

**IBM System/3
Model 6
RPG II Reference Manual**

Program Number 5703-RG1

Program Product

Preface

This publication is intended as a reference manual for programmers writing RPG II specifications for the IBM System/3 Model 6.

The manual describes the program documentation you can use to help write, test, and maintain RPG II programs. It also contains information common to all RPG coding sheets, and describes the seven types of RPG specifications in the order required by the RPG II compiler.

Specifications for each coding sheet are described column by column as a programmer would write them. Information in every column description is presented in this order:

1. List of possible entries.
2. General discussion of the use of the column and considerations for all possible entries.
3. Specific discussion of each entry.
4. Charts and examples.

The section following the column-by-column descriptions of the RPG specifications contains supplementary information about subjects previously described. Two sample programs are described in *Appendix A*. *Appendix B* lists the RPG II error messages for System/3 Model 6.

Appendixes A through G cover the following topics: Sample programs, RPG II error messages for Model 6, a summary of RPG II specifications, programming tips for saving core and improving performance, bytes of generated code for calculations, IBM-written subroutines, and OCL and halts.

Related Publications

- *IBM System/3 Model 6 Introduction*, GA21-9122
- *Introduction to RPG II*, GC21-7514
- *IBM System/3 Disk Concepts and Planning Guide*, GC21-7571
- *IBM System/3 RPG II Disk File Processing Programmer's Guide*, GC21-7566
- *IBM System/3 RPG II Additional Topics Programmer's Guide*, GC21-7567

Fourth Edition (September, 1972)

This is a major revision of, and obsoletes, SC21-7517-2 and Technical Newsletter SN21-5159. Information on the continuation record has been expanded, and additional programming tips have been supplied. Changes are indicated by a vertical line to the left of the change. New or extensively revised illustrations are indicated by the symbol ● to the left of the title.

This edition applies to version 7, modification 0 of Program Product 5703-RG1 and to all subsequent versions and modifications until otherwise indicated in new editions or Technical Newsletters. Changes are continually made to the specifications herein; before using this publication in connection with the operation of the IBM Systems, consult the latest IBM System/3 Newsletter, Order Number GN20-2228, for the editions that are applicable and current.

Requests for copies of IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

A form for reader's comments is provided at the back of the publication. If the form has been removed, comments may be addressed to IBM Corporation, Publications, Department 245, Rochester, Minnesota 55901.

Contents

INTRODUCTION	1	Column 16 (File Designation)	31
Using RPG II	1	Primary Files	31
Definitions of Terms	4	Secondary Files	31
Machine Requirements	4	Chained Files	31
General RPG II Object Program Logic	4	Record Address Files	31
COMMON ENTRIES	9	Table or Array Files	32
Columns 1–2 (Page)	9	Demand Files	32
Columns 3–5 (Line)	9	Column 17 (End of File)	32
Example	10	Column 18 (Sequence)	33
Column 6 (Form Type)	11	Column 19 (File Format)	33
Column 7 (Comments)	11	Columns 20–23 (Block Length)	33
Columns 75–80 (Program Identification)	11	Block Length for Disk Records	33
Control Card Specifications	11	Columns 24–27 (Record Length)	35
All Other Source Program Specifications	11	Record Length for Disk Files	35
CONTROL CARD SPECIFICATIONS	13	Record Length for Keyboard Files	35
Columns 1–2 (Page)	13	Column 28 (Mode of Processing)	35
Columns 3–5 (Line)	13	Consecutive	36
Column 6 (Form Type)	14	Sequential by Key	36
Columns 7–9 (Core Size To Compile)	14	Sequential Within Limits	36
Column 10 (Object Output)	14	Random by Relative Record Number or Key	36
Column 11 (Listing Options)	14	Random by ADDRUT File	37
Columns 12–14 (Core Size to Execute)	14	Columns 29–30 (Length of Key Field or Record Address Field)	37
Columns 13–14	15	Column 31 (Record Address Type)	37
Column 12	15	Column 32 (File Organization or Additional I/O Area) File Organization	37
Example	15	Additional Input/Output Area	38
Column 15 (DEBUG)	16	ADDRUT Files	38
Columns 16–20	16	Columns 33–34 (Overflow Indicators)	42
Column 21 (Inverted Print)	16	Columns 35–38 (Key Field Starting Location)	42
Columns 22–25	16	Column 39 (Extension Code)	42
Column 26 (Alternate Collating Sequence) Alternate Collating Sequence	16	Columns 40–46 (Device)	42
Columns 27–36	17	Cathode Ray Tube	43
Column 37 (Inquiry)	21	Console Files	44
Columns 38–40	22	Printer Files	44
Column 41 (1P Forms Position)	22	SPECIAL Device Support	44
Column 42	22	File Description Specifications for SPECIAL Device	45
Column 43 (File Translation)	22	Columns 47–52	45
File Translation	23	Column 53 (Continuation Lines–K)	45
Column 44 (Punch MFCU Zeros)	26	Columns 54–59	45
Columns 45–47	27	Name of Label Exit	45
Column 48 (Shared I/O)	27	Continuation Line Option	45
Column 49 (Field Print)	27	Columns 60–65 (Core Index)	45
Columns 50–74	27	Column 66 (File Addition) Adding Records to a File (A)	47
Columns 75–80 (Program Identification)	27	Loading Records in an Unordered Sequence (U) Examples	48
FILE DESCRIPTION SPECIFICATIONS	29	Column 67	52
Columns 1–2 (Page)	29	Columns 68–69 (Number of Extents)	52
Columns 3–5 (Line)	29	Column 70	53
Column 6 (Form Type)	30	Columns 71–72 (File Condition)	53
Columns 7–14 (Filename)	30	Columns 73–74	53
Column 15 (File Type)	30	Columns 75–80 (Program Identification) File Description Charts	53
Input Files	30	Example	53
Output Files	30	EXTENSION SPECIFICATIONS	63
Update Files	30	Columns 1–2 (Page)	64
Combined Files	30	Columns 3–5 (Line)	64
Display Files	31	Column 6 (Form Type)	64
		Columns 7–10	64

Columns 11–18 (From Filename)	64	Columns 53–58 (Field Name)	93
Columns 19–26 (To Filename)	64	Field Names	93
Columns 27–32 (Table or Array Name)	65	Special Words (PAGE, PAGE1, PAGE2)	94
Table Name	65	Example	95
Array Name	65	Columns 59–60 (Control Level)	96
Example	66	Assigning Control Level Indicators	96
Columns 33–35 (Number of Entries Per Record)	67	Split Control Fields	97
Example	67	Examples	98
Columns 36–39 (Number of Entries Per Table or Array)	68	Columns 61–62 (Matching Fields)	102
Columns 40–42 (Length of Entry)	69	Matching Fields	102
Examples	69	Sequence Checking	105
Column 43 (Packed or Binary Field)	70	Examples	106
Column 44 (Decimal Positions)	70	Columns 63–64 (Field Record Relation)	107
Column 45 (Sequence)	70	Record Identifying Indicators (01–99)	107
Columns 46–57	71	Control Level (L1–L9) and Matching Record (MR)	
Columns 58–74 (Comments)	71	Indicators	108
Columns 75–80 (Program Identification)	71	External Indicators (U1–U8)	108
		Halt Indicator (H1–H9)	108
		Examples	108
LINE COUNTER SPECIFICATIONS	73	Columns 65–70 (Field Indicators)	112
Columns 1–2 (Page)	74	Assigning Indicators in Columns 65–70	114
Columns 3–5 (Line)	74	Numeric Indicators (01–99)	114
Column 6 (Form Type)	74	Halt Indicators (H1–H9)	114
Columns 7–14 (Filename)	74	Columns 71–74	114
Columns 15–17 (Line Number–Number of Lines Per Page)	74	Columns 75–80 (Program Identification)	114
Columns 18–19 (Form Length)	74		
Columns 20–22 (Line Number–Overflow Line)	74	CALCULATION SPECIFICATIONS	115
Columns 23–24 (Overflow Line)	74	Columns 1–2 (Page)	116
Columns 25–74	74	Columns 3–5 (Line)	116
Columns 75–80 (Program Identification)	74	Column 6 (Form Type)	116
		Columns 7–8 (Control Level)	116
INPUT SPECIFICATIONS	75	Control Level Indicators (L0, L1–L9)	116
Columns 1–2 (Page)	76	Last Record Indicator (LR)	117
Columns 3–5 (Line)	76	Subroutine Lines (SR)	117
Column 6 (Form Type)	76	AND/OR Lines (AN, OR)	117
Columns 7–14 (Filename)	76	Examples	117
Columns 14–16	76	Columns 9–17 (Indicators)	122
Columns 15–16 (Sequence)	76	AND/OR Lines (AN, OR)	122
Alphabetic Characters	76	AND Relationship	122
Numeric Characters (01–99)	76	Field Indicators (01–99)	122
Examples	77	Command Keys Indicators (KA–KN, KP, KQ)	122
Column 17 (Number)	80	Record Identifying Indicators (01–99)	122
Example	80	Resulting Indicators (01–99)	122
Column 18 (Option)	80	Control Level Indicators (L1–L9)	123
Example	80	Last Record Indicator (LR)	123
Columns 19–20 (Record Identifying Indicator, **, TR)	81	Matching Record Indicator (MR)	123
Record Identifying Indicators	81	Halt Indicator (H1–H9)	123
Look Ahead Fields	82	External Indicators (U1–U8)	123
Spread Cards	82	Overflow Indicators (OA–OG, OV)	123
Columns 21–41 (Record Identification Codes)	85	Relationship Between Columns 7–8 and Columns 9–17	123
Position	85	Examples	124
Not (N)	85	Columns 18–27 (Factor 1) and Columns 33–42 (Factor 2)	127
C/Z/D	86	Literals	128
Character	86	Columns 28–32 (Operation)	130
AND Relationship	87	Columns 31–32	132
OR Relationship	87	Columns 43–48 (Result Field)	132
Examples	88	SPACEn, SKIPnn, and EJECT	132
Column 42	89	Field Name, Table Name, Array Name, or Array Element	132
Column 43 (Packed or Binary Field)	89	Columns 49–51 (Field Length)	133
Unpacked Decimal Format (Blank)	89	Positioning the Print Element and Setting Tab Stop Positions	133
Packed Decimal Format (P)	90	Result Field Length	133
Binary Format (B)	91		
Columns 44–51 (Field Location)	93		
Column 52 (Decimal Positions)	93		

Column 52 (Decimal Positions)	133	MR (Matching Record Indicator)	182
Column 53 (Half Adjust)	135	OA–OG, OV (Overflow Indicators)	182
Column 54–59 (Resulting Indicators)	136	L1–L9 (Control Level Indicators)	182
Test Results	136	L0 Indicator	183
Allowing Command Keys to be Pressed (SET)	137	LR (Last Record Indicator)	183
Setting Indicators (SETON, SETOF)	137	U1–U8 (External Indicators)	183
Example	138	Ledger File Processing	184
Columns 60–74 (Comments)	138	Spacing on the Ledger Card	186
Columns 75–80 (Program Identification)	138	Printing on the Ledger Card	186
		Ejecting the Ledger Card	186
OUTPUT-FORMAT SPECIFICATIONS	139	Types of Ledger Files	186
Columns 1–2 (Page)	140	Examples	189
Columns 3–5 (Line)	140	Ledger Card Overflow	194
Column 6 (Form Type)	140	Look Ahead	196
Columns 7–14 (Filename)	140	Specifications	202
Columns 14–16	140	Example	202
Column 15 (Type)	140	Multifile Processing	204
Heading Records (H)	140	Example	204
Detail Records (D)	141	Operation Codes	210
Total Records (T)	141	Arithmetic Operations	212
Exception Records (E)	141	Move Operations	213
Columns 16–18 (Add a Record)	141	Move Zone Operations	218
Column 16 (Stacker Select/Fetch Overflow)	141	Compare and Testing Operations	218
Ledger Card Eject (E)	141	Bit Operations	221
Fetch Overflow (F)	142	Setting Indicators	228
Columns 17–22 (Spacing and Skipping)	142	Branching Within RPG II	228
Column 17 (Space Before)	143	Branching to External Subroutines	231
Column 18 (Space After)	143	Look-Up Operations (LOKUP)	232
Columns 19–20 (Skip Before)	143	Subroutine Operations	241
Columns 21–22 (Skip After)	143	Programmed Control of Input and Output	241
Spacing and Skipping for Printer Files	143	Debug Operation	263
Spacing for Ledger Files	144	Overflow Indicators	264
Spacing and Skipping for CRT Files	144	Automatic Page Formatting	264
Spacing for Console Files	144	Continuous Listings	265
Columns 23–31 (Output Indicators)	144	Using Overflow Indicators to Control Page Formatting	265
AND or OR Lines	146	Using Overflow Indicators in Control Level	
Command Key Indicators (KA–KN, KP, KQ)	146	Indicators	268
Overflow Indicators (OA–OG, OV)	146	Overflow Printing with EXCPT Operation Code	270
First Page Indicator (1P)	147	Assigning Overflow Indicators	270
Halt Indicators (H1–H9)	147	RPG II Object Program Logic (Detailed)	272
External Indicators (U1–U8)	147	Subroutines	276
Examples	147	Coding Subroutines	276
Columns 32–37 (Field Name)	153	Tables and Arrays	281
Field Names	153	Rules for Creating Table or Array Input Records	284
Special Words	153	Defining Tables and Arrays	284
Examples	156	Loading Tables and Arrays	285
Column 38 (Edit Codes)	158	Searching Tables and Arrays	290
Column 39 (Blank After)	161	Using Arrays	290
Columns 40–43 (End Position in Output Record)	162	Modifying Contents of Tables and Arrays	292
Repeating Output Fields (*PLACE)	162	Table and Array Output	293
Column 44 (Packed or Binary Field)	162	Example of Using Tables	294
Columns 45–70 (Constant or Edit Word)	163	Examples of Building and Using Arrays	297
Constant	163		
Edit Words	163	APPENDIX A. RPG II SAMPLE PROGRAMS	307
Columns 71–74	173	Sample Program 1 (SAMPL1)	307
Columns 75–80 (Program Identification)	173	Specifications	307
		Sample Program 2 (SAMPL2)	310
SUPPLEMENTARY INFORMATION	175	Specifications	310
Indicators	175	Example Programs	313
01–99 (Field Indicators, Record Identifying Indicators,		Example Program 1	313
Resulting Indicators, and Conditioning Indicators)	178	Example Program 2	316
Examples	178	Example Program 3	321
KA–KN, KP, KQ (Command Key Indicators)	179		
H1–H9 (Halt Indicators)	179	APPENDIX B. RPG II ERROR MESSAGES	325
1P (First Page Indicator)	180	Message Format	325

APPENDIX C. SUMMARY OF RPG II SPECIFICATIONS	395
Information Common to All Forms	395
Control Card Specifications	395
File Description Specifications	397
Extension Specifications	399
Line Counter Specifications	400
Telecommunications Specifications	400
Input Specifications	403
Calculation Specifications	404
Output-Format Specifications	406
APPENDIX D. PROGRAMMING TIPS	409
Core Saving Techniques	409
Overlay Process	409
Creating the Overlays	409
Special Open/Close	409
Saving Core	409
Performance Improvement Techniques	412
APPENDIX E. BYTES OF GENERATED CODES FOR CALCULATIONS	419
APPENDIX F. IBM-WRITTEN SUBROUTINES	425
In-Line Inquiry Subroutine (SUBR95)	425
APPENDIX G. RUNNING AN RPG II PROGRAM—HALTS AND OCL	427
RPG II Halt Procedures	427
Operation Control Language for RPG II	427
INDEX	429

USING RPG II

RPG II consists of a symbolic programming language and a compiler program. The general operations involved in preparing a report using RPG II are shown in Figure 1. The circled numbers in Figure 1 refer to the numbers in the following text:

1. The programmer analyzes the report requirements to determine the format of the input files and the layout of the finished report. For example, he determines what fields in the input records are to be used, what calculations are to take place, where the data is to be located in the output records, and how many and what kind of totals must be accumulated.
2. After the programmer has analyzed the requirements for the report, he provides the RPG II compiler with information about these requirements. This information is coded on RPG specification sheets. The five types of specification sheets are designed so that one specification line represents one record in the source program.
 - a. Control Card Specifications. The programmer provides special information about his program and describes his System/3 Model 6 to the RPG II compiler by making entries on this sheet.
 - b. File Description Specifications. The programmer describes all files used by the object program (input files, output files, table or array files, etc.) through entries on the File Description sheet.
 - c. Extension Specifications. If the programmer uses record address files, tables, or arrays in his object program, he furnishes information about them through entries on the Extension sheet.
 - d. Line Counter Specifications. If the programmer uses printer files in his object program, he can provide information about the number of lines to be printed on the forms that are used. This is done by making entries on the Line Counter sheet.
 - e. Input Specifications. The programmer describes input records (record layout, fields used, etc.) entered from all input devices except the keyboard by making entries on the Input sheet.
 - f. Calculation Specifications. The programmer states what processing is to be done (add, subtract, multiply, divide, etc.) by means of entries on a Calculation sheet. The Calculation sheet is also used to describe fields entered from the keyboard, and to control certain input and output functions related to those fields.
 - g. Output-Format Specifications. The programmer defines the layout of the desired report (print positions, headings, etc.) by making entries on the Output-Format sheet.
3. After the specifications have been written on the appropriate forms, the data on the forms is entered into the system. Each line of specifications is considered one input record in your source program. The proper arrangement of records in a source program is shown in Figure 2.
4. These source program specifications must begin with RPG II control card specifications. The source program specifications are entered into the system and processed by the RPG II compiler under control of the Disk System Management Program. At the end of the processing run (compilation), the object program is produced and stored on disk. This object program contains all the machine instructions required to prepare the desired report.
5. The programmer can now leave the object program on disk to be processed later, or he can proceed directly to processing the object program.
6. When processing is done, the input data files and the object program are read into the system. This final operation (object run) results in the desired report.

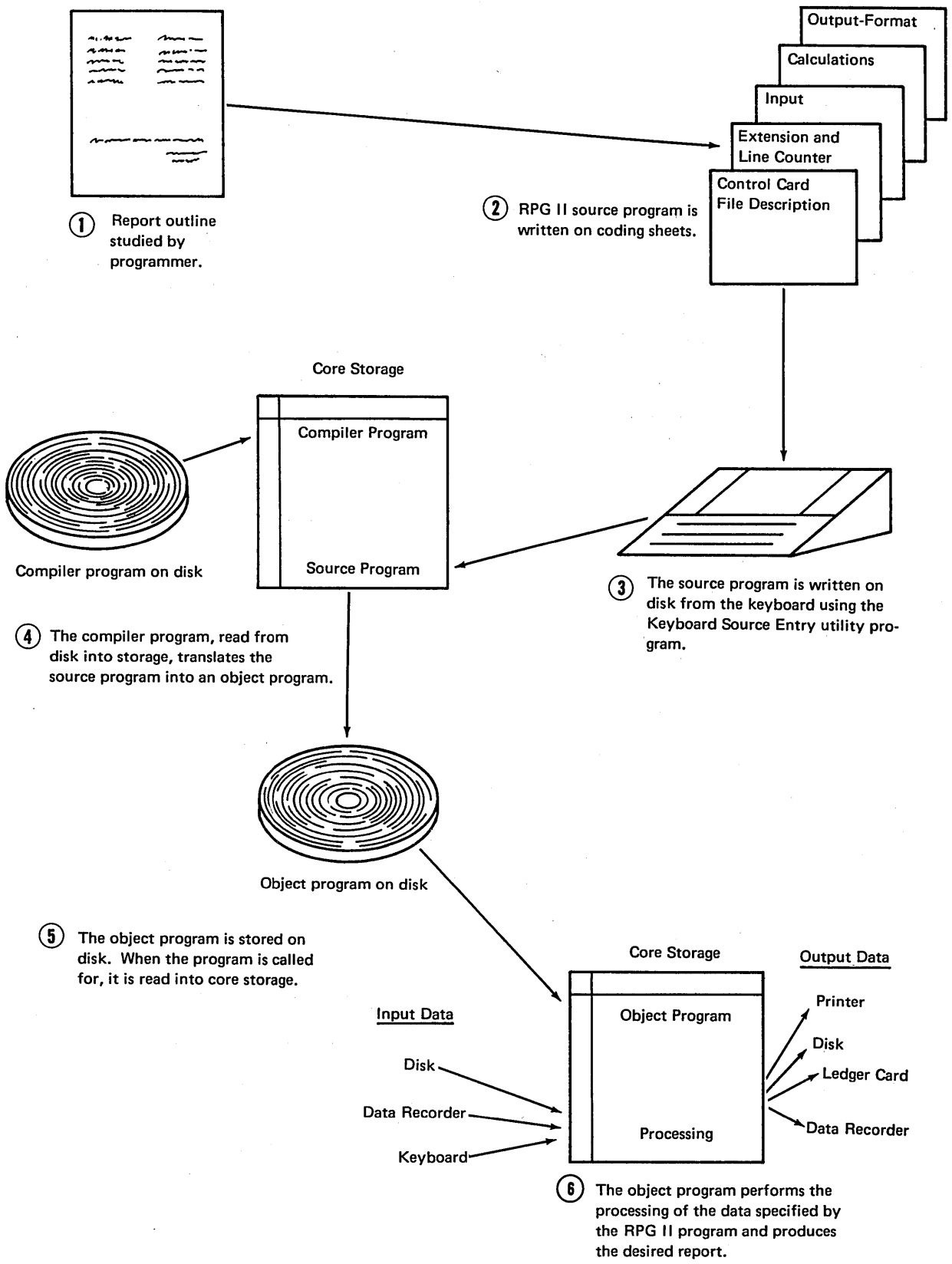
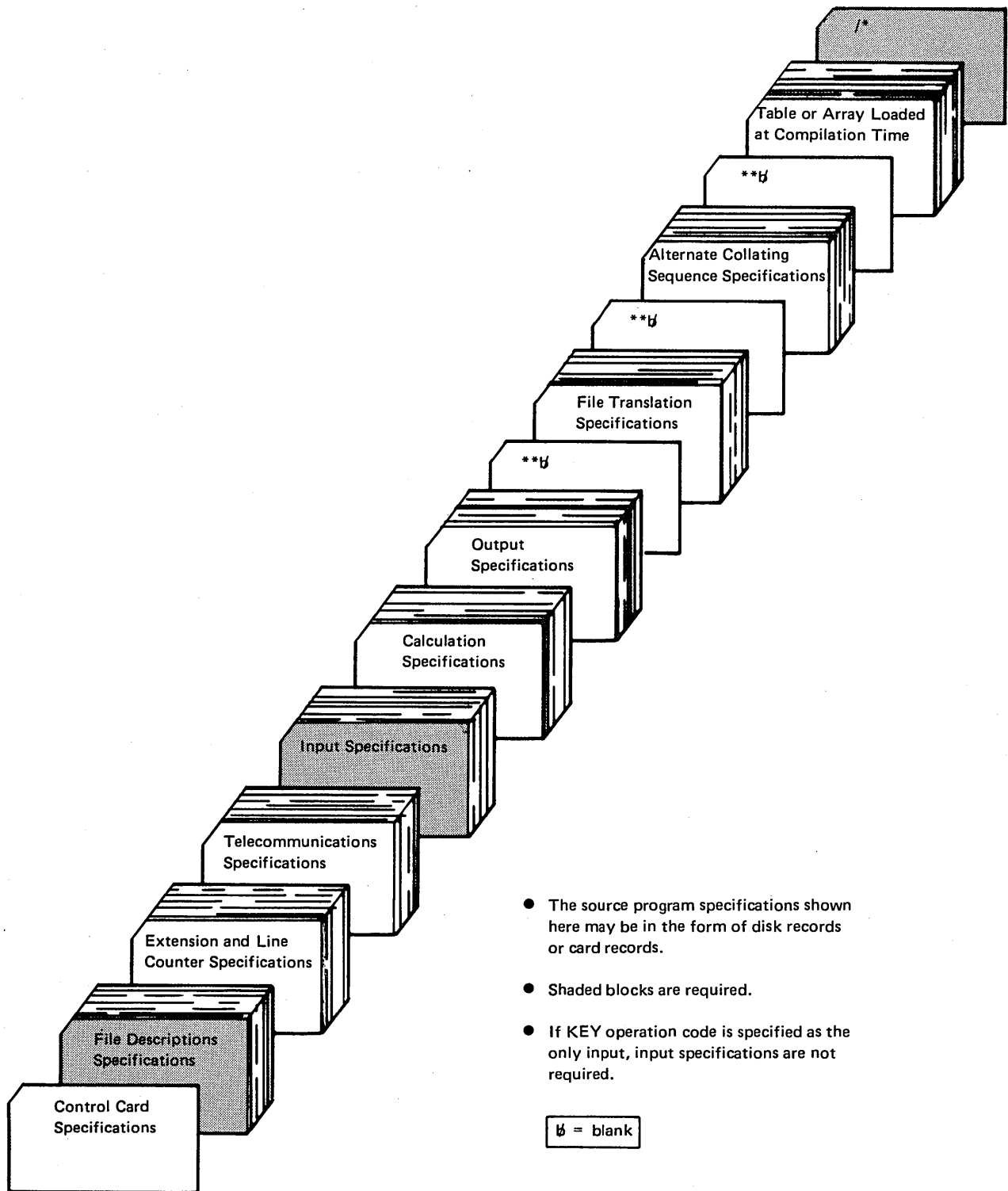


Figure 1. Preparation of a Report Using RPG II



• Figure 2. Arrangement of the RPG II Source Program

DEFINITIONS OF TERMS

EBCDIC (Extended Binary-Code-Decimal Interchange Code) Notation: The 256-character machine code used in the IBM System/3 Model 6.

Alphabetic Characters: The 26 alphabetic EBCDIC characters and the three EBCDIC characters, #, \$, and @.

Numeric Characters: The EBCDIC characters 0-9.

Special Characters: The 217 EBCDIC characters not defined as alphabetic or numeric.

Alphameric Characters: Any of the 256 EBCDIC characters.

Alphameric Fields: All fields for which a decimal-positions specification has not been made in the appropriate column of the specifications forms. Alphameric fields can contain alphabetic, numeric, or special characters.

Numeric Fields: All fields having a decimal positions specification in the appropriate columns of the specifications forms.

Valid RPG II Names: The following rules apply to names used in RPG II programs:

- RPG II filenames can be from 1-8 characters long; RPG II field names can be from 1-6 characters long.
- The first character of either a filename or a field name must be alphabetic (see preceding definition of alphabetic characters). 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.

MACHINE REQUIREMENTS

The minimum machine requirements for compiling and executing an RPG II program are:

- IBM 5406 Processing Unit, Model B2 (8K bytes) (includes the keyboard with eight command keys)
- IBM 5213 Printer or IBM 2222 Printer (includes ledger card device)
- IBM 5444 Disk Storage Drive

The following optional features are supported:

- 12K or 16K bytes of core storage
- IBM 5496 Data Recorder, Model 1 with System/3 Model 6 Attachment Feature
- IBM 129 Card Data Recorder with Card Input/Output Attachment Feature
- An additional IBM 5444 Disk Storage Drive
- IBM 2265 Display Station, Model 2 (Cathode Ray Tube) (used with the IBM 5406 Processing Unit, 12K or 16K bytes of core storage)
- 8 additional command keys

GENERAL RPG II OBJECT PROGRAM LOGIC

Each object program generated by the RPG II Compiler uses the same general program logic. For each record to be processed, the compiler goes through the same general program cycle (program cycle refers to all the logic functions performed for each data record read).

