assumed to be blank.

RPG–0397 FILE DESCRIBED AS 'ADD' TYPE FILE. EACH OUTPUT RECORD LINE MUST HAVE 'ADD' IN COLUMNS 16-18. ASSUME 'ADD'.

Severity: Warning

Specification Type: O

Explanation: The ADD function (A in column 66) was specified in the file description specifications for this file, but ADD was not specified in columns 16 through 18 of the output sheet for each record type output line to be written.

RPG–0398 COLUMNS 54-59, INVALID FOR DEVICE, OR WRONG ENTRY. ASSUME BLANK.

Severity: Terminal

Specification Type: F

Explanation: Columns 54 through 59 contain an entry for a file that was not assigned to a SPECIAL device (SPECIAL in columns 40 through 46).

RPG–0399 INVALID ENTRY IN COLUMNS 54-59.

Severity: Terminal

Specification Type: F

Explanation: The entry in columns 54 through 59 of your file description specifications for a SPECIAL file is neither SUBRxx (x = any alphabetic character) nor SRyzzz (y = one of 15 valid characters; z = one of 16 valid characters). Enter the name of the user-written subroutine (SUBRxx) or IBM-written subroutine (SRyzzz) that will perform the input/output operations for the SPECIAL file.

RPG–0400 INVALID MODE OF PROCESSING ENTRY IN COLUMN 28.

Severity: Terminal

Specification Type: F

Explanation: The entry in column 28 is not R, L, or blank. R is assumed for valid file type or mode of processing.

RPG–0401 ONLY ONE TABLE/ARRAY PER FILENAME ALLOWED FOR THIS DEVICE.

Severity: Terminal

Specification Type: E

RPG–0403 INVALID LENGTH OF KEY FIELD IN COLUMNS 29-30, ASSUME 3.

Severity: Terminal

Specification Type: F

Explanation: The length of key field entry in columns 29 and 30 is not specified properly. The entry must be 29 or less for unpacked keys, or 8 or less for packed keys.

RPG–0404 INVALID RECORD ADDRESS TYPE ENTRY IN COLUMN 31, ASSUME A.

Severity: Terminal

Specification Type: F

Explanation: The entry in column 31 is not A, P, I, or blank, or the entry is I when sequential within limits is specified.

RPG–0405 INVALID KEY START LOCATION ENTRY IN COLUMNS 35-38, ASSUME 1.

Severity: Terminal

Specification Type: F

Explanation: Columns 35 through 38 do not contain a number from 1 to 4096 for an indexed file.

RPG–0406 INVALID MAIN STORAGE INDEX ENTRY IN COLUMNS 60-65, ASSUME BLANK.

Severity: Warning

Specification Type: F

Explanation: Columns 60 through 65 do not contain a number from 6 to 9999 for an indexed file processed randomly.

RPG–0407 INVALID FILE ADDITION OR UNORDERED ENTRY IN COLUMN 66, ASSUME A.

Severity: Terminal

Specification Type: F

Explanation: The file addition or unordered load entry in column 66 is not A, U, or blank.

RPG–0408 NUMBER OF EXTENTS ENTRY IN COLUMNS 68-69 IS INVALID OR NOT ALLOWED WITH THIS DEVICE.

Severity: Terminal

Specification Type: F

Explanation: Columns 68 and 69 must be blank for nondisk devices. If the device is disk, the columns contain an entry that is not 01 or blank. Blank is assumed.

RPG–0409 ENTRY OF K MADE IN COLUMN 31 FOR RECORD ADDRESS TYPE, ASSUME A.

Severity: Warning

Specification Type: F

Explanation: An entry of K is not allowed in column 31 for record address type.

RPG–0410 EXTENSION SPECIFICATION LINE BLANK.

Severity: Terminal

Specification Type: Not applicable

Explanation: An E was specified in column 39 of a file description line, but no extension specifications were found in the program.

RPG–0411 RESERVED COLUMNS 71-74 ARE NOT BLANK.

Severity: Warning

Specification Type: T

RPG–0451 CONTINUATION, K IN COLUMN 53, INVALID FOR DEVICE OR FILE TYPE.

Severity: Terminal

Specification Type: F

Explanation: Continuation is allowed only for SPECIAL, WORKSTN, or delete-capable disk files. An A in column 66 of the file description specifications is invalid for a delete-capable, output file load. Continuation is ignored.

RPG–0452 ENTRY IN COLUMNS 54-59 OF A CONTINUATION RECORD IS INVALID OR MISSING.

Severity: Terminal

Specification Type: F

Explanation: For a SPECIAL file you must enter a table or array name in columns 54 through 59. The table or array must be defined on the extension specifications. Blanks are invalid. The continuation record is ignored. For a WORKSTN file, you can enter the following keywords: NUM, ID, IND, SAVDS, SLN, FMTS, INFSR, or INFDS. For a disk file, the valid entry is RECNO.

RPG–0453 CONTINUATION ENTRY IN COLUMNS 54-59 IS REPEATED FOR A FILE. SECOND ENTRY IS IGNORED.

Severity: Warning

Specification Type: F

Explanation: For a SPECIAL file only one continuation is allowed. For a WORKSTN file, each keyword (NUM, ID, IND, SAVDS, SLN, FMTS, INFSR, INFDS) may appear only once. For a disk file, RECNO may appear only once. The first keyword is used, and later duplications are dropped.

RPG–0455 COLUMNS 7-52 AND 66-72 OR 68-72 ON FMTS LINE ARE NOT BLANK FOR A CONTINUATION LINE, ASSUME BLANK.

Severity: Warning

Specification Type: F

Explanation: An FMTS continuation line can contain a format load member name in columns 60-67.

RPG–0456 IMPROPER OR NO CONTINUATION LINE FOUND FOR DIRECT FILE LOAD OR ADD TO RANDOM FILE.

Severity: Terminal

Specification Type: F

Explanation: File addition (A in column 66) for delete-capable file processed randomly or direct file load of delete-capable file requires a file description specifications continuation line with the keyword RECNO and associated value.

RPG–0461 INVALID ENTRY IN COLUMN 70, ASSUME BLANK.

Severity: Warning

Specification Type: F

Explanation: Column 70 must be blank.

RPG–0462 CONTINUATION, K IN COLUMN 53, INVALID FOR MAIN FILE DESCRIPTION LINE, ASSUME BLANK.

Severity: Warning

Specification Type: F

Explanation: K is valid only on a continuation file description specification.

RPG–0500 FROM NAME INVALID OR MISSING FOR RECORD ADDRESS FILE.

Severity: Terminal

Specification Type: E

Explanation: The from filename entry in columns 11 through 18 is missing or not specified properly for a record address file.

RPG–0502 FROM FILENAME IS A RECORD ADDRESS FILE THAT IS USED MORE THAN ONCE.

Severity: Terminal

Specification Type: E

Explanation: The record address file named in columns 11 through 18 is used more than once in the extension specifications. Only one record address file is allowed in a program.

RPG–0503 TO FILE NAME FOR A RECORD ADDRESS FILE TYPE IS EITHER 1-NOT A PRIMARY, SECONDARY, OR DEMAND FILE, OR 2-MISSING, INVALID OR A NON-DISK FILE.

Severity: Terminal

Specification Type: E

Explanation: The to filename entry in columns 19 through 26 must be a primary or secondary disk file, in order to be processed by a record address file.

RPG–0504 TO FILENAME IS INCORRECT FILE TYPE.

Severity: Terminal

Specification Type: E

Explanation: The filename specified in columns 19 through 26 is not an input, output, or update file.

RPG–0510 LENGTH GIVEN FOR BINARY FIELD IS NOT 2 OR 4, ASSUME 2.

Severity: Terminal

Specification Type: I, O

RPG–0511 PACKED LENGTH GREATER THAN 8 FOR A FIELD, TABLE OR ARRAY.

Severity: Terminal

Specification Type: I, O

RPG–0516 MORE THAN 7 AN/OR LINES SPECIFIED.

Severity: Terminal

Specification Type: C

Explanation: More than seven consecutive AN/OR lines are specified in the calculation specifications. Specify up to seven consecutive AN, OR, or AN/OR lines to condition an operation.

RPG–0517 AN/OR LINES OUT OF ORDER.

Severity: Terminal

Specification Type: C

Explanation: The line immediately following a line with an operation code is an AN/OR line. Remove the AN/OR entry in columns 7 and 8 from the first line in an AN/OR group.

RPG–0518 NO INDICATORS GIVEN WITH AN/OR LINES.

Severity: Terminal

Specification Type: C

Explanation: At least one indicator must be given in an AN or OR line.

RPG–0519 COLUMNS 18-59 ARE INVALID WITH AN/OR LINES, OR OPERATION CODE IS MISSING WITH INDICATORS PRESENT. ASSUME BLANK.

Severity: Terminal

Specification Type: C

Explanation: Only the last line of a group of AN/OR lines can have entries in columns 18 through 59, or indicators are specified in columns 7 through 17, but no operation is specified in columns 28 through 32.

RPG–0520 THIS LINE IS NOT AN AN/OR LINE AND PREVIOUS LINE HAS NO OPERATION CODE OR THIS LINE HAS NO INDICATORS AND NO OPERATION CODE.

Severity: Terminal

Specification Type: C

Explanation: This line is not an AN/OR line, and the previous line has no operation code specified. If this line should be an AN/OR line, enter an AN/OR entry in columns 7 and 8; if this line should have had an operation code (an operation code must be entered in the last line of a group of AN/OR lines), make the proper operation code entry in columns 28 through 32.

RPG–0521 MINUS INDICATOR NOT ALLOWED FOR TEST BIT OPERATION OF ONLY ONE BIT.

Severity: Warning

Specification Type: C

Explanation: Columns 56 and 57 (minus) must be blank when only 1 bit is specified for a TESTB operation. Blank is assumed.

RPG–0522 ALL THREE RESULTING INDICATORS ARE THE SAME.

Severity: Warning

Specification Type: C

Explanation: Usually the same indicator is used for only one or two of the conditions. Make sure the proper resulting indicator entries have been made in columns 54 through 59. The indicator specified will be set on each time the calculation is executed.

RPG–0523 A NEGATIVE FACTOR FOR THE SQUARE ROOT OPERATION IS NOT ALLOWED.

Severity: Terminal

Specification Type: C

Explanation: The entry specified in factor 2 of an SQRT operation is negative.

RPG–0524 WHOLE ARRAYS ARE NOT ALLOWED AS FACTOR 1 WITH CHAIN OR LOKUP OPERATION CODE.

Severity: Terminal

Specification Type: C

RPG–0525 OPERATION CODE IS INVALID FOR DEVICE TYPE OR MODE OF PROCESSING.

Severity: Terminal

Specification Type: C

Explanation: The CHAIN operation can be specified only for disk files processed randomly; READ may not be used with the KEYBORD file.

RPG–0526 FACTOR 1 MUST BE A 2 POSITION ALPHAMERIC FIELD WITH ACQ/REL/NEXT/POST OPERATION CODES.

Severity: Terminal

Specification Type: C

RPG–0527 WHOLE ARRAYS OR TABLES CANNOT BE SPECIFIED IN FACTOR 2 WITH THE ENDSR OP CODE.

Severity: Terminal

Specification Type: C

RPG–0528 FACTOR 2 MUST BE A 6 POSITION ALPHAMERIC FIELD OR ARRAY ELEMENT WITH THE ENDSR OP CODE.

Severity: Terminal

Specification Type: C

RPG–0529 FACTOR 2 INVALID LITERAL, BLANKS ASSUMED.

Severity: Warning

Specification Type: C

RPG–0541 FILE DESIGNATION IS INVALID FOR ADDROUT FILE, ASSUME R.

Severity: Terminal

Specification Type: F

Explanation: The file designation entry in column 16 is not R for an addrout file.

RPG–0543 LENGTH OF KEY, COLUMNS 29-30, OR LENGTH OF KEY AND KEY START LOCATION GREATER THAN RECORD LENGTH.

Severity: Terminal

Specification Type: F

Explanation: The key field entry in columns 29 and 30 must be less than 29 characters and must be less than the record length. The sum of the key field starting location (columns 35 through 38) plus the key length must not exceed the record length. Key field length of 03 is assumed, key field starting location of 01 is assumed.

RPG–0544 LENGTH OF RECORD ADDRESS FIELD OR KEY FIELD, COLUMNS 29-30, BLANK OR INVALID, ASSUME 3.

Severity: Terminal

Specification Type: F

RPG–0548 FILE ADDITION/DELETION INVALID FOR FILE TYPE, DEVICE, OR MODE OF PROCESSING, ASSUME BLANK.

Severity: Terminal

Specification Type: F, O

Explanation: File addition/deletion can be specified for sequential, direct, and indexed output files on disk only.

RPG–0549 KEY FIELD START LOCATION IS BLANK OR EXCEEDS RECORD LENGTH.

Severity: Terminal

Specification Type: F

Explanation: Columns 35 through 38 are blank, or the entry specified exceeds the record length in your file description specifications.

RPG–0550 NO MORE THAN 20 FILE DESCRIPTION SPECIFICATIONS ALLOWED.

Severity: Terminal

Specification Type: F

RPG–0551 RECORD LENGTH MISSING OR INVALID FOR DISK FILE, ASSUME 256.

Severity: Terminal

Specification Type: F

Explanation: The record length entry in columns 24 through 27 is missing. It can be a number from 1 to 4096.

RPG–0552 NO MORE THAN 15 DEMAND AND CHAIN FILES MAY BE USED IN ONE PROGRAM.

Severity: Terminal

Specification Type: F

RPG–0553 MAIN STORAGE INDEX IS INVALID FOR DEVICE TYPE OR MODE OF PROCESSING.

Severity: Warning

Specification Type: F

Explanation: Main storage index can be specified in columns 60 through 65 for indexed disk files processed randomly. Blank is assumed.

RPG–0554 ADD SPECIFIED ON THE FILE DESCRIPTION SPECIFICATIONS BUT ADD NOT REFERENCED ON OUTPUT.

Severity: Terminal

Specification Type: Not applicable

Explanation: Column 66 contains an A, but record addition ADD in columns 16 through 18 is not specified in your output specifications.

RPG–0555 NO ADD SPECIFIED ON FILE DESCRIPTION.

Severity: Terminal

Specification Type: Not applicable

Explanation: ADD is specified in columns 16 through 18 of your output specifications, but the ADD function was not specified in file description specifications (column 66) for this file.

RPG-0556 FILE DESIGNATED FOR DELETE NOT IN UPDATE MODE.

Severity: Terminal

Specification Type: F

Explanation: Only update files (U in column 15) can have records deleted.

RPG-0557 MASK FOR BIT OPERATION IS NOT 0-7.

Severity: Terminal

Specification Type: C

RPG-0558 INVALID USE OF, OR MISSING, RESULTING INDICATORS WITH THIS OPERATION CODE. ASSUME INVALID RESULTING INDICATORS BLANK.

Severity: Warning

Specification Type: C

Explanation: For CHAIN, columns 56 through 59 must be blank. It is suggested that columns 54 and 55 contain an indicator that can be tested for record not found.

For READ, columns 54 and 55 must be blank. It is suggested that columns 58 and 59 contain an indicator that can be tested for end of file. If the READ is to a WORKSTN file, columns 56 and 57 can contain an indicator that is set on if an exception/error condition occurs.

For ACQ, REL, and NEXT columns 54, 55, 58, and 59 must be blank. It is suggested that columns 56 and 57 contain an indicator that can be tested for an unsuccessful acquire, release, or next operation, respectively.

RPG-0559 FACTOR 2 OR THE RESULT FIELD IS INVALID FOR SPECIFIED OPERATION CODE.

Severity: Terminal

Specification Type: C

Explanation: MOVEA operation code: (1) either factor 2 or the result field must contain the name of an array; (2) both factor 2 and the result field may contain the name of an array (the same name cannot be used for both).

XFOOT and SORTA operation codes: factor 2 must be a whole array. Make the proper entries in columns 33 through 42 and/or columns 43 through 48.

RPG-0560 MODE OF PROCESSING COLUMN 28 GIVEN BUT NOT ALLOWED, ASSUME BLANK.

Severity: Terminal

Specification Type: F

Explanation: The mode of processing entry specified in column 28 is invalid. An entry is allowed only for limits or for random processing of disk files.

RPG-0561 KEY FIELD START LOCATION COLUMNS 35-38 GIVEN BUT NOT ALLOWED, ASSUME BLANK.

Severity: Terminal

Specification Type: F

Explanation: The key field start location entry specified in columns 35 through 38 is invalid. Place the proper entry in columns 35 through 38 of file description specifications for indexed files only.

RPG-0562 FILE TYPE FOR FROM FILENAME AND/OR TO FILENAME INVALID WITH TABLE/ARRAY.

Severity: Terminal

Specification Type: T

Explanation: The from filename and/or the to filename specified is invalid. Make sure the from filename specified in columns 11 through 18 of extension specifications is an input file and that the to filename in columns 19 through 26 is an output file.

RPG–0564 RECORD LENGTH IS NOT AT LEAST TWICE THE KEY LENGTH.

Severity: Terminal

Specification Type: F

RPG–0565 COLUMN 31 INVALID FOR DEVICE TYPE.

Severity: Terminal

Specification Type: F

Explanation: The entry in column 31 is valid for update, chained output (or addrout) disk files only.

RPG–0566 INVALID USE OF DEVICE AS FROM FILENAME.

Severity: Terminal

Specification Type: E

Explanation: The file named in columns 11 through 18 of extension specifications is not assigned to the disk or console.

RPG–0567 TABLE RECORD SIZE GREATER THAN FROM FILENAME DEVICE RECORD SIZE.

Severity: Terminal

Specification Type: E

Explanation: Table or array record length specified exceeds the maximum record allowed for the device. Make the table or array record length equal to or less than the maximum record length for the device.

RPG–0568 LENGTH OF KEY FIELD OR RECORD ADDRESS LENGTH, COLUMNS 29-30, GIVEN BUT NOT ALLOWED, ASSUME BLANK.

Severity: Terminal

Specification Type: F

Explanation: Length of key field or record address length specified in columns 29 and 30 is invalid for this file type.

RPG–0569 ENTRY OF I IN COLUMN 32 NOT GIVEN FOR AN INDEXED FILE, ASSUME I.

Severity: Terminal

Specification Type: F

RPG–0570 LOOK AHEAD WITH NUMERIC SEQUENCE OR LOOK AHEAD FOLLOWS A NUMERIC RECORD.

Severity: Terminal

Specification Type: I

Explanation: A look-ahead record type (** in columns 19 and 20) cannot be specified on the same line as a numeric sequence entry in columns 15 and 16.

RPG–0571 MORE THAN ONE LOOK AHEAD RECORD IN A FILE, OR LOOK AHEAD FIELD NOT VALID IN A DATA STRUCTURE.

Severity: Terminal

Specification Type: I

RPG–0572 LOOK AHEAD CANNOT BE THE ONLY RECORD IN A FILE.

Severity: Terminal

Specification Type: I

Explanation: Look-ahead records specified do not follow other file or record type specifications.

RPG–0573 MULTIPLE RECORD ADDRESS FILES DEFINED.

Severity: Terminal

Specification Type: F

Explanation: Only one record address file is allowed per program.

RPG–0574 EXTERNAL INDICATOR, COLUMNS 71-72, NOT THE SAME AS RECORD ADDRESS FILES.

Severity: Terminal

Specification Type: F

Explanation: The record address file and the file it is used to process are not conditioned by the same external indicator.

RPG–0575 NO INPUT SPECIFICATIONS FOUND FOR THIS FILE.

Severity: Terminal

Specification Type: Not applicable

Explanation: Input specifications are required for this file, but none were supplied.

RPG–0576 COMPILE TIME TABLE DATA FOUND. COMPILE TIME TABLE OR ARRAY NOT SPECIFIED IN EXTENSION SPECIFICATIONS.

Severity: Warning

Specification Type: E

Explanation: No extension specifications were supplied for compile time table. Table data is not processed.

RPG–0577 ONLY ONE FILE ASSOCIATED WITH RECORD ADDRESS FILE IS ALLOWED IN A PROGRAM.

Severity: Terminal

Specification Type: F, E

Explanation: More than one record address file or more than one file associated with a record address file is defined in this program.

RPG–0578 A RECORD ADDRESS FILE OR A FILE ASSOCIATED WITH A RECORD ADDRESS FILE IS REQUIRED BUT NOT DEFINED.

Severity: Terminal

Specification Type: F

RPG–0579 FIRST 1P LINE NOT FOR PRINTER. ASSUME COLUMN 41 IN CONTROL SPECIFICATION BLANK.

Severity: Warning

Specification Type: O

Explanation: Forms alignment is requested but the first 1P line is not specified for a printer file. Column 41 of the control specifications is assumed to be blank; therefore, no forms alignment is done.

RPG–0580 REFERENCED A MATCH LEVEL WHICH IS NOT VALID, OR DEFINED A LEVEL MORE THAN ONCE.

Severity: Terminal

Specification Type: I

Explanation: Valid match levels are M1 through M9.

RPG–0581 MISSING OR INVALID AN/OR ENTRY IN COLUMNS 7-8.

Severity: Terminal

Specification Type: C

RPG–0582 THE RELATIVE RECORD NUMBER FOR THE CHAIN OPERATION MUST BE NUMERIC WITH ZERO DECIMAL.

Severity: Warning

Specification Type: C

Explanation: The decimal positions are ignored.

RPG–0583 BINARY LENGTH SPECIFIED GREATER THAN 9. ASSUME 9.

Severity: Terminal

Specification Type: I

RPG–0584 THIS MATCH LEVEL WAS REFERENCED PREVIOUSLY IN THIS RECORD GROUP.

Severity: Terminal

Specification Type: I

RPG–0585 CHAIN OR DEMAND FILE SPECIFIED, BUT APPROPRIATE OPERATION CODE NOT FOUND IN THE CALCULATION SPECIFICATIONS.

Severity: Terminal

Specification Type: C

RPG–0586 MORE THAN ALLOWABLE TABLE/ARRAY NAMES USED IN THE PROGRAM.

Severity: Terminal

Specification Type: E

Explanation: More than 70 compile-time tables and/or arrays were defined or a total of more than 75 tables or arrays were defined in this program.

RPG–0587 IF FACTOR 1 OR FACTOR 2 IS A WHOLE ARRAY, THE RESULT FIELD MUST BE A WHOLE ARRAY.

Severity: Terminal

Specification Type: C

RPG–0588 TESTB, BITON AND BITOF MAY NOT REFERENCE AN ENTIRE ARRAY.

Severity: Terminal

Specification Type: C

RPG–0589 RESULT FIELD MUST BE A ONE POSITION ALPHAMERIC FIELD. IF FACTOR 2 IS A FIELD NAME, IT MUST BE A ONE POSITION ALPHAMERIC FIELD.

Severity: Terminal

Specification Type: C

Explanation: The result field is not a 1-byte alphameric field for TESTB, BITON, and BITOF, or factor 2 is a field name but not a 1-byte alphameric entry.

RPG–0590 WHENEVER HIGH IS USED IN A MOVE ZONE OPERATION, IT MUST REFERENCE AN ALPHAMERIC FIELD.

Severity: Terminal

Specification Type: C

Explanation: The high portion of a move zone instruction does not reference an alphameric field.

RPG–0591 LENGTH OF FIELD IN FACTOR 1 NOT EQUAL TO KEY LENGTH OF FILE SPECIFIED IN FACTOR 2.

Severity: Terminal

Specification Type: C

Explanation: The length of the field in factor 1 of a SETLL or CHAIN operation is not equal to the key field length specified in factor 2.

RPG–0592 FOR SEQUENTIALLY PROCESSED UPDATE FILE–T ENTRY IN COLUMN 15 IS INVALID OR L0-L9 INDICATOR USED WITH E IN COLUMN 15.

Severity: Terminal

Specification Type: O

Explanation: Total output cannot be specified for update files processed sequentially.

RPG–0593 TABLE/ARRAY NAME MISSING FOR 'TO' AND/OR 'FROM' FILENAME.

Severity: Warning

Specification Type: E

Explanation: No table name was specified in columns 27 through 32 for a table load operation (from filename in columns 11 through 18) or for a table output operation (to filename in columns 19 through 26).

RPG–0594 'TO' FILENAME MAY NOT BE USED WITH EXECUTION TIME TABLE/ARRAY.

Severity: Terminal

Specification Type: E

Explanation: An array output operation (to filename in columns 19 through 26) must not be specified for execution-time arrays.

RPG–0595 COLUMNS 27-32 AND 46-51 MUST BE BOTH TABLE OR BOTH ARRAY NAMES.

Severity: Terminal

Specification Type: E

Explanation: For alternating tables, columns 27 through 32 and 46 through 51 do not both contain table names; or columns 27 through 32 and 46 through 51 do not both contain array names for alternating arrays.

RPG–0597 END POSITION SPECIFIED FOR *PLACE LESS THAN TWICE THAT OF HIGHEST PREVIOUSLY SPECIFIED FIELD END POSITION, OR END POSITION GREATER THAN 256.

Severity: Terminal

Specification Type: O

RPG–0598 ALPHAMERIC TABLE/ARRAY SPECIFIED AS PACKED. ASSUME NUMERIC.

Severity: Terminal

Specification Type: E

RPG–0599 LENGTH OF ELEMENT FOR BINARY TABLE/ARRAY NOT SPECIFIED AS 4 OR 9. DEFAULT TO 4 IF LENGTH SPECIFIED IS LESS THAN 4, OTHERWISE DEFAULT TO 9.

Severity: Terminal

Specification Type: E

RPG–0603 INPUT SPECIFICATIONS INVALID FROM KEYBORD FILE.

Severity: Terminal

Specification Type: I

Explanation: Records to be supplied from the KEYBORD file are specified in your input specifications. Specify the input records for a KEYBORD file with the KEY operation code in calculation specifications, not in input specifications.

RPG–0604 INVALID OR MISSING MESSAGE INDICATOR SPECIFIED ON SET OR KEY OPERATION.

Severity Terminal

Specification Type: C

Explanation: Message indicator entries in columns 31 and 32 are missing, or the entry specified is not in the range 01-99.

RPG–0609 KEYBORD SPECIFIED AS PRIMARY FILE, BUT FUNCTION NEVER USED.

Severity: Terminal

Specification Type: C

Explanation: KEYBORD is specified as the primary file, but it is not used. The SET/KEY operation code must be used with a KEYBORD file.

RPG–0610 ONLY COMMAND KEY INDICATORS VALID AS RESULTING INDICATORS ON SET OPERATION.

Severity: Terminal

Specification Type: C

Explanation: The resulting indicator entries specified in columns 54 through 59 of a SET operation are not command key indicators.

RPG–0614 OPERATION REQUIRES KEYBORD SPECIFICATION ON FILE DESCRIPTION.

Severity: Terminal

Specification Type: C

Explanation: The KEYBORD file is not defined in your file description specification, but KEY and SET operations are used.

RPG–0616 KEYBORD AND WORKSTN MUST HAVE PRIMARY OR DEMAND DESIGNATION, ASSUME DEMAND.

Severity: Terminal

Specification Type: F

Explanation: A KEYBORD or WORKSTN file is not specified as a primary or a demand file (P or D in column 16).

RPG–0617 RESULT FIELD MAY NOT BE AN ARRAY.

Severity: Terminal

Specification Type: C

RPG–0618 CANNOT HAVE ANY OTHER PRIMARY OR SECONDARY FILES WHEN KEYBORD OR WORKSTN IS SPECIFIED AS PRIMARY.

Severity: Terminal

Specification Type: F

Explanation: When KEYBORD or WORKSTN are specified as primary files, no other file can be specified as primary or secondary. Remove the secondary file designation entry (column 16).

RPG–0625 FACTOR 1 MUST BE NUMERIC FOR CHAIN OPERATION WHEN FACTOR 2 FILENAME HAS PACKED KEYS.

Severity: Terminal

Specification Type: C

RPG–0631 FACTOR 1 MUST HAVE THE SAME LENGTH WHEN PACKED AS LENGTH OF PACKED KEYS FOR FACTOR 2 FILENAME.

Severity: Terminal

Specification Type: C

Explanation: The entry in factor 1 of a CHAIN operation is not the same length when packed as the record keys in the file named in factor 2.

RPG–0632 SET OR KEY OPERATION SPECIFIED WITH NO FUNCTION.

Severity: Terminal

Specification Type: C

Explanation: The SET or KEY operation code was entered in columns 28 through 32 without specifying the SET or KEY operation to be performed. Specify the function of the SET or KEY operation you want to perform (columns 7 through 27, 33 through 39).

RPG–0645 IMPROPER USE OF THE PACK/UNPACK FEATURE FOR LIMITS FILE PROCESSING.

Severity: Terminal

Specification Type: E

Explanation: The unpacked key length must be one or two less than twice the packed key length.

RPG–0646 WHOLE ARRAY IN RESULT FIELD INVALID FOR SPECIFIED OPERATION CODE.

Severity: Terminal

Specification Type: C

Explanation: An entire array cannot be specified as the result field for the operation specified. Make the result field a field (which must be numeric for XFOOT), an array element, or a table element.

RPG–0647 UNEQUAL KEY LENGTHS SPECIFIED FOR KEYS OF IDENTICAL FORMAT.

Severity: Warning

Specification Type: E

Explanation: The key length of a limits file (record address file) should be equal to the key length of the file to be processed by limits. The key length of the file processed by limits is assumed as the key length of the limits file.

RPG–0650 COMPILE TIME ARRAYS OVERLAPPING ARRAYS IN THE SAME DATA STRUCTURE.

Severity: Terminal

Specification Type: E

Explanation: Two or more compile time arrays have been defined in the same data structure such that they overlap. The from and to positions for the definitions of these arrays in the input specifications should be changed to correct the overlap.

RPG–0652 COMPILE TIME ARRAY IS NOT VALID IN DATA STRUCTURE SPECIFIED AS SAVDS.

Severity: Terminal

Specification Type: I

Explanation: If a data structure is defined as SAVDS in a program, a compile-time array cannot be defined as part of this data structure.

RPG–0653 THE SAME DATA STRUCTURE CANNOT BE SPECIFIED AS THE SAVDS AND AS THE DISPLAY STATION LOCAL DATA AREA.

Severity: Terminal

Specification Type: I

Explanation: The same data structure cannot be specified as the SAVDS and as the display station local data area in the same program.

RPG–0654 DEFINED LENGTH OF THE DATA STRUCTURE IN THE RECORD IS LESS THAN THE HIGH TO POSITION OF FIELDS DEFINING THE CONTENTS OF THE DATA STRUCTURE. ASSUME LENGTH OF DATA STRUCTURE IN RECORD TO BE CORRECT.

Severity: Terminal

Specification Type: I

Explanation: If a data structure is defined as a field in a record, this definition overrides the definition of the fields describing the contents of the data structure. If the data structure is not defined previously as a field in a record, the length of the data structure is assumed to be the highest to position of a field in the data structure.

RPG–0655 OVERLAPPING FIELDS IN A DATA STRUCTURE CANNOT BE USED IN THE CALCULATION SPECIFICATION.

Severity: Terminal

Specification Type: C

Explanation: Overlapping fields in a data structure cannot be used in the same calculation specification. If factor 1, factor 2, or the result field references an array or array element with a variable subscript, the entire array is used to determine whether overlap exists. The same array name can be referenced in the appropriate factors of a calculation specification without violating the overlap rule.

RPG–0656 RECORD INFORMATION OR DATA STRUCTURE SPECIFIED ON SAME SPECIFICATION AS FIELD TYPE ENTIRES FOR A DATA STRUCTURE. ASSUME COLUMNS 7-43 BLANK.

Severity: Terminal

Specification Type: I

Explanation: Columns 7 through 43 must be blank for a field specified in a data structure.

RPG–0657 SEQUENCE ENTRY IN COLUMNS 15-16 MUST BE BLANK FOR A DATA STRUCTURE.

Severity: Terminal

Specification Type: I

Explanation: If a data structure is specified in columns 19 and 20, columns 15 and 16 must be blank.

RPG–0658 COLUMNS 21-74 MUST BE BLANK FOR A DATA STRUCTURE.

Severity: Terminal

Specification Type: I

Explanation: Columns 21 through 74 must be blank if a data structure is specified correctly in columns 19 and 20 of the same input specification.

RPG–0659 DATA STRUCTURE DEFINED AS THE DISPLAY STATION LOCAL DATA AREA MUST NOT EXCEED LENGTH OF 256. LENGTH OF 256 IS ASSUMED.

Severity: Terminal

Specification Type: I

RPG–0660 DATA STRUCTURE NAME INVALID IN CALCULATIONS EXCEPT WITH RLABL OR WITH RESULT FIELD OF POST OPERATION.

Severity: Terminal

Specification Type: I

RPG–0707 RESULT FIELD MUST BE EITHER A 6 OR 12 POSITION NUMERIC FIELD WHEN USING THE TIME OPERATION CODE.

Severity: Terminal

Specification Type: C

RPG–0720 COLUMN 15 NOT C FOR WORKSTN FILE, ASSUME C.

Severity: Warning

Specification Type: F

RPG–0721 MORE THAN 1 WORKSTN FILE IS INVALID.

Severity: Terminal

Specification Type: F

RPG–0722 CONSOLE/KEYBORD/CRT FILES ARE NOT ALLOWED WITH WORKSTN FILES.

Severity: Terminal

Specification Type: F

RPG–0725 COLUMNS 60-65 DO NOT CONTAIN A VALID FIELD NAME FOR A CONTINUATION RECORD.

Severity: Terminal

Specification Type: F

Explanation: The keyword in columns 54 through 59 was ID, SLN, SAVDS, INFDS, or INFSR but columns 60 through 65 did not contain a valid field, label, or data structure name. Or, the keyword in columns 54 through 59 was RECNO, but columns 60 through 65 did not contain a valid field name. Or, the keyword in columns 54 through 59 was FMTS, but columns 60 through 67 did not contain a valid format load member name or the keyword *NONE.

RPG–0726 COLUMNS 64-65 DO NOT CONTAIN A VALID VALUE FOR NUM OR IND KEYWORDS.

Severity: Terminal

Specification Type: F

Explanation: With IND or NUM a two-digit numeric value must be entered in columns 64 and 65.

RPG–0727 INVALID DEFINITION FOR FIELD ASSOCIATED WITH KEYWORD.

Severity: Terminal

Specification Type: F

Explanation: The field associated with the keyword RECNO is not defined as a seven-position, numeric field with zero decimal positions.

RPG–0728 BLOCK LENGTH MUST BE BLANK FOR WORKSTN, ASSUME BLANK.

Severity: Warning

Specification Type: F

RPG–0729 COLUMNS 54 AND 55 MUST CONTAIN A VALID RESULTING INDICATOR AND COLUMNS 56-59 MUST BE BLANK FOR THIS OP CODE.

Severity: Terminal

Specification Type: C

Explanation: The SHTDN op code requires a valid indicator in columns 54 and 55. No other indicators can be specified in columns 56 through 59 nor can columns 54 and 55 be blank.

RPG–0730 THE RESULT FIELD MUST BE A DATA STRUCTURE NAME WHEN USING THE POST OPERATION CODE.

Severity: Terminal

Specification Type: C

RPG–0799 ERROR FILE FULL.

Severity: Terminal

Specification Type: Not applicable

Explanation: Too many errors were made in this program. The errors are listed on the output listing.

RPG–0801 AND LINE IS INVALID FOR THIS DEVICE, COLUMNS 14-16.

Severity: Terminal

Specification Type: I

Explanation: Use of AND lines in record ID codes is not valid with this device.

RPG–0802 RECORD IDENTIFICATION CODES IN COLUMNS 35-41 ARE NOT ALLOWED WITH THIS DEVICE.

Severity: Terminal

Specification Type: I

Explanation: Record ID codes in columns 21 through 34 only are valid for this device. Entries in columns 35 through 41 are not allowed.

RPG–0803 INVALID OR BLANK ENTRIES IN COLUMNS 21-34 FOR THIS DEVICE.

Severity: Terminal

Specification Type: I

Explanation: One of the following error conditions exists:

- Column 24 does not contain a 1.
- Column 25 is not blank.
- Column 26 does not contain a C, and/or two record ID codes are used.
- Column 31 does not contain a 2.
- Column 32 is not blank.
- Column 33 does not contain a C.

RPG–0804 INCONSISTENT RECORD IDENTIFICATION CODE USAGE ON 'OR' LINE.

Severity: Terminal

Specification Type: I

Explanation: The number of record ID codes on an OR line must be the same as its associated record line.

RPG–0805 RECORD IDENTIFICATION CODES CAN ONLY BE USED ONCE.

Severity: Terminal

Specification Type: I

Explanation: A record ID code character can only be used once in a file.

RPG–0806 INVALID OR NO RECORD IDENTIFICATION INDICATOR DEFINED, ASSUME 10.

Severity: Terminal

Specification Type: I

Explanation: A record line for a CONSOLE file must have a record ID indicator between 01 and 10 assigned.

RPG–0807 FIELD LENGTH DEFINED FOR RECORD IDENTIFICATION IS NOT LENGTH OF IDENTIFICATION.

Severity: Terminal

Specification Type: I

Explanation: A field was defined starting in column 1 but was not equal to the length of the record ID code for the CONSOLE file. If the record ID code is given a field name, the field can only be as long as the ID code.

RPG–0808 INVALID START POSITION FOR FIRST FIELD TO BE PROMPTED.

Severity: Terminal

Specification Type: I

Explanation: The start position of the first field to be prompted for does not start immediately following the record ID for the CONSOLE file.

RPG–0809 INVALID FROM OR TO ENTRY IN COLUMNS 44-51 FOR FIELD PROMPT DESCRIPTION.

Severity: Terminal

Specification Type: I

Explanation: Prompt format description entry for the CONSOLE file is invalid for one of the following reasons:

- From entry (columns 44 through 47) is 1, and to entry (columns 48 through 51) is greater than 2.
- Field prompt description overlaps previously defined field prompt. From position is less than previous to position, and the to position is greater than the previous to position.
- From entry is greater than previous to position plus 1.

RPG–0810 FIELD RECORD RELATION INDICATORS, COLUMNS 63-64, ARE INVALID WITH A PROMPT FIELD.

Severity: Terminal

Specification Type: I

Explanation: Field record relation indicators, columns 63 and 64, must be left blank on a field prompt definition for a CONSOLE file.

RPG–0811 LENGTH OF PROMPT FIELD IN COLUMNS 44-51 EXCEEDS LIMIT. MAXIMUM FIELD LENGTHS–ALPHAMERIC 66, NUMERIC 15.

Severity: Terminal

Specification Type: I

RPG–0813 RECORD IDENTIFICATION INDICATORS MUST BE UNIQUE.

Severity: Terminal

Specification Type: I

Explanation: A record identification indicator can be used only once for a CONSOLE file.

RPG–0815 A CONSOLE FILE MUST HAVE AT LEAST 1 FIELD DEFINED FOR WHICH THERE IS A PROMPT.

Severity: Terminal

Specification Type: I

Explanation: A CONSOLE file was defined with no prompted fields. Define at least one field in the CONSOLE file other than a record identification field.

RPG–0816 FACTOR 2 MUST BE A CONSOLE FILE FOR A VALID RESULT FIELD ENTRY.

Severity: Terminal

Specification Type: C

Explanation: The ERASE function applies only to a CONSOLE file, which must be specified in factor 2. The job is terminated, and the entire specification line is ignored.

RPG–0817 FACTOR 2 GIVEN ON SET OPERATION WITHOUT RESULT FIELD ENTRY 'ERASE'.

Severity: Terminal

Specification Type: C

Explanation: Factor 2 on a set operation requires an associated keyword (ERASE) specified in the result field (columns 43 through 48). The job is terminated, and the entire specification is ignored.

RPG–0818 INVALID ENTRY IN COLUMNS 49-51.

Severity: Warning

Specification Type: C

Explanation: The entry in columns 49 through 51 is not valid for a SET operation code. The entry in columns 49 through 51 is ignored by the system.

RPG–0819 LENGTH OF FACTOR 1 OR RESULT FIELD MAY NOT EXCEED 40 CHARACTERS IF 6X40 WINDOW SPECIFIED OR 79 CHARACTERS IF FULL SCREEN SPECIFIED.

Severity: Terminal

Specification Type: C

Explanation: A record length of 40 or less was specified for the KEYBORD file but factor 1 on a SET or KEY operation or the result field on a KEY operation is greater than 40 characters. Or, if a record length of greater than 40 was specified, factor 1 or the result field is greater than 79 characters.

RPG–0821 MULTIPLE CONSOLE FILES DEFINED BUT ONLY ONE IS ALLOWED. DEFAULT IS TO DISK.

Severity: Terminal

Specification Type: F

RPG–0822 CONSOLE CAN ONLY BE AN INPUT FILE, DEFAULT IS TO INPUT.

Severity: Terminal

Specification Type: F

RPG–0823 CONVERSATIONAL FILES ARE NOT ALLOWED, DEFAULT IS TO INPUT.