Every program cycle involves three basic logic steps:

1. Reading information (input).
2. Performing calculations (processing).
3. Recording results (output).

Within a program cycle, these basic logic steps can be divided into numerous substeps in which the input determines when calculation and output operations occur. According to RPG program logic, calculation and output operations are performed at two different times in a cycle: total time and detail time.

Total calculations are specified by placing an L indicator in columns 7–8 of the Calculation sheet. Total output operations are specified by placing a T in column 15 of the Output sheet. The appropriate control level indicator should be entered in columns 23–31 of the Output sheet to distinguish between output operations performed for different control levels.

Total calculation and total output operations are normally performed on data accumulated for a group of related records which form a control group. Such operations are normally done only after a control break has occurred. A control break occurs when the control field of the record just read is different from the control field of the previous record. Whenever a record is read, a check is made to determine if information in a control field (when one has been specified) is different from the control field information on the previous record.

A change in the control field information indicates that all records from a particular control group have been read and a new group is starting. When all records from a group have been read (shown by control level indicators being turned on), calculation and output operations are done using information accumulated from all records in that group. Information on the record that started the new control group is not used in these total operations; only information from records in the previous control group is used.

Those calculations not conditioned by L indicators in columns 7-8 of the Calculation sheet are detail calculations. Detail output operations are specified by placing an H or D in column 15 of the Output sheet. Detail calculation and detail output operations are normally performed for individual data records. These operations are done for each record, provided all conditioning indicators are satisfied. When any one of the following conditions is met, detail-time calculation and output operations are done:

1. All total calculation and total output operations have been completed.
2. No total operations are to be done (the information in the control field has not changed).

Total operations are performed before detail operations. This prevents data from the first record in a new control group from being accumulated in the totals for the previous group. Total operations are performed only on data accumulated from previous records. Detail operations on the record that caused the control break are done after total operations are finished.

Figure 3 shows specific steps in the general flow of RPG program logic. A program cycle begins with step 1 and continues through step 9, then begins again. Steps 6 and 7 are known as total time; steps 9 and 1 are known as detail time. The following statements describe each step:

- Step 1. Before the first record is read, the program prints all heading or detail lines (those having an H or D in column 15 of the Output sheet) on the first page. This is done only if all conditioning indicators are satisfied.
- Step 2. All control level and record identifying indicators are turned off.
- Step 3. A record is read and identified by the object program. The appropriate record identifying indicator is turned on.
- Step 4. The record just read is examined to determine whether or not a control break has occurred. (A control break occurs when the control field of the record just read is different from the control field of the previous record.)
- Step 5. If a control break has occurred, the proper control level indicator and all lower control level indicators are turned on, except L0 which is always on.
- Step 6. A check is made to determine if any of the control level indicators that are on are used in columns 7-8 to condition total calculations.
- Step 7. Total calculation operations (those conditioned by control level indicators used in columns 7-8 of the Calculation sheet) are performed if the appropriate control level indicator is on.
- Step 8. Total lines (those having a T in column 15 of the Output sheet) are written according to output specifications.
- Step 9. Data from the record read at the beginning of the cycle (step 3) is now made available for use in detail calculation and output operations.
- Step 10. All detail calculation operations (those not conditioned by control level indicators specified in columns 7-8 of the Calculation sheet) are performed on the data from the record read at the beginning of the cycle.

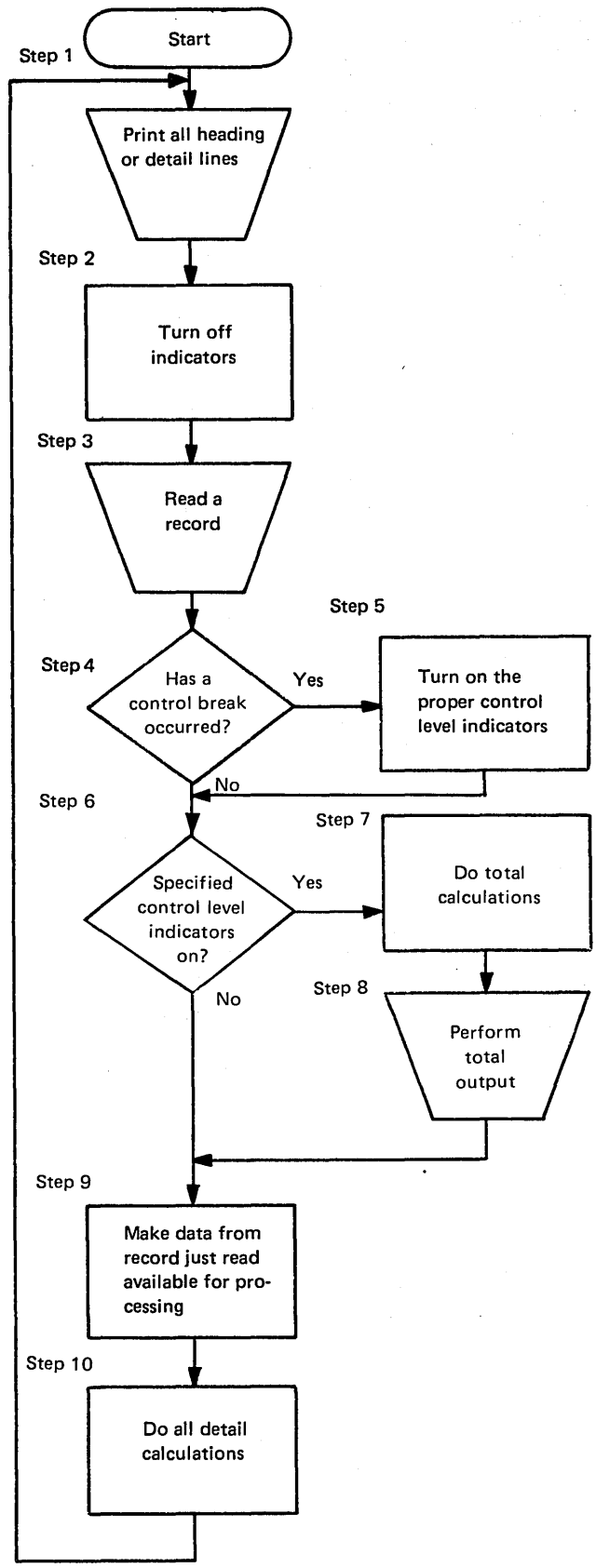


Figure 3. General Program Cycle

The first and last program cycles of a job are somewhat different from the normal cycle just described. Before the first record is read, lines conditioned by the first page (1P) indicator are printed. Any heading lines having no conditioning or all negative conditioning indicators are also printed at this time. Heading lines printed before the first record is read consist of constant information or fields for reserved words, such as PAGE and UDATE. In addition, total operations are bypassed until after the first record has been completely processed, even though a control break may occur. Steps 6 and 7 in the program are not done.

When the last record to be processed is read, the last record (LR) indicator is turned on. This automatically

causes all control level indicators to turn on. Total operations are performed and the job ends. Steps 8 and 9 in the program cycle are not done.

Knowledge of RPG logic is essential for anyone writing RPG II programs. For relatively simple jobs involving a single input file, knowledge of the general logic presented here is sufficient. For more complex jobs, a more detailed understanding is required. See *Supplementary Information, Detailed RPG II Program Logic* for a detailed flowchart and explanation of the program logic.

Common Entries

This section defines entries that are common to all RPG coding sheets. Each coding sheet contains the following entries.

1. Columns 1–2 (Page)
2. Columns 3–5 (Line)
3. Column 6 (Form Type)
4. Column 7 (Comments)
5. Columns 75–80 (Program Identification)

COLUMNS 1–2 (PAGE)

<i>Entry</i>	<i>Explanation</i>
01–99	Page number

Columns 1–2 in the upper right corner of each sheet are used to number the specifications sheets for your job. You can use more than one of each type of sheet if you need to, but keep all sheets of the same type together. When all specifications sheets are filled out, arrange them in the following order:

1. Control Card and File Description
2. Extension and Line Counter
3. Telecommunications

4. Input
5. Calculation
6. Output-Format

Number the sheets in ascending order.

COLUMNS 3–5 (LINE)

<i>Entry</i>	<i>Explanation</i>
Any numbers	Line numbers

Columns 3–5 are used to number the lines on each page. Columns 3–4 are preprinted on each sheet, so in most cases line numbering is already done for you. For instance, the Control Card and File Descriptions sheet contains line numbers for lines 01–07. If you need more than 7 lines on one sheet, enter 08 in columns 3–4 below line 07. Then 09 can be entered if it is required. The blank areas below the preprinted numbers can be used to insert a line between two lines you have completed (see *Example*).

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

Example

Figure 4 shows the insertion of a line. To show that a line belongs between line 02 and line 03, a 1 is placed in column 5 (any number 1–9 can be used). Line 021 should be inserted between 02 and 03. All lines inserted between existing lines should be written after the last line with a

printed line number. A maximum of nine lines can be inserted between two specification lines.

Note: If punched cards are used, the cards with insert lines must be placed in sequence before they are read.

IBM International Business Machines Corporation																																																																						Form X21-9083 Printed in U.S.A.																				
RPG CALCULATION SPECIFICATIONS																																																																																										
Date _____																																																																						Page <table border="1" style="display: inline-table; width: 20px;"><tr><td>1</td><td>2</td></tr></table>	1	2																		
1	2																																																																																									
Program _____																																																																						Program Identification <table border="1" style="display: inline-table; width: 100px;"><tr><td>75</td><td>76</td><td>77</td><td>78</td><td>79</td><td>80</td></tr></table>	75	76	77	78	79	80														
75	76	77	78	79	80																																																																																					
Punching Instruction <table border="1" style="display: inline-table; width: 100px;"><tr><td>Graphic</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>Punch</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>																																																																						Graphic										Punch										
Graphic																																																																																										
Punch																																																																																										
Programmer _____																																																																																										
Line	Form Type	Control Level (L, D, S, LR, SR)	Indicators			Factor 1	Operation	Factor 2	Result Field	Field Length	Decimal Positions	Resulting Indicators	Comments																																																																													
			And	And							Plus																																																																															
			Not	Not							Minus																																																																															
											Zero																																																																															
											Compare																																																																															
											High 1 > 2																																																																															
											Low 1 < 2																																																																															
											Equal 1 = 2																																																																															
											Lookup																																																																															
											Table (Factor 2) is																																																																															
											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																			
01	C					FACT1	ADD	FACT2	FACT2	82		30																																																																														
02	C			30			Z-SUB	FACT2	CFLD	82																																																																																
03	C					CFLD	COMP	1000.0				313233																																																																														
04	C																																																																																									
05	C																																																																																									
06	C																																																																																									
07	C																																																																																									
08	C																																																																																									
09	C																																																																																									
10	C																																																																																									
11	C																																																																																									
12	C																																																																																									
13	C																																																																																									
14	C																																																																																									
15	C																																																																																									
021	C			N30			Z-ADD	FACT2	CFLD																																																																																	
	C																																																																																									
	C																																																																																									
	C																																																																																									
	C																																																																																									

Figure 4. Insertion of Coding Lines

COLUMN 6 (FORM TYPE)

<i>Entry</i>	<i>Explanation</i>
H	Header record (control specifications)
F	File Description specifications
E	Extension specifications
L	Line counter specifications
T	Telecommunication specifications
I	Input specifications
C	Calculation specifications
O	Output-Format 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 record. The header record must always be the first record in the RPG II source program (see *Figure 2*).

COLUMN 7 (COMMENTS)

<i>Entry</i>	<i>Explanation</i>
*	Comment line

You often want to write comments that will help you understand or remember what you are doing in a certain section of coding. RPG II allows you to use an entire line for these comments. The comment line is identified by placing an asterisk in column 7. Any characters in the character set may 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. A comment line cannot be written in the control card specifications line.

Note: To be compatible with other RPG systems, the specification sheets show only 80 record positions used for RPG II coding.

COLUMNS 75–80 (PROGRAM IDENTIFICATION)

<i>Entry</i>	<i>Explanation</i>
Any valid program name	Program identification (the first character cannot be #, \$, or @ and no special character can be used in the name)
Blank	RPGOBJ

Control Card Specifications

Columns 75–80 of the control card specifications are used to name your object program and to identify each record in the object program. The program name (columns 75–80) is used in a program directory that contains the location of your program on disk. In addition, the compiler places the first four characters from the control card specifications (columns 75–78) in positions 89–92 of each record in your object program.

You can use any combination of alphabetic and numeric characters in columns 75–80; however, the first character must be alphabetic. Blanks must not appear between characters in the program name. The program name you specify should be unique. If columns 75–80 are left blank, the compiler assumes the entry is RPGOBJ. (The compiler uses positions 93–96 of each object program record for consecutive numbering of the records.)

All Other Source Program Specifications

Columns 75–80 of all source program specification sheets, except the Control Card, can contain any characters. These columns may use the program name in the control card 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.

COLUMN 6 (FORM TYPE)

An H must appear in column 6. A control record with an H in position 6 must be entered for every program even if all other control record positions are blank.

COLUMNS 7-9 (CORE SIZE TO COMPILE)

Columns 7-9 are not used. Leave them blank. Any entry in these columns is ignored by the compiler. The program is compiled in the available core storage.

COLUMN 10 (OBJECT OUTPUT)

<i>Entry</i>	<i>Explanation</i>
Blank	Object program is written temporarily in the object library. The system halts only when severe errors are found.
D	Object program is written temporarily in the object library. The system halts for both warning errors and severe errors. The operator can continue the job after a halt occurs for a warning error.
C	Object program is written permanently in the object library.
P	Object program is punched into cards.

Column 10 is used to indicate the output you want as a result of compiling the source program. The object program is usually written in the same object library in which the compiler resides, unless specified to another unit via OCL.

You will usually want the object program written temporarily in the object library until you have tested your program. When a program is written permanently in the object library, it deletes all programs temporarily written in the object library. (Every object program written permanently in the object library must be assigned a valid program name in columns 75-80 of the control card specifications.)

No object program is produced when severe (terminal) errors are present in the source statements.

If NOHALT is specified on your OCL for this job, no halts will be issued for either warning or terminal errors. When terminal errors are found, end-of-job occurs.

COLUMN 11 (LISTING OPTIONS)

<i>Entry</i>	<i>Explanation</i>
Blank	1. The object program is produced (if no severe errors are found). 2. The program listing is printed.
B	1. The object program is produced (if no severe errors are found). 2. The program listing is not printed.
P	1. The object program is produced (if no severe errors are found). 2. A partial program listing is printed, which includes the source program, information on indicator usage, and diagnostics.

Column 11 provides for listing options at the time your source program is compiled. If any severe errors are found during compilation, the system halts after completing the listing (provided a listing is to be printed).

The blank entry is the usual case, producing an object program (if no severe errors are found) and a source program listing. The RPG II listing consists of the source program listing, table array information, indicator usage information, the relative location of fields and their attributes, unreferenced field names, diagnostics, a core usage map, and a statement defining the total number of library sectors required for the object program. The core usage map lists the identification, the start address, and the size of each uniquely identifiable segment of code in the object program, and defines the amount of core required for execution.

The B entry means that no program listing is printed; however, an object program is produced. This entry can be used if you want to produce an object program for which you already have a listing.

The P entry means that a partial listing is printed, which includes the source program, information concerning indicator usage, and diagnostics. This can be used if you don't need a complete listing of the program. Excluded from this printout are table/array information, field information, a core usage map, and disk storage information.

COLUMNS 12–14 (CORE SIZE TO EXECUTE)

) Columns 12–14 define the core storage available for program execution (not including core requirements for the Supervisor). The entry must end in column 14.

This entry can be different from the core storage available for object program generation if your program is executed on a different system from the one that compiled your program.

The maximum core storage available for System/3 Disk RPG II compilers is 64K. If a program size is generated that exceeds X'FFFF' (65,535 bytes) at any time during compilation, compilation will cease. This terminal halt will occur before an attempt is made by the compiler to generate overlays.

Columns 13-14

<i>Entry</i>	<i>Explanation</i>
Blank	The core storage available for object program execution is the same as that used to compile the program.
01-13	The core storage available for program execution (if different from core storage available for object program generation).

Use columns 13-14 to specify some multiple of 1K bytes of storage (K=1,024).

Column 12

<i>Entry</i>	<i>Explanation</i>
Blank,0	No additional 256-byte increments are needed.
Q	One additional 256-byte increment is needed.
H	Two additional 256-byte increments are needed (512 bytes).
T	Three additional 256-byte increments are needed (768 bytes).

Column 12 may be used to specify additional 256-byte increments of storage. These increments allow an extra 1/4K, 1/2K, or 3/4K of storage to be available in addition to the storage specified in columns 13-14.

Example

The following chart shows examples of the possible entries that can be made in columns 12-14 and the amount of storage that would be made available for that entry:

<i>Entry</i>	<i>Available Bytes</i>
004	4,096
Q04	4,352 (4,096 + 256)
H04	4,608 (4,096 + 512)
T04	4,864 (4,096 + 768)
005	5,120

If the system used for program execution is different from that used for compilation, subtract the amount of core storage occupied by the supervisor from the total core storage of the system used for execution.

COLUMN 15 (DEBUG)

<i>Entry</i>	<i>Explanation</i>
Blank	DEBUG operation is not performed.
1	DEBUG operation is performed.

Use column 15 to indicate whether or not the DEBUG operation is performed. In order to perform a DEBUG operation:

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

See *Operation Codes, Debug Operation* under *Supplementary Information*.

COLUMNS 16-20

Columns 16-20 are not used. Leave them blank.

COLUMN 21 (INVERTED PRINT)

<i>Entry</i>	<i>Explanation</i>
Blank	Domestic format.
I	World Trade format.
J	World Trade format (zero is written to the left of the decimal comma).
D	United Kingdom format.

Use column 21 to describe the format and punctuation used in numeric and UDATE fields. The blank entry specifies that the System Date is in the domestic format of month/day/year for UDATE fields (10/15/70) and a decimal point for numeric fields (183.55). The I entry specifies that the System Date is in the World Trade format of day.month.year for UDATE fields (15.10.70) and a decimal comma for numeric fields (185,55). The J entry specifies that the System Date is in the same World Trade format as the I entry. When the J entry is used, zero balances are always written or punched with one zero to the left of the decimal comma (such as 0.00). Also, this leading zero appears for the J entry when there is some value in the field but no value to the left of the decimal comma (such as 0.04 or 0,10).

If an edit code is specified on the Output sheet, and the edit code is to print zero balances, a zero balance field will always have a zero to the left of the decimal comma—the edit code cannot suppress it.

The D entry specifies the United Kingdom format of day/month/year for UDATE fields (15/10/70), and a decimal point for numeric fields (183.55).

Note: The input for UDATE must be in the format expected as output. For example, if D (United Kingdom format) is specified in column 21, the input format must be DD/MM/YY.

Figure 6 shows how the entries in column 21 are used to format numeric data.

COLUMNS 22-25

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

COLUMN 26 (ALTERNATE COLLATING SEQUENCE)

<i>Entry</i>	<i>Explanation</i>
Blank	Normal collating sequence is used.
S	Alternate collating sequence is used.

Use column 26 only if you are altering the normal collating sequence. The following text contains a description of the specifications required to alter the normal collating sequence.

		Domestic Format	World Trade Format		United Kingdom Format
Unedited Data	Number of Decimal Positions	Column 21 Blank	Column 21		Column 21 D
			I	J	
072770	—	7/27/70	27.07.70	27.07.70	27/07/70
0034567	2	345.67	345,67	345,67	345.67
00345	3	.345	,345	0,345*	.345
00034	2	.34	,34	0,34*	.34
00000	2	.00	,00	0,00*	.00

● Assume that an edit code is used for zero-suppression.

* No edit code can remove leading zero.

Figure 6. Specifying the Format of Numeric and UDATE Fields on the Control Sheet

Alternate Collating Sequence

Every alphabetic, numeric, or special character holds a special position in relation to all other characters. This special order is known as the collating sequence. System/3 uses a collating sequence based on the way characters are represented in the machine (Figure 7).

You may change this collating sequence if you wish. If you want characters to appear in a sequence other than the one used by System/3 or if you want two or more characters to have the same position in the sequence (this means they are considered equal), you must describe an alternate collating sequence. The alternate collating sequence table is printed out with the compiled program. (Figure 2 shows the order of the RPG II source program including an alternate collating sequence table.)

Note: An alternate collating sequence applies to:

1. Matching fields and sequence checking.
2. Alphameric compare operations (COMP).

Defining Alternate Collating Sequence

To define an alternate collating sequence, you must first indicate that a sequence other than the normal one is to be used. Do this by entering an S in column 26 of the RPG II control card specifications.

A table also must be entered which lists the changes you wish to make in the normal collating sequence. This is a special table and requires no file description or extension entries. The following entries are needed for each table record entered:

Positions 1–6: Enter ALTSEQ to indicate that you are altering the normal sequence.

Positions 7–8: Leave these positions blank.

Positions 9–10: Enter the hexadecimal equivalent of the character being taken out of sequence. The table in Figure 7 lists characters and their hexadecimal equivalents.

Collating Sequence	Character	Hexadecimal Equivalent
1	Blank	40
2	Ç	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	,	6B
18	%	6C
19	_ (underscore)	6D
20	>	6E
21	?	6F
22	:	7A
23	#	7B
24	@	7C
25	'	7D
26	=	7E
27	"	7F
28	A	C1
29	B	C2
30	C	C3
31	D	C4
32	E	C5

Collating Sequence	Character	Hexadecimal Equivalent
33	F	C6
34	G	C7
35	H	C8
36	I	C9
37	}	D0
38	J	D1
39	K	D2
40	L	D3
41	M	D4
42	N	D5
43	O	D6
44	P	D7
45	Q	D8
46	R	D9
47	S	E2
48	T	E3
49	U	E4
50	V	E5
51	W	E6
52	X	E7
53	Y	E8
54	Z	E9
55	0	F0
56	1	F1
57	2	F2
58	3	F3
59	4	F4
60	5	F5
61	6	F6
62	7	F7
63	8	F8
64	9	F9

Figure 7. Normal Collating Sequence and Hexadecimal Equivalents of Characters

Positions 11–12: Enter the hexadecimal equivalent of the character that is replacing the character taken out of sequence.

Positions 13–16, 17–20, 21–24, etc: These positions are used the way positions 9–12 are used. The first two positions give the character to be replaced by the character specified in the next two positions. There may be as many four-position entries as can be contained in the record. Additional records may be used with the above format. The first blank position terminates the record. An `**b` or `/*` ends the table.

The alternate sequence table records must be preceded by a record with `**b` in columns 1–3. The remaining positions of the record may be used for comments. The alternate collating sequence records must follow the RPG II specifications and file translation records, if used. Figure 2 shows the arrangement of records in an RPG II source program.

Translation Table and Alternate Collating Sequence Coding Sheet

The Translation Table and Alternate Collating Sequence Sheet (Figure 8) can be used for coding an alternate collating sequence. It helps you to more easily determine the entries needed for the alternate collating sequence table input records.

Causing Characters to Be Considered Equal

If you want one character to be considered the same as another character, the characters must hold the same position in the collating sequence. For example, you may wish a blank to be considered as a zero. Therefore, you need to define an alternate collating sequence in which the blank is the same as the zero because it holds the same position in the sequence. The alternate collating sequence input record for this example looks like this:

<i>Record Position</i>	<i>Entry</i>
1–6	ALTSEQ
7–8	Blanks
9–12	40F0 (blank takes the zero's position)

Now whenever a blank is read and used in a compare, it is considered as a zero. Thus, if you were comparing numbers to 0036 to find an equal condition, 0036 and `b36` (where `b`=blank) both compare equal to 0036.

Altering the Normal Collating Sequence

You may alter the normal collating sequence in a number of ways. For example, you may insert a character between two existing characters, you may take a character out of the sequence, or you may change characters (put A where Z is and Z where A is). Regardless of how you alter the sequence, you must specify every character that is to be changed by the alteration. For example, if you want the dollar sign (\$) to be positioned in the collating sequence between A and B, the normal sequence is changed as follows:


<i>Normal Sequence</i>	<i>Altered Sequence</i>
A	A
B	\$
C	B
D	C
E	D
F	E
G	F
H	G
I	H
	I

Notice on the Translation Table and Alternate Collating Sequence Coding Sheet that there are many characters between I and `}`, R and S, Z and O. These characters can be represented in the computer and on records by a certain code. However, they have no printable graphic symbol. Due to this particular arrangement of graphics, nongraphics, graphics, etc. in the collating sequence, a character, when inserted between A and B, changes only the position of graphics B–I. All other graphics are not affected. B–I all move down one position causing the I to take the place of the nongraphic represented by hexadecimal CA. This does not matter, however, since the original character CA cannot be printed anyway. See Figure 9 for the entries on the Translation Table and Alternate Collating Sequence Coding Sheet.

The alternate sequence input record is as follows:

<i>Record Position</i>	<i>Entry</i>
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 is given a new position held by no other printable character)

Figure 8. Translation Table and Alternate Collating Sequence Coding Sheet



International Business Machines Corporation

TRANSLATION TABLE AND ALTERNATE COLLATING SEQUENCE CODING SHEET

Form X21-9096
Printed in U.S.A.

Code	System/3 Graphic	Entry	Replaced By/Takes Place Of	Code	System/3 Graphic	Entry	Replaced By/Takes Place Of	Code	System/3 Graphic	Entry	Replaced By/Takes Place Of	Code	System/3 Graphic	Entry	Replaced By/Takes Place Of
00000000		30		00110011		33		01100110		66		10011001		99	
00000001		01		00110100		34		01100111		67		10011010		9A	
00000002		02		00110101		35		01101000		68		10011011		9B	
00000011		03		00110110		36		01101001		69		10011100		9C	
00000100		04		00110111		37		01101010		6A		10011101		9D	
00000101		05		00111000		38		01101011		6B		10011110		9E	
00000110		06		00111001		39		01101100	%	6C		10011111		9F	
00000111		07		00111010		3A		01101101	-	6D		10100000		A0	
00001000		08		00111011		3B		01101110	>	6E		10100001		A1	
00001001		09		00111100		3C		01101111	?	6F		10100010		A2	
00001010		0A		00111101		3D		01110000		70		10100011		A3	
00001011		0B		00111110		3E		01110001		71		10100100		A4	
00001100		0C		00111111		3F		01110010		72		10100101		A5	
00001101		0D		01000000	Blank	40		01110011		73		10100110		A6	
00001110		0E		01000001		41		01110100		74		10100111		A7	
00001111		0F		01000010		42		01110101		75		10101000		A8	
00010000		10		01000011		43		01110110		76		10101001		A9	
00010001		11		01000100		44		01110111		77		10101010		AA	
00010010		12		01000101		45		01110000		78		10101011		AB	
00010011		13		01000110		46		01111001		79		10101100		AC	
00010100		14		01000111		47		01111010	:	7A		10101101		AD	
00010101		15		01001000		48		01111011	#	7B		10101110		AE	
00010110		16		01001001		49		01111100	@	7C		10101111		AF	
00010111		17		01001010	ç	4A		01111101	'	7D		10100000		B0	
00011000		18		01001011	.	4B		01111110	=	7E		10100001		B1	
00011001		19		01001100	<	4C		01111111	"	7F		10100010		B2	
00011010		1A		01001101	()	4D		10000000		80		10100011		B3	
00011011		1B		01001110	+	4E		10000001		81		10101000		B4	
00011100		1C		01001111		4F		10000010		82		10101001		B5	
00011101		1D		01010000	&	50		10000011		83		10101010		B6	
00011110		1E		01010001		51		10000100		84		10101011		B7	
00011111		1F		01010010		52		10000101		85		10111000		B8	
00100000		20		01010011		53		10000110		86		10111001		B9	
00100001		21		01010100		54		10000111		87		10111010		BA	
00100010		22		01010101		55		10001000		88		10111011		BB	
00100011		23		01010110		56		10001001		89		10111100		BC	
00100100		24		01010111		57		10001010		8A		10111101		BD	
00100101		25		01011000		58		10001011		8B		10111110		BE	
00100110		26		01011001		59		10001100		8C		10111111		BF	
00100111		27		01011010	!	5A		10001101		8D		11000000		C0	
00101000		28		01011011	\$	5B		10001110		8E		11000001	A	C1	
00101001		29		01011100	*	5C		10001111		8F		11000010	B	C2	
00101010		2A		01011101)	5D		10010000		90		11000011	C	C3	
00101011		2B		01011110	:	5E		10010001		91		11000100	D	C4	
00101100		2C		01011111	[]	5F		10010010		92		11000101	E	C5	
00101101		2D		01100000	-	60		10010011		93		11000110	F	C6	
00101110		2E		01100001	/	61		10010100		94		11000111	G	C7	
00101111		2F		01100010		62		10010101		95		11001000	H	C8	
00110000		30		01100011		63		10010110		96		11001001	I	C9	
00110001		31		01100100		64		10010111		97		11001010		CA	
00110010		32		01100101		65		10011000		98		11001011		CB	

Code	System/3 Graphic	Entry	Replaced By/Takes Place Of
11001100			CC
11001101			CD
11001110			CE
11001111			CF
11010000	⌋		DD
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

TRANSLATION TABLE AND ALTERNATE COLLATING SEQUENCE CODING SHEET

Code	System/3 Graphic	Entry	Replaced By	Code	System/3 Graphic	Entry	Replaced By	Code	System/3 Graphic	Entry	Replaced By	Code	System/3 Graphic	Entry	Replaced By
00110011		33		01100110		66		10011001		99		11001100			
00110100		34		01100111		67		10011010		9A		11001101			
00110101		35		01101000		68		10011011		9B		11001110			
00110110		36		01101001		69		10011100		9C		11001111			
00110111		37		01101010		6A		10011101		9D		11010000	I		
00111000		38		01101011		6B		10011110		9E		11010001	J		
00111001		39		01101100	%	6C		10011111		9F		11010010	K		
00111010		3A		01101101	-	6D		10100000		A0		11010011	L		
00111011		3B		01101110	>	6E		10100001		A1		11010100	M		
00111100		3C		01101111	?	6F		10100010		A2		11010101	N		
00111101		3D		01110000		70		10100011		A3		11010110	O		
00111110		3E		01110001		71		10100100		A4		11010111	P		
00111111		3F		01110010		72		10100101		A5		11011000	Q		
01000000	Blank	40		01110011		73		10100110		A6		11011001	R		
01000001		41		01110100		74		10100111		A7		11011010			
01000010		42		01110101		75		10101000		A8		11011011			
01000011		43		01110110		76		10101001		A9		11011100			
01000100		44		01110111		77		10101010		AA		11011101			
01000101		45		01111000		78		10101011		AB		11011110			
01000110		46		01111001		79		10101100		AC		11011111			
01000111		47		01111010	:	7A		10101101		AD		11000000			
01001000		48		01111011	#	7B		10101110		AE		11000001			
01001001		49		01111100	@	7C		10101111		AF		11000010	S		
01001010	¢	4A		01111101	'	7D		10110000		B0		11000011	T		
01001011	.	4B		01111110	=	7E		10110001		B1		11000100	U		
01001100	<	4C		01111111	"	7F		10110010		B2		11000101	V		
01001101	{	4D		10000000		80		10110011		B3		11000110	W		
01001110	+	4E		10000001		81		10110100		B4		11000111	X		
01001111		4F		10000010		82		10110101		B5		11010000	Y		
01010000	&	50		10000011		83		10110110		B6		11010001	Z		
01010001		51		10000100		84		10110111		B7		11010100			
01010010		52		10000101		85		10110000		B8		11010101			
01010011		53		10000110		86		10111001		B9		11011000			
01010100		54		10000111		87		10111010		BA		11011001			
01010101		55		10001000		88		10111011		BB		11011100			
01010110		56		10001001		89		10111100		BC		11011101			
01010111		57		10001010		8A		10111101		BD		11100000	0		
01011000		58		10001011		8B		10111110		BE		11100001	1		
01011001		59		10001100		8C		10111111		BF		11100010	2		
01011010	↑	5A		10001101		8D		11000000		C0		11100011	3		
01011011	\$	5B	C2 (B)	10001110		8E	\$ takes B's position.	11000001	A	C1		11101000	4		
01011100	*	5C		10010000		8F		11000010	B	C2	C3 (C)	11101001	5		
01011101	j	5D		10010001		91		11000011	C	C3	C4 (D)	11110100			
01011110	:	5E		10010010		92		10001000	D	C4	C5 (E)	11110101			
01011111	∩	5F		10010011		93		10001001	E	C5	C6 (F)	11110110			
01100000	-	60		10010100		94		10001010	F	C6	C7 (G)	11111001	9		
01100001	/	61		10010101		95		10001011	G	C7	C8 (H)	11111010			
01100010		62		10010110		96		10001000	H	C8	C9 (I)	11111011			
01100011		63		10010111		97		10001001	I	C9	CA	11111100			
01100100		64		10010110		98		10001010		CA		11111101			
01100101		65		10011000		99		10001011		CB		11111110			
												11111111			

(no printable character)

Figure 9. Altering the Collating Sequence

COLUMNS 27-36

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

COLUMN 37 (INQUIRY)

Entry	Explanation
Blank	Program cannot be interrupted (does not recognize an inquiry request).
B	Program can be interrupted (does recognize an inquiry request).
I	Program is an inquiry program that can only be executed when an inquiry request is made.

The system allows certain programs to be interrupted while they are being processed. A request for interruption is called an inquiry request (made by moving the inquiry request switch on the keyboard to ON). Programs are usually interrupted to permit another program to run. Control is then given back to the first program. The program to be loaded by an inquiry request must be an object program on disk and the OCL statements for the program must be entered from the keyboard.

A blank entry in column 37 indicates that the program cannot be interrupted (does not recognize an inquiry request).

A B entry indicates that the program can be interrupted (will recognize an inquiry request).

An I entry indicates that the program is an inquiry program that is executed only when the inquiry request is made. Usually this type of program is read in only when a B-type program is interrupted. In this case the I-type program will not recognize an inquiry request. However, if an I-type program is loaded in the normal manner (not because of a program interrupt), it can only be executed when an inquiry request is made. While this program is running, it will not recognize an inquiry request.

The RPG inquiry request is outlined in these steps:

1. Only a B-type program will recognize an inquiry request.
2. When the program recognizes an inquiry request, a Roll-Out routine moves the interrupted program from main storage to disk.
3. The program for which the interrupt was requested must be loaded normally. The interrupting program may be any type (blank, B, or I). This interrupting program cannot be interrupted.
4. After the interrupting program is executed, the interrupted program moves back into main storage using a Roll-In routine. The interrupted program begins execution at the point of interruption and terminates in a normal manner.

Programming Consideration: When the interrupted program is brought back into storage, any field lights that were on when the program was interrupted are not turned on again.

Note: An inquiry request can also be made by using IBM-written subroutine SUBR95 instead of roll-out/roll-in. For information on this method, see Appendix F.

COLUMNS 38–40

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

COLUMN 41 (1P FORMS POSITION)

<i>Entry</i>	<i>Explanation</i>
Blank	First 1P line is printed only once.
1	First 1P line can be printed repeatedly.

Use column 41 only when the first output line conditioned by the first page (1P) indicator is written to a printer file.

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. You may not want to print several lines of your report before you get the forms positioned correctly. In this case, you have the option of repeatedly printing the first line conditioned by the first page (1P) indicator by pressing the program start key. Each time the 1P line is printed, the program halts so you can reposition the forms if needed. When the forms are positioned correctly, the operator can continue the program by using the restart option. The page counter is not incremented until after the forms have been positioned correctly.

COLUMN 42

Column 42 is not used. Leave it blank.

COLUMN 43 (FILE TRANSLATION)

<i>Entry</i>	<i>Explanation</i>
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, combined, or update file is in a form not usable by your program. When file translation is specified for an update or combined file, both the input and output portion of the file is translated.

An F in column 43 indicates either or both of the following: (1) the character code used in the input data must be translated into a form that can be used by your program, or (2) the output data must be in a character code different from that used by your program.

The specification for forming a file translation table are described in the following text.

File Translation

At some time you may wish to use data that is in a character code different from the character code used in System/3. RPG II allows you to translate a different character code used as input into the System/3 code. You may also have the System/3 code translated for output into a different character code. This capability is called file translation.

File translation is specified by entering file translation table records. This is a special table and requires no File Description or Extension sheet entries. The file translation records must directly follow the RPG II Specifications in the source program (see Figure 2). The file translation table is printed with the compiled program.

File translation may be specified for input, output, update, and combined files. For input files, a different character code is translated into the System/3 code. For output files, the System/3 code is translated into a different character code. For update and combined files, the input data in a different code is translated into the System/3 code, then translated back to the different code for output.

In the following text, any character represented by the System/3 code will be known as an internal character; any character represented by a different code will be known as an external character.

Specifications for File Translation

You must first indicate that there are files to be translated. Do this by entering an F in column 43 of the RPG II control card specifications. Table input records must also be used to specify how the translation is to be done. The following entries are needed for each file translation table input record used.

Position 1–6: Enter *FILES to indicate that all input, output, update, and combined files are to undergo translation (both the input and output portions of update and combined files will be translated). Then complete your file translation input record by making the entries listed in the following text, beginning with positions 9–10. All files will be translated according to the table specified beginning in position 9.

If only certain files are to be translated, they must be named individually in positions 1–8. (The *FILES entry is not made in positions 1–6.)

Positions 1–8: Enter the filename of the input, output, update, or combined file to be translated (both the input and output portions of update and combined files will be translated). Then use the specifications listed, beginning with positions 9–10.

Positions 9–10: Enter the hexadecimal equivalent of the external character. This is the character in a different character code to be translated from input data or for output data.

Positions 11–12: Enter the hexadecimal equivalent of the internal character. This is the character in the System/3 code which represents internally the external input or output character.

Positions 13–16, 17–20, and 21–24, etc: These groups of positions are used the same way as positions 9–12 are used. For instance, columns 13–14 will contain the hexadecimal equivalent of the external character, and columns 15–16 will contain the hexadecimal equivalent of the related internal character.

All table records for one file must be kept together. The file translation table input records must be preceded by one record with ** \emptyset in positions 1–3. The remaining positions of this record may be used for comments.

Example

Assume that while working for a department store, you must process sales slips for all items sold. Each sales slip contains a punched and printed record of the actual, or wholesale, cost of its associated item along with a retail price.

Obviously, wholesale cost must remain confidential, so the store uses individual letters of a code-name in place of numbers comprising wholesale costs.

A typical code-name generally consists of a combination of letters that can be easily remembered by the store's personnel. The only restriction, however, is that the code-name must contain ten different letters, one for each of the numbers zero through nine.

Using the code-name BUCKINGHAM to represent numbers one through nine and zero, the letter B represents the number 1; letter U represents number 2, etc. Letter M represents zero. Individual letters are combined to represent each item's wholesale cost. Thus a wholesale cost of BBU.CC translates as 112.33; that is, one hundred twelve dollars and thirty-three cents.

In the following chart, hexadecimal equivalents of each letter in the word BUCKINGHAM are listed along with the hexadecimal equivalents of numbers one through nine and zero.

<i>Letter in Code-name (external character)</i>	<i>Hexadecimal Equivalent</i>	<i>Number</i>	<i>Hexadecimal Equivalent (internal character)</i>
B	C2	1	F1
U	E4	2	F2
C	C3	3	F3
K	D2	4	F4
I	C9	5	F5
N	D5	6	F6
G	C7	7	F7
H	C8	8	F8
A	C1	9	F9
M	D4	0	F0

Hexadecimal equivalents are merely a different way of representing the 8-bit code that the computer examines to recognize individual characters in your language.

See Figure 10. Note that if letters BBU were read and never translated, hexadecimal equivalents C2, C2, and E4 would be used by System/3. As a result, it would be impossible to perform an arithmetic operation involving the wholesale cost, BBU. Therefore, with the aid of file translation, the computer replaces the letters BBU with numbers.

The file translation specifications for letters in the word BUCKINGHAM are:

Record Position	Entry
1-6	*FILES
7-8	Blank
9-12	C2F1
13-16	E4F2
17-20	C3F3
21-24	D2F4
25-28	C9F5
29-32	D5F6
33-36	C7F7
37-40	C8F8
41-44	C1F9
45-48	D4F0

International Business Machines Corporation

TRANSLATION TABLE AND ALTERNATE COLLATING SEQUENCE CODING SHEET

Entry	Replaced By	Code	System/3 Graphic	Entry	Replaced By	Code	System/3 Graphic	Entry	Replaced By	Code	System/3 Graphic	Entry
33		01100110		66		10011001		99		11001100		CC
34		01100111		67		10011010		9A		11001101		CD
35		01101000		68		10011011		9B		11001110		CE
36		01101001		69		10011100		9C		11001111		CF
37		01101010		6A		10011101		9D		11010000	}	DO
38		01101011		6B		10011110		9E		11010001	J	D1
39		01101100	%	6C		10011111		9F		11010010	K	D2
3A		01101101	-	6D		10100000		AD		11010011	L	D3
3B		01101110	>	6E		10100001		A1		11010100	M	D4
3C		01101111	?	6F		10100010		A2		11010101	N	D5
3D		01110000		70		10100011		A3		11010110	O	D6
3E		01110001		71		10100100		A4		11010111	P	D7
3F		01110010		72		10100101		A5		11011000	Q	D8
40		01110011		73		10100110		A6		11011001	R	D9
41		01110100		74		10100111		A7		11011010		DA
42		01110101		75		10101000		A8		11011011		DB
43		01110110		76		10101001		A9		11011100		DC
44		01110111		77		10101010		AA		11011101		DD
45		01111000		78		10101011		AB		11011110		DE
46		01111001		79		10101100		AC		11011111		DF
47		01111010	:	7A		10101101		AD		11100000		E0
48		01111011	#	7B		10101110		AE		11100001		E1
49		01111100	@	7C		10101111		AF		11100010	S	E2
4A		01111101	'	7D		10110000		B0		11100011		E3
4B		01111110	=	7E		10110001		B1		11100100	U	E4
4C		01111111	''	7F		10110010		B2		11100101		E5
4D		10000000		80		10110011		B3		11100110	W	E6
4E		10000001		81		10110100		B4		11100111	X	E7
4F		10000010		82		10110101		B5		11101000	Y	E8
50		10000011		83		10110110		B6		11101001	Z	E9
51		10000100		84		10110111		B7		11101010		EA
52		10000101		85		10111000		B8		11101011		EB
53		10000110		86		10111001		B9		11101100		EC
54		10000111		87		10111010		BA		11101101		ED
55		10001000		88		10111011		BB		11101110		EE
56		10001001		89		10111100		BC		11101111		EF
57		10001010		8A		10111101		BD		11110000	0	F0
58		10001011		8B		10111110		BE		11110001	1	F1
59		10001100		8C		10111111		BF		11110010	2	F2
5A		10001101		8D		11000000		CO		11110011	3	F3
5B		10001110		8E		11000001		C1		11110100	4	F4
5C		10001111		8F		11000010	B	C2		11110101	5	F5
5D		10010000		90		11000011		C3		11110110	6	F6
5E		10010001		91		11000100	D	C4		11110111	7	F7
5F		10010010		92		11000101	E	C5		11111000	8	F8
60		10010011		93		11000110	F	C6		11111001	9	F9
61		10010100		94		11000111	G	C7		11111010		FA
62		10010101		95		11001000	H	C8		11111011		FB
63		10010110		96		11001001	I	C9		11111100		FC
64		10010111		97		11001010		CA		11111101		FD
65		10011000		98		11001011		CB		11111110		FE
										11111111		FF

E4, which, if translated, would represent the number 2, is the letter U in the code used by System/3.

C2, which, if translated, would represent the number 1, is the letter B in the code used by System/3.

Figure 10. Differences in Character Codes

Only the letters of the previous example will be specified for translation. All other characters will be handled in the normal manner. Remember that these letters will be translated for all fields in the file, not just the code-name field, BUCKINGHAM.

Translation Table and Alternate Collating Sequence Coding Sheet

You will find this coding sheet helpful for determining the correct entries you wish to make in the file translation table input record. Figure 11 shows the entries made on the sheet for the previous example.

COLUMN 44 (PUNCH MFCU ZEROS)

Entry	Explanation
Blank	Leading zeros are removed.
1	Leading zeros are used.

This column applies only to output on the DATA96. If the column is left blank, all numeric output fields on the DATA96 will be zero suppressed to the units position. Enter a 1 in column 44 when you wish to have leading zeros in fields punched or printed by the DATA96.

If an edit word or edit code is defined for fields to be printed or punched on the DATA96, the edit word or code will override column 44.

International Business Machines Corporation Form X21-9090
Printed in U.S.A.

TRANSLATION TABLE AND ALTERNATE COLLATING SEQUENCE CODING SHEET

Code	System/3 Graphic	Entry	Replaced By	Code	System/3 Graphic	Entry	Replaced By	Code	System/3 Graphic	Entry	Replaced By	Code	System/3 Graphic	Entry	Replaced By
00110011		33		01100110		66		10011001		99		11001100		CC	
00110100		34		01100111		67		10011010		9A		11001101		CD	
00110101		35		01101000		68		10011011		9B		11001110		CE	
00110110		36		01101001		69		10011100		9C		11001111		CF	
00110111		37		01101010		6A		10011101		9D		11010000	}	DD	
00111000		38		01101011		6B		10011110		9E		11010001	J	DD	
00111001		39		01101100	%	6C		10011111		9F		11010010	K	D2	F4
00111010		3A		01101101	%	6D		10100000		A0		11010011	L	D3	
00111011		3B		01101110	%	6E		10100001		A1		11010100	M	D4	E0
00111100		3C		01101111	%	6F		10100010		A2		11010101	N	D5	F6
00111101		3D		01110000		70		10100011		A3		11010110	O	D6	
00111110		3E		01110001		71		10100100		A4		11010111	P	D7	
00111111		3F		01110010		72		10100101		A5		11011000	Q	D8	
01000000	Blank	40		01110011		73		10100110		A6		11011001	R	D9	
01000001		41		01110100		74		10100111		A7		11011010	DA		
01000010		42		01110101		75		10101000		A8		11011011	DB		
01000011		43		01110110		76		10101001		A9		11011100	DC		
01000100		44		01110111		77		10101010		AA		11011101	DD		
01000101		45		01111000		78		10101011		AB		11011110	DE		
01000110		46		01111001		79		10101100		AC		11011111	DF		
01000111		47		01111010	:	7A		10101101		AD		11100000	EO		
01001000		48		01111011	#	7B		10101110		AE		11100001	E1		
01001001		49		01111100	@	7C		10101111		AF		11100010	S	E2	
01001010	¢	4A		01111101	*	7D		10110000		B0		11100011	T	E3	
01001011	.	4B		01111110	*	7E		10110001		B1		11100100	U	E4	F2
01001100	<	4C		01111111	#	7F		10110010		B2		11100101	V	E5	
01001101	(4D		10000000		80		10110011		B3		11100110	W	E6	
01001110	+	4E		10000001		81		10110100		B4		11100111	X	E7	
01001111		4F		10000010		82		10110101		B5		11101000	Y	E8	
01010000	&	50		10000011		83		10110110		B6		11101001	Z	E9	
01010001		51		10000100		84		10110111		B7		11101010	EA		
01010010		52		10000101		85		10111000		B8		11101011	EB		
01010011		53		10000110		86		10111001		B9		11101100	EC		
01010100		54		10000111		87		10111010		BA		11101101	ED		
01010101		55		10001000		88		10111011		BB		11101110	EE		
01010110		56		10001001		89		10111100		BC		11101111	EF		
01010111		57		10001010		8A		10111101		BD		11110000	0	F0	
01011000		58		10001011		8B		10111110		BE		11110001	1	F1	
01011001		59		10001100		8C		10111111		BF		11110010	2	F2	
01011010	†	5A		10001101		8D		11000000		C0		11110011	3	F3	
01011011	‡	5B		10001110		8E		11000001	A	C1	F9	11110100	4	F4	
01011100	*	5C		10001111		8F		11000010	B	C2	F1	11110101	5	F5	
01011101)	5D		10010000		90		11000011	C	C3	F3	11110110	6	F6	
01011110	:	5E		10010001		91		11000100	D	C4		11110111	7	F7	
01011111	∟	5F		10010010		92		11000101	E	C5		11111000	8	F8	
01100000	-	60		10010011		93		11000110	F	C6		11111001	9	F9	
01100001	/	61		10010100		94		11000111	G	C7	F7	11111010		FA	
01100010		62		10010101		95		11001000	H	CB	F8	11111011		FB	
01100011		63		10010110		96		11001001	I	CB	F5	11111100		FC	
01100100		64		10010111		97		11001010	CA	CB		11111101		FD	
01100101		65		10011000		98		11001011	CB	CB		11111110		FE	
												11111111		FF	

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

This is the hexadecimal equivalent of the System/3 character that will be substituted for the character that is to be translated.

51758

Figure 11. Specifications for File Translation Input Records

COLUMNS 45-47

Columns 45-47 are not used. Leave them blank.

COLUMN 48 (SHARED I/O)

<i>Entry</i>	<i>Explanation</i>
1	All disk files share a single input/output area.
Blank	Each disk file uses a separate input/output area.

Column 48 applies to disk files only. Enter a *1* in this column to indicate that all disk files in the program share a single input/output area.

Normally an RPG II program uses one input/output area for each file. An entry in column 48 allows all disk files to use one input/output area. By specifying a shared input/output area, you can reduce the amount of core storage needed to process a program. This is particularly important if a program is so large that it cannot run in the core storage you have available. However, the use of a shared input/output area increases the time required to process your program. Therefore, before you indicate that all disk files are to share one input/output area, be sure that the program would otherwise exceed the capacity of the system.

Note 1: A shared input/output area cannot be specified for multi-volume files (entry greater than 01 in columns 68-69 of the File Description sheet).

Note 2: Additional input/output areas (entry in column 32 of the File Description sheet) cannot be specified for disk files using a shared input/output area.

COLUMN 49 (FIELD PRINT)

<i>Entry</i>	<i>Explanation</i>
1	Your system has the special feature that allows printing in either direction across the page, but printing for this job is to be done in one direction only (left to right).

Use column 49 to override the special feature that allows printing in both directions across the page. Leave this column blank if the special print feature is not installed, or if it is installed and you want to use it for this job.

COLUMNS 50-74

Columns 50-74 are not used. Leave them blank.

COLUMNS 75-80 (PROGRAM IDENTIFICATION)

See *Common Entries*.

COLUMN 6 (FORM TYPE)

An F must appear in column 6.

COLUMNS 7–14 (FILENAME)

Use columns 7–14 to assign a unique filename to every file used in your program. Every file must be named, with the following exceptions:

1. Compile-time tables and arrays do not require a filename.
2. If multiple tables or arrays are read in at pre-execution time from the same device, only one filename is required. Note that all tables and arrays must be defined on the Extension sheet.

Filename can be from 1–8 characters long, and must begin in column 7. 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 may not appear between characters in the filename.

COLUMN 15 (FILE TYPE)

<i>Entry</i>	<i>Explanation</i>
I	Input file.
O	Output file.
U	Update file.
C	Combined file.
D	Display file.

Use column 15 to identify the way your program uses the file.

Input Files

Input files are records that a program uses as a source of data. When input files are described in a program, it indicates that records are to be read from the file. All input files must be further described on the Input sheet with the following exceptions:

1. Pre-execution-time tables and arrays and record address files are described on the Extension sheet. There is, however, a method of loading arrays using extension and input specifications (see *Supplementary Information, Arrays* for complete information).
2. Input files using the device name KEYBORD are further described on the Calculation sheet when the KEY operation is used.

Output Files

Output files are records written, punched, or printed by a program. All output files, except table output files, must be further described on the Output-Format sheet. Output table files are further described on the Extension sheet.

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. Update files must be further described on both the Input and Output-Format sheets; only the fields to be updated must be described on the Output-Format sheet. A chained file or a demand file may be updated at detail time, total time, or at exception output time. All other disk files can be updated only at detail time.

Combined Files

A combined file is both an input and an output file. A combined file can only be assigned to the ledger card device. 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-Format sheets.

Display Files

A display file allows you to print the contents of up to two fields used in your program. The DSPLY operation code must be used on the Calculation sheet in order to print a field or record directly from storage and/or to key data into a field or record in storage. Display files need only be described on the File Description sheet. The device name associated with a display file must be CONSOLE. See *Supplementary Information, Operation Codes, Display*.

COLUMN 16 (FILE DESIGNATION)

<i>Entry</i>	<i>Explanation</i>
P	Primary file.
S	Secondary file.
C	Chained file.
R	Record address file.
T	Table file (execution time tables or arrays).
D	Demand file.

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

Primary Files

A 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 *Supplementary Information, Multifile Processing* for more information on record selection in primary files.)

A primary file can be an input, update, or combined file. In programs that read records from only one file, that file is the primary file. Every program must have one, and only one, primary file.

Note: If the keyboard is specified as a primary input file, no other input files in the program can be specified as primary or secondary files.

Secondary Files

Secondary files apply to programs that do multifile processing. All files involved 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.

Note that table, chained, record address, and demand files are not involved in record selection in multifile processing. (See *Supplementary Information, Multifile Processing* for more information on primary and secondary files.)

Chained Files

A chained file is a disk file that uses the CHAIN operation code to do one of the following:

1. Read records randomly.
2. Load a direct file.

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

Record Address Files

A record address file is an input file that indicates to your program (1) which records are to be read from a disk file and (2) the order in which the records are to be read from the disk file. You cannot use more than one record address file in a program. All record address files must be further defined in extension specifications. Record address files contain record-key limits or disk addresses.

Record address files that contain record-key limits can be disk files, card files, or can be entered through the console. Files that contain limits are used with indexed files only. See *Column 28 (Mode of Processing), Sequential Within Limits* in this section for a complete discussion of this topic.