Severity: Terminal

Specification Type: T

Explanation: BSCA conversational files are not allowed. The file is assumed to be an input file.

RPG–0825 FOR CONSOLE THE RECORD LENGTH MUST BE TWICE AS LARGE AS THE RECORD ADDRESS FIELD LENGTH. DEFAULT IS 2 TIMES RECORD ADDRESS FIELD LENGTH.

Severity: Warning

Specification Type: F

RPG–0826 BOTH ITB AND TRANSPARENCY CANNOT BE SPECIFIED FOR A TRANSMITTING FILE; ITB WAS DROPPED.

Severity: Terminal

Specification Type: T

Explanation: Capability is not supported on system.

RPG–0828 WHOLE ARRAY NAME INVALID FOR FIELD NAME ENTRY IN COLUMNS 53-58 FOR CONSOLE FILE.

Severity: Terminal

Specification Type: I

Explanation: An array name was specified in columns 53 through 58 of the input specifications sheet. For CONSOLE files, whole array names must be entered in either of the following ways:

- Define the whole array as a subfield within a field for a CONSOLE record.
- Define each element of the array with an index, and place this entry in columns 53 through 58 of the input specifications sheet. The index must be an integer value.

RPG–0999 PROGRAM EXCEEDS MAIN STORAGE IN COLUMNS 12-14 OF CONTROL SPECIFICATION.

Severity: Warning

Specification Type: H

Explanation: The program requires more storage for execution than specified in columns 12 through 14 of the control specification. The program will run, but will use the amount of storage shown on the compiler listing next to MAIN STORAGE REQUIRED TO EXECUTE. You can increase the number specified in columns 12 through 14 of the control specifications to equal the amount of main storage required, or you can redesign your program to use less storage.

RPG II DISPLAYED MESSAGES

RPG II messages that appear on a display station display screen are described in the *Displayed Messages Guide.*

RPG II AUTO REPORT MESSAGES

NOTE 001 SOURCE PROGRAM IS MISSING. PROGRAM IS TERMINATED.

Severity: Terminal

Specification Type: Not applicable

Explanation: /* or ** was encountered as the first record in the source program.

NOTE 002 RPG II CONTROL SPECIFICATION IS MISSING. A CONTROL SPECIFICATION WITH BLANK ENTRIES IS CREATED.

Severity: Warning

Specification Type: H

NOTE 003 SOURCE PROGRAM CONTAINS MORE THAN ONE RPG II CONTROL SPECIFICATION. ALL BUT THE FIRST ARE DROPPED.

Severity: Warning

Specification Type: H

Explanation: Multiple RPG II control specifications are present in the source program. The copied member may contain a control specification.

NOTE 004 DUPLICATE FILENAMES ARE PRESENT ON THE FILE DESCRIPTION SPECIFICATIONS READ FROM THE SOURCE FILE. DUPLICATE IS DROPPED.

Severity: Warning

Specification Type: F

NOTE 005 REQUESTED LIBRARY MEMBER CANNOT BE FOUND. SPECIFICATION IS DROPPED.

Severity: Warning

Specification Type: /COPY

Explanation: The requested library member was not found for one of the following reasons:

- The wrong name was used for the member.
- The wrong library was specified.
- There are no records in the library member.

NOTE 006 DUPLICATE FILENAMES ARE PRESENT ON THE FILE DESCRIPTION SPECIFICATIONS READ FROM THE LIBRARY SOURCE MEMBER. DUPLICATE IS DROPPED.

Severity: Warning

Specification Type: F

Explanation: The library member can contain only one file description specification for a filename. The error occurred for one of the following reasons:

- One library source member contains two file description specifications for the same filename.
- More than one library source member has file description specifications for the same filename.

NOTE 007 TABLE AREA PROVIDED FOR INPUT OVERRIDES EXCEEDED. OVERRIDE FUNCTION IS DISCONTINUED FOR THIS /COPY.

Severity: Warning

Specification Type: /COPY

Explanation: The number of input field modifier statements following a /COPY exceeds the available space in the table. Those fields that could be overridden will be added to the file. Therefore, it is possible to have invalid specifications in the generated program. Auto report always handles at least 20 overrides. Place override statements first, followed by input fields to be added to the copied specifications. This allows all table space to be used for override statements.

NOTE 008 INVALID RPG II SPECIFICATION TYPE. SPECIFICATION IS DROPPED.

Severity: Warning

Specification Type: /COPY

Explanation: The error occurred for one of the following reasons:

- Position 6 does not contain an H, F, E, L, T, I, C, or O, and position 7 does not contain an asterisk.
- /COPY appears on an H (control) specification.

NOTE 009 INVALID OR UNDEFINED FILE FOR *AUTO LINES. PROGRAM IS TERMINATED.

Severity: Terminal

Specification Type: F, O

Explanation: The error occurred for one of the following reasons:

- *AUTO file is not a printer or line counter file.
- The file description specification for *AUTO file is missing.
- The names on the file description specification and *AUTO file do not match.

NOTE 010 TABLE AREA FOR FIELD NAMES USED ON *AUTO LINES EXCEEDED. SPECIFICATION IS DROPPED.

Severity: Warning

Specification Type: O

Explanation: Each field used in the H-*AUTO and D/T-*AUTO lines is placed in a table.

NOTE 011 TABLE AREA PROVIDED FOR FIELD NAMES EXCEEDED. NON-UNIQUE FIELD NAMES MAY BE GENERATED.

Severity: Warning

Specification Type: O

Explanation: Generated field names that end in 1-9 or R are added to the field name table.

NOTE 012 GENERATED TOTAL FIELD PREVIOUSLY DEFINED WITH DIFFERENT ATTRIBUTES. PREVIOUS DEFINITION IS USED.

Severity: Warning

Specification Type: O

Explanation: For the *AUTO specification, the generated total field was previously defined with either a different field length or a different number of decimal positions. The first or previous definition is used, and both the total field and the generated field names are printed with the error note number.

NOTE 013 *AUTO PREVIOUSLY USED FOR A DIFFERENT FILE. DROP ALL SPECIFICATIONS TO NEXT RECORD TYPE.

Severity: Warning

Specification Type: O

Explanation: *AUTO can be specified for only one file.

NOTE 014 POSITIONS 7-22 ARE NOT BLANK ON OUTPUT FIELD SPECIFICATION. BLANKS ARE ASSUMED.

Severity: Warning

Specification Type: O

Explanation: Positions 7 through 22 of output field specifications must be blank.

NOTE 015 INVALID INDICATOR. BLANKS ARE ASSUMED.

Severity: Warning

Specification Type: O

Explanation: Positions 24 and 25, 27 and 28, or 30 and 31 are not 01-99, KA-KN, KP-KY, L0-L9, MR, 1P, H1-H9, OA-OG, OV or blank. Positions 23, 26, or 29 are not N or blank. The invalid indicator is printed with the note number. Blanks are assumed.

NOTE 016 INVALID FIELD NAME. SPECIFICATION IS DROPPED.

Severity: Warning

Specification Type: O

Explanation: The field name is invalid for one of these reasons:

- The field name was not found.
- The field name was not defined.
- The array index contains a blank after the comma or a comma as the first character.

The specification is dropped, and the column headings for the field are also dropped.

NOTE 017 INVALID ENTRY IN POSITION 38 AND/OR 44. BLANKS ARE ASSUMED.

Severity: Warning

Specification Type: O

Explanation: Position 38 and position 44 must be blank for alphameric fields.

NOTE 018 INVALID ENTRY IN POSITION 39. BLANK IS ASSUMED.

Severity: Warning

Specification Type: O

Explanation: Position 39 was not a blank or B for any field on an H-*AUTO line.

NOTE 019 INVALID ENTRY IN POSITIONS 40-43. BLANKS ARE ASSUMED.

Severity: Warning

Specification Type: O

Explanation: The end position cannot be specified on field specifications in an H-*AUTO line.

NOTE 020 INVALID ENTRY IN POSITIONS 45-70. BLANKS ARE ASSUMED.

Severity: Warning

Specification Type: O

Explanation: A literal or edit word cannot be specified with an alphameric field.

NOTE 021 FIELD NAME WILL BE CONDITIONED BY THE INDICATOR N1P.

Severity: Warning

Specification Type: O

Explanation: A field name is used and the H-*AUTO line is unconditioned. If printed on the first page, the field may not contain meaningful data because the first record has not been read. An N1P indicator will be generated by the system for this specification.

NOTE 022 INVALID EDIT CODE, POSITION 38. BLANK IS ASSUMED.

Severity: Warning

Specification Type: O

Explanation: Position 38 is not one of the following valid edit codes: A, B, C, D, J, K, L, M, 1, 2, 3, 4, X, Y, or Z.

NOTE 023 INVALID ENTRY IN POSITION 44. BLANK IS ASSUMED.

Severity: Warning

Specification Type: O

Explanation: Position 44 is not used with *AUTO because packed and binary data cannot be used. Position 44 must be blank.

NOTE 024 CONDITIONING INDICATORS IN POSITIONS 23-31 ARE NOT BLANK FOR A TOTALING FIELD. A IN POSITION 39, ON A T *AUTO LINE. BLANKS ARE ASSUMED.

Severity: Warning

Specification Type: O

Explanation: The indicators specified on a totaling field in a T *AUTO specification are not used when generating specifications.

NOTE 025 LITERAL IN POSITIONS 45-70 HAS APOSTROPHE MISSING AT BEGINNING OR END. BLANKS ARE ASSUMED IN POSITIONS 45-70.

Severity: Warning

Specification Type: O

Explanation: The error occurred for one of these reasons:

- Position 45 is not an apostrophe, but 45 through 70 are not blank.
- Position 45 is an apostrophe, but there are no apostrophes in 46 through 70.
- There is an embedded single apostrophe (not paired) in positions 46 through 69.
- Positions through position 70 are not blank after the last apostrophe.

This specification is dropped if no field name is present.

NOTE 026 UNABLE TO DETERMINE IF FIELD OR RECORD SPECIFICATION. SPECIFICATION IS DROPPED.

Severity: Warning

Specification Type: O

Explanation: Position 15 is blank indicating it is a field specification, but positions 32 through 37 and 45 through 70 are also blank. This specification and its possible column headings are dropped.

NOTE 027 POSITIONS 38-44 ARE NOT BLANK WHEN A LITERAL IS SPECIFIED. BLANKS ARE ASSUMED.

Severity: Warning

Specification Type: O

Explanation: Positions 38 through 44 must be blank when a literal is specified on an H-*AUTO line.

NOTE 028 POSITIONS 7-13 ARE NOT BLANK ON AND/OR SPECIFICATION. BLANKS ARE ASSUMED.

Severity: Warning

Specification Type: O

Explanation: Positions 7 through 13 are not blank when 14 through 16 contain AND or 14 and 15 contain OR.

NOTE 029 SPACE AND/OR SKIP ENTRIES IN POSITIONS 17-22 ARE INVALID. BLANKS ARE ASSUMED.

Severity: Warning

Specification Type: O

Explanation: Position 17 and/or 18 do not contain 0-3 or blank, or positions 19 and 20 and 21 and 22 do not contain 01-84.

NOTE 030 POSITIONS 37-70 NOT BLANK ON RECORD SPECIFICATION. BLANKS ARE ASSUMED.

Severity: Warning

Specification Type: O

Explanation: Positions 37 through 70 must be blank for record types.

NOTE 031 INVALID ENTRY IN POSITION 38. BLANK IS ASSUMED.

Severity: Warning

Specification Type: O

Explanation: Edit code cannot be specified for a literal on D/T-*AUTO lines.

NOTE 032 END POSITION IN 40-43 IS INVALID. BLANKS ARE ASSUMED.

Severity: Warning

Specification Type: O

Explanation: Error occurred because either positions 40 through 43 contain invalid numbers or the end position exceeds the record length.

NOTE 033 GENERATED FIELD LENGTH EXCEEDS 15. 15 IS ASSUMED.

Severity: Warning

Specification Type: O

Explanation: Two positions have been added to a field specified with an A in position 39 in order to generate a total field. The length of the generated total field exceeds 15. Decrease the length of the field on the specification where definition of the field occurred.

NOTE 034 DEFINITION OF FIELD IS INVALID. DEFINITION IS NOT USED.

Severity: Warning

Specification Type: F, E, I, C

Explanation: This error occurred for one of these reasons:

- Length equals 0.
- Length is greater than 15 for a numeric field.
- Length entry is nonnumeric.
- Length is less than decimal position.
- Decimal position entry is nonnumeric.
- Position 43 is not P, B, or blank.

NOTE 035 ARRAY NAME SPECIFIED ON *AUTO LINE. SPECIFICATION IS DROPPED.

Severity: Warning

Specification Type: O

Explanation: This error occurred for one of these reasons:

- The field name in this H-*AUTO or D/T-*AUTO specification is an array name.
- A generated name for this D/T-*AUTO specification total field is an array name. In this case, the specification is dropped along with all its column headings.

In positions 32 through 37, enter the name of a numeric field that is to be accumulated. Both the total field name and the generated field array name are printed with the note number, or the specification is dropped.

NOTE 036 RECORD LENGTH FOR FILE WITH *AUTO LINES IS INVALID. ASSUME RECORD LENGTH OF 96.

Severity: Warning

Specification Type: F

Explanation: This error occurred for one of these reasons:

- Record length is 0.
- Record length is nonnumeric.
- Record length is blank.

NOTE 037 TOTALING, A IN POSITION 39, SPECIFIED FOR AN INVALID FIELD NAME. ASSUME POSITION 39 IS BLANK.

Severity: Warning

Specification Type: O

Explanation: Position 39 of a D/T-*AUTO specification field has an A, yet the field name is one of the following:

- Blank
- A table name
- An indexed array name
- A page field

In positions 32 through 37, enter the name of a numeric field that is to be accumulated. The system drops the specification and all its column headings.

NOTE 038 TOTALING, A IN POSITION 39, SPECIFIED FOR AN ALPHAMERIC FIELD. ASSUME POSITION 39 IS BLANK.

Severity: Warning

Specification Type: O

Explanation: The D/T-*AUTO line field name is alphameric, and position 39 is an A. In positions 32 through 37, enter the name of a numeric field that is to be accumulated.

NOTE 039 POSITIONS 7-38 NOT BLANK FOR A COLUMN HEADING. BLANKS ARE ASSUMED.

Severity: Warning

Specification Type: O

Explanation: Position 39 is a C: therefore, positions 7 through 38 must be blank.

NOTE 040 INVALID ENTRY IN POSITION 39. BLANK IS ASSUMED.

Severity: Warning

Specification Type: O

Explanation: This error occurred for one of these reasons:

- Position 39 is B, but the field name is blank.
- This is a field specification of a D/T-*AUTO line, and position 39 is not A, B, C, 1-9, R, or blank.

NOTE 041 COLUMN HEADING, C IN POSITION 39, SPECIFIED BUT LITERAL NOT PRESENT. SPECIFICATION IS DROPPED.

Severity: Warning

Specification Type: O

Explanation: Position 39 is a C, and positions 45 through 70 are blank.

NOTE 042 EDIT CODE AND EDIT WORD ARE BOTH SPECIFIED. EDIT WORD IS DROPPED.

Severity: Warning

Specification Type: O

Explanation: Edit code in position 38 and edit word in positions 45 through 70 are both specified. Positions 45 through 70 are assumed blank.

NOTE 043 EDITING SPECIFIED FOR AN ALPHAMERIC FIELD ASSUME BLANKS IN POSITIONS 38 AND 45-70.

Severity: Warning

Specification Type: O

Explanation: Positions 38 and 45 through 70 must be blank for alphameric fields.

NOTE 044 INVALID ENTRY IN POSITION 16. BLANK IS ASSUMED.

Severity: Warning

Specification Type: O

Explanation: Position 16 is not F or blank.

NOTE 045 AND/OR SPECIFICATION OUT OF SEQUENCE. SPECIFICATION IS DROPPED.

Severity: Warning

Specification Type: O

Explanation: The AND/OR (positions 14 through 16) does not follow a record specification. Ensure that record identification entries in columns 15 through 31 precede AND or OR lines.

NOTE 046 MULTIPLE D/T *AUTO LINES SPECIFIED IN THE PROGRAM. DROP ALL SPECIFICATIONS TO NEXT RECORD TYPE.

Severity: Warning

Specification Type: O

Explanation: Only one D/T-*AUTO line is allowed in the program.

NOTE 047 COLUMN HEADING SPECIFICATION OUT OF ORDER. SPECIFICATION IS DROPPED.

Severity: Warning

Specification Type: O

Explanation: A field specification with a C in position 39 does not follow a specification with a C, B, A, or blank (with a field name in positions 32 through 37) in position 39.

NOTE 048 END POSITION INVALID FOR THIS SPECIFICATION TYPE. ASSUME BLANKS IN POSITIONS 40-43.

Severity: Warning

Specification Type: O

Explanation: The end position cannot be specified for these specifications if there is a C, R, or 1-9 in position 39.

NOTE 049 SPECIFIED END POSITION IS LESS THAN FIELD OR LITERAL LENGTH. ASSUME BLANKS IN POSITIONS 40-43.

Severity: Warning

Specification Type: O

Explanation: End positions (40 through 43) must be at least as large as the field or literal.

NOTE 050 MORE THAN THREE COLUMN HEADING LINES SPECIFIED. SPECIFICATION IS DROPPED.

Severity: Warning

Specification Type: O

Explanation: Only two consecutive *AUTO specifications may have a C in position 39.

NOTE 051 NO VALID TOTALING FIELDS SPECIFIED. ONLY ONE D/T OUTPUT LINE IS GENERATED. TOTAL LINE CONSTANTS, 1-9, R IN POSITION 39, ARE DROPPED.

Severity: Warning

Specification Type: O

Explanation: Position 39 of D/T-*AUTO line does not contain an A. Therefore no automatic totaling is done and no total lines are generated. Use a * specification with an A in position 39 if you want automatic totaling done.

NOTE 052 1-9, R IS INVALID IN POSITION 39. SPECIFICATION IS DROPPED.

Severity: Warning

Specification Type: O

Explanation: Position 39 contains 1-9, but the associated level indicator (L1-L9) was not defined on input specifications (positions 59 and 60), or this is a T-*AUTO line and the lowest level indicator used on the T-*AUTO line is of greater or equal level. If a 2 was specified in position 39, then L2 must be defined as a level indicator on the input specifications; if this is a T-*AUTO line, the lowest control level indicator present on the T-*AUTO line must be L1.

NOTE 053 INDICATORS NOT ALLOWED ON THIS SPECIFICATION TYPE. BLANKS ARE ASSUMED IN POSITIONS 23-31.

Severity: Warning

Specification Type: O

Explanation: Indicators cannot be used on:

- A field description following an H-*AUTO specification

- A field description with position 39 containing a 1-9 or R following a D/T-*AUTO line

NOTE 054 SPECIFIED END POSITION CAUSES OVERLAYS OF FIELDS OR LITERALS. BLANKS ARE ASSUMED IN POSITIONS 40-43.

Severity: Warning

Specification Type: O

Explanation: The length of the line up to this specification plus the length of the field/literal of this specification is greater than the specified end position. An end position will automatically be generated if positions 40 through 43 are left blank.

NOTE 055 I/O ERROR OCCURRED. PROGRAM IS TERMINATED.

Severity: Terminal

Specification Type: Not Applicable

Explanation: The additional information printed with the error describes the problem:

- (41) Permanent disk error.
- (44W) The number of tracks allocated for $WORK is too small.
- (44S) The number of tracks allocated for $SOURCE is too small.

NOTE 056 LIBRARY OR SOURCE MEMBER NAME IS INVALID. ENTRY IS DROPPED.

Severity: Warning

Specification Type: U and /COPY

Explanation: A /COPY statement cannot have a U or H in column 6 of the specification sheet. When using Auto Report Option Specifications, the specification sheet must have a U in column 6.

For the U specification, the library name can be F1, #LIBRARY, or a valid 8-character library name, and must begin in column 8. The member name can be up to 8 characters long and must be separated from the library name by a comma.

For the /COPY statement, the library name can be F1, #LIBRARY, or a valid library name, and must begin in column 13. The member name can be up to 8 characters long and must be separated from the library name by a comma.

NOTE 057 TOTALING SPECIFIED MORE THAN ONCE FOR THIS FIELD NAME. SPECIFICATION IS DROPPED.

Severity: Warning

Specification Type: O

Explanation: Totaling for any particular field name specified with an A in position 39 may be specified only once in each program.

NOTE 058 MAXIMUM NUMBER OF H *AUTO LINES EXCEEDED. DROP ALL SPECIFICATIONS TO THE NEXT RECORD TYPE.

Severity: Warning

Specification Type: O

Explanation: More than five H-*AUTO lines were specified.

NOTE 059 INVALID ENTRY IN POSITION 7 OF U SPECIFICATION. BLANK IS ASSUMED.

Severity: Warning

Specification Type: U

Explanation: Valid entries are C or blank.

NOTE 060 LIBRARY AND MEMBER NAME IN POSITIONS 8-24 ARE NOT BLANK. BLANKS ARE ASSUMED.

Severity: Warning

Specification Type: U

Explanation: Positions 8 through 24 must be blank if position 7 is blank.

NOTE 061 INVALID ENTRY IN DATE SUPPRESS, POSITION 27. BLANK IS ASSUMED.

Severity: Warning

Specification Type: U

Explanation: Valid entries are N and blank. Page/date is not suppressed. If position 27 is blank, the heading line generated by the first H-*AUTO specification will contain a date and page number.

NOTE 062 INVALID ENTRY IN ASTERISK SUPPRESS, POSITION 28. BLANK IS ASSUMED.

Severity: Warning

Specification Type: U

Explanation: Valid entries are N and blank. Asterisks will appear. An asterisk is printed on the first generated total line to the right of the highest end position generated from the D/T-*AUTO specification. One additional asterisk is printed on each higher-level line including the final total.

NOTE 063 AND/OR SPECIFICATION IS INVALID. SPECIFICATION IS DROPPED.

Severity: Warning

Specification Type: O

Explanation: This error occurred for one of these reasons:

- The main record specification does not have conditioning indicators.
- The AND/OR specification has no indicators.

You can use AND and OR specifications with *AUTO output indicators if you enter an output indicator on the first record description specification. Normal RPG II rules for AND and OR lines apply.

NOTE 064 D/T *AUTO LINE OVERFLOW WILL OCCUR WITH GENERATION OF ASTERISK INDICATION. ALL ASTERISKS ARE SUPPRESSED.

Severity: Warning

Specification Type: O

Explanation: One or more of the asterisks will cause overflow of the defined printer record length. Put an N in column 28 of the U specification to suppress the asterisk indication.

NOTE 065 POSITIONS FOLLOWING LIBRARY MEMBER NAME ARE NOT BLANK. BLANKS ARE ASSUMED.

Severity: Warning

Specification Type: U

Explanation: Positions 30 through 49 of the /COPY statement are not blank. This may be caused by an embedded blank in the name. In this case, characters up to the blank are considered the name. The rest of the characters are dropped.

NOTE 066 MORE THAN 19 AND/OR LINES CONDITION AN *AUTO LINE. THIS AND ALL FOLLOWING AND/OR SPECIFICATIONS ARE DROPPED.

Severity: Warning

Specification Type: O

Explanation: The RPG II language does not permit over 19 AND/OR lines on output specifications.

NOTE 068 NUMBER OF FILE DESCRIPTION SPECIFICATIONS EXCEEDS THE MAXIMUM ALLOWED. SPECIFICATION IS DROPPED.

Severity: Warning

Specification Type: F

Explanation: The maximum number of file description specifications allowed is 20.

NOTE 069 AUTOMATIC TOTALING OF THIS FIELD RESULTS IN GENERATED FIELD NAME CONFLICTS. ASSUME POSITION 39 BLANK.

Severity: Warning

Specification Type: O

Explanation: This error occurred for one of these reasons:

- A field name that was generated for totaling was previously defined as alphameric.
- Another field name, which is a duplicate through 5 characters to this field name, appears in the program and is used as a totaling field.

Both names are printed, which may cause incorrect format of the output line.

NOTE 070 GENERATED LINE IS TOO LONG. EXCESS IS DROPPED.

Severity: Warning

Specification Type: O

Explanation: Either the length of H-*AUTO line exceeds the record length or the length of D/T*AUTO line exceeds twice the record length, depending on which is specified.

NOTE 071 INVALID OUTPUT RECORD TYPE IN POSITION 15. SPECIFICATION IS DROPPED.

Severity: Warning

Specification Type: O

Explanation: Entry must be either H, D, T, or E.

NOTE 072 PAGE FIELD NOT AVAILABLE FOR USE IN PAGE HEADING. NO PAGE NUMBERING WILL OCCUR.

Severity: Warning

Specification Type: O

Explanation: Auto report uses one of the unused RPG II page fields (PAGE, PAGE1-PAGE7) for page numbering. If all the page fields have been used in the program, no page numbering will occur. Format of the output line may be incorrect.

NOTE 073 ERROR OCCURRED WHILE •ATTEMPTING TO PUT A LIBRARY SOURCE MEMBER IN THE LIBRARY. SOURCE WAS NOT PUT IN LIBRARY.

Severity: Warning

Specification Type: Not applicable

Explanation: This error occurred for one of these reasons:

- Library is full.
- Invalid operation (library may not be allocated).

The program is not cataloged.

NOTE 074 DUPLICATE LIBRARY SOURCE MEMBER NAME WAS FOUND IN THE LIBRARY. PREVIOUS MEMBER WAS REPLACED.

Severity: Warning

Specification Type: Not applicable

NOTE 075 ENTRIES IN COLUMNS 25-26, 29, AND/OR 31-74 ARE NOT BLANK, ASSUME BLANK.

Severity: Warning

Specification Type: U

NOTE 076 GENERATED END POSITION FOR TOTAL LINE CONSTANT, 1-9 OR R IN POSITION 39, EXCEEDS RECORD LENGTH. DROP ALL TOTAL LINE CONSTANTS.

Severity: Warning

Specification Type: O

Explanation: This error occurred for one of these reasons:

- The length of the constants for a particular level exceeds the record length.
- The first A-type field encountered has a beginning position greater than the record length.

NOTE 077 LEVEL INDICATORS USED ON T *AUTO LINE IS UNDEFINED. INDICATOR IS DROPPED.

Severity: Warning

Specification Type: O

Explanation: A control level indicator used on a T-*AUTO line must be defined in positions 59 and 60 of the input field specifications. The invalid indicator will be printed with the error note and no total lines will be generated.

NOTE 078 PUNCH OPTION IS INVALID

Severity: Warning

Specification Type: U

Explanation: Punching of a generated source deck is not supported. Use a valid entry in position 7 of the U specification. There is no punching since the option is invalid.

NOTE 079 D-*AUTO IS CONDITIONED BY MORE THAN 7 AN/OR LINES. ONLY THE FIRST 7 WILL APPLY.

Severity: Warning

Specification Type: C

Explanation: The indicators that condition a D-*AUTO line are used to condition the generated EXSR calculation specification needed for total rolling. RPG II will allow only seven lines of AN/OR conditioning indicators in the calculations.

NOTE 080 INVALID ENTRY IN LISTING OPTION, POSITION 30. BLANK IS ASSUMED.

Severity: Warning

Specification Type: U

Explanation: Valid entries are B, P, or blank.

RPG II LINKAGE EDITOR MESSAGES

Following is a list of RPG linkage editor messages. If you receive any of these terminal messages, call your program support representative.

RPG 4000 FIRST RECORD IN $WORK IS NOT A PHASE RECORD

RPG 4001 PHASE NAME ERROR

RPG 4002 ORIGIN ADDRESS SPECIFICATION IN ERROR

RPG 4003 MORE THAN 128 PHASES

RPG 4004 ESL TABLE EXCEEDS CORE

RPG 4005 WORK AREA OVERFLOW

RPG 4011 INVALID OR DUPLICATE PARAMETER ON PHASE RECORD

RPG 4013 DUPLICATE OPTION RECORD

RPG 4014 INVALID RECORD TYPE

RPG 4016 TEXT RECORDS OUT OF SEQUENCE

RPG 4018 DUPLICATE ESL RECORD FOUND

RPG 4020 TEXT RECORD 'LENGTH' EXCEEDS MAXIMUM ALLOWED

RPG 4021 TEXT RECORD RLD POINTS OUTSIDE TEXT AREA

RPG 4022 NO '/*' RECORD IN $WORK FILE

RPG 4023 RPG OVERLAY PROGRAM WITH ONLY ONE PHASE

RPG 4024 $SOURCE WORK FILE EXCEEDED

Explanation: The $SOURCE parameter on the command statement can be increased and the compile rerun. The $SOURCE parameter is the second parameter on the command statement.

Appendix C. WORKSTN Return Codes

Columns 23 through 26 of the WORKSTN INFDS data structure contain the following information:

Major Return Code	Minor Return Code	Explanation
00	00	Operation successful
01	00	New requestor successfully attached to program
02	00	Stop requested by system operator
04	00	Output exception occurred
08	00	Attempt to acquire terminal already attached–no error
11	00	Accept rejected–no invites (end-of-file)
18	00	Acquire failed temporarily
24	00	Terminal released by operator taking 2 option on inquiry display
28	00	Operation rejected–program previously released single requestor
32	00	Acquire failed–unauthorized user
34	01	Input rejected–buffer too small
38	00	Attempt to acquire terminal failed
40	00	Requested terminal offline
41	00	Print operation rejected–would cause more than eight spool files
42	00	Print operation rejected–file not valid
46	00	Print request from unlocked keyboard
48	00	Printer allocated on print operation
80	00	Permanent system error occurred

See *Specifications for the INFDS Data Structure* in Chapter 13 for more information about how the INFDS uses these return codes. For information on major and minor return codes that result from the use of the Interactive Communications Feature, see the *Interactive Communications Feature Reference Manual.*

Glossary

This glossary contains some terms that are used in this manual. Data processing terms are defined in *IBM Data Processing Glossary*, GC20-1699.

This glossary includes definitions developed by the American National Standard Institute (ANSI) and the International Organization for Standardization (ISO). This material is reproduced from the American National Dictionary for Information Processing, copyright 1977 by the Computer and Business Equipment Manufacturers Association, copies of which may be purchased from the American National Standards Institute, 1430 Broadway, New York, New York 10018.

ANSI definitions are identified by an asterisk. An asterisk to the right of the term indicates that the entire entry is reprinted from the *American National Dictionary for Information Processing*; where definitions from other sources are included in the entry, ANSI definitions are identified by an asterisk to the right of the item number.

$SOURCE file: The file into which the RPG II program reads the RPG II source programs.

$WORK file: The file required by the RPG II program while processing the RPG II source program.

access method: A technique for moving data between main storage and input/output devices.

accumulating: The process of totaling a particular field's values as records are being processed.

acquire: To assign a nonrequesting display station to a program.

add file: An indexed or sequential disk file defined as an input, output, or update file to which records are added if the appropriate entries are made on the file description and output specifications.

address: A name, label, or number that identifies a register, location in storage, or any other data source.

addressing: In data communications, the means by which the sending or control station selects the unit to which it will send a message.

addrout file: A record address disk file produced by the sort program. An addrout file contains the binary relative record numbers of records in a disk file and can be used to process input files and to update files that are designated as primary or secondary files.

alphabetic character: Any one of the letters A through Z, or one of the special symbols #, $, and @.

alphameric character: An alphabetic character, or one of the digits 0 through 9.

AND relationship: The specifying of conditioning indicators so that the operation is performed only when all conditions are met.

application program: A program used to perform a particular data processing task; for example, inventory control or payroll.

array: A series of elements with like characteristics. Like a table, an array can be searched for a uniquely identified element. Unlike a table, however, elements in an array can be accessed by their position relative to other elements. Contrast with *table*.

array file: An input file containing array entries.

ASCII: American National Standard Code for Information Interchange. Synonymous with USASCII.

auto-answer: In data communications, a machine feature that allows a station to respond to a call that it receives over a switched line without operator action.

auto-call: In data communications, a machine feature that allows a station to initiate a connection with another station over a switched line without operator action.

auto line: A line that is a part of the auto report specifications in an auto report program.

auto report: A function of the RPG II program product that simplifies the defining of formats for printed reports, and allows the inclusion of previously written statements in new programs. Auto report uses simplified specifications and standard RPG II specifications to generate a complete RPG II source program.

auto report option specifications sheet: An RPG II coding sheet used to identify the library member to be cataloged by the auto report program.

auto report program: A set of instructions (program) that use the RPG II auto report function. See *auto report*.

backup diskette: A diskette that contains information that was copied from another diskette or from disk. A backup diskette is used in case the original information is unintentionally altered or destroyed.

basic data exchange: A data file format for exchanging data on diskettes between systems or devices. Basic data exchange refers to data files only, not entire diskettes.

batch processing: A method of running jobs that does not require continuous operator attention; that is, processing that is not interactive. Contrast with *interactive processing*.

binary: Relating to, being, or belonging to a system of numbers having 2 as its base (for example, the binary digits 0 and 1).

binary synchronous communications (BSC): A flexible form of line control that provides a set of rules for transferring data over a communications line connecting two or more devices that use a communications adapter.

block: (1) A record or a collection of contiguous records recorded or processed as a unit. (2) On System/34, a 10-sector unit of disk storage that contains 2,560 bytes.

BSC: Binary synchronous communications.

BSCA: The device name specified on the file description specifications for a communications adapter.

byte: The representation of a character by 8 binary bits; the amount of storage required for one EBCDIC character.

calculation specifications sheet: An RPG II coding sheet used to describe processing to be done by the program.

called station: On a switched line in data communications, the location to which a connection is initiated.

calling station: On a switched line in data communications, the location that initiates a connection.

central station: See control station.

chained file: An input, output, or update disk file for which the CHAIN operation code is used to read records randomly or to load a direct file.

character: *A letter, digit, or other symbol that is used as part of the organization, control, or representation of data.

character set: A defined collection of graphic symbols.

Cmd (key): A display station function control key that, when pressed, causes System/34 to recognize the 14 keys on the top row of the keyboard as command function keys. See *command function keys.*

collating sequence: The order each character holds in relation to other characters according to the bit structure.

combined file: Used as both an input and output file. A combined file can be specified on the file description specifications only for a SPECIAL device or a WORKSTN device.

command display station: (1) A display station defined during system configuration as able to request and initiate jobs. (2) A display station that can be used for data entry or interactive processing. See also *data display station.*

command function keys: The 14 keys on the top row of the display station keyboard that are used with the Cmd key to request functions of program products and user programs. By using the uppercase shift key, 24 different key functions are available. In RPG II, the command function keys correspond to command key indicators KA through KN and KP through KY.

command statement: A statement that requests the performance of a particular function. A command statement always contains the name of the command and may include parameters or data. The two types of command statements are control commands and procedure commands. See *control command; procedure command.*

comments: Words or statements in a program that serve as documentation rather than as instructions to the compiler.

communications adapter: A hardware feature that enables System/34 to become a part of a data communications network.

compile: *To prepare a machine language program from a computer program written in another programming language by making use of the overall logic structure of the program, or generating more than one machine instruction for each symbolic statement, or both, as well as performing the function of an assembler.

compile-time table or array: A table or array compiled with the source program that becomes a permanent part of the object program. See also *execution-time array; preexecution-time table or array.*

compiler: A program that translates a series of instructions, written in a programming language, into a program the system can execute.

computer: An electronic device or group of interrelated devices capable of processing data, either separately or in conjunction with other interrelated devices.

conditioning: Using indicators to control when calculations or output operations are done.

consecutive processing: File processing that reads records in the order in which they exist in the file.

CONSOLE files: Files assigned to the device CONSOLE on the file description specifications. A CONSOLE file can be either an input data file or a record address file. Input to a CONSOLE file is received from only one command display station.

constant: A data item that does not change during execution of a program. This item represents itself and is actually used in processing rather than being a field name representing the data. For example, cost is a name representing a field containing data that changes. The constant 100 is actual data used that does not change.

control and file description specifications sheet: An RPG II coding sheet on which the programmer provides special information about the program, describes the system to the RPG II compiler, and describes the files used.

control break: A change in the contents of a control field.

control command: A command statement used by an operator to control system or display station operations. A control command does not run a procedure and cannot be used in a procedure. See also *command statement; procedure command.*

control field: One or more specified fields that are compared to determine the record sequence that identifies a record's relationship to other records (such as a part number in an inventory record). Control fields are compared from record to record to determine when certain operations are to be performed.

control group: A set of records all having the same control field information.

control level indicator: An indicator used to specify certain fields as control fields and to control which operations are performed at total time in the RPG II program cycle.

control statement: A specification that provides special information about the program and describes the system to the RPG II compiler.

control station: The primary or controlling computer in a multipoint data communications configuration. The control station controls the sending and receiving of data.

CRT: The device name specified on the file description specifications when the display screen is used as an output device for normal and exception output.

data: *A representation of facts, concepts, or instructions in a formalized manner suitable for communications, interpretation, or processing by human or automatic means.

data display station: (1) A display station that was defined during system configuration as only capable of being acquired by an executing program. A data display station cannot request or initiate jobs. (2) A display station that can be used for data entry only. See also *command display station.*

data link: The equipment and protocols used for data transmission over a communications line.

data link message: A message sent over a data link that is in the form of EBCDIC or ASCII code.

data processing: Performing a series of planned instructions on information to achieve a desired result.

delete-capable file: A file that allows deletion of records. It is defined by specifying the DFILE parameter on the FILE OCL statement that builds the file.

demand file: A file that can be specified as an input, update, or combined file and that is used with the READ or KEY operation code.

descending order: The arrangement of data in a specified field from high to low. See *collating sequence.*

detail record: An output record produced during the detail output operation of the RPG II program cycle.

detail time: A portion of the RPG II program cycle in which calculation and output operations for specified fields are performed for each record read.

diagnostic message: An output message that identifies RPG II specification errors and their severity.

digit: One of the characters 0 through 9.

direct file: A disk file in which records are assigned specific record positions. Regardless of the order in which records are put in a direct file, they always occupy the assigned position in the file. Direct files can be processed by the consecutive, random by relative record number, and addrout file processing methods.

disk: A flat, circular plate with a magnetic surface on which program libraries and data files can be stored.

disk file: An organized collection of related records on disk that are treated as a unit.

diskette: A thin, flexible magnetic disk permanently enclosed in a semirigid protective jacket.

display: When a display screen format is executed, all of the information on the display screen. See *display screen format.*

display screen: The part of a display station on which data, messages, or other information is displayed.

display screen format: A two-part table that defines a display presented by display station data management. Display screen formats are generated and placed in a library load member by the display screen format generator utility program ($SFGR).

display screen layout sheet: A form used to plan the location of data on the display screen.

display station: An input/output device containing a display screen on which data is displayed and an attached keyboard from which data is entered. A display station can be designated as the system console or as a command or data display station at system configuration time.

display station local data area: A 256-byte area on disk that can be used to pass information between jobs and job steps during a session. A separate local data area exists for each command display station.

displayed message: A message that appears on the display screen and is documented in the *Displayed Messages Guide.*

documentation: A written explanation of a program, its use, function, and operations.

EBCDIC: Extended binary coded decimal interchange code.

EBCDIC transparency: See *transparent text mode.*

edit: To punctuate a field by suppressing zeros and inserting commas, decimal points, dollar signs, or other constant information.

edit code: A number or letter indicating that editing should be done according to a predefined pattern. This includes zero suppression and punctutation.

element: The smallest addressable unit of a table or array.

end of file: The end of records in a file.

entry function key: Any of the following four keys that the operator can use to perform an enter function: Field Exit, Field -, Field +, or Enter/Rec Adv.

EOF: End of file. The end of records in a file.