Record address files that contain disk addresses can only be disk files. Record address files on disk that contain disk addresses are called ADDROUT (address output) files. They are produced by the Disk Sort program and can be used with any type of disk file. See *Column 28 (Mode of Processing), Random By ADDROUT File* in this section for a complete discussion of this topic.

Table or Array Files

A table file is an input file that contains table or array entries. Tables or arrays can be entered from the following devices: CONSOLE, DISK, or DATA96. The entries can be read into the program during the compilation or execution of the program. Only pre-execution-time table or array files are described on the File Description sheet. However, tables and arrays must be described in extension specifications.

Entries read during compilation become a permanent part of the program. Both compile and pre-execution time tables or arrays can be changed at execution time. Compile-time tables or arrays, however, can be permanently altered only by recompiling the program. Pre-execution-time tables can be permanently altered each time the program is executed.

Table files are not involved in record selection and processing. They are only a means of supplying entries for tables used by the program. When pre-execution-time table or array files are read during the execution of the program, the program reads all the entries from the table or array files before it begins record processing. All table and array files must be further defined in extension specifications.

Demand Files

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

COLUMN 17 (END OF FILE)

<i>Entry</i>	<i>Explanation</i>
E	All records from the file must be processed before the program can end.
Blank	<ol style="list-style-type: none"> 1. The program can end whether or not all records from the file have been processed. 2. If column 17 is blank for all files, all records from every file must be processed before the program can end.

Column 17 applies to programs that perform multifile processing. Use it to indicate whether the program can end before all records from the file are processed.

This column applies only to input, update, and combined files used as primary, secondary, or record-address files. The devices associated with column 17 are disk files (except those processed by record address files or processed randomly), data recorder files, and console files. The last record in data files entered from the console or the data recorder must contain a /* so that RPG II can detect end-of-file. In input files assigned to the console, be sure that the input record defined on the Input sheet is at least two positions long so that you can enter the /* at end of job.

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 in the descriptions of the files provides that indication.

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

Note: An entry cannot be made in column 17 for a keyboard file or a ledger file. In order to terminate the program with a primary keyboard file or a primary ledger file, the LR indicator must be set on by the calculation specifications.

COLUMN 18 (SEQUENCE)

<i>Entry</i>	<i>Explanation</i>
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.
Blank	No sequence checking is to be done.

Use column 18 to indicate whether or not the program is to check the sequence of records. Column 18 applies to input, update, or combined files used as primary or secondary files. The devices associated with column 18 are disk files (except those processed randomly), ledger files, data recorder files, and console files. Use columns 61–62 on the Input sheet 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:

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

COLUMN 19 (FILE FORMAT)

<i>Entry</i>	<i>Explanation</i>
F	Fixed-length records.

Column 19 must contain an F. This entry indicates that all records in the file are of the same length.

COLUMNS 20–23 (BLOCK LENGTH)

<i>Entry</i>	<i>Explanation</i>
1–4096	<ol style="list-style-type: none">1. Record length for disk, data recorder, or console files.2. Multiple of disk record length.3. Length of largest field keyed for keyboard files.4. Length of largest output record for CRT files.5. Begin print position for ledger files and printer files.
1	The file assigned to TRACTR2 is used as a carbon copy of a ledger card file.
Blank	The block length for this file is the same as the record length.

Columns 20–23 have a different use depending on the device named for the file. The block length entry must end in column 23, and leading zeros can be omitted. Columns 20–23 can be left blank for any file except printer (TRACTR1, TRACTR2) and ledger card files (Figure 13).

Block Length for Disk Records

Block length must be a number, either record length, or a multiple of record length. The maximum block length is 4096.

Block length does not affect the way records are written on disk. Its function is to specify the amount of core storage to use for input/output area.

Note: If record length is used in these columns for disk files, RPG II will assign an efficient block length.

DEVICE	COLUMNS 20-23 BLOCK LENGTH	COLUMNS 24-27 RECORD LENGTH	MAXIMUM RECORD LENGTH
Disk	Record length or a multiple of record length	Record length	4096
Data Recorder	Record length	Record length	96
Console	Record length	Record length	125
Keyboard	Length of largest field to be keyed	Length of largest field to be keyed	128
Ledger	Begin print position (minimum 83)	End print position (maximum 214)	132
Printer (220-position)	Begin print position	End print position	220
Printer (132-position)	Begin print position	End print position	132
TRACTR2 (carbon copy)	1	1	
CRT	Length of longest output record	Length of longest output record	64
Special	Record length or a multiple of record length	Record length	4096

Figure 13. Block Length and Record Length Entries

COLUMNS 24–27 (RECORD LENGTH)

<i>Entry</i>	<i>Explanation</i>
1–4096	<ol style="list-style-type: none"> Record length for disk files, data recorder files, or console files. Length of largest field keyed for keyboard files. Length of largest output record for CRT. End print position for ledger files or printer files.
1	The file assigned to TRACTR2 is used as a carbon copy of a ledger card file. (See <i>Columns 40–46, Printer Files</i>).

Columns 24–27 have a different use depending on the device named for the file. An entry must be made for all files. Entries in these columns must end in column 27, and leading zeros can be omitted (see *Figure 13*).

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. The record length can be shorter than the maximum length for the device, but no longer.

Record Length for Disk Files

Due to the sector size (256 positions), the most efficient record length for disk files is 256, or a multiple or sub-multiple of 256.

Record Length for Keyboard Files

The record length for keyboard files should be the length of the largest field to be keyed (the largest field length in columns 49–51 of the Calculation sheet when the KEY operation is used).

COLUMN 28 (MODE OF PROCESSING)

<i>Entry</i>	<i>Explanation</i>
L	Sequential within limits
R	<ol style="list-style-type: none"> Random by relative record number. Random by key. Random by ADDROUT file. Direct file load (random load).
Blank	<ol style="list-style-type: none"> Sequential by key. Consecutive.

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, or chained, the possible methods depend upon the organizations of the files (*Figure 14*). For the other types of files, consecutive processing is the only possible method.

PRIMARY AND SECONDARY FILES		
<u>Method</u>	<u>Column 28</u>	<u>Column 31</u>
Consecutive	Blank	Blank
By ADDROUT	R	I
Sequential By Key	Blank	A or P
Sequential Within Limits	L	A *

CHAINED FILES		
<u>Method</u>	<u>Column 28</u>	<u>Column 31</u>
Random By Relative Record Number	R	Blank
Random By Key	R	A
Direct File Load (Random Load)	R	Blank **

* When the sequential-within-limits method of processing is specified, the number of extents entry in columns 68-69 for the file must not be greater than 01.

** A direct file load requires an O in column 15 and a C in column 16.

Figure 14. Specifications Identifying Methods for Retrieving Records

Column 31 is used to further identify the method for the program. See *Column 31 (Record Address Type)* in this section.

Consecutive

The consecutive method applies only to sequential and direct files. 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. (When a direct file is loaded, such spaces are filled with blanks.)

The program reads records from the file until either the end of that file is reached or the program ends due to the end-of-file condition of another file. See *Column 17, End of File* in this section 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, secondary files or demand files.

Records are read in ascending key sequence (the order in which the record keys are arranged in the index portion of the file). The program reads records until all records in the file are processed or the program ends due to the end-of-file condition of another file. See *Column 17, End of File* for more information about the second condition.

Sequential Within Limits

The sequential within limits method applies only to indexed disk files used as primary and secondary files and demand files. A limits record consists of the lowest record key and the highest record key of the records in the indexed disk file which are to be read. Limits records are contained in a record address file. The record address file can be located on disk, punched on cards, or entered by the printer-key-board.

To process sequentially within limits, the program reads:

1. A limits record from the record address file.
2. 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 these two steps until either the end of the record address file is reached or the program ends due to the end-of-file condition of another file. See *Column 17, End of File* in this section for more information about the second condition.

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

1. 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.
2. The low record key must begin in position one of the record. The high record key must immediately follow the low record key. A record key can be from 1–29 characters in length.
3. The low record key and the high record key must be equal in length and each key must be equal in length to the key field length specified in columns 29–30. Leading zeros are not necessary in specifying numeric record keys.
4. An alphameric record key can contain blanks.

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

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

The two record keys in a limits record can be equal. Only one data record will be read in this case.

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

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 or written 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. 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. 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.

Records are read during the calculation phase of the program. Therefore, they can be executed during detail or total calculation. Note then, that fields of records read from chained update files can be read and altered during total calculations and the records can be updated (written back on the file with alterations) during total output; the same also applies to detail calculations and detail output.

Random By ADDRROUT File

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

When an RPG II program uses an ADDRROUT file, it reads a disk address from the ADDRROUT file. The program then locates and reads records located at that address in the original disk file. Records are read in this manner until either the end of the ADDRROUT file is reached or the program ends due to the end-of-file condition of another file. ADDRROUT files must be further described on the Extension sheet. See *Column 17, End of File* in this section for more information about the second condition.

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

<i>Entry</i>	<i>Explanation</i>
--------------	--------------------

1–29	Length of record key or disk address.
------	---------------------------------------

Columns 29–30 apply only to indexed disk files and record-address files. Use it to indicate:

1. The length of the record keys in indexed files and record-address files that contain limits.
2. The length of the disk addresses in ADDRROUT files.
3. The length of record keys in packed format.

All of the key fields in the records in an indexed file must be the same length. The maximum is 29 bytes, 8 bytes for record keys in packed format. All of the disk addresses contained in an ADDRROUT file are three characters long.

COLUMN 31 (RECORD ADDRESS TYPE)

<i>Entry</i>	<i>Explanation</i>
A	Record keys in unpacked format are used in processing or loading indexed files.
I	The file is being processed by using disk addresses from the ADDRROUT file, or the file is an ADDRROUT file consisting of disk addresses.
P	Record keys in packed format are used in processing or loading indexed files.
Blank	<ol style="list-style-type: none"> 1. Relative record numbers are used in processing sequential and direct files. 2. A sequential or direct file is being loaded. 3. Records are read consecutively.

Column 31 applies to disk files specified as input, update, or chained output files. It indicates the way in which records in the file are identified (see Figure 14). Together, columns 28 and 31 indicate:

1. The method by which records are read from the file.
2. A direct file load.

For ADDRROUT files, column 31 must contain an I, indicating that disk addresses 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 output specifications.

COLUMN 32 (FILE ORGANIZATION OR ADDITIONAL I/O AREA)

<i>Entry</i>	<i>Explanation</i>
I	Indexed file.
T	ADDRROUT file.
1–9	Sequential file or direct file. Use two input/output areas for the file.
Blank	Sequential file or direct file. Use one input/output area for the file.

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

File Organization

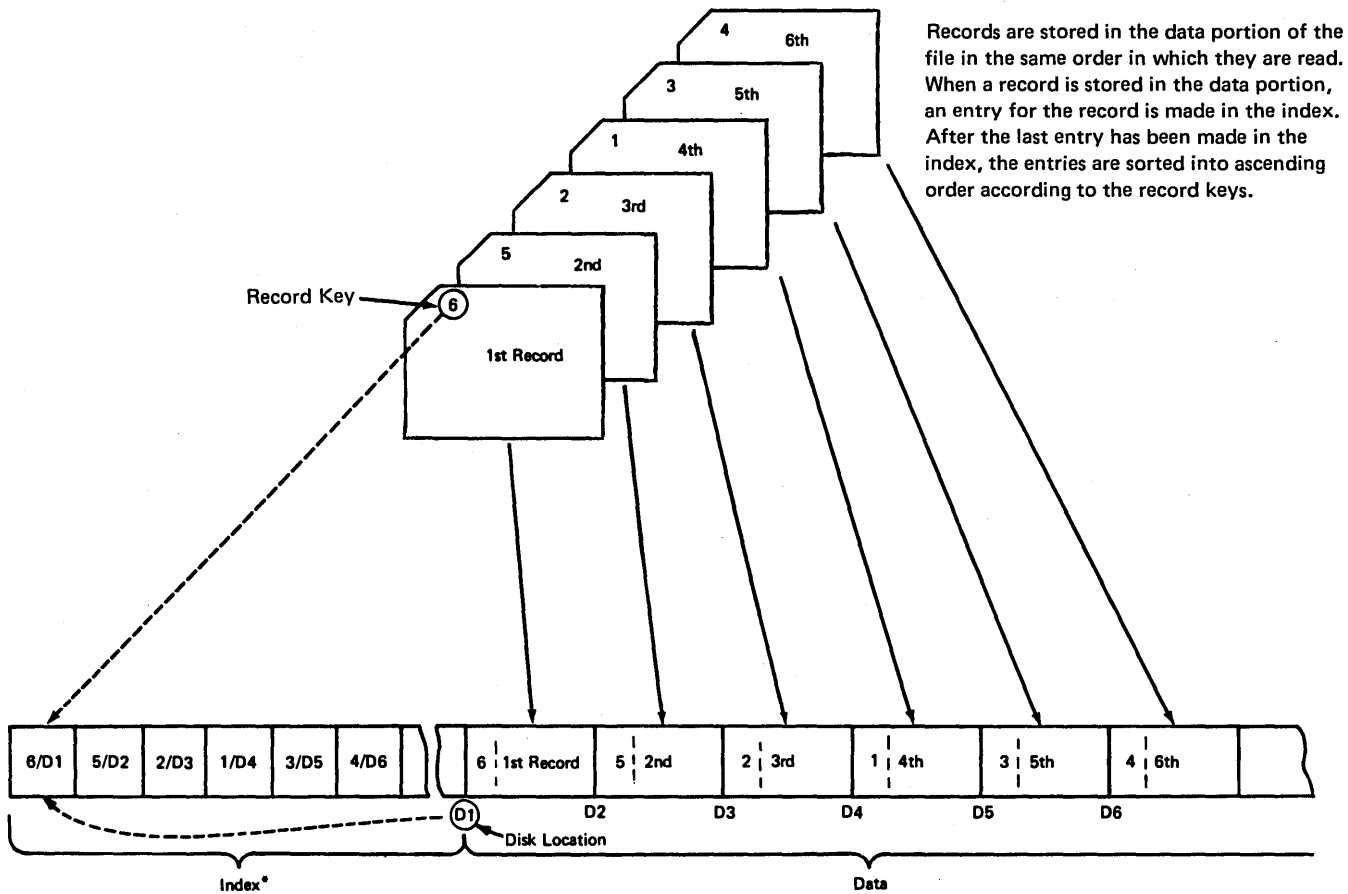
File organization is the arrangement of records in a file. The three types are indexed, direct, and sequential. Files organized in these ways are called indexed files, direct files, and sequential files, respectively.

Indexed Files

An indexed file is a disk file in which the location of records is recorded in a separate portion of the file called

an index. The index and its associated file occupy adjacent positions on disk. The index contains the record key and disk address of every record (Figure 15).

A record key is the information from the key field of a record. The record key can be used to identify the records of an indexed file. Record keys are always required in an indexed file. Indexed files can be loaded with the keys in ascending sequence or keys in non-ascending sequence. After a file is loaded in non-ascending key sequence, the keys in the index are placed in ascending sequence. See *File Description, Column 66* for a definition of the un-ordered load function.



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

ART: 55013

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

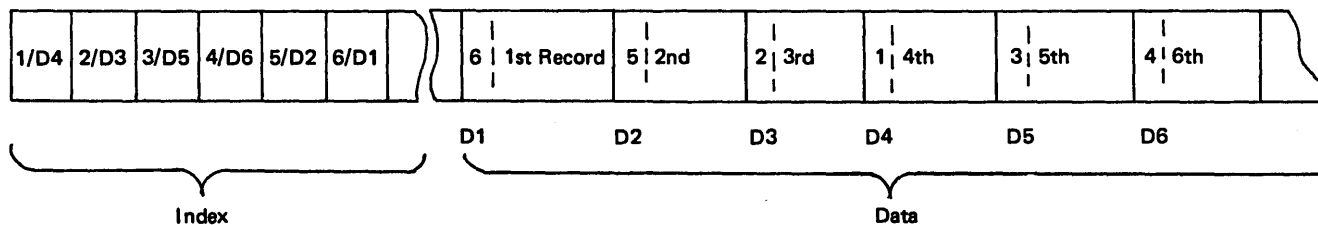
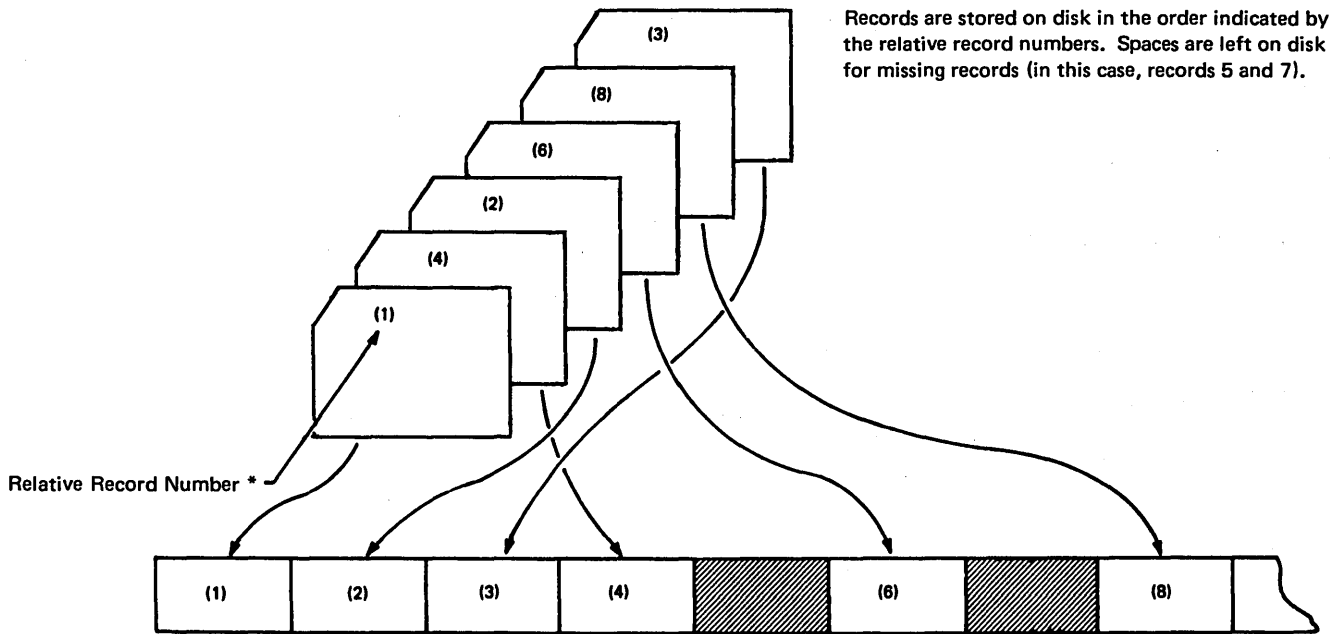


Figure 15. Indexed File Organization

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 (a specific disk address). Relative record numbers identify the relative position of a record within the file.

Before a direct file is loaded, the entire disk area for the direct file is cleared to blanks. Spaces are reserved in a direct file for records not available at the time the file is loaded (see *Figure 16*).



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

* The programmer usually derives relative record numbers from information in the records.

ART: 55010

Figure 16. Direct File Organization

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 17*).

Files other than disk files are always sequential files. Disk files can be sequential, direct, or indexed files.

Additional Input/Output Area

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

1. Direct disk files.
2. Sequential disk files.

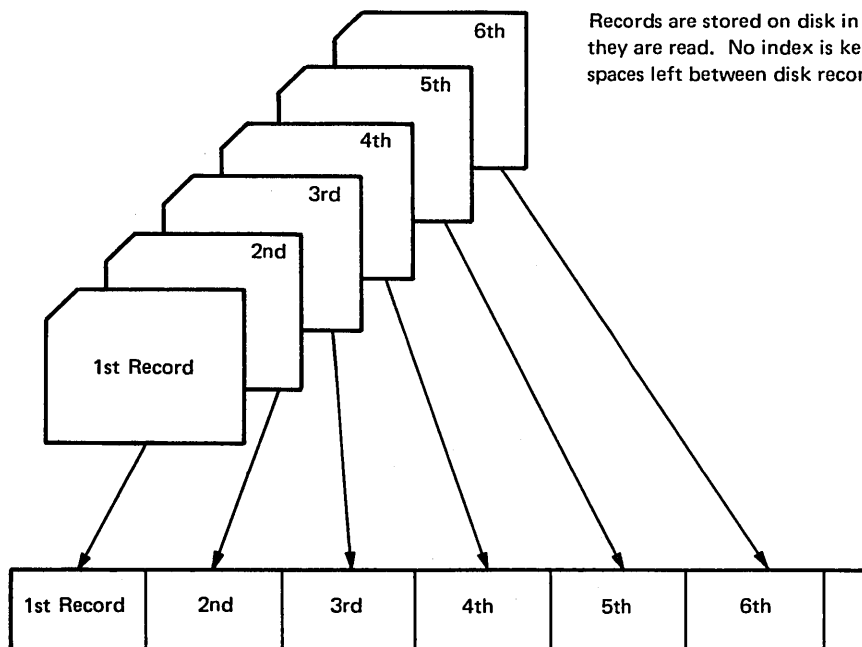
Additional input/output areas cannot be used for table or demand files or for disk files using a shared input/output area. Only disk files and input files assigned to the data recorder can use an additional I/O area.

The use of two I/O areas increases the efficiency of the program. However, it also increases the size of the program. Therefore, before you indicate that two areas are to be used for a file, be sure that the increase in size will not make your program exceed the capacity of your system.

Note: Additional I/O area cannot be specified for disk files with a shared input/output area (column 48 of the Control sheet). If both additional I/O and a shared input/output area are specified, additional I/O is dropped, and a warning message is printed.

ADDRROUT Files

When describing an ADDRROUT file, you must place a T in column 32. The ADDRROUT file must be a disk file.



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

Figure 17. Sequential File Organization

COLUMNS 33–34 (OVERFLOW INDICATORS)

<i>Entry</i>	<i>Explanation</i>
OA–OG, OV	An overflow indicator is used to condition records in the file. The indicator specified is the one used.
Blank	No overflow indicator is used.

Columns 33–34 apply to output files assigned to the printer and ledger card device and the combined file assigned to the ledger card device. Use these columns to indicate that you are using an overflow indicator to condition records being printed in the file.

Any overflow indicators used in a program must be unique for each file. (See *Supplementary Information, Ledger File Processing* for information on ledger card overflow.) The use of overflow indicators is described under *Overflow Indicators* in the Supplementary Information section. Note that only one overflow indicator can be assigned to a file. Do not assign overflow indicators to a console file.

COLUMNS 35–38 (KEY FIELD STARTING LOCATION)

<i>Entry</i>	<i>Explanation</i>
1–4096	Record position in which the key field begins.

Columns 35–38 apply to indexed disk files only. An entry must be made in these columns for an indexed disk file. This entry identifies the record position in which the key field begins. The key field of a record is the field that contains the information that identifies the record. The 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 number you place in these columns must end in column 38. Leading zeros can be omitted.

COLUMN 39 (EXTENSION CODE)

<i>Entry</i>	<i>Explanation</i>
E	Extension specifications further describe the file.
L	Line counter specifications further describe the file.

Column 39 applies only to (1) table and array files that are to be read during program execution, (2) record-address files, and (3) output files assigned to the printer. Use it to indicate whether the file is further described on the Extension sheet or the Line Counter sheet. Output files that are assigned to the printer can be described on the Line Counter sheet. Table, array, and record-address files must be described on the Extension sheet.

COLUMNS 40–46 (DEVICE)

<i>Entry</i>	<i>Explanation</i>
DISK	Disk.
KEYBOARD	Keyboard.
LEDGER	Ledger Card Device.
TRACTR1	132-position printer or primary print unit of 220-position printer.
TRACTR2	Secondary print unit of the 220-position printer.
BSCA	Binary Synchronous Communications Adapter
CONSOLE	Keyboard.
DATA96	Data Recorder (5496 or 129)
CRT65	Cathode Ray Tube.
SPECIAL	Device not supported by RPG II.

Use columns 40–46 to identify the input/output device to be used for the file. All entries must begin in column 40. The devices that can be used depend upon the form of the records (Figure 18).

For information about the RPG II telecommunications feature (BSCA), see the *IBM RPG II Telecommunications Programming Reference Manual, SC21-7507*.

FILE	FORM	POSSIBLE DEVICES
Primary Input Files	Keyed in by operator	KEYBOARD
Primary or Secondary Input Files	Cards	DATA96
	Disk	DISK
	Keyed in by operator*	CONSOLE
	SIOC	SPECIAL
Record Address Files Containing Record-Key Limits	TP Lines	BSCA
	Cards	DATA96
	Disk	DISK
	Keyed in by operator*	CONSOLE
Record Address Files Containing Disk Addresses (ADDROUT File)	Disk	DISK
Demand Files	Cards	DATA96
	Ledger card	LEDGER
	Disk	DISK
	Keyed in by operator	KEYBOARD
	Keyed in by operator*	CONSOLE
	SIOC	SPECIAL
Table Files	TP Lines	BSCA
	Cards	DATA96
	Disk	DISK
	Keyed in by operator*	CONSOLE
Chained Input Files	Disk	DISK
Update Files (Primary, Secondary, or Chained)	Disk	DISK
	SIOC	SPECIAL
Combined Files (Primary or Secondary)	Ledger card	LEDGER
	TP Lines	BSCA (only for conversational reply)
Output Files	Cards	DATA96
	Ledger card	LEDGER
	Disk	DISK
	Printed lines	CRT65
	Printed pages	TRACTR1 or TRACTR2 or CONSOLE
	SIOC	SPECIAL
	TP Lines	BSCA
Display File	Printed pages	CONSOLE

* Records are typed when they are keyed into the program.

● Figure 18. Device Assignment

Figure 19 shows the columns that can be used for the devices named. (TRACTR1 is TRACTR1 or TRACTR2.)

Note: The device names CONSOLE and KEYBOARD refer to the same physical device which includes the keyboard and the printer. Thus, you cannot assign files to both the CONSOLE and the KEYBOARD in the same program. Use CONSOLE when you want to use the DSPLY operation code; use KEYBOARD when you want to use the KEY or SET operation code.

Cathode Ray Tube

The Cathode Ray Tube can be used as an output device for normal and exception output. (See *Output-Format Specifications, Column 15* for more information on exception output.) Any alphanumeric character can be written on the CRT65. As many as 64 characters can be written across the width of the screen; a maximum of 15 such lines can appear on the screen at one time. Thus, you specify a maximum record length of 64 characters for CRT files.

Data moves onto the screen from top to bottom. As the bottom (sixteenth) line is written on the screen, the top line moves off. Data is written on the CRT screen at the normal output times (total and detail) or at calculation time for exception output.

The CRT65 is designed to be used when a response to an inquiry is needed. It cannot be used interchangeably with the printer as the 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. You can specify spacing before and after (0-3 entry in columns 17-18 of the Output sheet). You can specify a skip before to line 01 only (01 entry in columns 19-20 of the Output sheet). This specification is made whenever you wish to erase data from the CRT 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. You cannot specify a skip after (columns 21-22 of the Output sheet) for CRT files. When a line is printed on the CRT over a previous line, the previous line is erased. Edit codes, edit words, and output indicators can be specified for CRT files.