EOJ: End of job.

erase: To remove a unit of data.

error message: See *diagnostic message.*

execute: To cause an instruction, program, utility, or other machine function to be performed.

execution-time array: An array that is loaded by input specifications after actual execution begins. See *compile-time table or array; preexecution-time table or array.*

extension and line counter specifications sheet: An RPG II coding sheet used to provide information about record address, table, and array files used by the program and the number of lines to be printed on the forms that are used.

external indicators: Eight indicators (U1 through U8) that are normally set by the SWITCH OCL statement prior to processing. The indicators can be altered by the job during execution. External indicators are sometimes used to specify which files are to be used in multifile processing.

factor: In RPG programming, a field name or constant used in a calculation operation.

field: One or more bytes of related information in a record.

field indicator: An indicator used to indicate whether a given field in an input record is plus, minus, zero, or blank.

field length: The number of positions allowed for a given field, determined by the maximum length of information that will be entered in the field.

field name: In RPG programming, a combination of no more than 6 alphabetic or numeric characters that identify a field. The first character must be alphabetic, and no blanks can appear between characters.

figurative constant: An implied literal specified in the calculation specifications without a length definition. The length is determined by the fields with which the constant is used.

file: A set of related records. See *disk file; input file; output file; primary file; secondary file.* For types of file organization, see *sequential file; indexed file; direct file.* For types of file processing, see *consecutive; sequential by key; random by key; random by relative record number; addrout file.*

filename: The name associated with a file. A filename can be from 1 to 8 characters long. The first character must be alphabetic, and the remaining characters can be any combination of alphabetic or numeric characters. Blanks cannot appear between characters in a name.

first page (1P) indicator: An indicator used to specify which lines (such as headings) should be printed on the first page only.

generated program: A program that has been compiled.

group indication: In RPG II programming, the printing of control information for only the first record of a group of records containing identical control information.

half adjust: A method of rounding off a number by adjusting the last significant digit.

heading: A constant, usually printed at the top of the page, identifying the information or report on that page.

hex: Hexadecimal.

hexadecimal: Pertaining to a number system with a base of 16; valid digits range from 0 (zero) through F.

host: See control station.

ideographic: Consisting of both pictograms and graphics and often other types of symbols.

ideographic character: A pictogram or graphic that requires 2 bytes of storage.

ideographic character set: A character set that contains pictograms or graphics that can be used to represent ideas.

ideographic field: In a record, one or more ideographic characters of related information bracketed by S/O and S/I control characters.

ideographic support: The combination of hardware and software elements allow the use of ideographic data on a System/34.

index key: The field within a record that identifies that record in an indexed file. The index key and record location for each record in the file are stored in the file index.

indexed file: A file in which the position of each record is recorded in a separate portion of the file called an index. The index contains an index key and disk address for each record in the file. Indexed files can be processed by the consecutive, sequential by key, sequential within limits, random by key, or addrout file processing methods. Contrast with *direct file; sequential file.*

indicator: (1) A two-digit or 2-character entry on the specification sheets used to tell when certain operations are to be performed. (2) An internal switch used by the object program to remember when a certain event occurs and what to do when the event occurs. See *control level indicator; field indicator; first page indicator; last record indicator; overflow indicator; record identifying indicator; resulting indicator.*

input: (1) Information to be transferred from disk or keyboard to storage. (2) Data that is to be operated on (processed) by the computer.

input file: A set of records a program uses as source information.

input job queue: A list of jobs that are waiting to be processed by the system. This list is maintained on the disk. Each entry in the list references a procedure stored in a library on disk.

input specifications sheet: An RPG II coding sheet used to identify the different types of records in each input file and to describe the fields of each record.

inquiry: (1) A request (entered from a display station) for information in storage. See also *inquiry program.* (2) A request for information that puts the sytem into inquiry mode. (The operator initiates an inquiry by pressing the Attn key.)

inquiry file: The file into which an inquiry is made using the inquiry function of the customer program.

inquiry mode: A method of operation when the system is responding to an inquiry. (The operator puts the system in inquiry mode by pressing the Attn key.)

inquiry program: (1) A program that enables the operator to access information from a disk file. See *inquiry.* (2) A program that is executed while the system is in inquiry mode.

instruction: A statement that specifies an operation to be performed by the computer and the locations in storage of all data involved in that operation.

interactive processing: A method of processing in which each operator action causes a response from a system or program, as in an inquiry system or an airline reservation system.

intermediate block checking: In binary synchronous communications, a provision that allows validity checking of each logical record, rather than checking of the total buffer, when large buffers of data are received.

ITB: Intermediate block check.

keyboard: A systematic arrangement of keys by which commands, control statements, and data are entered into a computer.

KEYBORD: The device name for an input or demand file to be used with the KEY and/or SET operation codes, which control input and prompts to a command display station.

last record indicator: An indicator that signifies when the last data record is processed and that is used to condition all operations that are to be done at the end of job (EOJ).

library: An area on disk that can contain load members, procedure members, source members, and subroutine members. See also *system library; user library.*

library member: A named collection of records or statements in a library. See *load member; procedure member; source member; subroutine member.*

limits file: A record address file containing limits records when the sequential within limits processing method is used.

limits record: A record that consists of the lowest record key and the highest record key of the records in the indexed disk file that are to be read.

literal: A symbol or a quantity in a source program that is itself data, rather than a reference to data.

load member: A collection of instructions that the system can execute to perform a particular function, regardless of whether the function is requested by the operator or specified in an OCL statement. Load members are stored in a library.

look-ahead field: A field that allows the program to look at information in a field on the next record that is available for processing in any input or update file.

LR indicator: Last record indicator.

machine language: A language that can be interpreted and used by a computer.

main storage: (1) General purpose storage of a computer. (2) Program-addressable storage from which instructions can be executed and from which data can be loaded directly into registers.

manual answer: In data communications, operator actions to make a called station ready in response to a call received over a switched line. Contrast with *auto-answer.*

manual call: In data communications, operator actions taken to initiate a connection with a station over a switched line. Contrast with *auto-call.*

master file: A collection of permanent information, for example, a customer address file, that is often processed along with a transaction file.

match fields: In multifile processing, fields used to condition operations to be done when the fields of records in different files containing these fields match.

match level: The value assigned to the match field (M1 through M9). The match level identifies fields by which records are matched during multifile processing.

message identification code (MIC): A four-digit number that identifies a record in a message member. This number can be part of the message identifier.

message identifier: A field in the display or printout of a message that directs the user to the description of the message in a message guide or a reference manual. This field consists of up to four alphabetic characters, followed by a dash or a blank, followed by a four-digit number (the message identification code).

message load member: A special type of library load member from which the system support program product retrieves the text associated with a specified message identification code.

message member: A library load member in which each record contains a message.

message source member: A special type of library source member containing control and message text statements (MIC and text) required for creating a message load member.

MIC: Message identification code.

MRT: Multiple requestor terminal program.

multiple requestor terminal program: A program that can process requests from more than one requesting display station concurrently. Compare with *single requestor terminal program.*

nonswitched line: In data communications, a connection between systems or devices that does not have to be established by dialing. Contrast with *switched line.*

null response: The action of pressing the Enter/Rec Adv key without having previously keyed any data.

numeric characters: The digits 0 through 9.

object program: A set of instructions in machine language. The object program is produced by the compiler from the source program.

OCL: Operation control language.

operation: A defined action performed on one or more data items, such as adding, multiplying, comparing, or moving information.

operation code: A word or abbreviation specified on the calculation specifications sheet to identify an operation, such as SUB for subtract or ADD for addition.

operation control language (OCL): A programming language used to identify a job and its processing requirements to the System Support Program Product.

OR relationship: Specifying conditioning indicators so that the operation conditioned is performed when either one or both of the conditions are met.

output: Data delivered or ready to be delivered from a device or program, usually after some processing.

output file: A file containing the data that results from processing.

output specifications sheet: An RPG II coding sheet used to specify the records to be written in each output file and the format of the records.

overflow: The condition that occurs when the last line to be printed on the page has been passed.

overflow indicator: An indicator that signifies when the last line on a page has been printed or passed. It can be used to specify which lines are to be printed on the next page.

overflow line: The line specified to be the last line printed on a page.

overflow page: The new page after an overflow has occurred.

overlay: A program segment or phase that is loaded into main storage. It replaces all or part of a previously loaded segment.

packed data field: One byte is used to store two numeric digits. Bits 0 through 3 for one digit and bits 4 through 7 for the other.

packed decimal format: Each byte within a field represents two numeric digits except the rightmost byte, which contains one digit in bits 0 through 3 and the sign in bits 4 through 7. For all other bytes, bits 0 through 3 represent one digit; bits 4 through 7 represent one digit. For example, the decimal value +123 is represented as 0001 0010 0011 1111. Contrast with *zoned decimal format.*

packed field: A field that contains data in the packed decimal format.

packed key: An index key in the packed decimal format.

polling: In a multipoint environment for data communications, an invitation to send, transmitted from the primary station to a specific tributary station.

preexecution-time table or array: A table or array that is loaded at the same time as the object program, before actual execution of the program begins. See also *compile-time table or array; execution-time array.*

primary file: If specified, the main file from which a program first reads records. In multifile processing, the primary file is used to determine the order in which records are selected for processing.

printer: The output device that records information on paper in the form of printed characters.

printer spacing chart: A form used to plan the location of data in the printer output file.

procedure: A set of related OCL statements, and possibly utility control statements, that cause a specific function or set of functions to be performed. A procedure in a library is called a procedure member.

procedure command: A command statement that runs a procedure. A procedure command is a special form of the INCLUDE OCL statement. See also *command statement; control command.*

procedure member: A procedure that is stored in a library.

processing: The handling of input according to specific instructions or rules; performing a series of planned actions upon information (data) to achieve a desired result.

processing unit: The parts of a computer that perform the processing and control functions for the system, perform operations on data, and control output. These units include main storage, main storage processor, control storage, and control processor.

program: (1) A sequence of instructions to a computer, written in a special form the computer can interpret. A program tells the computer where to get input, how to process it, and where to put the results. (2) A set of instructions that tells the computer which operations are to be done and how to do them.

program cycle: A series of operations performed by the computer for each record read.

program listing: A computer printout that gives information about the source program, such as source statements, diagnostic messages, indicators used, storage address of fields and constants used.

program name: The name or code specified in columns 75 through 80 of the RPG II specifications sheets, which identify a program. You can name the program according to its function or use any letters and numbers to identify the program.

prompt: A message issued by a program that requests either information or an operator action to continue processing.

protected field: A field on a display in which operators cannot key data.

random by key: A method of processing chained files using the CHAIN operation code. Records to be processed are identified by record keys.

random by relative record number: A method of processing chained files using the CHAIN operation code. Relative record numbers are used to identify the records to be processed.

record: A collection of related data, treated as a unit. For example, one line of an invoice can comprise a record. A complete set of records could form a file.

record address file: An input file that indicates to your program which records are to be read from a disk file, and the order in which the records are to be read from the disk file.

record identification code: Characters placed in a record to identify that record type.

record identifying indicator: An indicator that identifies the type of record being processed during the current program cycle.

record key: A field within a record that identifies the record in a file.

record length: The total number of characters (bytes) in a record.

record type: The classification of records in a file. Records are classified according to a specific field or fields within each record. Records of the same type have the same fields in the same order and identical record identification codes.

region: The amount of main storage available for a task. Region size is specified by a REGION OCL statement, by the SET procedure, or by the $SETCF utility program.

relative record number: A number that specifies the location of a record in relation to the beginning of the file.

result field: The name of a field specified on the calculation specifications where the outcome of arithmetic calculations is kept.

resulting indicator: An indicator that signifies whether the result of a calculation is plus, minus, or zero; whether a field is greater than, less than, or equal to another field; or whether an element of a table or array was found.

right justify: The placement of data in a field with the least significant digit in the rightmost position.

RPG II: A commercially oriented programming language specifically designed for writing application programs that meet common business data processing requirements.

RPG II display format generator: A program within RPG II that uses the input specifications to generate source input to the $SFGR utility program, if the RPG II program contains a CONSOLE device.

RPG II program cycle: A series of operations performed by the computer for each record read.

RPG II source program: The program used as input to the RPG II compiler. The program is translated into machine language and stored in a library as a load member.

RPGR: RPG II display format generator.

S/I control character: See shift-in (S/I) control character.

S/O control character: See shift-out (S/O) control character.

search word: Data for which you want to find a match in a table or array. The search word is specified in the LOKUP statement.

secondary file: Any file other than the primary file used in multifile processing.

sequence checking: An RPG II function that checks the sequence of records in input, update, or combined files used as primary and secondary files.

sequential by key: A method of file processing that reads records in the order in which the record keys are arranged in the index portion of the file.

sequential file: A file in which records are entered one after the other; one in which there is no relationship between the contents of the records and their positions in the file. Contrast with *direct file; indexed file.* Sequential files can be processed by the consecutive, random by relative record number, and addrout file processing methods.

SEU (source entry utility): A part of the Utilities Program Product that is used to enter and update source and procedure members.

shift-in (S/I) control character: A character that indicates the end of a string of ideographic characters. The shift-in control character is represented by hex 0F.

shift-out (S/O) control character: A character that indicates the start of a string of ideographic characters. The shift-out control character is represented by hex 0E.

single requestor terminal program: A program that can have only one requesting display station at a time. Contrast with *multiple requestor terminal program.*

source member: A collection of records, such as RPG II specifications, that is used as input for a program. Source members are stored in a library.

source program: A set of instructions that represents a particular job as defined by the programmer. These instructions are written in a programming language, such as RPG II.

special character: A character other than a digit, a letter, or #, $, @. For example, *, +, and % are special characters.

specification sheets: Forms on which an RPG II program is coded and described. See *control and file description specifications sheet; extension and line counter specifications sheet; input specifications sheet; calculation specifications sheet; output specifications sheet; auto report option specifications sheets; telecommunications specifications sheet.*

station: A system or device that is capable of sending or receiving data over a communication line.

subroutine: In RPG II programming, (1) a group of instructions that are coded last on the calculation specifications and that can be referenced one or more times elsewhere in the calculation specifications; (2) a group of assembler instructions, assembled as a subroutine, that can be referenced by an RPG II program.

subroutine member: A subroutine that needs to be link edited (joined) before being loaded for execution. Subroutine members are stored in a library.

switched line: In data communications, a connection between a communication controller and a remote station, or between two stations, that is established by dialing. Contrast with *nonswitched line.*

system console: A display station designated to activate certain system functions, and control and monitor system operation, in addition to functioning as a command display station.

system input: The statements and commands that make up the system input stream.

system input device: The device from which the system input stream is being read.

system library: The library containing the members that are part of the System Support Program Product. The system library is labeled #LIBRARY and cannot be deleted from disk. See also *library; user library.*

table: A series of elements with like characteristics. Like an array, a table can be searched for a uniquely identified element. Unlike an array, however, elements in a table cannot be accessed by their position relative to other elements. Contrast with *array.*

table file: An input file containing table entries.

telecommunications specifications sheet: An RPG II coding sheet that describes the information necessary to establish and maintain the BSC link.

total operations: Operations performed only after a group of records has been processed.

total rolling: The transfer of accumulated totals from one field to another during a control break.

total time: That part of the RPG II program cycle in which calculation and output operations specified for a group of records are done.

transmission control characters: In data communications, nondata characters that are included in a message to control communication over a data link. For example, the sending station and the receiving station use transmission control characters to exchange status information; the receiving station uses transmission control characters to flag errors in data it receives.

transparent literal: A literal that begins with an apostrophe followed immediately by the shift-out (S/O) control character and that terminates with the shift-in (S/I) control character followed immediatedly by an apostrophe.

transparent text mode: In data communications, a mode of binary synchronous transmission in which only transmission control characters preceded by the DLE control character are acted upon as line control characters. All other characters having the same bit pattern as transmission control characters are transmitted as data.

tributary station: A secondary or noncontrolling device in a multipoint data communications configuration.

update file: Disk files from which a program reads a record, updates fields in the record, and writes the record back into the location it came from.

USASCII: United States American Standard Coded Information Interchange.

user library: A library created by the user. A user library is in addition to the system library and may contain any type of library member.

valid RPG II names: The following rules apply to names used in RPG II programs. (1) RPG II filenames can be from 1 to 8 characters long. (2) RPG II field or data structure names can be from 1 to 6 characters long. (3) The first character of either a filename, a field name, or a data structure name must be an alphabetic character. The remaining characters can be any combination of alphabetic and numeric characters. (4) Blanks cannot appear between characters in a name.

WORKSTN: The device name specified for a combined file that is used to communicate with one or more display stations.

zero suppression: The elimination of preceding zeros in a number. For example, 00057 when zero suppressed becomes 57.

zoned decimal format: Representation of a decimal value by 1 byte per digit. Bits 0 through 3 of the rightmost byte represent the sign; bits 0 through 3 of all other bytes represent the zone portion; bits 4 through 7 of all bytes represent the numeric portion. For example, in zoned decimal format, the decimal value +123 is represented as 1111 0001 1111 0010 1111 0011. Contrast with *packed decimal format*.

zoned field: A field that contains data in the zoned decimal format.

Index

/COPY statement specifications
  (see also auto report function)
  cataloged RPG II source specifications 15-22
  format of 15-22
  modifying copied specifications 15-26
    file description 15-26
    input field 15-28
  order of specifications included 15-22
  placement in auto report source program 15-22
  sorting of specifications 15-22
/EJECT 1-13
/SPACE 1-14
/TITLE 1-13
(/COPY function) 15-26
  using a WORKSTN device 13-32
& (ampersand)
  auto report copy function 15-26
  use in edit word 9-29
$ (fixed or floating dollar sign) 9-32
$SFGR utility program
  compiling display screen formats
    for a CONSOLE file . 12-1
    for a WORKSTN file 13-1, 13-12
* (asterisk)
  asterisk fill (asterisk protection) 9-28
  comment line 1-13
  indication of auto report total line 15-18
**
  alternate collating sequence table 2-9
  file translation table 2-13
  look-ahead fields 7-7
  tables or arrays 4-2
*AUTO output specifications
  (see also auto report function)
  accumulating totals 15-15
    resetting total fields to zero 15-18
  asterisk indication 15-18
  column headings 15-18
  detail printing 15-12
  end position entries 15-15, 15-30
  field description specifications
    A in column 39 15-15
    blank or B in column 39 15-13
    C in column 39 15-18
    1-9 or R in column 39 15-20
  field or literal on generated total line 15-20
  generated edit codes 15-14
    (see also edit codes)

*AUTO output specifications (continued)
  generated end positions 15-30
  generated RPG II specifications 15-5, 15-34
  generated total fields 15-16
    restrictions in naming fields 15-16
  group printing 15-21
  indicators, output 15-13
  number of printer files allowed 15-8
  record description entries 15-12
  report format 15-30
  spacing and skipping 15-13
  suppress asterisk indication 15-7
  total rolling 15-15
*AUTO page headings specifications
  (see also auto report function)
  centering headings 15-32
  date suppression 15-10
  editing 15-11
  field description entries 15-11
  format of page heading 15-30, 15-32
  generated date 15-10
  generated page number 15-10
  generated RPG II specifications 15-34
  indicators, output 15-10
  number of heading lines allowed 15-8
  number of printer files allowed 15-8
  page number 15-10
    suppression of 15-10
  placement of fields and literals in heading 15-30
  record description entries 15-10
  spacing and skipping 15-10
*AUTO specifications (see *AUTO output specifications, *AUTO page headings specifications)
*BLANK 8-13
*BLANKS 8-13
*CANCL 13-29
*DETC 13-29
*FILES 2-13
*GETIN 13-29
*INP keyword 13-27
*MODE keyword 13-27
*OPCODE 13-27
*OUT keyword 13-27
*PLACE special word
  description 9-17
  example 9-20
*RECORD 13-27
*STATUS 13-26
*suppress (auto report) 15-7
*ZERO 8-13
*ZEROS 8-13

A$$SUM subroutine (auto report) 15-34
accumulating totals (auto report) 15-15
ACQ (acquire) operation code
  bytes of generated code 16-10
  description 10-7, 13-17
  exception/error indicator 10-7, 13-17
acquire, definition of G-1
acquiring devices 13-4
add a record
  file description specifications entry 3-51
  output specifications entry (ADD) 9-4
add file, definition of G-1
ADD operation code
  bytes of generated code 16-9
  description 10-7
adding records to a file
  delete-capable direct and sequential files 3-52, 3-56
  example 3-53
  file description entry 3-51
  indexed file 3-51
  output specifications entry 9-4
  RECNO keyword 3-49
  sequential file 3-52
additional input/output area 3-22
address output file (see addrout file)
addressing characters (BSCA file) 6-6
addrout file
  (see also record address files)
  binary relative record number 3-17
  definition of G-1
  file description specification entries
    file organization (column 32) 3-20
    length of key field (columns 29-30) 3-17
    mode of processing (column 28) 3-8
    record address type (column 31) 3-19
    summary charts 3-58
  processing 3-17
adjusting arithmetic results (see half adjust)
alignment of printer forms 2-11
alphabetic character, definition of G-1
alphameric character, definition of G-1
alphameric field
  length 7-30
  moving (MOVE operations) 10-4
alphameric literal 8-12
alternate collating sequence
  coding sheet 2-7
  control specifications entry 2-4
  input record format 2-9
  operations affected 2-4
alternating format (see related tables)
ALTSEQ
  (see also alternate collating sequence)
  definition 2-9

ampersand (&)
  auto report copy function 15-26
  use in edit word 9-29
AN and OR lines
  calculation specifications entries 8-6
  examples 8-7, 8-8
AND and OR lines
  (see also output indicators, record identification codes)
  *AUTO output specifications 15-13
  *AUTO page headings specifications 15-10
  input specifications 7-2, 7-23, 7-24
  output specifications 9-2, 9-8
AND relationship
  calculation specifications 8-6
  input specifications 7-24
  output specifications 9-8
apostrophe
  use with edit word 9-28
  use with output constant 9-26
AREA field (SUBR21) 13-34
arithmetic operations
  (see also calculation operations, factor 1, factor 2, half adjust individual entries, operation codes)
  ADD (add) 10-7
  DIV (divide) 10-16
  length of fields in arithmetic operations 10-4
  MULT (multiply) 10-45
  MVR (move remainder) 10-45
  SQRT (square root) 10-64
  SUB (subtract) 10-64
  XFOOT (summing elements of an array) 10-69
  Z-ADD (zero and add) 10-69
  Z-SUB (zero and subtract) 10-69
arrangement of source program 1-3
array
  (see also table)
  *AUTO output field name 15-14, 15-21
    restriction with A in column 39 15-17
  *AUTO page headings 15-11
  adding records to a short array 14-9
  alternating 14-3
  auto report specifications
    format of compile-time arrays 15-1
    order in generated program 15-1
  binary format 4-6
  building (see loading)
  building via calculations (see execution-time array)
  calculating totals with arrays 14-14
  compile-time
    definition of 14-1
    rules for loading 14-4
  contained in a data structure 14-4
  creating input records 14-3
  decimal positions 4-6
  defining 14-3