File Description Specifications for SPECIAL Device

The following file description specifications apply to files assigned to SPECIAL devices.

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	1-9 or blank.
33-39	Must be blank.
40-46	SPECIAL
47-53	Must be blank.
54-59	Name of the user-written subroutine which will perform the input/output operations. The subroutine name must be in the form SUBRxx, where x is any alphabetic character.
60-70	Must be blank.
71-72	U1-U8 or blank.
73-74	Must be blank.

COLUMNS 47-52

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

COLUMN 53 (CONTINUATION LINES-K)

Entry	Explanation
K	Continuation Record

Continuation records provide additional information about the special file being defined. One continuation record can be specified for each special file. When specifying a continuation record, columns 54-59 (Continuation Line Option) must be coded. Figure 20 shows an example of the coding necessary on the File Description sheet for a continuation line.

COLUMNS 54-59

Name of Label Exit

Entry	Explanation
SUBRxx	Name of the user-written subroutine which will perform the I/O operation for a SPECIAL device (x = any alphabetic character).
SRyzzz	Name of the IBM written subroutine (6-character name in library is \$\$yzzz) which will perform the I/O operation for a device supported by SPECIAL (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).
Blank	No SPECIAL device is being used.

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

Columns 54-59 must contain an entry for each data file assigned to a SPECIAL device. These columns are used to specify the subroutine which will perform the input/output operations for a file assigned to a SPECIAL device. The subroutine name entered in columns 54-59 can be from 4 to 6 characters long. The first four characters must be SUBR; the remaining characters can be any alphabetic characters.

Continuation Line Option

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

COLUMNS 60-65 (CORE INDEX)

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

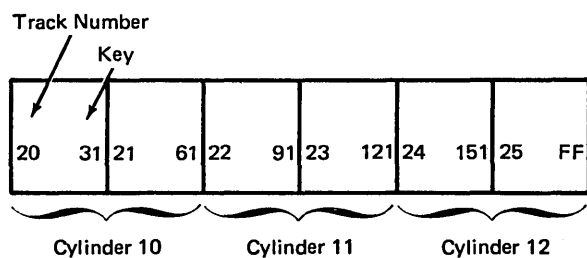


Figure 23. Core Index for INDEXT

The use of the core 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. With the core index, the system can find a specific record by searching only a small portion of the file index. Without the core index, however, all index entries which precede the record you want must be searched. Using the core index shown in Figure 23 the record with field 125 can be found in this manner:

(

(

(

- Search the core index until the first key field nearest to but lower than 125 is located. In this instance that key is 121; it has track 24 associated with it.
- Search track 24 in the file index until key 125 is located.
- Chain directly to the associated data record.

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

For efficient processing, the core index should be large enough to contain one entry (key and track number) for each track of index in the data file.

The storage area reserved for the core index is equal to key-field length plus 2, multiplied by the number of tracks in the file index. Therefore, for an indexed file having a key length of 4 and 10 tracks of file index, the most efficient core index requires 60 bytes of storage (4 plus 2 times 10).

If the storage space you specify in columns 60–65 is not large enough to contain one entry for each track of file index, the system will construct a table containing one entry for every cylinder of file index. Or, the core index might only contain one entry for every other cylinder. As the number of entries in the core index becomes fewer, the amount of processing time increases. If storage space is not enough for at least two index entries, the entry is ignored and no core index is used for this job.

COLUMN 66 (FILE ADDITION)

<i>Entry</i>	<i>Explanation</i>
A	New records will be added to the file.
U	Records are to be loaded for an indexed file in unordered sequence (non-ascending sequence).

Column 66 applies to consecutive and indexed disk files. This column indicates:

- The program is to add new records to the file (Example 1).
- The program is to load records in an unordered sequence (Example 2).

CALCULATING THE NUMBER OF BYTES REQUIRED FOR A CORE INDEX

To determine the number of bytes required for a core index, do the following calculations:

1. Key field length + 3 = length of each index entry in the data file.
2. $256^* \div$ length of index entry = number of entries per sector. (Round result down to nearest whole number.)
3. Number of records \div number of entries per sector = number of sectors. (Round result up to nearest whole number.)
4. Number of sectors \div 24** = number of tracks required for the file index. (Round result up to the nearest whole number.)
5. Key field length + 2 multiplied by the number of tracks (as determined in step 4) = number of bytes required for the core index.

* A sector contains 256 characters.

** Number of sectors per track.

Adding Records to a File (A)

Records added to a consecutive file are added at the end of the file.

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. The index is then reorganized so that the record keys (including the new ones) are in ascending order. File addition in column 66 cannot be specified for indexed files from which records are read using the sequential-within-limits method. Records added to an indexed file should be in ascending sequence.

After a file has been loaded on disk, it may be necessary to add records to the file. Records can be added at detail, total, or exception time during the program cycle. When chained files are specified with add, the records to be added may:

- Contain keys or record numbers that are above the highest presently in the file. In this case, the records constitute an extension of the file.
- Contain keys or record numbers 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 sequentially:

- The key of the record to be added must be lower than the key currently in process and higher than the preceding record.
- The file must be at end of file.

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

To add a record to a sequentially processed file, the program determines if the key of the record to be added is lower than the key currently in process and higher than the preceding record. If these requirements are not met, a halt occurs. Otherwise, the record is added.

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 have been loaded and an index built in the unordered sequence, the index is sorted into ascending sequence.

When an unordered load is specified, the number of extents entry in columns 68–69 for the file must not be greater than 01.

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

Column 15	Column 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.

* An A in column 66 requires an ADD entry in columns 16-18 of the Output-Format sheet.

If a large number of records are to be added to the file, the time required for the index sort can be decreased by allocating a special work file. This requires no special RPG II coding but does require a special OCL statement. For additional information and an example, see the *IBM System/3 Model 6 Operation Control Language and Disk Utility Programs Reference Manual*, GC21-7516.

Examples

Example 1

Figure 24 shows how records can be added to an indexed disk file. The new records are contained in another disk file, DISKIN. The file INDEXED is the existing disk file to which new records will be added. A printer file, PRINT, will provide a report showing all the records in DISKIN with an indication of which records were added to INDEXED and which records were not added.

On the File Description sheet, an A must appear in column 66 for the file INDEXED, and on the Output sheet ADD must appear in columns 16–18 for the new record to be added.

RPG OUTPUT - FORMAT SPECIFICATIONS

Date _____

Program _____

Programmer _____

Punching Instruction	Graphic						
	Punch						

Page 1 2

Program Identification 75 76 77 78 79 80

Line	Form Type	Filename	Type (H/D/T/E)			Space			Skip			Output Indicators			Field Name	Edit Codes	End Position in Output Record	Sterling Sign Position
			Stacks	Stacks	Stacks	Before	After	Before	After	Not	Not	Not	And	And				
01	O	INDEXED													KEY		10	
02	O														FIELD1		20	
03	O														FIELD2		30	
04	O														MORE		64	
06	O	PRINT																
07	O																	
08	O																	
09	O																	
10	O																	
11	O																	
12	O																	
13	O																	
14	O																	
15	O																	
16	O																	

Figure 24. File Addition (Part 2 of 2)

As defined on the Input sheet, all the records 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 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.

There may be records in DISKIN that do not belong in that file, or some records may have an error. These records are identified on the Input sheet as not having the character A in position 120. These records will 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 sheet, the constant 'RECORD NOT ADDED' is printed only on indicator 02, indicating a record that was not added to the disk file. In this manner, there will be a printed report of all records in DISKIN, and the records not added to INDEXED are identified by the constant 'RECORD NOT ADDED'.

Example 2

Figure 25 shows the coding for unordered loading of an indexed disk file from an unsequenced input disk file. 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 sheet indicates that an unordered load is to be done. The input file, INPUT, is described on the Input sheet as being unsequenced.

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 is built including the location of the record on disk. After the entire file has been loaded and an index entry has been constructed for each record, the index entries are sorted into ascending sequence.

IBM

International Business Machines Corporation

Form X21-9092
Printed in U.S.A.

RPG CONTROL CARD AND FILE DESCRIPTION SPECIFICATIONS

Date _____

Program _____

Programmer _____

Punching Instruction	Graphic								
	Punch								

Page **01**

Program Identification

75	76	77	78	79	80
----	----	----	----	----	----

Control Card Specifications

Line	Form Type	Core Size to Compile	Object Output Listing Options	Core Size to Execute	Debug	MFCM Stacking Sequence	Sterling	Input-Pence	Output-Pence	Inverted Print	360/20 2501 Buffer	Number Of Print Positions	Alternate Collating Sequence	Model 20	Model 20	Refer to the specific System Reference Library manual for actual entries.
01	H															

File Description Specifications

Line	Form Type	Filename	File Type	File Designation	End of File	Sequence	File Format	Block Length	Record Length	L/R	Mode of Processing	Device	Symbolic Device	Label S/N/E/M	Name of Label Exit	Extent Exit for DAM	Core Index	Continuation Lines	Option	Entry	File Addition/Unordered	Number of Tracks for Cylinder Overflow	Number of Extents	Tape Rewind	File Condition U1-U8	
02	F	INPUT	I	PE	F	96	96				DISK															
03	F	MASTER	O	F	F	256	64				DISK															

IBM

International Business Machines Corporation

Form X21-9094
Printed in U.S.A.

RPG INPUT SPECIFICATIONS

Date _____

Program _____

Programmer _____

Punching Instruction	Graphic								
	Punch								

Page **02**

Program Identification

75	76	77	78	79	80
----	----	----	----	----	----

Line	Form Type	Filename	Sequence	Number (1-N)	Option (O)	Record Identifying Indicator	Record Identification Codes									Field Location		Field Name	Control Level (L1-L9)	Matching Fields or Chaining Fields	Field Record Relation	Field Indicators			Sterling Sign Position		
							1	2	3	From	To	Plus	Minus	Zero or Blank													
01	I	INPUT	NS																								
02	I																										
03	I																										

IBM

International Business Machines Corporation

Form X21-9090
Printed in U.S.A.

RPG OUTPUT - FORMAT SPECIFICATIONS

Date _____

Program _____

Programmer _____

Punching Instruction	Graphic								
	Punch								

Page **03**

Program Identification

75	76	77	78	79	80
----	----	----	----	----	----

Line	Form Type	Filename	Type (H/D/T/E)	Space	:Skip	Output Indicators			Field Name	Edit Codes	Edit Codes				Sterling Sign Position			
						And	And	And			Commas	Zero Balances to Print	No Sign	CR		-		
01	O	MASTER	D															
02	O																	
03	O																	

Figure 25. Unordered Load of an Indexed File
File Description Specifications 51

COLUMN 67

Column 67 is not used. Leave it blank.

COLUMNS 68–69 (NUMBER OF EXTENTS)

Entry	Explanation
Blank	A single-volume file.
01–50	Number of volumes (disks) that contain the disk file.

When an entry is made in columns 68–69, it must end in column 69.

These columns define the number of volumes (disks) on which the disk file is located. A disk file must occupy consecutive cylinders on each volume. For instance, a disk file could not occupy cylinders 20–30 and 41–50 on one volume. The file could occupy 20–40 on that volume, or the data in cylinders 41–50 could be placed on another volume.

The number of volumes you can use depends on the mode of processing and the number of drives used. For single volume files the entry in columns 68–69 must be 01 or blank. The number of extents entry in columns 68–69 must not be greater than 01 when sequential processing within limits (L in column 28 and A in column 31) or an unordered load (U in column 66) is specified for the file.

For multi-volume files, determine the entry as follows:

1. **Consecutive Processing.** A disk file to be processed consecutively can be located on a fixed disk, a removable disk, or both if the entire file is online during processing. However, when portions of the file are offline during processing the file must be located on removable disks only. To determine the entry in columns 68-69 for a multivolume file, see Figure 26.
2. **Indexed Sequential or Random Processing by Keys.** A disk file to be processed indexed sequentially or randomly by keys can be located on a fixed disk, a removable disk, or both if the entire file is online during processing. However, when portions of the file are offline during processing, the file must be located on removable disks only. To determine the entry in columns 68–69 for a multivolume file, see Figure 26.

Note: For indexed random processing the volumes must be processed sequentially. However, the records on each volume may be processed randomly.

3. **Random Processing by Relative Record Number.** A disk file to be processed randomly by relative record number can be located on a fixed disk, a removable disk, or both. To process a multivolume disk file randomly by relative record number, the entire file must be available to the system at any given time. Therefore, the entire file must be online. To determine the entry in columns 68–69 for multivolume file, see Figure 26.

	ONE DRIVE	TWO DRIVES
	Maximum number of volumes allowed	Maximum number of volumes allowed
Consecutive processing, or indexed sequential or random processing by keys (offline—removable disks only)	50	50
Consecutive processing, or indexed sequential or random processing by keys (online—removable or fixed disks)	2	4
Random processing by relative record number (online—removable or fixed disks)	2	4

● Figure 26. Numbers of Volumes Allowed for Multi-Volume Files

COLUMN 70

Column 70 is not used. Leave it blank.

COLUMNS 71–72 (FILE CONDITION)

<i>Entry</i>	<i>Explanation</i>
U1–U8	The file is conditioned by the specified external indicator.
Blank	The file is not conditioned by an external indicator.

Columns 71–72 apply to input (excluding table input files), update, output, and combined files. These columns indicate whether or not the file is conditioned by an external indicator. 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 had been reached. (No records can be read from or written in the file.) See *Supplementary Information, Indicators, External Indicators* for more information.

COLUMNS 73–74

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

COLUMNS 75–80 (PROGRAM IDENTIFICATION)

See *Common Entries*.

FILE DESCRIPTION CHARTS

The File Description charts (Figures 27–37) are for:

1. Disk files and presented by disk file organization and processing method.
 2. Keyboard, ledger, data recorder, console, and printer files.
- 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.

Example

If you are updating an indexed disk file using the CHAIN operation code, see Figure 27 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, End of File, Sequence, Block Length, Cylinder Index in Core, Number of Extents, and File Condition.

File Description Specifications

Line	Form Type	Filename	File Type					Mode of Processing						Device	Symbolic Device	Name of Label Exit		Extent Exit for DAM		File Addition/Unordered					
			I/O/U/C/D	P/S/C/R/T/D	A/D	F/V/S/M/D	Block Length	Record Length	L/R	A/P/I/K	I/D/T or 2	Key Field Starting Location	Extension Code E/L			Label S/N/E/M	Option	Entry	A/U	R/U/N	File Condition U1-U8				
0 2	F		0	F									TRACTR1												
0 3	F																								
0 4	F																								
0 5	F																								
0 6	F																								
0 7	F																								
	F																								
	F																								

Device can be
TRACTR1 or
TRACTR2

Figure 35. Printer Files

File Description Specifications

Line	Form Type	Filename	File Type					Mode of Processing						Device	Symbolic Device	Name of Label Exit		Extent Exit for DAM		File Addition/Unordered					
			I/O/U/C/D	P/S/C/R/T/D	A/D	F/V/S/M/D	Block Length	Record Length	L/R	A/P/I/K	I/D/T or 2	Key Field Starting Location	Extension Code E/L			Label S/N/E/M	Option	Entry	A/U	R/U/N	File Condition U1-U8				
0 2	F		0	F									CRT65												
0 3	F																								
0 4	F																								
0 5	F																								
0 6	F																								
0 7	F																								
	F																								
	F																								

 SHADED COLUMNS
MUST BE BLANK

Figure 36. Cathode Ray Tube Files

File Description Specifications

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

Figure 37. SPECIAL Files

Extension specifications are needed to describe all record address files, tables, and arrays used in your job. Enter these specifications on the Extension and Line Counter sheet (Figure 38).

Record address files require entries on the Extension sheet in columns 11–26. Pre-execution time tables and arrays are described in columns 11–45. Compile time tables and arrays are described in columns 19–45. If an alternating table or array is to be specified with another table or array, it is described in columns 46–57 of the same line as the first. A maximum of 63 tables or arrays can be used in a program. Only 60 of these may be compile-time tables or arrays.

IBM

International Business Machines Corporation

RPG EXTENSION AND LINE COUNTER SPECIFICATIONS

Form X21-0091
Printed in U.S.A.

Date: _____ Page:

Program: _____

Punching Instruction:

Graphic								
Punch								

Programmer: _____

Extension Specifications

Line	Form Type	Record Sequence of the Chaining File							To Filename	Table or Array Name	Number of Entries Per Record	Number of Entries Per Table or Array	Length of Entry	P = Packed/B = Binary Decimal Positions Sequence (A/D)	Table or Array Name (Alternating Format)	Length of Entry	P = Packed/B = Binary Decimal Positions Sequence (A/D)	Comments
		Number of the Chaining Field			From Filename													
0 1	E																	
0 2	E																	
0 3	E																	
0 4	E																	
0 5	E																	
0 6	E																	
0 7	E																	
0 8	E																	
0 9	E																	
1 0	E																	

Line Counter Specifications

Line	Form Type	Filename	1		2		3		4		5		6		7		8		9		10		11		12	
			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
1 1	L																									
1 2	L																									
1 3	L																									

Figure 38. Extension and Line Counter Sheet

Figure 43 at the end of this chapter shows possible Extension sheet entries. See *Supplementary Information, Tables and Arrays* for a complete discussion of arrays.

COLUMNS 1–2 (PAGE)

See *Common Entries*.

COLUMNS 3–5 (LINE)

See *Common Entries*.

COLUMN 6 (FORM TYPE)

An E must appear in column 6.

COLUMNS 7–10

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

COLUMNS 11–18 (FROM FILENAME)

<i>Entry</i>	<i>Explanation</i>
Record Address Filename.	The name of the Record Address file.
Table or Array Filename.	Table or array loaded at pre-execution time.
Blank.	<ol style="list-style-type: none"> 1. Table or array loaded at compilation time if there is an entry in Number of Entries per Record (columns 33–35). 2. Array loaded at execution time (via input and/or calculations specifications) if there is no entry in Number of Entries per Record (columns 33–35).

Columns 11–18 are used 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 in the file description specifications. Leave columns 11–18 blank for compile time tables or arrays or for arrays loaded via input and /or calculation specifications.

These columns must contain the table or array filename of every pre-execution time table or array used in your program. When the table or array is loaded at compilation time, it is compiled along with the source program and included in the object program. Thus, a table file is not needed in addition to the object program every time the program is run. Only those tables and arrays which do not change often should be compiled with the program.

When tables or arrays are being compiled with the program, table file records must always follow the RPG II source program. A record with `**b` in positions 1–3 is needed to separate the RPG II source program from the table or array records. Tables or arrays must be separated from each other by records with `**b` in positions 1–3 (Figure 2). Because `**b` in positions 1–3 indicate the end of a table, you must not specify `**b` in positions 1–3 of your table input records. This would cause the system to stop processing the table. A record with `/*` in positions 1 and 2 must follow the table file input records. Each table or array loaded at pre-execution time must be followed by a record with `/*` in positions 1–2.

Short tables (tables that are not full) can be compiled with the program, but a warning is issued. See *Columns 36–39* for a discussion of short tables.

COLUMNS 19–26 (TO FILENAME)

<i>Entry</i>	<i>Explanation</i>
Name of an input or update file.	The file processed via the Record Address file named under From Filename.
Name of an output file.	The output file to which a table or array is to be written or punched.

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

If a record address file named under From Filename (columns 11–18) the name of the input or update file that contains the data records to be processed must be entered under To Filename (columns 19–26). Do not enter the record address filename in these columns.

If you wish a table or array to be written, use columns 19–26 to enter the filename of the output file you will use to do this. This output file must have been previously named in the file description specifications.

A table or array can be written on only one output device. Leave columns 19–26 blank if you do not want the table or array written.

If a table or array is assigned to an output file, it is automatically written at the end of the job after all other records have been written.

Since the table or array will be written in the same format in which it was entered, you may want to rearrange the output table or array through output-format specifications. If this is done, the table or array will be written in the specified format at the end of this job, but will also be written in the same format in which it was entered. You may format table or array output by using exception lines to write one item at a time (see *Supplementary Information, Operation Codes, Exception*). Tables or arrays should be written only after all records have been processed (Last Record indicator is on).

COLUMNS 27–32 (TABLE OR ARRAY NAME)

<i>Entry</i>	<i>Explanation</i>
Table or Array name.	Name of each table or array used in the program.

Use columns 27–32 to name your table or array. No two tables or arrays may 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 be given a name. The entire table name can be from 3–6 characters long, and

must begin with the letters TAB. After the letters TAB, 1–3 alphabetic or numeric characters may be used (no special characters allowed). Blanks may not appear between characters in the table name. Any name in columns 27–32 which does not begin with TAB is considered an array name. The table name 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 Result Field (on the Calculation sheet) with LOKUP operation, it refers to the entire table. When the table name is used with any other operation code, it refers to the table item last selected from the table by a LOKUP operation. See *Supplementary Information, Operation Codes, Look-up and Tables and Arrays*.

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

If two tables are in alternating format in one table file, the table whose item appears first must be named in columns 27–32. The second table is named in columns 46–51 (see *Example*).

Array Name

Every array used in your program must be given a name. An array name cannot begin with the letters TAB. This array name is used throughout the program. See *Supplementary Information, Tables and Arrays* for more information on forming array names.

Example

Figure 39, insert A, shows two tables (TABA and TABB) described in alternating format. An item for TABA appears

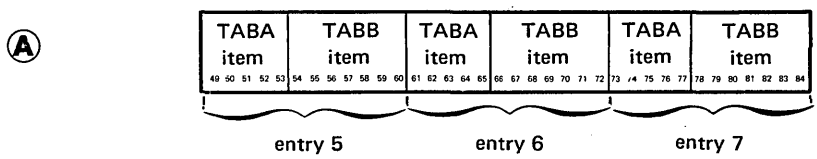
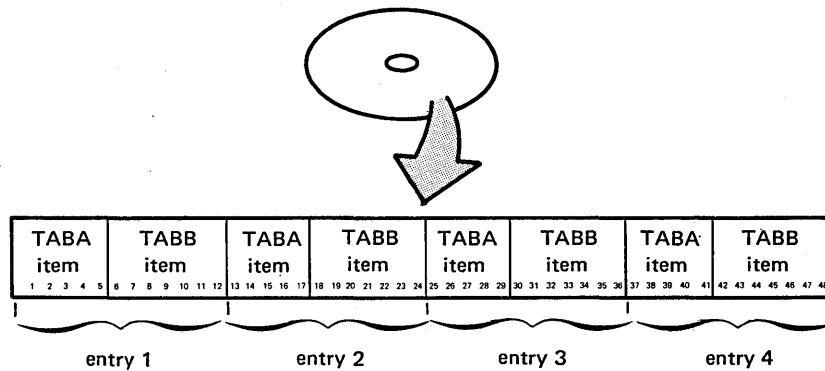
first. Thus, in insert B, TABA is named in columns 27–32. of the Extension sheet; TABB is named in columns 46–51.

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

-----Corresponding Table Items

5 7
Positions Positions

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 machine in alternating format. Corresponding items from the two tables are considered as one entry.

Figure 39. Related Tables (Part 1 of 2)

RPG EXTENSION AND LINE COUNTER SPECIFICATIONS

Punching Instruction	Graphic Punch								
----------------------	---------------	--	--	--	--	--	--	--	--

Extension Specifications

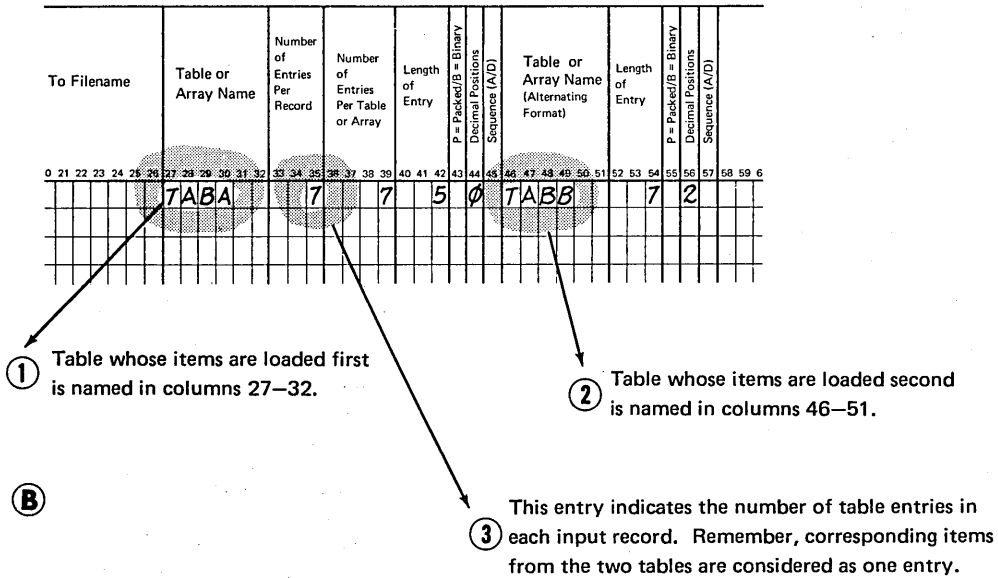


Figure 39. Related Tables (Part 2 of 2)

COLUMNS 33–35 (NUMBER OF ENTRIES PER RECORD)

Entry	Explanation
1–999	Number of table or array entries found in each table or array input record.

Indicate in columns 33–35 the exact number of table or array entries in each table or array input record. Every table or array input record except the last must contain the same number of entries as indicated in columns 33–35. The last record may contain fewer entries than indicated, but never more.

When two tables or arrays are described in one file, each table or array input record must contain the corresponding items from each table or array written in alternating format. These table items are considered as one entry (see Example). The number entered must end in column 35. Corresponding items from tables or arrays in alternating format must be on the same record. Comments may be entered on table input records in positions following table entries.

When loading an array the following must be considered:

- To load a pre-execution time array, a filename must be entered in columns 11–18 and an entry must be made in Number of Entries per Record (columns 33–35).
- To load an array at compile time, the filename entry (columns 11–18) must be blank, but an entry must be made in Number of Entries per Record (columns 33–35).
- To load an execution time array (via the input and/or calculations specifications), the filename (columns 11–18) entry must be blank and the Number of Entries per Record (columns 33–35) must be blank.

Example

Figure 39, insert B, shows table entries for the two tables, A and B, entered in alternating format. A1 and B1, the corresponding items in tables A and B, are considered one entry. Even though there are 14 table items listed, there are only 7 table entries.

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

Entry Explanation

1-9999 Maximum number of table or array entries.

Use columns 36-39 to indicate the maximum number of table items which can be contained in the table named in columns 27-32, or the maximum number of array items which can be contained in the array named in columns 27-32. This number may apply to one table or to two tables in alternating format. Any number entered in these columns must end in column 39.

Since the number of items for two tables or arrays entered in alternating format must be the same, the entry in these columns also gives the number items in a second table or array (columns 46-51).

If your table or array is full, this entry gives the exact number of items in it. However, if the table or array is not full, the entry gives the number of items that can be put into it (Figure 40). A table or array that is not full is known as a short table or array.