- array (continued)
  - definitions of terms 14-1
  - dynamic arrays (see execution-time array)
  - editing 14-10
  - element 14-1
  - element as result field 8-20
  - entries
    - length of 4-6
    - number of 4-5
    - number per record 4-3
  - examples of using 14-14
  - execution-time
    - definition of 14-1
    - rules for loading 14-4
  - extension specifications 4-1
    - summary chart 4-10
  - file
    - description of 3-5
    - sequence 4-8
  - file description specifications entry 3-4
  - filename (when required on file description specifications) 3-2
    - formatting output 14-14
  - full array, definition of 14-1
  - index 14-7
    - field 14-14
    - fixed 14-14
  - length of entry 4-6
  - loading
    - compile-time 14-4
    - execution-time 14-4
    - from more than one record 14-7
    - from one record 14-4
    - placement in source program 14-4
    - preexecution-time 14-4
    - via input or calculation specifications
      - LOKUP operation 10-32
  - maximum allowed per program 4-1
  - modifying contents 14-9
  - moving (MOVEA operation code) 10-40
  - name
    - and index 14-7
    - as data structure subfield 7-21
    - as field name in *AUTO output 15-14, 15-17, 15-21
    - as field name in *AUTO page headings 15-11
    - as result field 8-20
    - extension specifications 4-2
    - file description specifications 3-2
    - how to form 4-3
    - input specifications 7-30
    - rules for 14-7
  - number of entries per array 4-5
  - number of entries per record 4-3
- array (continued)
  - order in source program 14-4
    - formatting 14-14
    - printing of elements 14-20
  - packed format 4-6
  - preexecution-time 14-1, 14-4
  - printing elements of 14-20
  - referencing in calculations 14-8
  - referencing in RLABL operation 10-48
  - related, definition of 14-1
  - relationship of columns 27-32, 11-18, and 33-85 (extension specifications) 4-5
  - searching (see LOKUP operation code)
  - sequence of specifications 14-3
  - short, definition of 4-6
  - square root (SQRT operation) 10-64
  - summing elements of (XFOOT operation) 10-69
  - using field indexes 14-14
  - using fixed indexes 14-14
  - using name and index 14-7
  - using name only 14-7
  - XFOOT operation 10-69
- assembler language
  - input/output subroutine 3-36
  - subroutines
    - considerations for programmer 10-53
    - linkage to 10-18
- asterisk (*) comment line 1-13
- asterisk (*) indication
  - on auto report total lines 15-18
  - suppression of 15-7
- asterisk fill (asterisk protection) 9-28
- attributes, program 13-1
- AUTO command statement 18-8
- AUTO procedure 18-9
  - use of display station local data area 18-8
- auto report copy function (see /COPY statement specifications)
- auto report function
  - (see also *AUTO output specifications, *AUTO page headings specifications, /COPY statement specifications, auto report option specifications)
  - A$$SUM subroutine 15-34
  - compilation of source program 18-8
  - definition of G-1
  - examples of 15-45
  - generated specifications
    - calculation 15-34
    - format of 15-1
    - order of 15-5
    - output 15-34
  - halts 18-10
  - messages B-54
  - operations of 15-4
  - programming aids 15-39
  - sample program 19-21

auto report output function (see *AUTO output specifications)
auto report page heading function (see *AUTO headings specifications)

B-type program 2-9
back space key (see function control keys)
BEGSR (begin subroutine) operation code 10-7
binary field
  comparison with packed and zoned decimal fields 7-28
  for table or array file 4-6
  format 7-29
  input specifications 7-26
  output specifications 9-26
binary relative record number 3-17
  (see also addrout file)
binary synchronous communications (BSC)
  definition of G-2
  description 6-1
binary synchronous communications adapter
  (see also BSCA file, telecommunications specifications)
  BSCA device name 3-28
bit operations 10-5
  (see also BITON, BITOF and TESTB)
bit testing (TESTB) 10-67
BITOF (set bit off) operation code
  bytes of generated code 16-9
  description 10-8
BITON (set bit on) operation code
  bytes of generated code 16-9
  description 10-8
blank after
  *AUTO output 15-15
    generated for total fields 15-18
  *AUTO page headings 15-11
  output specifications 9-22
block length
  file description specifications entry 3-7
  relation to record length entry 3-8
branching operations 10-6
  see also GOTO operation code, TAG operation code)
branching to assembler language subroutine 10-18
BSC (binary synchronous communications) 6-1
BSCA file
  addressing characters 6-6
  block length 3-7
BSCA file (continued)
  description of 3-29
  file description specifications 6-6
  polling characters 6-6
  record length 3-7
  wait time 6-5
buffer erasing, CONSOLE 8-20
bypass (halt recovery option) 18-11
bytes (see packed or binary fields)
bytes of generated code for calculations 16-9

C/Z/D (character/zone/digit)
  in record identification code 7-23
calculation
  bytes of generated code 16-9
  detail time 1-6
  factors 1 and 2 8-12
  generated by auto report 15-5, 15-34
  indicators
    conditioning 8-6
    control level 8-2
    in AND relationship 8-6
    resulting 8-23
  operations
    (see also operation codes)
    coding techniques 16-8
    summary of 8-14, 10-4
  order in auto report generated program 15-5
  specifications 8-1
    general description 1-5
    order of 8-1
    relationship between columns 7-8 and 9-17 8-11
    summary A-19
  subroutines 10-6
    coding of 10-19
    operation codes 10-6
  total time 1-6
cataloged RPG II source specifications used by /COPY function 15-22
cataloging auto report generated program 15-7
CHAIN operation code 10-11
  (see also direct file load, random processing)
  bytes of generated code 16-10
  specifying storage index 3-49
chained file
  description of 3-5
  number allowed per program 3-5
  random processing 3-11
character
  (see also record identification codes)
  collating sequence 2-6

character (continued)
  considering characters equal 2-5
  grouping by zone and digit 7-23
  hexadecimal equivalents 2-6
  in record identification code 7-22
  structure (see collating sequence, packed or binary field)
    unprintable character option 2-14
checking sequence (see sequence checking)
clear key (see function control keys)
Cmd (see command key)
code (see edit code, record identification code)
codes, operation (see operation codes)
coding sheet (see calculation specifications, control specifications, display screen format specifications, display screen layout sheet, extentsion specifications, file description specifications, input specifications, line counter specifications, output specifications, telecommunications specifications, translation table and alternate collating sequence coding sheet)
coding subroutines 10-19
collating sequence 2-6
  (see also alternate collating sequence)
column headings
  (see also *AUTO page headings specifications)
  centering of 15-30
  overflow of D/T-*AUTO print lines 15-32
  spacing and skipping 15-30
combined file
  definition G-2
  description 3-4
  translation 2-11
command display station (definition) G-2
command function keys (see command keys)
command key indicators (KA-KN, KP-KY)
  (see also SET operation code, WORKSTN file)
  conditioning calculation operations 8-6
  conditioning output operations 9-8, 9-11
  corresponding command keys 1-12, 10-54
  general description 1-12
  setting indicators 1-12
  specified by SET operation 10-54
command keys
  Cmd 10-54
  corresponding indicators 10-54
  definition G-2
  enabling/disabling 13-32
  end of file (CONSOLE) 3-6
  null response 10-54
  resetting 10-54
  using SET operation to specify ones
    to be pressed 8-23, 10-54
  with a WORKSTN file 1-12
command statements
  AUTO 18-8
  RPG 18-1
  RPGR 18-4
  RPGX 18-4
comments
  /COPY statement 15-22
  for program documentation 1-13
  on calculation specifications 8-25
  on extension specifications 4-10
  on table input records 4-3
  order in auto report generated program 15-5
  use of asterisk (*) 1-13
common entries on specification sheets 1-12
COMP (compare) operation code
  bytes of generated code 16-9
  description 10-13
compare and testing operations 10-5
compare, arithmetic
  alternate collating sequence 2-4
  description 10-13
compilation
  auto report program 18-8
  halts 18-11
  RPG II format generator 18-4
  source program 18-1
compile-time arrays
  definition 4-2, 14-1
  rules for loading 14-4
compile-time table
  definition 4-2, 14-1
  rules for loading 14-4
compiler listing, spacing of 1-13
compiler messages B-1
compiler program 1-1
conditioning files 3-57
  (see also external indicators)
conditioning indicators (01-99)
  bytes of generated code 16-10
  calculation 8-6
  general description 1-9
  output 9-8
configuration for data communications network 6-2
consecutive processing of files 3-9
CONSOLE file
  array name as input field name 7-30
  as record address file 3-29
  block length 3-7
  buffer, erasing 10-57, 12-7
  description of 3-30
  display screen format member
    generation of 12-1
    name 2-15, 12-2
    RPGR command statement 18-4
  end of 3-6
  ERASE function 8-20, 10-57
  file description specifications chart 3-62
  how to specify subfields 12-7, 12-8
  record identifying indicators 12-6
  record length 3-7
  restrictions 12-1

- CONSOLE file (continued)
  - sample program 19-1
  - sequence checking 3-6
  - specifications 12-5
  - used with KEYBORD/CRT 12-1
- constants
  - (see also edit word, literal)
  - *AUTO output
    - A in column 39 15-15
    - blank or B in column 39 15-13
    - C in column 39 15-18
    - 1-9 or R in column 39 15-20
  - *AUTO page headings 15-11
  - definition G-3
  - rules for use on output specifications 9-26
- continuation line 3-45
- continuation line option
  - DISK device 3-49
  - name of label exit 3-46
  - SPECIAL device 3-45
  - WORKSTN device 3-45, 13-14
- continuation record (see continuation line)
- continue (halt recovery option) 18-11
- control break
  - (see also control field, control group, control level indicators)
    - description 1-6
  - how to avoid unwanted 7-37
  - on first cycle 7-36
- control field
  - (see also control break, control level indicators)
  - assigning on input specifications 7-34
  - definition G-3
  - description 1-6
  - split 7-36, 7-39
- control group
  - definition G-3
  - description 1-6
  - (see also control break, control field, control group, control elvel indicators)
- control level
  - (see also control break, control field, control group, control level indicators)
  - calculation specifications 8-2, 8-11
  - input specifications 7-34
- control level indicator (L1-L9)
  - assigning to input fields 7-34
  - auto report
    - adding to copied input specifications 15-28
    - determining levels of generated totals 15-16
    - effect in group printing 15-22
  - calculation specifications
    - columns 7-8 8-2
- control level indicator (L1-L9) (continued)
  - calculation specifications (continued)
    - columns 9-17 8-6
    - relation between entries 8-11
  - general description 1-11
  - record identifying indicator 7-7
  - with subroutines 10-6
- control specifications 2-1
  - general description 1-4
  - summary A-1
- controlled cancel (halt recovery option) 18-11
- controlling calculations 8-2
- controlling output 9-8
- copy function, auto report (see /COPY statement specifications)
- CR (negative balance symbol) 9-30
- creating a direct file (see direct file load)
- creating a source library member 1-1
- creating overlays 16-1
- cross-reference listing
  - format of 18-10
  - identification of 2-15
  - parameter for 18-1
  - RPGX command 18-4
- CRT file
  - block length 3-7
  - description 3-32
  - file description specifications 3-62
  - record length 3-7
  - spacing and skipping 9-8
- currency symbol (see also fixed currency symbol and floating currency symbol)
- cycle, program
  - detailed description 17-1
  - general description 1-5
  - matching record 11-2
  - with a WORKSTN file 13-6, 13-7

- D-*AUTO
  - (see also *AUTO output specifications)
  - overflow of print lines 15-33
- D specifications, summary of A-37
- data communications programming 6-1
  - (see also telecommunications)
- data display station (definition) G-3
- data format (see character, packed or binary field)
- data structure
  - as result field 8-20
  - description 7-17
  - display station local data area 7-6
  - examples 7-18
  - INFDS subfields 7-21, 13-25

- data structure (continued)
  - length 7-22
  - overlapping subfields 8-12
  - referenced in RLABL operation 10-48
  - restrictions 7-22
  - subfield names 7-21
- date edit 2-3
- date format 2-3
- date generated for *AUTO page headings 15-10
  - suppressing the date 15-7
- date option 2-3
- date suppression (auto report) 15-7
- DEBUG operation code
  - control specifications entry 2-3
  - description 10-15
  - format of records 10-15
- debugging WORKSTN programs 13-50
- decimal comma, definition of 2-4
- decimal data format (see packed decimal format, zoned decimal format)
- decimal period, definition of 2-4
- decimal positions
  - calculation specifications 8-22
  - extension specifications 4-6
  - generated total fields 15-16
  - input specifications 7-30
  - with move remainder (MVR) operation 10-45
  - with square root (SQRT) operation 10-64
- defining a field in calculations (result field) 8-20
- defining an alternate collating sequence 2-4
- definition of terms G-1
- DEL
  - (see also deleting record from a file)
  - description 9-4
- delete-capable file
  - adding records to 3-52, 3-56
  - definition 6-4
  - direct file load 3-21
- deleting records from a file
  - DEL 9-4
  - RECNO keyword 3-49
- demand file
  - (see also READ operation code)
  - description of 3-5
  - number allowed per program 3-5
- description 15-6
- detail calculation time 1-6
- detail lines (see *AUTO output specifications)
- detail output record 9-2
- detail output time 1-6
- detail printing, auto report (see *AUTO output specifications)
- detail time 1-6
- detailed object program logic 17-1
- device
  - assignment chart 3-31
  - file description entry 3-29
  - special support 3-33
  - with WORKSTN file, (display stations & SSP-ICF sessions) 13-1
- device names (see BSCA, CONSOLE, CRT, DISK KEYBORD, PRINTER, SPECIAL, WORKSTN)
- digit
  - (see also character, record identification code)
  - grouping by zone and digit 7-25
- direct file
  - adding records (delete-capable) 3-52
  - adding records (differences from sequential and indexed) 3-21
  - creating (loading) delete-capable 3-21
  - creating (loading) nondelete-capable 3-20
  - description 3-20
  - processing methods 3-9, 3-60
  - summary chart 3-60
  - synonym records 3-21
- DISK device 3-33
- disk file
  - (see also direct file, indexed file, sequential file)
  - block length 3-7
  - device entry 3-29
  - organization (see file organization)
  - processing methods 3-9, 3-58
  - record length 3-7
  - sequence checking 3-6
- display format (see display screen format)
- display screen format
  - for CONSOLE file 12-2
  - for WORKSTN file 13-9
    - input field 13-11
    - name on output specifications 9-28, 13-21
    - number of 2-14, 13-14
    - output field 13-11
    - output/input field 13-11
    - steps in creating 13-12
  - load member name
    - defining with FMTS continuation option 3-48, 13-16
    - for CONSOLE file 2-15, 12-2
    - for WORKSTN file 2-15
  - name on output specifications 9-28, 13-21
  - number of 2-14, 13-14
  - specifications 13-10
    - example 13-32
    - summary A-33
  - with KEY operation 10-25

display screen format specifications 13-10
display screen layout sheet 13-9
display station
  attaching to a program 13-4
  definition G-4
display station local data area
  definition G-4
  description 7-6
  how to define 7-6
  input specification entries 7-6
  SUBR21 to input and output with multiple display stations 13-34
  use by RPG II procedures 18-1, 18-8
DIV (divide) operation code
  (see also MVR operation code)
  bytes of generated code 16-9
  description 10-16
dollar sign in edit words 9-32
DS
  (see also data structure)
  input specifications entry 7-7
dual input/output area 3-22
duplicate character value, processing of 13-31
duplicate field names on /COPY modifier statement 15-28
duplicate records (see synonym records)
dynamic array (see execution-time array)

edit codes 9-21
  *AUTO output specifications
    A in column 39 15-17
    blank or B in column 39 15-14
    generated K edit code 15-14, 15-17
    1-9 or R in column 39 15-21
  *AUTO page headings specifications 15-11
  date field 9-25
    control spec 2-3
    control specifications 9-24
    relation to column 21 of
  effect of inverted print 2-4, 9-25
  effect on end position 9-25
  examples 9-23
  zero suppression 9-21
edit words 9-28
  considerations 9-29
  examples 9-33
  formatting 9-30
  replaceable characters 9-29
  rules for 9-28
  with edit code 9-28
edit, date 2-3
editing nonprinter files 9-22

enable/disable
  command keys 13-32
  function control keys 13-32
end of file
  file description specifications entry 3-6
  for CONSOLE file 3-6
  for WORKSTN file 13-7
  records processed 3-6
end position
  blank 9-22
  for *AUTO output specifications 15-15, 15-30
  in output record 9-22
ENDSR (end subroutine) operation code
  description 10-16
  factor 2 entries for INFSR 13-29
entry (see arrays, tables)
entry function keys, definition of G-4
ERASE function 8-20
erasing CONSOLE file buffer 8-20, 12-7
error handling (see exception/error handling for WORKSTN)
error indicators 13-23
  (see also ACQ, REL, READ, NEXT operation codes)
error return codes 13-26
error, sequence
  record type 7-2
  recovery from file sequence error 3-6
exception/error condition codes 13-26
exception/error handling for WORKSTN
  description 13-21
  error indicators 13-23
  INFDS data structure 13-25
  INFSR subroutine 13-29
exception/error indicators (see error indicators)
exception/error processing subroutine (see INFSR subroutine)
exception output records 9-2
  (see also EXCPT operation code)
EXCPT (exception) operation code 10-16
execute subroutine (EXSR) operation code 10-19
execute, storage size to 2-2
execution
  halts 18-11
  object program 18-11
  storage size 2-2
execution-time array
  (see also array)
  description 14-1
  rules for loading 14-4
  specifications 4-10
EXIT (exit to an external subroutine)
  (see also RLABL operation code)
  operation code 10-18