If a table or array is to be compiled, it 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. The short table or array is completed in storage by filling it with blanks or zeros (for alphameric or numeric tables or arrays, respectively). A pre-execution time table or array need not be full. All pre-execution time tables and arrays must be followed by a record with /* in positions 1 and 2 when they are loaded.

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

TABPRT (Part Number)	TABAMT (Price)	TABPRT (Part Number)	TABAMT (Price)
001	127.62	001	127.62
002	198.32	002	198.32
003	.27	003	.27
004	.01	004	.01
005	1.98	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 machine, TABPRT and TABAMT will be full (20 entries fill the table).

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

International Business Machines Corporation
EXTENSION AND LINE COUNTER SPECIFICATIONS

Punching Instruction	Graphic								
	Punch								

Page

1	2
---	---

Extension Specifications

Table or Array Name	Number of Entries Per Record	Number of Entries Per Table or Array	Length of Entry	P = Packed/B = Binary Decimal Positions Sequence (A/D)	Table or Array Name (Alternating Format)	Length of Entry	P = Packed/B = Binary Decimal Positions Sequence (A/D)
TABPRT	12	20	3	0	TABAMT	5	2

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

Figure 40. Table Entries (Number Per Table)

COLUMNS 40–42 (LENGTH OF ENTRY)

Entry	Explanation
1–15	Length of a numeric entry.
1–256	Length of an alphabetic entry.

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

All table or array items must have the same number of characters. It is almost impossible, however, for every item to be the same length. Therefore, add zeros or blanks before numeric items to make them the same length. Add blanks after alphabetic items (see *Examples, Example 1*).

If two tables or arrays are entered in alternating format, the specification in columns 40–42 applies to the table or array whose item appears first in the record (see *Examples, Example 2*).

The maximum length of a numeric table or array item is 15 characters. The maximum length of an alphabetic table or array item is 256 characters for disk records, up to 125 characters for console records, and up to 96 characters for punched cards.

See *Arrays or Tables* under *Supplementary Information* for more information.

Examples

Example 1: Figure 41 shows a table, called TABMO, which lists the months of the year. The name SEPTEMBER, having nine characters, is the longest entry. Because the lengths of the entries must be the same, blanks are added to the remaining names to make each of them nine characters long.

JANUARY
FEBRUARY
MARCH
APRIL
MAY
JUNE
JULY
AUGUST
SEPTEMBER
OCTOBER
NOVEMBER
DECEMBER

List of Months

JANUARYbb
FEBRUARYb
MARCHbbbb
APRIL bbbb
MAYbbbbbb
JUNE bbbbb
JULY bbbbb
AUGUSTbbb
SEPTEMBER
OCTOBERbb
NOVEMBERb
DECEMBERb

Table of Months

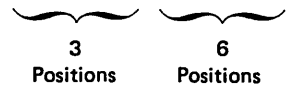
All entries must have the same length. Those items that are not as long as the longest item must be padded with blanks (b).

Figure 41. Length of Table Entries

Example 2: Figure 42 shows entries for two tables entered in alternating format, TABCOD and TABAMT. Each item in TABCOD is three characters long; each item in TABAMT is six characters long. Since TABCOD is entered in the machine first, its length, 3 is specified in columns 40–42. The length of items in TABAMT is indicated in columns 52–54.

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

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



(A)

COLUMN 43 (PACKED OR BINARY FIELD)

Entry	Explanation
Blank	Data for table or array is in unpacked 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 pre-execution time table or array is in packed or binary format. Leave column 43 blank if the field is unpacked. See *Input Specifications, Column 43* for more information on packed or binary formats.

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.

Column 44 must always have an entry for a numeric table or array. If the items in a table or array have no decimal positions, enter a 0.

If two tables or arrays are entered in alternating format, the specification in this column applies to the table or array containing the item which appears first on the record.

International Business Machines Corporation

PG EXTENSION AND LINE COUNTER SPECIFICATIONS

Punching Instruction	Graphic Punch								

Page

Extension Specifications

ame	Table or Array Name	Number of Entries Per Record	Number of Entries Per Table or Array	Length of Entry	P = Packed/B = Binary Decimal Positions Sequence (A/D)	Table or Array Name (Alternating Format)	Length of Entry	P = Packed/B = Binary Decimal Positions Sequence (A/D)
	TABCOD	10	10	3	P	TABAMT	6	2

The length of the table item which is entered in the machine first must

(B) appear in columns 40–42.

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.

Figure 42. Length of Corresponding Table Items

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. The program will halt after compilation. If a pre-execution 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 items are entered starting with the lowest data item (according to the collating sequence) and proceeding to the highest. Descending order means that the table or array items are entered starting with the highest data item and proceeding 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 item which appears first on the record.

When you are searching a table or array for an item (LOKUP) and wish to know if the item is high or low compared with the search word, your table or array must be in either ascending or descending order. See *Supplementary Information, Operation Codes, Look-up* for more information. When a specific sequence has been specified, RPG II checks the data in the table or array to see if it really is in that sequence.

An execution time array (built in input and/or calculation specifications) will not be sequence checked. However, an A or D entry must be specified if a High or Low look-up operation is performed.

COLUMNS 46–57

Use columns 46–57 only when describing a second table or array which is entered in an alternating format with another table or array. Usually, the second table or array corresponds with the table or array named in columns 27–32. All fields in this section have the same significance and require the same entries as the fields with corresponding titles in columns 27–45. See the previous discussion on those columns for information about correct specifications.

Leave these columns blank for a single table or array.

COLUMNS 58–74 (COMMENTS)

Enter any information you wish in columns 58–74. The comments you use should help you understand or remember what you are doing in 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*.

RPG EXTENSION AND LINE COUNTER SPECIFICATIONS

Date _____

Program _____

Programmer _____

Punching Instruction	Graphic						
	Punch						

Page

1	2
---	---

Program Identification

75	76	77	78	79	80
----	----	----	----	----	----

Extension Specifications

Line	Form Type	Record Sequence of the Chaining File		To Filename	Table or Array Name	Number of Entries Per Record	Number of Entries Per Table or Array	Length of Entry	P = Packed/B = Binary Decimal Positions Sequence (A/D)	Table or Array Name (Alternating Format)	Length of Entry	P = Packed/B = Binary Decimal Positions Sequence (A/D)	Comments
		Number of the Chaining Field	From Filename										
01	E			← Output file →	← Compile time table →					← Alternating table →			} Tables
02	E			← Table file →	← Output file →					← Pre-execution time table →			
03	E												} Arrays
04	E			← Output file →	← Compile time array →					← Alternating array →			
05	E			← Array file →	← Output file →					← Pre-execution time array →			
06	E									← Execution time array →			
07	E												} Record Address Files
08	E			← R.A. file →	← Input or update file →								
09	E												
10	E												

Line Counter Specifications

Line	Form Type	Filename	1		2		3		4		5		6		7		8		9		10		11		12	
			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
11	L																									
12	L																									
13	L																									

- The shaded columns must be blank for the file named.
- For tables and all arrays except execution time arrays, columns 19–26 and columns 46–57 are optional.
- Execution arrays are loaded via input and/or calculation specifications.
- For record address files, columns 11–26 must have entries.

Figure 43. Possible File Entries for Extension Specifications

COLUMNS 1–2 (PAGE)

See *Common Entries*.

COLUMNS 3–5 (LINE)

See *Common Entries*.

COLUMN 6 (FORM TYPE)

An L must appear in column 6.

COLUMNS 7–14 (FILENAME)

Use columns 7–14 to identify the output file to be written on the printer. The filename must begin in column 7.

Any filename entered in these columns must be a filename previously defined on the File Description sheet. The output device assigned to the file on the File Description sheet must be a printer.

COLUMNS 15–17 (LINE NUMBER–NUMBER OF LINES PER PAGE)

<i>Entry</i>	<i>Explanation</i>
1–112	Number of printing lines available is from 1–112.

Columns 15–17 specify the exact number of lines available on the form or page to be used. The entry must end in column 17. Leading zeros are not necessary.

COLUMNS 18–19 (FORM LENGTH)

<i>Entry</i>	<i>Explanation</i>
FL	Form length.

Columns 18–19 must contain the entry FL. This entry indicates that the preceding entry (columns 15–17) is the form length.

COLUMNS 20–22 (LINE NUMBER – OVERFLOW LINE)

<i>Entry</i>	<i>Explanation</i>
1–112	A line number from 1–112 is the overflow line.

Columns 20–22 specify the line number that is the overflow line. The entry must end in column 22. Leading zeros may be omitted.

When the line which you have 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 occur 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, you have to specify the overflow line high enough on the page to allow all these lines to print. You know the data you will be printing out after the overflow line is reached. Thus, you can judge what line should be the overflow line on this basis. See *Overflow Indicators* under *Supplementary Information* for more information.

COLUMNS 23–24 (OVERFLOW LINE)

<i>Entry</i>	<i>Explanation</i>
OL	Overflow line.

Columns 23–24 must contain the entry OL. This indicates that the preceding entry (columns 20–22) is the overflow line.

COLUMNS 25–74

Columns 25–74 are not used. Leave them blank.

COLUMNS 75–80 (PROGRAM IDENTIFICATION)

See *Common Entries*.

COLUMNS 1–2 (PAGE)

See *Common Entries*.

COLUMNS 3–5 (LINE)

See *Common Entries*.

COLUMN 6 (FORM TYPE)

An I must appear in column 6.

COLUMNS 7–14 (FILENAME)

Column 7–14 identify the input, update, or combined file you are describing. The filename must begin in column 7. Use the same filename given in the file description specifications. The filename must appear on the first line that contains information concerning the records in that file.

COLUMNS 14–16

Columns 14–16 may contain the characters AND, and columns 14–15 may contain the characters OR. These AND/OR lines are used to indicate a relationship between record identifying indicators or record types. For further information see *AND Relationship* and *OR Relationship* for columns 21–41 in this chapter.

COLUMNS 15–16 (SEQUENCE)

<i>Entry</i>	<i>Explanation</i>
Any two alphabetic characters.	Do not check for special sequence.
01–99	Check for special sequence.

Columns 15–16 may contain a numeric entry which assigns a special sequence to different record types in a file. If different types of records do not need to be in any special order, use two alphabetic characters. Within one file, all record types having alphabetic entries in columns 15–16 must be specified before those types with numeric entries. Columns 15–16 must contain an alphabetic entry for chained files.

Alphabetic Characters

If you do not wish to check for a special sequence of record types, enter any two alphabetic characters in columns 15–16 (see *Examples, Example 1*). Alphabetic characters must be used for chained files and look ahead records.

Numeric Characters (01–99)

Use columns 15–16 to assign sequence numbers to different types of records within a file. Your job may require that one record type (identified by a record identification code) must appear before another record type within a sequenced group. For instance, you may want a name record before an address record. You must provide a record identification code for each type of record and then number the record types in the order that they should appear. The program will check this order as the records are read. The first record type must have the lowest sequence number (01), the next record type should be given a higher number, etc. (See *Examples, Example 2*).

Numeric sequence numbers only ensure that all records of record type 01 precede all records of record type 02, etc., in any sequenced group. The sequence numbers do not ensure that records within a record type are in any certain order. Numeric sequence numbers have no relationship with control levels, nor do they provide for sequence checking of data in fields of a record (see *Examples, Example 3*).

Note: Numeric sequence is not allowed on demand files.

Gaps in sequence numbers are allowed, but the numbers used must be kept in ascending order. The first sequence number must be 01.

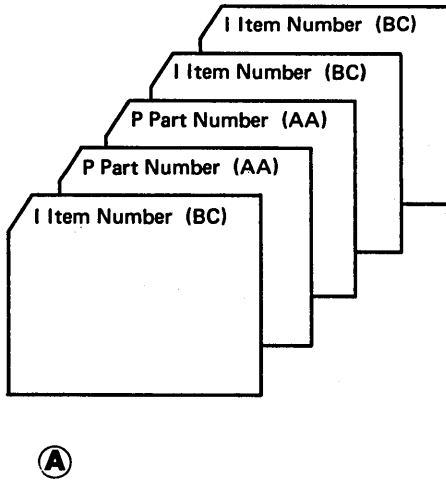
A record type out of sequence causes the program to stop. The program may be restarted by pressing the start key. The record that causes the halt is bypassed and the next record is read from the same file.

Records in an OR line cannot have a sequence entry in these columns. The entry in these columns from the previous line also applies to the card in the OR line. See *Columns 53–58* for information on OR relationships.

Examples

Example 1: Figure 46, insert A, shows a file having two types of records (part number and item number) which may appear in any order. Since they are not to be checked for

sequencing, they are assigned two alphabetic characters (AA and BC, respectively) instead of numbers. See Figure 46, insert B, for the coding of this example.



IBM International 8
RPG INPUT

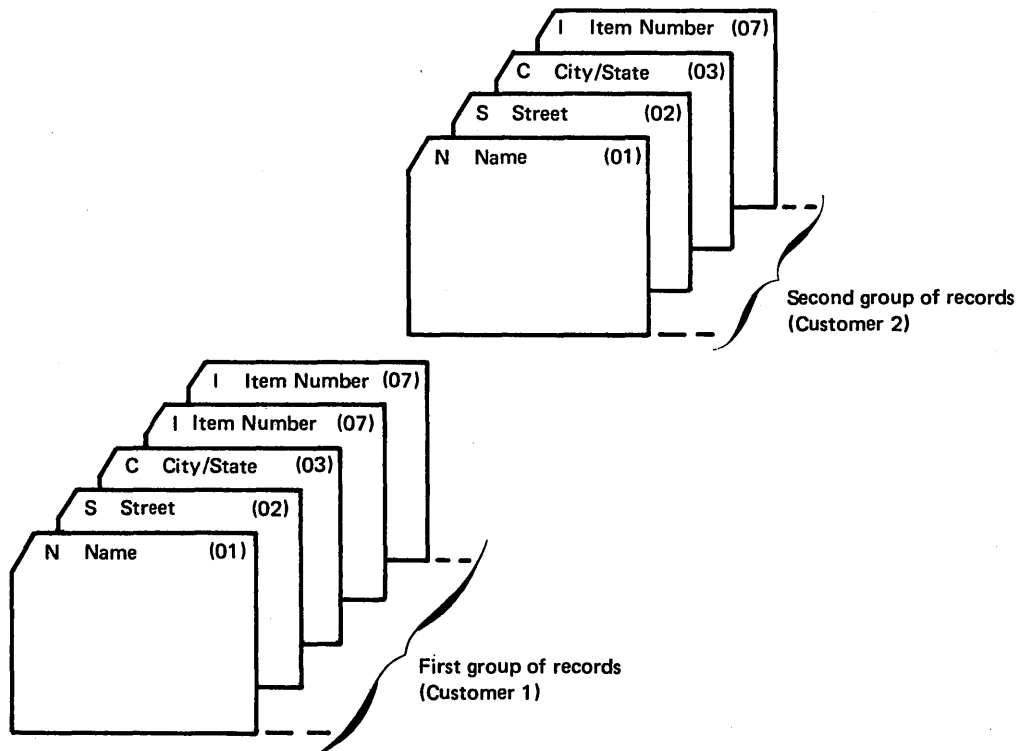
Date _____
 Program _____
 Programmer _____

				Punching Instruction		Graphic Punch																													
Line	Form Type	Filename	Sequence	Number (1-N) Option (O)	Record Identifying Indicator	Record Identification Codes																													
						1			2																										
						Position	Not (N) C/Z/D Character		Position	Not (N) C/Z/D Character	Posit																								
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		
0 1	I	CARDA	AA	01	I	CP																													
0 2	I																																		
0 3	I																																		
0 4	I																																		
0 5	I																																		
0 6	I																																		
0 7	I		BC	02	I	CI																													
0 8	I																																		
0 9	I																																		

Figure 46. Unsequenced Card Type in a File

Example 2: Figure 47, insert A, shows the order of four different types of records within a file. 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.) See Figure 47, insert B, for the coding of this example.



A

Line		Form Type	Filename	Sequence Number (1-N) Option (O)	Record Identifying Indicator or	Record Identification Codes									Field Location		Field Name	Control Level (L1-L9)	Matching Fields or Chaining Fields	Field Record Relation	Field Indicators			Sterling Sign Position
						1			2			3			From	To	Decimal Positions				Plus	Minus	Zero or Blank	
						Position	Not (N) C/Z/D Character		Position	Not (N) C/Z/D Character		Position	Not (N) C/Z/D Character											
01	I		CUST	011		1	CN																	
02	I			0210				1	CS					5	25		NAME							
03	I			031		1	CC							2	26		STREET							
04	I			041		1	CI							5	21		CTYST							
05	I			051		1	CS							10	16		ITEM							
06	I			061		1	CC							18	23		QTY							
07	I			0710		1	CI																	
08	I			081		1	CC																	
09	I			091		1	CC																	
10	I			101		1	CC																	

B

Figure 47. Sequence Checking of Record Types

Example 3: Figure 48 shows three groups of four different record types. 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 which you specify in columns 15–16 will not catch this mistake since the sequence entry does not cause the data on the record to be checked. See Figure 47, insert B, for the coding of this example.

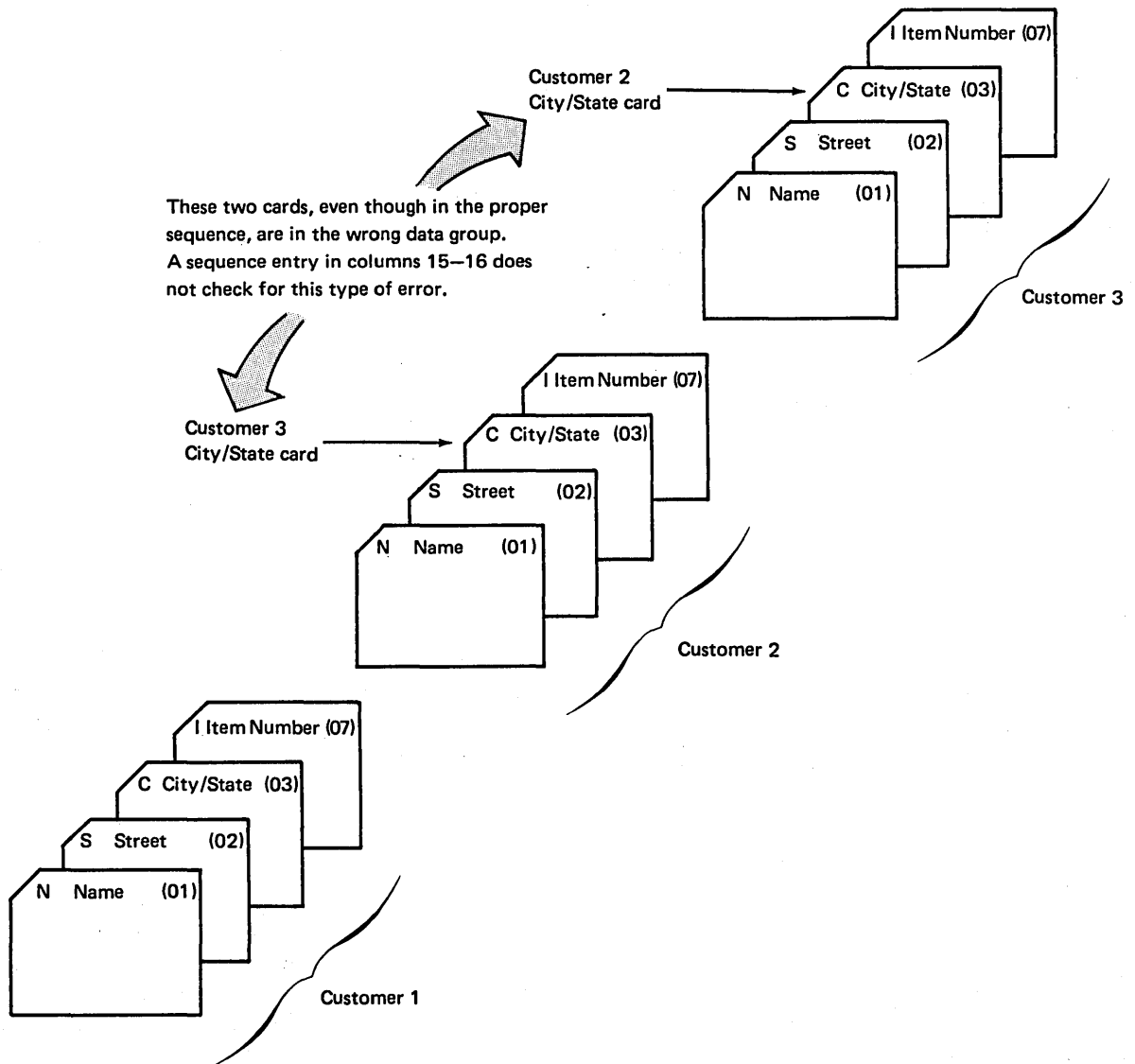


Figure 48. Correct Card Sequence (Incorrect Data Within Groups)

COLUMN 17 (NUMBER)

Entry	Explanation
Blank	Record types are not being sequence checked (columns 15–16 have alphabetic entries).
1	Only one record of this type is present in the sequenced group.
N	One or more records of this type may be present in the sequenced group.

Use column 17 only if sequence checking is to be done (columns 15–16 contain a numeric entry). Often, when sequence checking, you may have more than one record of a particular type within the sequenced group (see *Example*). Thus you must indicate by an entry in column 17 that a certain number of records of one type may be found in the sequence group.

OR lines (columns 14–15 have the letters OR) should not have an entry in this column. It is assumed that the number of records of this type to be found in the sequenced group is the same as the number entered in column 17 of the previous line. See *Columns 53–58* for more information on OR lines.

Example

Figure 49 shows a sequence record file in which there is more than one record per type in a group. The record type called item number appears three times.

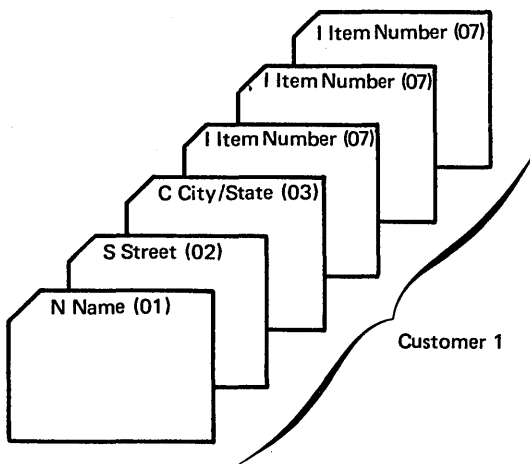


Figure 49. Sequenced Card File (More than one Record Per Type in a Group)

Refer to Figure 47, insert B, for the coding of this example. There is no reason for a name, street, or city/state record to appear more than once in one group. A 1 is entered in column 17 to indicate that these record types appear only once in each group. However, since one person may have purchased more than one item, there may be two or more item number records per group; an N is entered in column 17 for this field.

COLUMN 18 (OPTION)

Entry	Explanation
Blank	Record type must be present if sequence checking is specified.
O	Option. Record type may or may not be present if sequence checking is specified.

Column 18 is used only when record types are being sequence checked (columns 15–16 contain a numeric entry). A blank entry specifies that a record of this record type must be present in each sequenced group.

The O entry specifies that a record of this record type may or may not be present in each sequenced group (see *Example*). If all record types are optional, no sequence errors will be found.

OR lines should not have an entry in this column. The entry in this column on the previous line also applies to this record in the OR relationship. See *Columns 53–58* for more information on OR lines.

Example

Figure 50 shows a sequenced card file in which a card type may be optional. For instance, the street or item number records may not be included. Since it is not always necessary to have a street address, this record is optional. Suppose this job required a list of all items purchased during one month by the individual named in the name record. It is possible that a person might not buy anything during the month. In this case, there would be no item record; therefore, the item record would also be optional. See Figure 47, insert B, for a coding example.

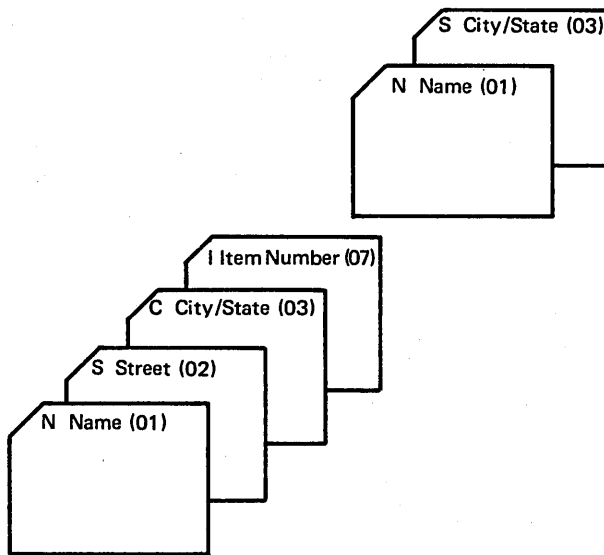


Figure 50. Sequenced Card File (Optional Record Types)

COLUMNS 19–20 (RECORD IDENTIFYING INDICATOR, **, TR)

<i>Entry</i>	<i>Explanation</i>
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.
TR	Spread cards.

Columns 19–20 may be used for three purposes:

1. To specify record identifying indicators.
2. To indicate a look ahead field.
3. To specify the trailer portion of spread cards.

Record Identifying Indicators

Use columns 19–20 to assign an indicator to each record type. When you have different types of records within a file, you often want to do different operations for each record type. Therefore, you must have some way of knowing which type of record has just been read. To do this, you assign different record identifying indicators to each record type. Whenever a record type is selected to be processed next, its corresponding identifying indicator is turned on. (All other record identifying indicators are off at this time unless chained files or demand files are being processed; in this case, several indicators may be on at the same time.) This indicator signals throughout the rest of the program cycle which record type has just been selected.

When a control level indicator used as a record identifying indicator is turned on to indicate the type of record read, only that one control level indicator is turned on. All lower control level indicators remain unchanged.

Because the record identifying indicator is on for the rest of the program cycle, you may use it to condition calculation operations (see *Columns 9–17* under *Calculation Specifications*) and output operations (see *Columns 23–31* under *Output-Format Specifications*).

Record identifying indicators do not have to be assigned in any order.

You may assign the same indicator to two or more different record types provided you want the same operations performed on these types. Do this by using the OR relationship (see *Columns 21–41*).

No record identifying indicator may be specified in the AND line of an AND relationship. Record identifying indicators for OR lines may be specified for every record type in the OR relationship that requires special processing. See *Columns 21–41* for information on AND lines. See *Columns 53–58* for information on OR lines.

If the RPG II Telecommunications feature is used, a description of the permanent error indicators can be found in the *IBM System/3 RPG II Telecommunications Programming Reference Manual, SC21-7507*.

Look Ahead Fields

Use asterisks in columns 19–20 to indicate that fields named in columns 53–58 in the following specifications are look ahead fields. Look ahead fields apply only to disk records and punched cards. A look ahead field allows you to look at information in a field on the next record that is available for processing in any input, update, or combined file. Because of this capability you are able to use the information from the look ahead field to determine what operation should be done next.