EXSR (execute subroutine) operation code
  bytes of generated code 16-10
  description 10-19
extension code (file description specifications entry) 3-29
extension specifications
  description 4-1
  general description 1-4
  possible file entries 4-10
  summary A-9
external indicators (U1-U8)
  as output indicators 9-13
  general description 1-12
  set by SWITCH OCL statement 3-57
  setting and restoring (see SUBR20)
  setting on and off 10-62
  used to condition files 3-57
  used to control calculations 8-11

factor 1
  as search word 10-30
  entries for 8-12
  in arithmetic operations 10-4
  in compare operations 10-13
factor 2
  entries for 8-12
  for ENDSR operation 10-16
  in arithmetic operations 10-4
  in compare operations 10-13
fetch overflow 9-5
  (see also overflow)
field
  alphameric 7-30, 9-13
  binary 4-6, 7-29, 9-26
  blanking 9-22
  control 1-6, 7-34
  key 3-17, 3-29
  length 7-30, 10-4
  look-ahead 7-10
  match 7-36, 11-1
  numeric 7-30, 9-13
  packed 7-27
  record address 3-17, 7-8
  result 8-20
  zeroing 9-22, 10-69
field description entries
  (see also *AUTO output specifications)
  input specifications 7-26
  output specifications 9-8
field indicators (01-99)
  explanation of 7-43
  general description 1-9
  halt indicators as 7-45
  how to assign 7-45
  numeric 7-45
  output specifications 9-8
field indicators (01-99) (continued)
  used to control calculations 8-6
field length
  alphameric 7-30
  arithmetic operations 10-4
  calculation specifications 8-21
  compare operations 10-13
  input specifications 7-30
  key 3-17
  numeric 7-30
  output specifications 9-22
  record address 3-17
  relation to decimal positions 8-22
field location entry (input specifications) 7-30
field name
  as result field 8-20
  definition G-5
  generated by auto report 15-16
  in an OR relationship 7-32
  on input specifications 7-30
  on output specifications 9-13
  referenced in RLABL operation 10-48
  rules for 7-32
  special words 9-13
    (see also *PLACE, PAGE, PAGE1-PAGE7, UDATE, UDAY, UMONTH, UYEAR)
field record relation
  indicators used 7-39
  rules for use of 7-39
figurative constants
  description 8-13
  with arrays 14-9
file
  (see also end of file, file designation, file organization, file processing, multifile processing)
  addrout 3-17, 3-28
  array (see array)
  chained 3-5
  combined 3-4
  CONSOLE 3-30
  demand 3-5
  direct 3-20
  indexed (see indexed file)
  input 3-2
  number allowed on file description specifications 3-1
  output 3-2
  primary 3-4
  record address 3-5
  secondary 3-4
  sequential (see sequential file)
  sharing considerations 3-17
  table (see table)
  update 3-2
  WORKSTN 13-1
file addition
  (see also adding record)
  for direct file 3-20
  for indexed file 3-51

file addition (continued)
for sequential file 3-52
file and record-type identification entries
input specifications 7-2
output specifications 9-2
file condition 3-57
(see also external indicators)
file description specifications
for BSCA file 6-6
for CONSOLE file 12-5
general description 3-1
maximum number of files allowed 3-1
summary A-4
summary charts for device entries 3-58
using the copy function 15-26
file designation 3-4
(see also chained file, demand file, primary file, record address file, secondary file, table file)
file format 3-6
file information data structure (see INFDS Data structure)
file organization (see indexed files, direct files, sequential files)
file processing (see addrout file processing, consecutive processing, random by relative record number or key processing, sequential by key processing, sequential within limits processing)
file sharing 2-10
(see also inquiry function)
file translation
description 2-11
placement of table in source program 1-3
table records 2-13
file type 3-2
(see also combined file, input file, output file, update file)
filename
definition G-5
file description specifications 3-2
from 4-2
input specifications 7-2
line counter specifications 5-2
output specifications 9-2
telecommunications specifications 6-3
to 4-2
first page (1P) forms position 2-11
first page (1P) indicator
as output indicator 9-13
general description 1-11
restrictions 1-11
fixed currency symbol 9-29, 9-32
fixed length format 3-6
floating currency symbol
auto report 15-21
examples 9-32
with edit code 9-22, 9-28
flowchart
detailed program logic 17-4
general program logic 1-8
flowchart (continued)
WORKSTN logic 13-5, 13-7
FMTS continuation line option 3-48, 13-16
FORCE operation code
bytes of generated code 16-10
description 10-19
form length
(see also line counter specifications)
default value 5-1
description 10-46
how to specify 5-2
form type (column 6) 1-13
format name (see display screen format)
format, date 2-3
format, display screen
(see also display screen format)
for CONSOLE file 12-2
for WORKSTN file 13-9
with KEY operation 10-25
format, file 3-6
formats, number of 2-14
formatting edit words 9-30
formatting report (auto report) 15-30
forms position, 1P 2-11
from filename (extension specifications) 4-2
full table or array 14-1
function control keys
as exception 13-21
enabling/disabling 13-32

general object program logic 1-8
general storage saving techniques 16-4
generated calculation code 16-9
generated RPG II program, auto report
calculations 15-5, 15-34
comments 15-5
date 15-10
edit codes 15-14, 15-17
examples 15-44
field names 15-16
format of generated specifications 15-1
group printing 15-21
order of specifications 15-5
output specifications 15-5, 15-34
page number 15-10
reformatting *AUTO page headings 15-32
source of specifications 15-22
subroutine (A$$SUM) 15-34
total fields 15-16
generated specifications, auto report 15-1, 15-34
generated total fields, auto report 15-16
generation of object program (see compilation)

- glossary (definition of terms) G-1
- GOTO (branch to) operation code
  - (see also TAG operation code)
  - bytes of generated code 16-10
  - description 10-22
- group indication
  - definition G-5
  - description 15-52
- group operations (see total operations)
- group printing, auto report 15-21
- grouping characters by zone and digit 7-25

- H-*AUTO record description 15-10
  - (see also *AUTO page headings specifications)
- half adjust 8-22
- halt indicators (H1-H9)
  - as output indicators 9-13
  - as record identifying indicator 7-7
  - assigning on input specifications 7-7
  - field indicator 7-43
  - field record relation 7-39
  - general description 1-9
  - to control calculations 8-6
- halt recovery procedures 18-11
- halts, auto report 18-10
- header specifications (see control specifications)
- heading (H) output records 9-2
- headings (see *AUTO page heading specifications, column headings)
- help key (see function control keys)
- hexadecimal equivalents of characters 2-6

- I-type program 2-9
- IBM-supplied procedures
  - AUTO 18-9
  - RPG 18-3
  - RPGR 18-7
  - RPGX 18-6
- IBM-written subroutines (see SUBR01, SUBR08, SUBR20, SUBR21, SUBR22, SUBR23, SUBR25, SUBR95)
- ID continuation line option 13-16
- identification
  - of programs 2-15
  - of record types 7-22
- ideographic support 20-1
- immediate cancel (halt recovery option) 18-11
- IND continuation line option 13-15
- index key 3-20
- index, array 14-7
- index, storage 3-49
- indexed file
  - addition of records 3-51
  - addrout processing 3-17
  - consecutive processing 3-9
  - file addition functions 3-56
  - file description summary chart 3-58
  - general information 3-20
  - key field 3-29
  - key length 3-17
  - loading 3-17
  - random processing 3-11
  - sequential by key processing 3-9
  - sequential within limits processing 3-10
  - storage index 3-49
  - unordered load 3-56
- indication of total lines, auto report 15-18
- indicators
  - calculation specifications
    - AND/OR relationship 8-6
    - conditioning (columns 9-17) 8-6
    - control level (columns 7-8) 8-2
    - resulting 8-23
  - command key (KA-KN, KP-KY)
    - calculation specifications 8-6
    - general description 1-12
    - output specifications 9-8, 9-11
    - setting 1-12
  - conditioning
    - calculations 8-6
    - general description 1-9
    - output 9-8
  - control level (L1-L9, L0) indicators
    - assigning to input fields 7-34
    - calculation specifications entries 8-2, 8-6
    - field record relation 7-39
    - general description 1-11
    - input specifications entries 7-7, 7-34, 7-39
    - L0 8-2
    - output specifications entries 9-8
    - record identifying indicator 7-7
    - with auto report 15-16, 15-22, 15-28
    - with subroutines 10-6
  - external (U1-U8)
    - conditioning files 3-57
    - controlling calculations 8-11
    - field record relation 7-39
    - general description 1-12
    - output specifications 9-13
    - set by SWITCH OCL statement 3-57
    - setting and restoring (see SUBR20)
  - field (01-99)

- indicators (continued)
  - field (0I-99) (continued)
    - assigning on input specifications 7-45
    - calculation specifications 8-6
    - general description 1-9
    - output specifications 9-8
  - field record relation 7-39
  - file conditioning 3-57
  - first page (1P)
    - as output indicator 9-13
    - general description 1-11
    - restrictions 1-11
  - general description 1-9
  - halt (H1-H9)
    - assigning on input specifications 7-7
    - calculation specifications 8-11
    - field indicator 7-43
    - field record relation 7-39
    - general description 1-9
    - output specifications 9-13
  - input specifications
    - control level 7-34
    - field 7-43
    - field record relation 7-39
    - record identifying 7-7
  - last record (LR)
    - calculation specifications 8-2, 8-11
    - general description 1-11
    - input specifications 7-7
    - output specifications 9-8
    - record identifying indicator 7-7
  - level zero (L0)
    - calculation specifications 8-2
    - output specifications 9-8
    - setting indicator restriction 10-5
  - matching record (MR)
    - (see also multifile processing)
    - assigning match fields 7-36, 11-1
    - calculation specifications 8-11
    - field record relation 7-39
    - general description 1-11
    - output specifications 9-8
    - when turned on 11-1
  - negative 9-8
  - output
    - (see also *AUTO output specifications)
    - AND/OR lines 9-2, 9-8
    - columns 23-31 9-8
    - command key 9-11
    - first page 9-13
    - halt 9-13
    - overflow 9-11
    - restriction in use of negative indicators 9-8
    - restriction in use of N1P 15-13
  - overflow (OA-OG, OV)
    - assigning on file description specifications 3-28

- indicators (continued)
  - overflow (OA-OG, OV) (continued)
    - calculation specifications 8-11
    - conditioning auto report page headings 15-10
    - external 9-13
    - fetch overflow routine 9-5
    - general description 1-11
    - output specifications 9-11
    - relation to program cycle 9-14
    - restriction with exception lines 9-11
    - setting of 9-14
  - permanent error 6-4
  - record available 6-5
  - record identifying (01-99)
    - assigning on input specifications 7-7
    - calculation specifications 8-6
    - field record relation 7-39
    - general description 1-9
  - referenced in RLABL operation 10-48
  - resulting (01-99)
    - calculation specifications 8-23
    - general description 1-9
    - use with ACQ operation code 10-7
    - use with LOKUP operation code 10-30
    - use with NEXT operation code 10-46
    - use with READ operation code 10-47
    - use with REL operation code 10-47
    - use with TESTB operation code 10-67
    - valid entries 8-23
  - setting (SETON, SETOF) 8-25, 10-5
  - summary chart A-29
  - valid A-28
- INFDS
  - continuation line option 13-16
  - purpose of 13-21
  - specifications for 13-25
- INFSR
  - continuation line option 13-16
  - purpose of 13-21
  - specifications for 13-29
- inline inquiry subroutine (SUBR95) 2-10
- input field modifier statement (auto report) 15-28
- input file
  - description 3-2
  - for KEYBORD device 3-2
  - translation 2-11
- input job queue 1-1, 18-2
- input/output (I/O) area
  - additional 3-22
  - input/output subroutine (linkage for SPECIAL device) 3-34
  - shared 2-14
- input/output, programmed control of 10-6
- input records
  - number allowed in program 7-6
  - option 7-6

input specifications 7-1
  general description 1-5
  summary A-16
  using the copy function 15-28
inputting and outputting the display station local data area (see SUBR21)
inquiry function 2-9
  (see also file sharing, SUBR95)
inserting new records (see adding records to a file)
Interactive Communications Feature 13-1
intermediate block checking (ITB) 6-4
inverted print 2-4
  (see also edit codes)
ITB (intermediate block checking) 6-4

K (1, 024 bytes of main storage) 2-2
K edit code (generated by auto report) 15-14, 15-17
KA-KN, KP-KY indicators (see command key indicators)
key
  (see also indexed files)
  random processing by 3-11
  sequential processing by 3-9
key field
  length 3-17
  restriction for binary 3-29
  starting location 3-29
KEY operation code
  bytes of generated code 16-10
  description 10-25
  with SET operation 10-30
key sorting 3-17
KEYBORD file
  as a demand file 3-5
  as a primary file 3-4
  block length 3-7
  description 3-33
  description of input data 3-33
  file description specifications summary 3-61
  format of display 3-8
  record length 3-7
  with KEY operation 10-25
keywords
  *CANCL 13-29
  *DETC 13-29
  *GETIN 13-29
  *INP 13-27
  *MODE 13-27
  *OPCODE 13-27
  *OUT 13-27
  *RECORD 13-27

keywords (continued)
  *STATUS 13-26
  FMTS 13-16
  ID 13-16
  IND 13-15
  INFDS 13-16
  INFSR 13-16
  NUM 13-15
  RECNO 3-49
  SAVDS 13-15
  SLN 13-16

label exit, name of 3-46
last record (LR) indicator
  calculation specifications
    columns 7-8 8-2
    columns 9-17 8-6
  general description 1-11
  input specifications 7-7
  output specifications 9-8
  record identifying indicator 7-7
length of
  array entry 4-6
  array name 14-7
  block 3-7
    relation to record length 3-8
  data structure 7-22
  field
    alphameric 7-30
    arithmetic operations 10-4
    compare operations 10-13
    numeric 7-30
    relation to decimal positions 8-22
  form 5-1, 5-2
  key field 3-17
  record 3-7
  record address field 3-17
  result field 8-20
  table entry 4-6
level zero (L0) indicator
  calculation specifications 8-2
  general description 1-11
  output specifications 9-8
  restriction on setting 10-5
level, control (see control level)
levels of totals 15-16
library specification
  for auto report generated source program 15-7
  for object program 2-2
limits processing 3-10
  (see also sequential within limits processing)
limits record 3-10
line (columns 3-5) 1-13

line counter specifications
  description 5-1
  general description 1-4
  summary A-11
line number
  on specifications sheets 1-13
  overflow 5-2
linkage (input/output) for SPECIAL device 3-34
linkage editor messages B-65
linkage to assembler subroutines 10-18
  (see also EXIT operation code, RLABL operation code)
linking to external subroutines 10-6
  (see also EXIT operation code, RLABL operation code)
list options
  for auto report 15-8
  for RPG II object program 2-2
listing, cross-reference (see cross-reference listing)
literal, alphameric 8-12
literal, numeric 8-13
loading
  arrays 14-4
  direct file 3-20
  indexed files 3-20
  tables 14-4
  unordered 3-56
local data area (see display station local data area)
location of field (input specifications) 7-30
logic of RPG II program
  detailed 17-1
  general 1-5
LOKUP (lookup) operation code
  bytes of generated code 16-10
  description 10-30
look-ahead field
  description 7-10
  input specifications entry 7-7
lookup operation (see LOKUP operation code)
LR (last record) indicator
  calculation specifications
    columns 7-8 8-2
    columns 9-17 8-6
  general description 1-11
  input specifications 7-7
  output specifications 9-8
  record identifying indicator 7-7
LR total line 15-16, 15-20
L0 indicator
  calculation specifications 8-2
  general description 1-11
  output specifications 9-8
  setting indicator restriction 10-5
L1-L9 (control level) indicators
  assigning to input fields 7-34
L1-L9 (control level) indicators (continued)
  calculation specifications
    entries 8-2, 8-6
  field record relation 7-39
  general description 1-11
  input specifications
    entries 7-6, 7-34, 7-39
  LO 8-2
L1-L9 total line 15-16, 15-20

machine instructions 1-1
main overlay 16-1
main storage
  size to compile 2-2
  size to execute 2-2
  usage map 16-3
match fields
  (see also multifile processing)
  alternate collating sequence 2-4
  assigning values (M1-M9) to 11-1, 11-2
  description 7-36
  example 11-4
  in multifile processing 11-1
  used for sequence checking 7-39
match levels (M1-M9) 7-36
matching record (MR) indicator
  (see also multifile processing)
  assigning match fields 7-36, 11-1
  calculation specifications 8-6
  field record relation 7-39
  general description 1-11
  output specifications 9-8
  when turned on 11-1
matching record logic 11-2
matching records processing 11-3
message identification code (MIC)
  description 10-57
  with a WORKSTN file 13-49
message member, user 10-58
message retrieve subroutine (SUBR23) 10-53
method (mode) of processing 3-8
  (see also processing, mode of)
method of operation, auto report 15-4
MHHZO (move high to high zone) operation code 10-38
MHLZO (move high to low zone) operation code 10-38
MIC (see message identification code)
MLHZO (move low to high zone) operation code 10-38
MLLZO (move low to low zone) operation code 10-38
mode of processing (see addrout file processing, consecutive processing, random by relative record number or key processing, sequential by key processing, sequential within limits processing)

modifier statements (/COPY function)
  file description specifications 15-26
  input field specifications 15-28
modifying contents of tables and arrays 14-9
modifying copied specifications (/COPY function) 15-26
modifying file description specifications (/COPY function) 15-26
modifying input field specifications (/COPY function) 15-28
move array (MOVEA) operation code 10-40
move operation codes
  bytes of generated code 16-12
  MOVE 10-38, 10-38
  MOVEL (move left) 10-43, 10-43
move remainder (MVR) operation code
  description 10-45
move zone operation codes
  MHHZO (move high to high zone) 10-38
  MHLZO (move high to low zone) 10-38
  MLHZO (move low to high zone) 10-38
  MLLZO (move low to low zone) 10-38
MOVEA (move array) operation code 10-40
MOVEL (move left) operation code 10-43, 10-43
MRT program (see multiple requestor terminal program)
MULT (multiply) operation code
  bytes of generated code 16-9
  description 10-45
multifile logic 11-2
multifile processing
  assigning match field values 7-36, 11-1
  end of file entry 3-6
  FORCE operation code 10-19
  logic 11-2
  match fields 11-1
  no match fields 11-1
  normal selection, three files 11-6
multiple input/output areas 3-22
multiple requestor terminal (MRT) program 13-3
MVR (move remainder) operation code
  description 10-45
M1-M9 (match field values) 7-36

N (not)
  calculation specifications 8-6
  output specifications 9-8
    restriction 9-8
  record identification code 7-22
name
  array 4-3, 14-7
  field
    calculation specifications 8-12, 8-20
    input specifications 7-30
    output specifications 9-13
  program 2-15
  result field 8-20
  table 4-3
name of label exit 3-46
negative balance (CR) 9-30
negative indicators 9-8
negative numbers 7-26
negative square root 10-64
NEXT (next) operation code
  bytes of generated code 16-10
  description 10-46, 13-19
  exception/error indicator 10-46, 13-19
nonprint characters 2-14
not (N)
  calculation specifications 8-6
  output specifications 9-8
    restriction 9-8
  record identification code 7-23
null response 10-54
NUM continuation line option 13-15
number of entries per record (table or array) 4-3
number of entries per table or array 4-5
number of formats 2-14
number of lines per page 5-2
number, negative 7-26
numbering report pages (PAGE special words)
  input specifications 7-32
  output specifications 9-13
  restarting numbering sequence 9-17
numeric (01-99) indicators (see conditioning indicator, field indicator, field record relation, record identifying indicator, resulting indicator)
numeric fields
  auto report
    centering column headings 15-32
    editing 15-17
  conversion 7-27
  moving 10-4
  sign of 7-26
  testing 7-43
numeric literal
  calculation specifications 8-13
  inverted print option 2-4

OA-OG, OV (overflow) indicators
  assigning on file description specifications 3-28
  calculation specifications 8-11
  conditioning auto report page headings 15-10