Through the use of a look ahead field, you are able to:

1. Determine when you are processing the last record of a control group.
2. Extend the RPG II matching record capability.

See *Look Ahead* under *Supplementary Information* for information on when and how to use look ahead fields.

Spread Cards

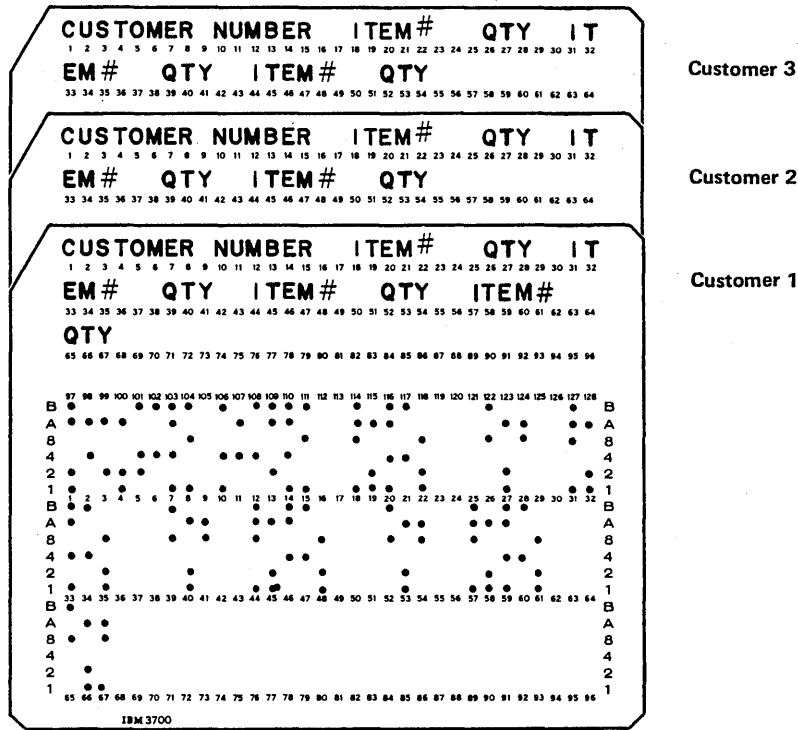
Certain jobs require you to keep data files which contain a header card and a separate card for each item or transaction being recorded. Thus, for a billing job you may have a data file with cards like these for each customer:

ITEM#	QTY	
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		
ITEM#	QTY	
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		
ITEM#	QTY	
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		
CUSTOMER NUMBER		
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		
ITEM#	QTY	
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		
ITEM#	QTY	
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		
CUSTOMER NUMBER		
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		
ITEM#	QTY	
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		
ITEM#	QTY	
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		
CUSTOMER NUMBER		
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 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96		
B 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128		
A		A
S		S
4		4
2		2
1		1
B 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		B
A		A
8		8
4		4
2		2
1		1
B 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		B
A		A
8		8
4		4
2		2
1		1
B 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		B
A		A
8		8
4		4
2		2
1		1

Header cards and transaction cards

You can store more data on each card and, in turn greatly reduce the amount of time required to process such a file by using the spread card capability of RPG II. Instead of using a header card and a separate card for each item or transaction, you can specify a spread card with a header portion followed by trailer portions which contain the item or transaction data. A trailer portion can consist of as many fields as are necessary; however, the same fields must appear in each trailer portion. A trailer portion must not be split between two records.

Thus a data file for a billing job like the one shown previously may have spread cards like these:



Spread cards containing header and trailer portions. CUSTOMER NUMBER is the header portion; each set of ITEM# and QTY fields is a trailer portion.

Specifications

Spread cards can only be specified for input card files designated as primary or secondary. No look ahead fields can be described for spread cards. You can describe a maximum of 255 valid TR lines (TR in columns 19–20) in a program.

Specify spread cards as follows:

1. Describe the fields in the header portion of the spread card on separate specification lines immediately following the proper file and record type entries. The header portion is all positions up to the first trailer in the record. Any record identification codes specified for the header/trailer record must have the position entry contained within the header portion of the record. If a numeric entry is made in columns 15–16 of the specification line containing the file and record type entries, an N must be entered in column 17 of the same line.

Describe fields in the header portion as you would any normal RPG II field. You are required to describe only those fields in the header portion that are used later in the program. If no field in the header portion is used, you can omit the header field specification and specify the TR line described below immediately following the file and record type entries.

2. Enter TR in columns 19–20 of a specification line to indicate that the fields in the first trailer portion are described in the specification lines that follow. Leave columns 7–18 and 21–74 of the TR line blank.
3. Describe the fields in the first trailer portion on separate lines immediately following the TR line. Leave columns 7–43, 59–62, and 71–74 of the trailer specifications blank. Describe the fields in the trailer portion as you would any normal RPG II field.

You are required to describe only those fields in the first trailer portion that are used later in the program. Be sure, however, that you describe the fields that indicate the start and end position of the first trailer portion.

Since all trailer portions must be the same length and must include the same fields, you need only describe the first one. The compiler uses this trailer specification to calculate how many trailer portions the record contains and to determine the start and end portion of each.

Processing Spread Cards

The following considerations apply when processing spread cards:

1. One trailer portion from a spread card is processed per program cycle. The system treats that trailer portion, along with its associated header portion, as one logical record.
2. The next spread card is read when:
 - the system has processed all trailer portions in the current record.
 - the system encounters a trailer portion in the card being processed which is entirely blank.

Example: Figure 51 shows the input specifications needed to describe a file (CARDIN) containing spread cards. Each card in the file contains a header portion in positions 3–8 which is made up of the customer number field (CUSTNO). The header is followed by a number of trailer portions. Each trailer is made up of an item number field (ITMNO), a field indicating the size of the item (SIZE), and a quantity field (QTY).

C/Z/D

Entry	Explanation
C	Entire character.
Z	Zone portion of character.
D	Digit portion of character.

Use columns 26, 33, and 40 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) may be used (see *Examples, Example 3 and Example 4*). 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 *Character, Character Grouping by Zone or Digit* in the following text.

Character

Use any alphabetic character, special character, or digit in columns 27, 34, and 41 to identify the character that was used in the record to serve as the code or part of the code.

Character Grouping by Zone or Digit

When selecting characters for record identification purposes on a digit or zone only basis, it must be understood that all characters having the same zone or digit will be 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. It is the 8-bit code that is tested to see if the character meets the requirements of the record identifying character on the input specifications. Figure 52 lists the character grouping for zone or digit only in the Character/Zone/Digit columns (26, 33, or 40) and Character columns (27, 34, or 41) of the input specifications.

As an example, a digit only entry in C/Z/D and an A in character would cause all records having a / (slash), A, J, or 1 in the specified column to be selected.

Using the same letter A but now selecting records on a zone only basis, & and A-I meet the requirements and are selected.

Character Grouping by Zone (Z)		Character Grouping by Digit (D)	
ϕ . < (+ 	& A B C D E F G H I	blank & - (minus)]] 0	H Q Y 8
/ A J 1	- (minus)) J K L M N O P Q R	B K S 2	ϕ ↑ :
↑ \$ *) :]	S T U V W X Y Z	C L T 3	. \$, #
/ (comma) % (underscore) > ?	blank 0 1 2 3 4 5 6 7 8 9	D M U 4	< * % @
: # @ (apostrophe) = ≠	E N V 5	E N V 5	() (underscore) (apostrophe)
	F O W 6	F O W 6	+ ; > =
	G P X 7	G P X 7]] ? ≠

Figure 52. Characters Interpreted as Having the Same Zone or Digit

Negative Numbers

Negative numbers have a different character structure than positive numbers because negative numbers are formed by punching a minus sign with the number on the data recorder. Numbers 0–9 have only digit portions. A minus sign is a B zone punch. Thus, when the zone (minus sign) and the digit (0–9) are punched together, a letter is formed. Therefore, negative numbers are represented by characters J–R, which appear in the computer as hexadecimal D1–D9, respectively.

AND Relationship

A maximum of three identifying characters may be described in one specification line. Thus, if the identification code consists of more than three characters, an AND line must be used. This means that the first three identifying characters are described in the first line. The additional identifying characters are described in as many following lines as are needed. Write the word AND in columns 14–16 to indicate an AND line (see *Examples, Example 1*).

A maximum of twenty AND lines may 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 will turn on.

OR Relationship

A particular record type may be identified by two different codes. If this is the case, OR lines must be used to indicate that either one of the codes may be present to identify the record. A maximum of twenty OR lines may appear for each record sequence if no AND lines are used. Write the word OR in columns 14–15 to indicate an OR line (see *Examples, Example 2*).

Note: If AND lines and OR lines are combined, the total number of AND and OR lines for one record sequence cannot exceed twenty.

Examples

Example 1: Figure 53, insert A, shows a record identification code consisting of five characters. The first character is located in position 1, the other four in positions 93, 94, 95, and 96. Since only three identifying characters may be described on one line, the word AND must be used on the next line to indicate that the last two characters of the code are part of the preceding record identification entries.

assigned resulting indicator 12 can be identified by two different codes. The record can be identified by a code consisting of a 5 in position 1 and a 6 in position 2 or a code consisting of a 6 in position 1.

Example 3: In Figure 53, insert A, the entry in column 32 indicates that the digit 9 must not be present in position 93 for record type 12.

Example 2: Figure 53, insert B, shows the use of an OR line to describe record type identification codes. The record

Example 4: Figure 53, insert A, shows that only the zone portion of the character T located in position 94 is part of the identifying code. In position 96 only the digit portion of the character E is part of the code.

IBM International Business Machines Corporation Form X21-9094 Printed in U.S.A.

RPG INPUT SPECIFICATIONS

Date _____ Program _____ Programmer _____

Page 1 2 Program Identification 75 76 77 78 79 80

Line	Form Type	Filename	Sequence	Number (1-1)	Option (O)	Record Identifying Indicator	Record Identification Codes									Field Location		Field Name	Control Level (1-1-L9)	Matching Fields or Chaining Fields	Field Record Relation	Field Indicators			Sterling Sign Position								
							1			2			3			From	To					Plus	Minus	Zero or Blank									
							Position	Not (N)	C/Z/D Character	Position	Not (N)	C/Z/D Character	Position	Not (N)	C/Z/D Character	Decimal Positions	Field Name					Control Level (1-1-L9)	Plus	Minus		Zero or Blank							
0 1	1	PAYROLL BC				12	1	C5		93	N	9	94	Z	T																		
0 2	1							95	C9		96	D	E																				

A

IBM International Business Machines Corporation Form X21-9094 Printed in U.S.A.

RPG INPUT SPECIFICATIONS

Date _____ Program _____ Programmer _____

Page 1 2 Program Identification 75 76 77 78 79 80

Line	Form Type	Filename	Sequence	Number (1-1)	Option (O)	Record Identifying Indicator	Record Identification Codes									Field Location		Field Name	Control Level (1-1-L9)	Matching Fields or Chaining Fields	Field Record Relation	Field Indicators			Sterling Sign Position							
							1			2			3			From	To					Plus	Minus	Zero or Blank								
							Position	Not (N)	C/Z/D Character	Position	Not (N)	C/Z/D Character	Position	Not (N)	C/Z/D Character	Decimal Positions	Field Name					Control Level (1-1-L9)	Plus	Minus		Zero or Blank						
0 1	1	PAYROLL BC				12	1	C5		2	C6																					
0 2	1							1	C6																							

B

Figure 53. Record Identification Codes

COLUMN 42

Column 42 is not used. Leave it blank.

Note: Column 43 begins the field description entries (columns 43–74) which must begin one line below the file and record identification entries (columns 7–42) for each file.

COLUMN 43 (PACKED OR BINARY FIELD)

Entry	Explanation
Blank	Field is in unpacked decimal format or is alphameric.
P	Field is in packed decimal format on disk.
B	Field is in binary format on disk.

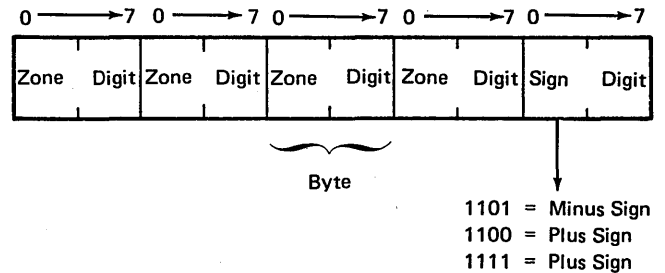
Column 43 is used to indicate that a numeric field is in packed decimal or binary format. Packed decimal or binary fields can only be read from or written on disk. Numeric data fields in packed decimal or binary format must be converted to the unpacked decimal format before they can be processed by the system. This conversion ignores decimal points.

Column 43 must contain a P if the input field named in columns 53–58 is in packed decimal format. Column 43 must contain a B if the input field named in columns 53–58 is in binary format.

Any array which was read in packed or binary format should have an entry in column 43 of the Input sheet. In this case, the From and To columns in the Input sheet should define the positions the array occupies in the record in the packed or binary format. The unpacked decimal length of each array element is defined on the Extension sheet.

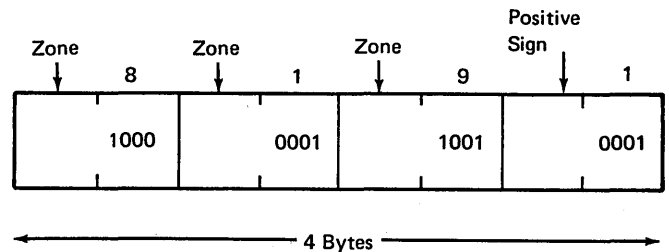
Unpacked Decimal Format (Blank)

Unpacked decimal format means that each byte of storage, whether on disk or in the computer, can contain one character. That character may be a decimal number or it may be an alphabetic or special character. In the unpacked decimal format, each byte of storage is divided into a four-bit zone portion and a four-bit digit portion. The unpacked 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 is used to indicate whether the decimal number is positive or negative. In unpacked decimal format, the zone portion is included for each digit in a decimal number; however, only the low-order zone portion serves as the sign. The decimal number 8,191 looks like this in unpacked decimal format:

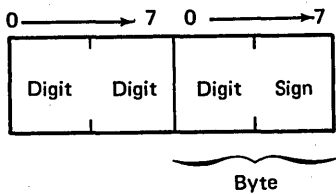


Once data has been read into the computer, it must be represented in the unpacked decimal format before it can be processed. Thus, it is perfectly correct to store data on disk and read it into the computer in the unpacked decimal format. This eliminates the need to convert the input field since it is already in the required format. However, if you desire more efficient use of disk storage space, you may store numeric data (decimal numbers) on disk in either the packed decimal or the binary format. Only numeric fields can be represented in the packed decimal or binary format.

Packed Decimal Format (P)

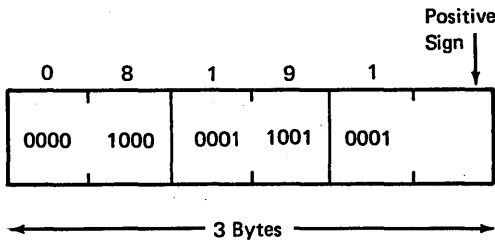
Packed decimal format means that a byte of disk storage (except for the low-order byte) can contain two decimal numbers. Since many of the fields in your disk files are made up of decimal numbers, you can conserve disk space by storing these fields in the packed decimal format. This format allows you to get almost twice as much data into a byte as you can using the unpacked 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:

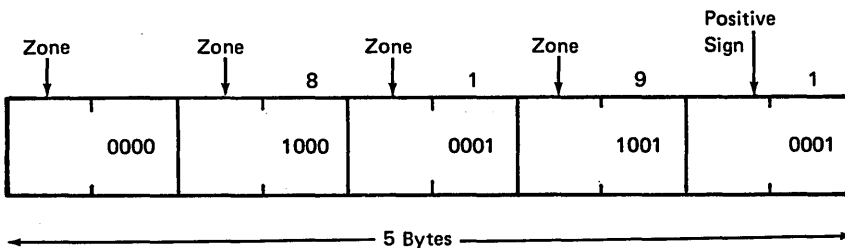


The sign portion of the low-order byte is used to indicate whether the numeric value represented in the digit portions is positive or negative. In the packed decimal format, the sign is included for each decimal number; the zone portion is not given for each digit in the number. Compare how the decimal number 8,191 is represented in packed decimal format with its unpacked representation shown before (Figure 54).

Packed Decimal Format:



Unpacked Decimal Format:



Since data must be represented in unpacked decimal format once it is inside the computer, you must give the RPG II program an indication when input fields are in another format. A P entry 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 unpacked format.

Packed key fields can be up to 8 bytes long. The following chart shows the packed equivalents for unpacked fields up to 15 bytes long:

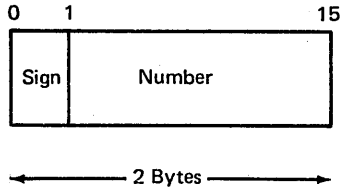
Unpacked Length in Bytes	Packed Length in Bytes
15 14	8
13 12	7
11 10	6
9 8	5
7 6	4
5 4	3
3 2	2
1	1

Figure 54. Packed Decimal and Unpacked Decimal Representation of 8,191

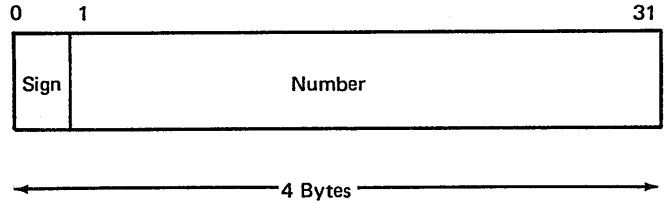
Binary Format (B)

Binary format means that two bytes of disk storage can contain up to four decimal numbers, and that four bytes of disk storage can contain up to nine decimal numbers. 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 two or four bytes in length.

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 two bytes of disk storage. For each 2-byte binary field stored on disk, the system automatically sets aside four bytes of core 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 four bytes of disk storage. For each 4-byte binary field stored on disk, the system automatically sets aside nine bytes of core storage to accommodate the field when it is unpacked. A 4-byte field in binary format looks like this:

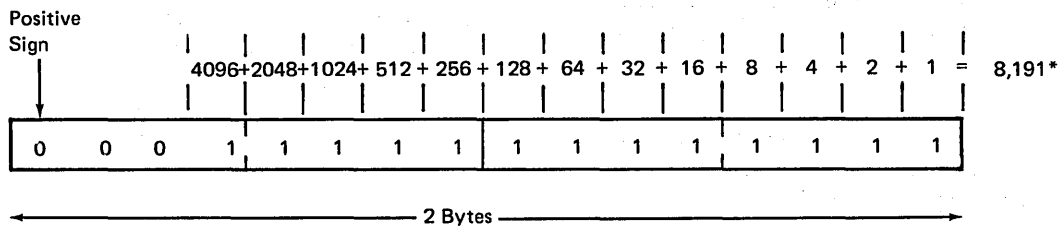


In each case, the sign portion of the high-order byte is used to indicate whether the numeric value is positive (sign bit off) or negative (sign bit on). Notice that, in the binary format, the zone position of the decimal number is not given. Compare how the decimal number 8,191 is represented in binary format with packed and unpacked representation (Figure 55).

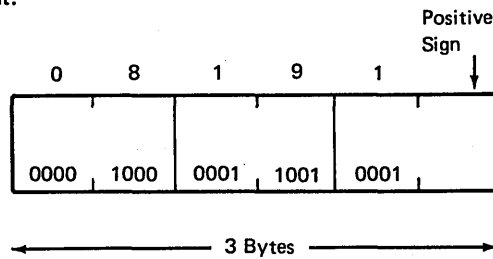
Since data must be represented in unpacked decimal format once it is inside the computer, you must give the RPG II program an indication when input fields are in another format. A B entry in column 43 indicates that the input field is in the binary format and that the system must convert this field to the required unpacked format.

Note: Although packed and binary fields require less disk storage space, the conversion routines needed to handle such data will increase the object program size.

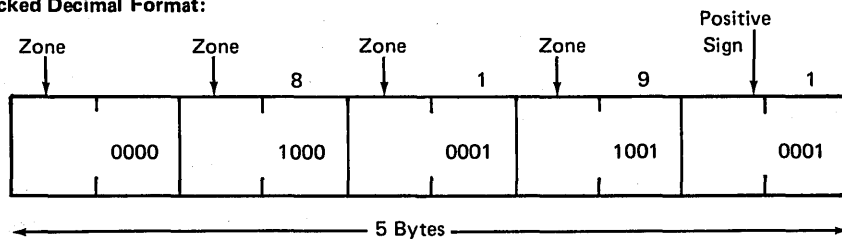
Binary Format:



Packed Decimal Format:



Unpacked Decimal Format:



* The numeric value of a positive binary number is obtained by adding the value of the bits that are on (1). The sign bit is not included. The numeric value of a negative binary number is obtained by adding the values of the bits that are off (0), plus one. The sign bit is not included.

Figure 55. Binary, Packed, and Unpacked Representation of 8,191

COLUMNS 44–51 (FIELD LOCATION)

<i>Entry</i>	<i>Explanation</i>
1–4096	Beginning of a field (From) or end of a field (To).

Use columns 44–51 (From and To) to describe the location on the record of each field containing input data named in columns 53–58 (Field Name). Enter the number of the record position in which the field begins in columns 44–47. Enter the number of the record position in which the field ends in columns 48–51.

A single-position field is defined by putting the same number in both From (columns 44–47) and To (columns 48–51). If a field of more than one position is defined, the number entered in columns 44–47 must be smaller than the number entered in columns 48–51.

The maximum field length for an unpacked numeric field is 15 positions (8 if the field is packed, 4 if it is binary). The maximum field length for an alphameric field is 256 characters. The maximum length for a ledger card control field is 9.

Entries in these columns must end in columns 47 and 51. Leading zeros may be omitted.

Note: From one to nine positions are allowed for a ledger card control number. Therefore, a From or To entry for a ledger card control field cannot exceed 9.

COLUMN 52 (DECIMAL POSITIONS)

<i>Entry</i>	<i>Explanation</i>
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 53–58. Column 52 must always have an entry when the field named in columns 53–58 is numeric. If you wish 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)

<i>Entry</i>	<i>Explanation</i>
1–6 alphameric characters	Field name, array name, or array element.
PAGE, PAGE1, PAGE2	Special words.

Use columns 53–58 to name a field, array, or array element found on your input records. If you are referencing an array, additional entries may be needed in these columns (see *Arrays* under *Supplementary Information*). Use this name throughout the program whenever you refer to this field. You must indicate the names of the fields for all types of records. However, you should name only the fields that you will use.

Field Names

A field name can be from 1–6 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 may not 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 may 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 found in different locations in each record type. Duplicate field names should not be used if matching fields are specified in your program.

Numeric fields may have a maximum length of 15 characters. Alphameric fields may have a maximum length of 256 characters. However, fields which are read in from a card are limited to the length of one punched card.

Fields that are used in arithmetic operations (see *Operation Codes* under *Supplementary Information*) or fields that are edited or zero suppressed (see *Column 38* and *Columns 45–70* under *Output-Format Specifications*) must be defined as numeric. This means that column 52 must have a decimal position entry.

A separate line is used for each field description.

Example

Figure 57 shows how the use of OR lines can save duplicate coding. The two different record types (one identified by a 5 in column 1, the other by a 6 in column 1) both have identical fields which must be described. Figure 57, insert

A, shows one way of doing this. Figure 57, insert B, shows the use of OR lines to do the same thing with less coding. The coding in Figure 57, insert B, says that all four fields can be found on either the record type identified by the 5 in column 1 or the record type with a 6 in column 1.

IBM International Business Machines Corporation Form X21-9094 Printed in U.S.A.

RPG INPUT SPECIFICATIONS

Date _____ Program _____ Programmer _____

Punching Instruction	Graphic					
	Punch					

Page

1	2
---	---

 Program Identification

75	76	77	78	79	80
----	----	----	----	----	----

Line	Form Type	Filename	Sequence Number (1-N) Option (O)	Record Identifying Indicator	Record Identification Codes									Field Location		Field Name	Control Level (L1-L9)	Matching Fields or Chaining Fields	Field Record Relation	Field Indicators			Sterling Sign Position	
					1			2			3			From	To					Decimal Positions	Plus	Minus		Zero or Blank
					Position	Not (N)	C/Z/D	Position	Not (N)	C/Z/D	Position	Not (N)	C/Z/D											
01	I	SALES	AA	14	1	C5																		
02	I												5	8	DEPT									
03	I												9	14	EMPNO									
04	I												46	500	ITEM									
05	I												66	700	COST									
06	I		BB	15	1	C6																		
07	I												5	8	DEPT									
08	I												9	14	EMPNO									
09	I												46	500	ITEM									
10	I												66	700	COST									



IBM International Business Machines Corporation Form X21-9094 Printed in U.S.A.

RPG INPUT SPECIFICATIONS

Date _____ Program _____ Programmer _____

Punching Instruction	Graphic					
	Punch					

Page

1	2
---	---

 Program Identification

75	76	77	78	79	80
----	----	----	----	----	----

Line	Form Type	Filename	Sequence Number (1-N) Option (O)	Record Identifying Indicator	Record Identification Codes									Field Location		Field Name	Control Level (L1-L9)	Matching Fields or Chaining Fields	Field Record Relation	Field Indicators			Sterling Sign Position	
					1			2			3			From	To					Decimal Positions	Plus	Minus		Zero or Blank
					Position	Not (N)	C/Z/D	Position	Not (N)	C/Z/D	Position	Not (N)	C/Z/D											
01	I	SALES	AA	14	1	C5																		
02	I		OR	15	1	C6																		
03	I												5	8	DEPT									
04	I												9	14	EMPNO									
05	I												46	500	ITEM									
06	I												66	700	COST									



Figure 57. Record Types with Identical Fields

COLUMNS 59–60 (CONTROL LEVEL)

Entry *Explanation*

L1–L9 Any control level indicator.

Use columns 59–60 to assign control level indicators to input fields. Control level indicators may not be associated with a chained file or a demand file. Control level indicators are used to specify the point at which specified operations are to be performed. You may assign a control level indicator to any field; this field is then known as a control field. It is checked for a change in information. When information in the control field 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 under Calculation Specifications* or *Columns 23–31 under Output-Format Specifications*).

There are nine different control levels (L1–L9). When a certain 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 others should determine how you assign indicators. For example, the type of data which demands a subtotal has a lower control level indicator than data which needs a final total. A field containing department numbers is given a higher control level indicator than a field containing employee numbers (see *Examples, Example 1*).

Control level indicator L0, since it is always on, cannot be assigned to a control field. Nevertheless, you may use it to condition operations (see *Columns 7–8 under Calculation Specifications*).

Normally, control level indicators are used to:

1. Condition certain total calculations to be performed when the information in the control field changes.
2. Condition certain total output operations to be done after totals have been accumulated for one control group.
3. 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:

1. 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 *Examples, Example 2*.
2. In the same record type, record positions in control fields assigned different control level indicators may overlap (Figure 58). However, the total number of positions assigned as control fields must not be greater than 144. In Figure 58 for example, a total of 15 positions is assigned to control levels.