- OA-OG, OV (overflow) indicators (continued)
  - fetch overflow routine 9-5
  - general description 1-11
  - output specifications 9-11
  - relation to program cycle 9-14
  - restriction with exception lines 9-11
  - setting of 9-14
- object program
  - execution 18-11
  - identification 2-15
  - listing options 2-2
  - logic
    - detailed 17-1
    - general 1-5
  - output 2-2
  - size to execute 2-2
- OCL (operation control language) 18-11
- OP field
  - for SUBR20 13-31
  - for SUBR21 13-34
- open/close, special 16-2
- operation codes
  - arithmetic
    - (see also half adjust)
    - ADD 10-7
    - DIV (divide) 10-16
    - MULT (multiply) 10-45
    - MVR (move remainder) 10-45
    - SQRT (square root) 10-64
    - SUB (subtract) 10-64
    - XFOOT (summing elements of an array) 10-69
    - Z-ADD (zero and add) 10-69
    - Z-SUB (zero and subtract) 10-69
  - bit
    - BITOF (set bit off) 10-8
    - BITON (set bit on) 10-8
    - TESTB (test bit) 10-67
  - branching
    - (see also linkage to external subroutines)
    - GOTO 10-22
    - TAG 10-64
  - bytes of generated code 16-9
  - compare and testing
    - COMP (compare) 10-13
    - TESTZ (test zone) 10-69
  - DEBUG 10-15
  - linkage to external subroutines
    - EXIT 10-18
    - RLABL (RPG II label) 10-48
  - lookup (LOKUP) 10-30
  - move
    - MOVE 10-38
    - MOVEA (move array) 10-40
    - MOVEL (move left) 10-43
  - move zone
    - MHHZO (move high to high zone) 10-38
    - MHLZO (move high to low zone) 10-38
- operation codes (continued)
  - move zone (continued)
    - MLHZO (move low to high zone) 10-38
    - MLLZO (move low to low zone) 10-38
  - programmed control of input and output
    - CHAIN 10-11
    - EXCPT (exception) 10-16
    - FORCE 10-19
    - KEY 10-25
    - NEXT 10-46
    - READ 10-47
    - SET 10-54
    - SETLL 10-60
  - setting indicators
    - SETOF (set off) 10-62
    - SETON (set on) 10-62
  - shutdown (SHTDN) 10-62
  - subroutine
    - BEGSR (begin subroutine) 10-7
    - ENDSR (end subroutine) 10-16
    - EXSR (exit subroutine) 10-19
  - summary table A-26
  - time of day (TIME) 10-69
  - WORKSTN
    - ACQ (acquire) 10-7
    - REL (release) 10-47
- operation of auto report 15-4
- operations
  - (see also operation codes)
  - detail 1-6
  - total 1-6
- operator options for halt recovery 18-11
- option specifications
  - summary A-25
- option specifications, auto report
  - *suppress 15-7
  - date suppress 15-7
  - list option 15-8
  - source 15-7
  - source member reference 15-8
- option, date 2-3
- options, listing 2-2
- OR relationship
  - calculation specifications 8-6
  - input specifications 7-2, 7-24
  - output specifications 9-2, 9-8
- order of generated specifications (auto report) 15-5
- output
  - detail 1-6, 9-2
  - exception 9-2, 10-16
  - heading 9-2
  - table or array 14-10
  - total 1-6, 9-2
- output field
  - end position 9-22
  - name 9-13
  - repeating with *PLACE 9-17
- output file

output file (continued)
description 3-2
translation 2-11
output function (see *AUTO output specifications , *AUTO page headings specifications)
output indicators
(see also *AUTO output specifications, *AUTO page heading specifications)
AND/OR lines 9-2, 9-8
columns 23-31 9-8
command key 9-11
external 9-13
first page 9-13
halt 9-13
overflow 9-11
restriction in use of negative indicators 9-8
restriction in use of N1P 15-13
output records
detail 9-2
end position 9-22
exception 9-2
heading 9-2
total 9-2
output specifications
description 9-1
general description 1-5
generated by auto report 15-5
summary A-22
overflow
automatic 3-28
fetched 9-5
line 5-2
default 5-1
printing 9-14
spacing and skipping 9-11
with indicator specified 3-28
overflow indicator (OA-OG, OV)
assigning on file description specifications 3-28
calculation specifications 8-11
conditioning auto report page headings 15-10
fetch overflow routine 9-5
general description 1-11
output specifications 9-11
relation to program cycle 9-14
restriction with exception lines 9-11
setting of 9-14
overlapping data structure subfields 8-12
overlay
area 16-1
creating 16-1
how to find 16-5
process 16-1
reducing the size 16-5
override operation 13-30
overriding fields in a format 13-30
packed decimal format
(see also binary format, zoned decimal format)
description 7-27
length in bytes 7-29
packed field
(see also packed decimal format)
for table or array file 4-6
input specifications 7-27
output specifications 9-26
packed keys 3-19
page (columns 1-2) 1-12
PAGE fields, use by auto report 15-10
page headings
(see also *AUTO page headings specifications)
generated by auto report 15-30
reformatting *AUTO 15-32
page numbering (see PAGE, PAGE1-PAGE7)
PAGE, PAGE1-PAGE7
input specifications 7-32
output specifications 9-17
resetting to zero 9-18
performance improvement techniques 16-8
permanent error indicator 6-4
placement of headings and fields, auto report 15-30
polling characters for BSCA file 6-5
position (record identification code) 7-23
positioning printer forms 2-11
POST operation code
bytes of generated code 16-10
description 10-46
preexecution-time table or array
(see also array, table)
description 14-1
rules for loading 14-4
primary files 3-4
print key (see function control keys)
print, inverted 2-4
printer
block length 3-7
description of device 3-32
form length 5-1
record length 3-7
spacing and skipping 9-7
procedures
AUTO 18-9
RPG 18-3
RPGR 18-7
RPGX 18-6
processing methods
by addrout file 3-17
consecutive 3-9
for direct disk files 3-60
for indexed files 3-58
for sequential files 3-59
random by relative record number or key 3-11

processing methods (continued)
  sequential by key 3-9
  sequential within limits 3-10
processing, mode of (see processing methods)
program attributes (see multiple requestor terminal program, single requestor terminal program)
program cycle
  (see also program logic)
  detailed 17-1
  general description 1-5
  with WORKSTN file 13-6, 13-7
program halt, nonprint character 2-14
program identification
  for CONSOLE 2-15
  for WORKSTN 2-15
  from control specifications 1-14, 2-15
  of cross-reference listing 2-15
  used for display screen format name
program listing 2-2
program logic
  detailed 17-1
  general description 1-5
  with WORKSTN file 13-6, 13-7
programmed control of input and output 10-6
programming aids
  discussion 16-1
  for auto report 15-39
prompt format for CONSOLE file 12-5

random by relative record number or key processing 3-11
random processing
  by addrout file 3-17
  by CHAIN operation code 10-11
  by key 3-11
  by relative record number 3-11
RCODE field
  for SUBR20 13-31
  for SUBR21 13-32
READ (read) operation code
  bytes of generated code 16-10
  description 10-47
  exception/error indicator 10-47
  used with demand files 3-5
RECNO keyword
  deleting records 9-4
  description 3-49
  direct file load (delete-capable file) 3-21
record addition
  file description specifications entry 3-51
record addition (continued)
  indexed file 3-51
  output specifications entry 9-4
  sequential file 3-52
record address file
  (see also addrout file)
  description 3-5
  extension specifications entry 4-2
  format of records 3-5
  length of key field 3-17
  name of display screen format (CONSOLE file) 12-2
record address type 3-19
record available indicator 6-5
record description specifications
  *AUTO output specifications 15-12
  *AUTO page headings specifications 15-10
  input specifications 7-2
  output specifications 9-2
record identification code
  description 7-22
  for CONSOLE file 12-6
  input specifications entries 7-23
record identifying (01-99) indicators
  assigning on input specifications 7-7
  controlling calculations 8-6
  field record relation 7-39
  general description 1-9
record key 3-20
record length
  file description specifications entry 3-7
  maximum allowed per device 3-7
record type 7-7, 7-22
record, limits 3-10
records, alternate collating sequence table 2-9
records, creating table or array 14-3
records, file translation table 2-13
reducing the overlay size 16-5
region size used for compilation 2-2
REL (release) operation code
  bytes of generated code 16-10
  description 10-47, 13-19
  exception/error indicator 10-47, 13-19
related tables or arrays 14-1
relative record number
  description 3-11
  length 3-17
relative record number field (see RECNO)
release a display station (on output) 9-5
remainder (see MVR operation code)
repeating
  operations (see GOTO operation code, TAG operation code)
  output fields with *PLACE 9-17
  output lines with EXCPT 9-2

report
- formatting with auto report 15-30
- preparation 1-1
- return codes, WORKSTN file C-1

result field
- contents following KEY operation 10-25
- entries 8-20
- in arithmetic operations 10-4
- length 8-21
- testing the value of 8-23

resulting (01-99) indicators
- calculation specifications 8-23
- general description 1-9
- use with ACQ operation code 10-7
- use with CHAIN operation code 10-11
- use with LOKUP operation code 10-30
- use with NEXT operation code 10-46
- use with READ operation code 10-47
- use with REL operation code 10-47
- use with TESTB operation code 10-67
- valid entries 8-23

RLABL (RPG II label) operation code 10-48
- referencing a data structure 10-48
- referencing a field 10-48
- referencing a table or array 10-48
- referencing an indicator 10-48

roll down key (see function control keys)
roll up key (see function control keys)
rolling totals, auto report 15-15
root segment 16-1
rounding numbers in result field (half adjust) 8-22
RPG command 18-1
RPG II format generator
- automatic generation for CONSOLE file 12-1
- command statement 18-4

RPG II names, valid G-11
RPG II printed messages B-1
RPGR command 18-4
RPGX command 18-4

S specification, summary of A-33
sample programs
- auto report 19-21
- using a CONSOLE device 19-1
- using a WORKSTN file 13-32, 19-33

SAVDS continuation line option 13-15
screen design aid 13-9
screen format generator (see $SFGR utility)
SDA (see screen design aid)
search word for LOKUP operation 10-30
secondary files 3-4
- (see also multifile processing)

sequence
- assigning, numbers 7-2
- for record types 7-2

sequence checking
- alternate collating sequence 2-4
- file description specifications entry 3-6
- for fields in a record 7-39
- for records 3-6
- match field values 7-36

sequence, collating 2-6
sequential by key processing 3-9
sequential file
- addition of records 3-52
- description 3-21
- processing methods 3-59

sequential within limits processing 3-10
SET operation code
- bytes of generated code 16-10
- description 10-54
- specifying command keys 8-25
- with KEY 10-60

SETLL (set lower limits) operation code
- bytes of generated code 16-10
- description 10-60

SETOF (set off) operation code 8-25, 10-62
SETON (set on) operation code 8-25, 10-62
setting and restoring external indicators (see SUBR20)
setting indicators 10-5
shared file considerations 3-17
shared input/output area 2-14
short tables or arrays 14-1
SHTDN (shutdown) operation code
- bytes of generated code 16-11
- description 10-62

shutdown operation code 10-62
sign
- binary format 7-29
- numeric 9-13
- packed decimal format 7-27
- zoned decimal format 7-26

single requestor terminal (SRT) program
- changing to an MRT 13-1
- description 13-2

size to compile 2-2
size to execute 2-2
size, region 2-2
skip after 9-6
skip before 9-6
skipping
- for printer or CRT file 9-6
- with auto report 15-30

SLN continuation line option 13-16
SORTA
- bytes of generated code 16-10
- description 10-63

sorting keys 3-17
source member reference (auto report) 15-7

source program
  compiling 1-1, 18-1
  entering on disk 1-1
space after 9-6
space before 9-6
spacing
  for printer or CRT file 9-6
  special character, definition of G-10
  with auto report 15-30
special device support (see SPECIAL file)
SPECIAL file
  block length 3-7
  continuation line option 3-46
  description 3-33
  file description specifications 3-36
  linkage for assembler input/output subroutine 3-34
  record length 3-7
  system input subroutine (SUBR01) 3-45
special open/close 16-2
special words (see *PLACE, PAGE, PAGE1-PAGE7, UDATE, UDAY, UMONTH, UYEAR)
specifications
  calculation 8-1
  common entries 1-12
  control 2-1
  extension 4-1
  file description 3-1
    summary charts 3-58
  input 7-1
  line counter 5-1
  order of 1-3
  output 9-1
  overview 1-4
  summary A-1
split control field 7-36
SQRT (square root) operation code 10-64
SR (see subroutine lines)
SRT program (see single requestor terminal program)
SRyzzz (for SPECIAL device) 3-36, 3-46
SSP-ICF (see Interactive Communications Feature)
storage index 3-49
storage map 16-1
storage saving techniques
  general 16-4
  overlay process 16-1
storage size to compile 2-2
storage size to execute 2-2
SUB (subtract) operation code
  bytes of generated code 16-9
  description 10-64
subfields
  data structure 7-21
  overlapping 8-12
subroutine operation codes
  BEGSR (begin subroutine) 10-7
  ENDSR (end subroutine) 10-16
  EXSR (execute subroutine) 10-19
subroutines
  (see also EXIT operation code, RLABL operation code)
  (see also subroutine operation codes)
  A$$SUM (auto report) 15-34
  assembler language 3-34
  calculation specifications entry (SR) 8-2
  coding for RPG II 10-19
  KEY and SET operations 10-58
  linkage for user-written input/output 3-34
    generated DTF 3-35
  linkage to external 10-18
SUBRxx
  for external subroutine 10-18
  for SPECIAL file 3-36
SUBR01 (system input subroutine for SPECIAL device) 3-45
SUBR08 (1255 MICR subroutine) 3-45
SUBR20 (setting and restoring external indicators) 13-32
SUBR21 (inputting and outputting the display station local data area) 13-34
SUBR22 (reading a work station utility transaction file) 3-37
SUBR23 (message retrieve subroutine) 10-53
SUBR25 (1255 MICR subroutine) 3-45
SUBR95 (inline inquiry subroutine) 2-10
summary tables
  edit codes 9-23
  extension 4-10
  file description 3-58
  indicators A-29
  operation codes A-26
  specifications A-1
  valid indicators A-28
suppressing asterisks indication (auto report total line) 15-7
suppressing date and page (auto report) 15-7
suppression of zero 9-21
synonyms, definition of 3-21
sysin (see SUBR01)
system input subroutine (see SUBR01)

T-*AUTO (see *AUTO output specifications)
table
  (see also array, LOKUP operation code)
  *AUTO output field name 15-14, 15-21
    restriction with A in column 39 15-17
  *AUTO page headings 15-11
  addition of records 14-9
  alternating 14-3

table (continued)
auto report specifications
format of compile-time tables 15-1
order in generated program 15-1
binary format 4-6
building (see loading)
compile-time
definition of 14-1
rules for loading 14-4
creating input records 14-3
decimal positions 4-6
defining 14-3
definition of terms 14-1
element 14-1
entries
length of 4-6
number of 4-5
number per record 4-3
examples of using 14-12
extension specifications 4-1
summary chart 4-10
file description 3-5
file description entry 3-4
filename (when required on file description specifications) 3-2
full table 14-1
length of entry 4-6
loading
compile-time 14-4
placement in source program 14-4
preexecution-time 14-4
LOKUP operation 10-30
maximum allowed per program 4-1
modifying contents 14-9
name
as field name in *AUTO output 15-11, 15-14, 15-17
as field name in *AUTO page headings 15-21
as result field 8-20
extension specifications 4-2
file description specifications 3-2
how to form 4-3
number of entries per record 4-3
number of entries per table 4-5
order in source program 14-4
output 14-10
packed format 4-6
preexecution-time 14-1, 14-4
referenced in RLABL operation 10-48
related 14-1
rules for creating records 14-3
searching 14-7
sequence of specifications 14-3
short 4-6
TAG operation code 10-64
telecommunications specifications
description 6-1
identification of station 6-3

telecommunications specifications (continued)
placement in source program 1-3
transparency 6-3
type
of code 6-2
of control 6-2
of station 6-2
test bit operation code 10-67
test zone operation code 10-69
TESTB (test bit) operation code
bytes of generated code 16-9
description 10-67
testing fields (see field indicators)
testing result field 8-23
TESTZ (test zone) operation code
bytes of generated code 16-9
description 10-69
TIME operation code
bytes of generated code 16-10
description 10-69
TNAME field
for SUBR20 13-31
for SUBR21 13-31
to filename 4-2
total fields, auto report
asterisk indication 15-18
description 15-16
resetting to zero 15-18
total operations 1-6
total output records 9-2
total rolling, auto report 15-15
total time 1-6
translation table and alternate collating sequence coding sheet 2-4
translation, file (see file translation)
transparent literal 2-15

UDATE
date format 2-3
description 9-21
example of editing 9-25
UDAY 9-21
UMONTH 9-21
unordered load (indexed file) 3-56
update file
(see also updating files)
description 3-2
translation 2-11
updating files
CAUTION 3-3
restriction 3-3
user message member 10-58
UYEAR 9-21
U1-U8 (see external indicators)

- valid indicators A-28
- wait time for BSCA file 6-5

- work station utility transaction file, reading and updating a (SUBR22) 3-37
- WORKSTN file
  - $SFGR and RPG II input buffers 13-19
  - $SFGR and RPG II output buffers 13-22
  - ACQ operation code 10-7, 13-17
  - as a primary file 3-4
  - command key indicators 1-12, 8-6, 13-14
  - command keys, enabling/disabling 13-32
  - continuation line options 3-47, 13-15
  - creating display screen formats 13-9
  - data returned to the program 13-6
  - debugging hints 13-50
  - definition
    - of input field 13-11
    - of output field 13-11
    - of output/input field 13-11
  - description 3-33
  - duplicate character value, processing 13-31
  - end of file 13-8
  - exception/error processing subroutine (see INFSR subroutine)
    - file information data structure (see INFDS)
  - FMTS option 13-16
  - format name on output specifications 9-28
  - function control keys, enabling/disabling 13-32
  - ID option 13-16
  - IND option 13-15
  - INFDS data structure 13-25
  - INFDS option 13-16
  - INFSR option 13-16
  - INFSR subroutine 13-29
  - input processing 13-5
  - input record, first 13-17
  - NEXT operation code 10-46, 13-19
  - number of formats 2-14, 13-14
  - override operation 13-30
  - program attributes 13-1
  - program logic 13-5, 13-7
  - READ operation 10-47
  - record length 13-14
  - REL (release) operation 10-47, 13-19
  - releasing display station at output time 9-5
  - restrictions 13-11
  - return codes C-1

- WORKSTN file (continued)
  - SAVDS option 13-15
  - saving fields or indicators 13-49
  - SLN option 13-16
  - Z edit code 13-21

- XFOOT (summing elements of an array) operation code
  - bytes of generated code 16-10
  - description 10-69

- Y edit code
  - control specifications entries (columns 19-21) 2-3
  - specification 9-21
  - with date option entry 2-3

- Z (zone) 7-22
- Z-ADD (zero and add) operation code
  - bytes of generated code 16-9
  - description 10-69
- Z edit code (with WORKSTN file) 13-21
- Z-SUB (zero and subtract) operation code
  - bytes of generated code 16-9
  - description 10-69
- zero (blanking) fields
  - blank after
    - auto report 15-11
    - output specifications 9-22
  - subtract operation 10-64
- zero suppression 9-21
- zone
  - (see also move zone operations)
  - character grouping 7-23
- zoned decimal format 7-26
  - length in bytes 7-29

- 01-99 indicators (see conditioning indicators, field indicator, field record relation, record identifying indicators, resulting indicators)
- 1-9 or R in column 39 (auto report) 15-20
- 1P (first page) forms position 2-11

1P (first page) indicator
as output indicator 9-2
general description 1-11
restrictions 1-11

READER'S COMMENT FORM

Please use this form only to identify publication errors or request changes to publications. Technical questions about IBM systems, changes in IBM programming support, requests for additional publications, etc, should be directed to your IBM representative or to the IBM branch office nearest your location.

Error in publication (typographical, illustration, and so on). **No reply.**

Page Number *Error*

Inaccurate or misleading information in this publication. Please tell us about it by using this postage-paid form. We will correct or clarify the publication, or tell you why a change is not being made, provided you include your name and address.

Page Number *Comment*

IBM may use and distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever. You may, of course, continue to use the information you supply.

Name ______________________________

Address ______________________________

____

- No postage necessary if mailed in the U.S.A.

IBM System/34
RPG II
Reference Manual

SC21-7667-4

SC21-7667-4

Fold and tape

Please do not staple

Fold and tape

NO POSTAGE
NECESSARY IF
MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 40 ARMONK, N. Y.

POSTAGE WILL BE PAID BY . . .

IBM CORPORATION
General Systems Division
Development Laboratory
Publications, Dept. 532
Rochester, Minnesota 55901

Fold and tape

Please do not staple

Fold and tape

IBM

International Business Machines Corporation

General Systems Division
4111 Northside Parkway N.W.
P.O. Box 2150
Atlanta, Georgia 30301
(U.S.A. only)

General Business Group/International
44 South Broadway
White Plains, New York 10601
U.S.A.
(International)

Cut Along Line

IBM System/34 RPG II Reference Manual (File No. S34-28) Printed in U.S.A. SC21-7667-4

READER'S COMMENT FORM

IBM System/34
RPG II
Reference Manual

SC21-7667-4

Please use this form only to identify publication errors or request changes to publications. Technical questions about IBM systems, changes in IBM programming support, requests for additional publications, etc, should be directed to your IBM representative or to the IBM branch office nearest your location.

Error in publication (typographical, illustration, and so on). **No reply.**

Page Number *Error*

Inaccurate or misleading information in this publication. Please tell us about it by using this postage-paid form. We will correct or clarify the publication, or tell you why a change is not being made, provided you include your name and address.

Page Number *Comment*

IBM may use and distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever. You may, of course, continue to use the information you supply.

Name ________________________________

Address ______________________________

● No postage necessary if mailed in the U.S.A.

Cut Along Line

Fold and tape

Please do not staple

Fold and tape

NO POSTAGE
NECESSARY IF
MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 40 ARMONK, N. Y.

POSTAGE WILL BE PAID BY . . .

IBM CORPORATION
General Systems Division
Development Laboratory
Publications, Dept. 532
Rochester, Minnesota 55901

Fold and tape

Please do not staple

Fold and tape

IBM

International Business Machines Corporation

General Systems Division
4111 Northside Parkway N.W.
P.O. Box 2150
Atlanta, Georgia 30301
(U.S.A. only)

General Business Group/International
44 South Broadway
White Plains, New York 10601
U.S.A.
(International)

IBM®

International Business Machines Corporation

General Systems Division
4111 Northside Parkway N.W.
P.O. Box 2150
Atlanta, Georgia 30301
(U.S.A. only)

General Business Group/International
44 South Broadway
White Plains, New York 10601
U.S.A.
(International)

IBM System/34 RPG II Reference Manual (File No. S34-28) Printed in U.S.A. SC21-7667-4

SC21-7667-4