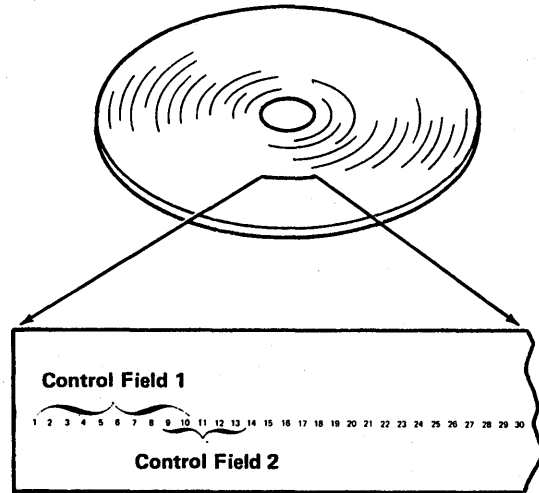


Figure 58. Overlapping Control Fields in a Disk Record

3. Field names are ignored in control level operations. Therefore, fields from different record types which have been assigned the same control level indicator may have the same name.
4. Control levels need not be written in any sequence. L2 entry can appear before L1. Also, there may be gaps in the control levels assigned.
5. When numeric control fields with decimal positions are compared to see if 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.
6. If a field is specified as numeric, only the digit portion is used to determine if a control break has occurred. This means that a field is always considered to be positive. A -5 is considered equal to a +5.
7. 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). This means that when numeric control fields are compared to see if the information has changed, only the digit portion of each character is compared.
8. Control fields are initialized to hexadecimal zeros.
9. A control break is highly probable after the first record containing a control field is read. The control fields in this record are compared to an area in storage which contains hexadecimal zeros. Since fields from two different records are not being compared, total calculations and total output operations are bypassed for this cycle. A control break does occur then, but it is not considered to be a true control break.
10. Total calculations and total output operations are bypassed until the first cycle following a cycle involving a record with control fields specified.
11. If different record types in a file do not have the same number of control fields, unwanted control breaks may occur. See *Examples, Example 3* for a method of avoiding unwanted control breaks.
12. A control field cannot be specified as binary (B 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 in the input specifications and treated as one control field (see *Examples, Example 4*). Some special rules for split control fields are:

1. For one control level indicator, a field may 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.
2. The length of the portions of a split control field may vary for different record types if the field names are different. However, the total length of the portions must always be the same.
3. No other specification lines may come between lines which describe split control fields.
4. If one section of a split control field is numeric, the whole field is considered numeric.
5. A numeric split control field may 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.
6. A split control field cannot be made up of a packed decimal field and an unpacked decimal field. Both portions of the control field must be packed or both must be unpacked.

Note: Additional rules applying to control level indicators when used with indicators in the Field Record Relation columns are discussed in *Columns 63–64*.

Date _____
 Program _____
 Programmer _____

Punching Instruction	Graphic						
	Punch						

Line	Form Type	Filename	Sequence Number (L-N)	Option (O)	Record Identifying Indicator	Record Identification Codes									Field Location		Field Name	Control Level (L1-L9)	Matching Fields or Chaining Fields	Field Record Relation	Field Indicators			Sterling Sign Position													
						1			2			3			From	To					Plus	Minus	Zero or Blank														
						Position	Not (N) C/Z/D	Character	Position	Not (N) C/Z/D	Character	Position	Not (N) C/Z/D	Character	Stacked Select	P = Packed/B = Binary					Decimal Positions																
01	I	EMPLREP	AB		10					1	CA																										
02	I													5	10	EMPLNO	L1																				
03	I													11	30	NAME	L1																				
04	I													80	80	DIVSON	L3																				
05	I													33	33	SHIFT	L3																				
06	I													2	5	DEPT	L2																				
07	I		CD	20		1	CA																														
08	I													5	10	EMPLNO	L1																				
09	I													11	14	DEPT	L2																				
10	I													60	70	HRSWKD																					
11	I													80	80	DIVSON	L3																				
12	I																																				

Figure 59. Control Level Indicators (Two Record Types)

Examples

Example 1: Figure 59 shows the assignment of three indicators. 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 corporation; L2 is assigned to it. The employee field has the lowest control level indicator (L1) assigned.

Example 2: Figure 59 shows that the same control level indicators may 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 columns in length, DEPT is 4, and DIVSON is 1.

Example 3: Different record types normally contain the same number of control fields. However, some applications require a different number of control fields in some records. This is shown in Figure 60, insert A. The salesman records contain only the L2 control field. The item records contain both L2 and L1 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 (see Figure 60, insert B).

Figure 60, inserts C and D, contains excerpts from a program that processes the input shown in Figure 60, insert A, and prevents the unwanted control break from occurring. The corrected output produced is shown in Figure 60, insert B.

Line 01 of the Calculation 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-Format sheet). Detail calculations are then processed for the item record and line 02 of the Calculation sheet sets indicator 11 off. This allows the normal L1 control break to occur.

A	(L2)	
	Salesman Number	Salesman Name
	2 — 3	4 — 16

Salesman Record

B	(L2)	(L1)	
	Salesman Number	Item Number	Amount
	2 — 3	4 — 6	7 — 9

Item Record

Ⓐ

01	DICK LOVE				
			*	←	Unwanted Control Break
	100	3			
	100	2			
		5	*		
	101	4			
		4	*		
		9	**		
02	CAL WINBUSH				
			*	←	Unwanted Control Break
	100	6			
	100	2			
		8	*		
	101	3			
		3	*		
		11	**		
		20			

Output Showing Unwanted Control Level Break

01	DICK LOVE			
	100	3		
	100	2		
		5	*	
	101	4		
		4	*	
		9	**	
02	CAL WINBUSH			
	100	6		
	100	2		
		8	*	
	101	3		
		3	*	
		11	**	
		20		

Corrected Output

Ⓑ

Figure 60. Unwanted Control Breaks (Part 1 of 3)

IBM

International Business Machines Corporation

Form X21-9094
Printed in U.S.A.

RPG INPUT SPECIFICATIONS

Date _____

Program _____

Programmer _____

Punching Instruction	Graphic						
	Punch						

Page	1	2
------	---	---

Program Identification	75	76	77	78	79	80
------------------------	----	----	----	----	----	----

Line	Form Type	Filename	Sequence Number (1-N)	Option (O)	Record Identifying Indicator	Record Identification Codes									Field Location				Field Name	Control Level (L1-L9)	Matching Fields or Chaining Fields	Field Record Relation	Field Indicators			Sterling Sign Position
						1			2			3			From	To	Decimal Positions	Plus					Minus	Zero or Blank		
						Position	Not (N) C/Z/D Character	Character	Position	Not (N) C/Z/D Character	Character	Position	Not (N) C/Z/D Character	Character												
01	I	SALES	AA	Ø1	1	CA																				
02	I												2	3	L2FLD	L2										
03	I												4	16	NAME											
04	I		BB	Ø2	1	CB																				
05	I												2	3	L2FLD	L2										
06	I												4	6	L1FLD	L1										
07	I												7	9Ø	AMT											
08	I																									

IBM

International Business Machines Corporation

Form X21-9093
Printed in U.S.A.

RPG CALCULATION SPECIFICATIONS

Date _____

Program _____

Programmer _____

Punching Instruction	Graphic						
	Punch						

Page	1	2
------	---	---

Program Identification	75	76	77	78	79	80
------------------------	----	----	----	----	----	----

Line	Form Type	Control Level (L1-L9, SR)	Indicators			Factor 1	Operation	Factor 2	Result Field	Field Length	Decimal Positions	Half Adjust (H)	Resulting Indicators			Comments	
			And	And	Not								Arithmetic	Plus	Minus		Zero
			Not	Not	Not								High 1 > 2	Low 1 < 2	Equal 1 = 2		
01	C		Ø1				SETOM								11		
02	C		Ø2				SETOP								11		
03	C		Ø2		AMT		ADD	L1TOT							5Ø		
04	C	L1			L1TOT		ADD	L2TOT							5Ø		
05	C	L2			L2TOT		ADD	LRTOT							5Ø		
06	C																
07	C																

©

Figure 60- Unwanted Control Breaks (Part 2 of 3)

IBM

International Business Machines Corporation

Form X21-8094
Printed in U.S.A.

RPG INPUT SPECIFICATIONS

Date _____
 Program _____
 Programmer _____

Punching Instruction	Graphic								
	Punch								

Page

1	2
---	---

 Program Identification

75	76	77	78	79	80
----	----	----	----	----	----

Line	Form Type	Filename	Sequence	Number (1-N)	Option (O)	Record Identifying Indicator or ***	Record Identification Codes												Field Location		Field Name	Control Level (L1-L9)	Matching Fields or Chaining Fields	Field Record Relation	Field Indicators			Sterling Sign Position													
							1			2			3			From	To	Decimal Positions	Plus	Minus					Zero or Blank																
							Position	Not (N)	C/Z/D	Character	Position	Not (N)	C/Z/D	Character	Position	Not (N)	C/Z/D									Character	Stacker Select		P = Packed/B = Binary												
0 1	I	MASTER	AA	01																																					
0 2	I																	28	31		CUSNO	L4																			
0 3	I																	15	20		ACCTNO	L4																			
0 4	I																	50	52		REGNO	L4																			

All portions of a split control field must be assigned the same control level indicator.

Figure 61. Split Control Fields

Example 4: Figure 61 shows a split control field made up of three portions. The control level indicator (L4) which is used for all three portions indicates that they are all to be treated as one control field. The field can be pictured as follows:

```

CUSNO      ACCTNO  REGNO
1 6 7 9   8 6 5 3 9 7   1 1 1
```

The first field assigned the same control level indicator begins the control field; the last ends it.

COLUMNS 61-62 (MATCHING FIELDS)

Entry	Explanation
M1-M9	Any matching level.

Use columns 61-62 to specify matching fields and sequence checking. Matching fields and sequence checking cannot be specified for chained files or demand files.

An entry in columns 61-62 indicates:

- Matching fields and sequence checking when you have two or more input, update, or combined files with match fields.

- Sequence checking only when you have just one input, update, or combined file.

Matching Fields

Specify matching fields when you wish to compare records from two or more input, update, or combined files in order to determine when records match. Records can be matched by matching one field, many fields, or entire records. You can indicate as many as nine matching fields (M1-M9). Whenever the contents of the match fields from records of the primary file are the same as the contents of the match fields from a secondary file, the matching record (MR) indicator turns on. M1-M9 are used only to identify fields by which records are matched. The values M1-M9 are not indicators but do cause MR to turn on when a match occurs. You can then use the MR indicator to condition those operations which are to be done only when records match (see *Columns 9-17 under Calculation Specifications* and *Columns 23-31 under Output-Format Specifications*).

Assigning Matching Field Values (M1–M9)

The following considerations apply to assigning matching field values:

1. Sequence checking is automatically done for all record types with matching field specifications. The contents of the fields to which M1–M9 have been assigned are checked for correct sequence. An error in sequence stops the program. The record which caused the halt is not processed. When the machine is restarted, the next record from the same file is read. Thus, all matching fields must be in the same order, either all ascending or all descending (see *Column 18* under *File Description Specifications*).
2. Not all files used in the job must have matching fields. Not all record types within one file must have matching fields either. However, at least one record type from two files must have matching fields if files are ever to be matched.
3. The same number of matching fields must be specified for all record types which are used in matching. The same matching record values must also be used for all types (see *Examples, Example 1*).
4. All match fields given the same matching record value (M1–M9) must be the same length and type (alphameric or numeric).
5. Record positions of different matching fields may overlap, but the total length of all fields must not exceed 144 characters.
6. If more than one matching field is specified for a record type, all the fields are combined and treated as one continuous matching field (see *Examples, Example 2*). They are combined according to descending sequence (M9–M1) of matching record values.
7. Matching fields may not be split. This means that the same matching field value cannot be used twice for one type of record.
8. Matching fields may be either alphameric or numeric. However, all matching fields given the same matching record value (M1–M9) are considered numeric if any one of those matching fields is described as numeric.
9. 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.
10. Only the digit portions of numeric match fields are compared. Even though a field is negative, it is considered to be positive since the sign of the numeric field is ignored. Thus, a -5 will match with a +5.
11. Whenever more than one matching record value is used, all match fields must match before the MR indicator turns on. For example, if matching fields M1, M2, 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 *Examples, Example 1*).
12. Field names are ignored in matching record operations. Therefore, fields from different record types which have been assigned the same match level may have the same name.
13. If you have defined an alternate collating sequence for your program, alphameric fields are matched according to the sequence you have specified.
14. A matching field cannot be specified as binary (B in column 43).

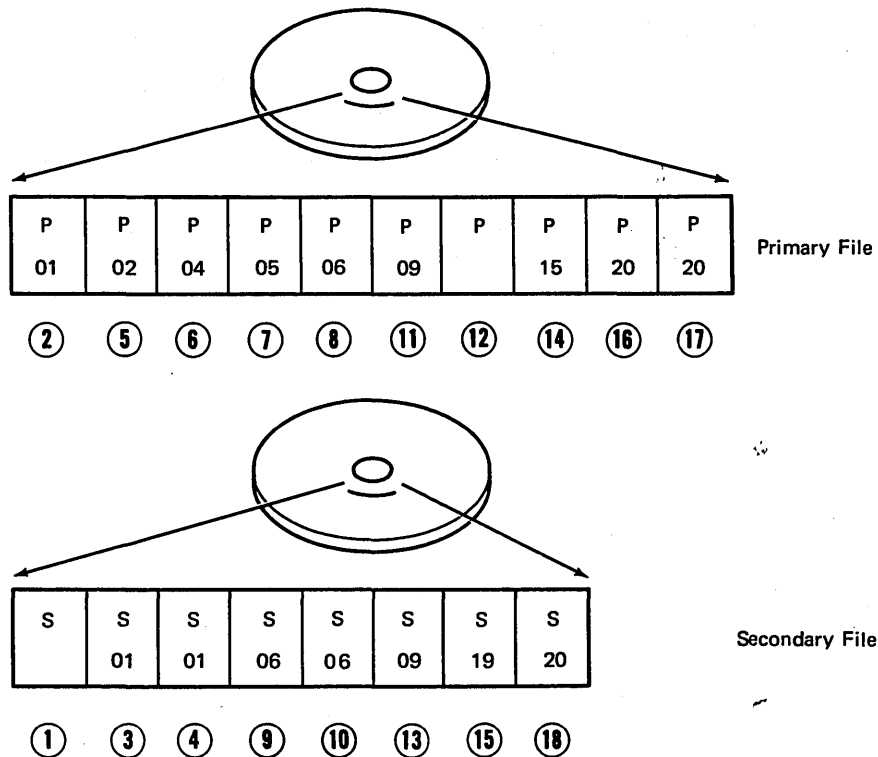
Note: Additional rules applying to matching records when used with entries in the Field Record Relation columns are discussed in *Columns 63–64*.

Processing Matching Records

Matching records for two or more files are processed in the following manner:

1. 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. (Remember, the record identifying indicator which 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.)

2. Whenever records from ascending files do not match, the record having the lowest match field content is processed first (Figure 62). Whenever records from descending files do not match, the record having the highest match field content is processed first.



The circled numbers represent the order of record selection from the two files. The following notes clarify the reasons for the selection of various records.

- ① This record is selected first because it has no match field.
- ② When a match occurs, the primary record is always selected first.
- ⑤ When records have no match, the record with the lowest match field is selected, regardless of file.
- ⑫ A record with no match field is processed immediately after the record it follows, regardless of file or match condition.

Figure 62. Processing Order According to Matching Records

3. A record type which has no matching 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 (Figure 62).
4. The matching of records makes it possible to enter data from primary records into their matching secondary records since the primary record is processed before the matching secondary record. However, the transfer of data from secondary records into matching primary records can only be done through look ahead fields (see *Look Ahead* under *Supplementary Information*).

For additional information on matching records from more than two files, see *Operation Codes, Force*, under *Supplementary Information*.

Sequence Checking

Make an entry in columns 61–62 when you want to sequence check records within one input, update, or combined file. This entry causes sequence checking of the data in the fields to which M1–M9 have been assigned (see *Columns 15–16* for sequence checking of record types). You may use as many as nine fields (M1–M9) to sequence check. The sequence (ascending or descending) of your record file must be specified in the file description specifications (see *Column 18* under *File Description Specifications*). An entry in columns 61–62 indicates that the records are to be checked to see if they really are in the sequence specified (see *Examples, Example 3*).

IBM International Business Machines Corporation

Form X21-9094
Printed in U.S.A.

RPG INPUT SPECIFICATIONS

Date _____

Program _____

Programmer _____

Punching Instruction	Graphic										Page	1	2	Program Identification	75	76	77	78	79	80
	Punch																			

Line	Form Type	Filename	Sequence Number (1-N)	Option (O)	Record Identifying Indicator or **	Record Identification Codes												Field Location		Field Name	Decimal Positions	Control Level (L1-L9)	Matching Fields or Chaining Fields	Field Record Relation	Field Indicators			Sterling Sign Position																			
						1				2				3				From	To						Plus	Minus	Zero or Blank																				
						Position	Not (N)	C/Z/D	Character	Position	Not (N)	C/Z/D	Character	Position	Not (N)	C/Z/D	Character																														
0 1	I	MASTER																																													
0 2	I															1	20	NAME																													
0 3	I															11	15	NUM																													
0 4	I															17	19	DEPT			M1																										
0 5	I															20	22	REGION			M2																										
0 6	I															23	25	DIVISION			M3																										
0 7	I																																														

Figure 64. Match Fields (Sequence Checking Within a File)

The matching field from record 1 is compared with the matching field from record 2. If the file is specified to be in ascending sequence, the records are in order since 005025003 is higher than 003051008. However, if the file is specified as having a descending sequence, record 2 is out of order.

COLUMNS 63-64 (FIELD RECORD RELATION)

Entry	Explanation
01-99	Record identifying indicator assigned to a record type.
L1-L9	Control level indicator previously used.
MR	Matching record indicator.
U1-U8	External indicator previously set.
H1-H9	Halt indicator previously used.

Columns 63-64 have several uses which are discussed after these general rules:

1. All fields, including matching or control fields, that have no field record relation specification should come before those that do.
2. All fields related to one record type (that is, having the same Field Record Relation entry) should be entered as a group in specification lines following one another for more efficient use of core storage. These fields could, however, be entered in any order.

3. All portions of a split control field must be assigned the same field record relation indicator and must be entered as a group in specification lines following one another (see *Examples, Example 3*). For more information on split control fields, see *Columns 59-60*.
4. When used with match or control 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.
5. When any match value (M1-M9) is specified without field record relation, all match values used must be specified once without field record relation. If all match fields are not common to all records, a dummy match field should be used.

Record Identifying Indicators (01-99)

Columns 63-64 are commonly used when several record types have been specified in an OR relationship. Fields which have no field record relation indicator are associated with all the record types in the OR relationship. This is fine when all record types have the same fields. But if the record types in the OR relationship have some fields that are the same and some that are not the same, you do not want to associate every field with all records. Therefore, you must have some way of relating a field to a certain record. To do this, place in columns 63-64 the record identifying indicator found in columns 19-20 of the record type on which the field is found (see *Examples, Example 1*).

Control fields (indicated by entries in columns 59–60) and matching fields (indicated by entries in columns 61–62) may also be related to a particular record type in an OR relationship by a field record relation entry. Control fields or matching fields that are not related to any particular record type in the OR relationship by the 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 matching fields have the same matching level entry, it is possible to assign a field record relation indicator to just one of the control fields or to just one of the matching fields. In this case, only the specification having 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 matching field, the specification without a field record relation indicator is used. Control fields and matching fields cannot have an L1–L9 or MR entry in columns 63–64.

When any match value (M1–M9) is specified without field record relation, all match values used must be specified once without field record relation. If all match fields are not common to all records, a dummy match field should be used (see *Example, Example 2*).

Control Level (L1–L9) and Matching Record (MR) Indicators

Another situation for which you may use these columns is when you wish to accept and use data from a particular field only when a certain condition (such as matching records or a control break) occurs. You indicate the conditions under which you accept data from a field by indicator L1–L9 or MR. MR cannot be used with demand files. Data from the field named in columns 53–58 is accepted only when the indicator is on.

External Indicators (U1–U8)

You may also use these columns to condition a specification by an external indicator (U1–U8). The external indicator, which you set prior to processing, controls whether or not the specification is done. When the indicator is on, the specification is done; when it is off, the specification is not done.

External indicators are primarily used when file conditioning is done by an entry in columns 71–72 in the file description specifications. However, they may also be used to condition when a specification should or should not be done even though file conditioning is not specified. See *Indicators, External Indicators* under *Supplementary Information*.

Halt Indicator (H1–H9)

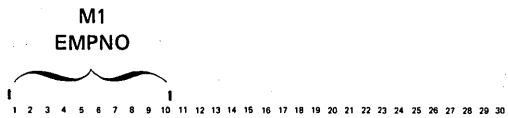
A halt indicator is used to relate a field to a record that is in an OR relationship and also has a halt indicator specified in column 19–20.

Examples

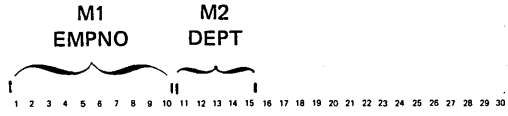
Example 1: Figure 65 shows how record identifying indicators are used to relate a field to a record. The file contains two different types of records, one identified by a 5 in position 1 and the other by a 6 in position 1. FLDC is related by record identifying indicator 14 to the record type which is 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 and FLDB are found on both types since they are not related to any one type by a record identifying indicator.

Example 2: Figure 66 shows how dummy match fields are used in field record relation. Three different record types are found in the input file (see insert A). All three contain a match field in positions 1–10. Two of them have a second match field. Insert B shows the correct field record relation specifications. Since M1 is found on all record types, it can be specified without an entry in columns 63–64. If one match value (M1–M9) is specified without field record relation entries, all match values must be specified once without

field record relation entries. Since 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).



Record Identifying Indicator 01



Record Identifying Indicator 02



Record Identifying Indicator 03

(A)

			International Business Machines Corporation																																						Form X21-9094 Printed in U.S.A.								
			RPG INPUT SPECIFICATIONS																																						Page 1 2		Program Identification 75 76 77 78 79 80						
			Punching Instruction															Graphic					Punch																										
Line	Form Type	Filename	Sequence	Number (I-N)	Option (O)	Record Identifying Indicator	Record Identification Codes									Field Location				Field Name	Control Level (L1-L9)	Matching Fields or Chaining Fields	Field Record Relation	Field Indicators			Sterling Sign Position																						
1-30	31-40	41-50	51-60	61-70	71-80	1	2		3		From	To	Decimal Positions	Plus	Minus	Zero or Blank																																	
01	I	DISK	AB	Ø1			1	C1																																									
02	I		OR	Ø2			1	C2																																									
03	I		OR	Ø3			1	C3																																									
04	I													1	100	EMPNO	M1																																
05	I													11	150	DUMMY	M2																																
06	I													17	150	DEPT	M2Ø2																																
07	I													21	250	DEPT	M2Ø3																																
08	I																																																
09	I																																																
10	I																																																

(B)

Figure 66. Dummy Match Fields

Example 3: Split control fields on one record type must have the same record relation entry. Figure 67, insert A, shows several record types with split control fields in each. The record identified by a 1 in position 95 has two split control fields:

FLD1A and FLD1B
FDL2A 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

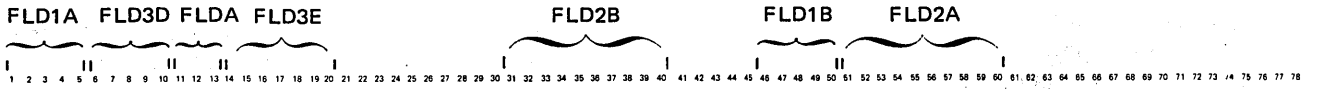
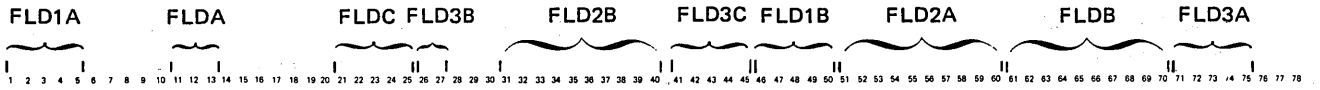
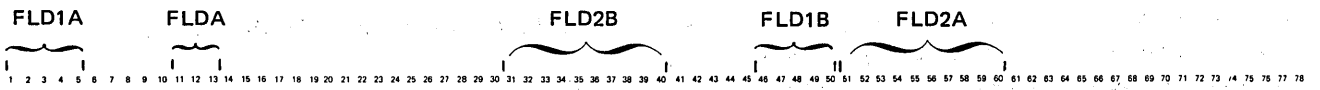
All portions of the split control field must be assigned the same control level indicator and all must have the same field record relation entry. Figure 67, insert B, shows the field record relation required for the three record types.

COLUMNS 65–70 (FIELD INDICATORS)

<i>Entry</i>	<i>Explanation</i>
01–99	Numeric indicator.
H1–H9	Halt indicator (when checking for an error condition in the data).

Columns 65–70 can be used to check the condition of numeric fields. The three conditions are:

1. Plus (columns 65–66). Any valid indicator entered here is turned on if the numeric field named in columns 53–58 is greater than zero.
2. Minus (columns 67–68). Any valid indicator entered here is turned on if the numeric field in columns 53–58 is less than zero.
3. Zero or blank (columns 69–70). Any valid indicator entered here is turned on if a numeric field named in columns 53–58 is all zeros or if an alphameric field is all blanks. A numeric field which is all blanks will turn on an indicator specified for all zeros. However, if an alphameric field is all zeros, the field will not turn on an indicator specified for all blanks.



A

IBM International Business Machines Corporation Form X21-9094
Printed in U.S.A.

RPG INPUT SPECIFICATIONS

Date _____ Program _____ Programmer _____

Punching Instruction	Graphic										
	Punch										

Page 1 2 Program Identification 75 76 77 78 79 80

Line	Form Type	Filename	Sequence Number (1-N)	Option (O)	Record Identifying Indicator C* U*	Record Identification Codes									Field Location		Field Name	Control Level (L1-L9)	Matching Fields or Chaining Fields	Field Record Relation	Field Indicators			Sterling Sign Position
						1			2			3			From	To					Plus	Minus	Zero or Blank	
						Position	Not (N) C/Z/D	Character	Position	Not (N) C/Z/D	Character	Position	Not (N) C/Z/D	Character										
01	I	DISK	BC		91	95	C1							1	5	FLD1A	L1							
02	I		OR		92	95	C2							46	50	FLD1B	L1							
03	I		OR		93	95	C3							11	13	FLDA	L2							
04	I													51	60	FLD2A	L3							
05	I													31	40	FLD2B	L3							
06	I													71	75	FLD3A	L4	92						
07	I													26	27	FLD3B	L4	92						
08	I													41	45	FLD3C	L4	92						
09	I													61	70	FLDB		92						
10	I													21	25	FLDC		92						
11	I													6	10	FLD3D	L4	93						
12	I													14	20	FLD3E	L4	93						
13	I																							
14	I																							
15	I																							
	I																							
	I																							
	I																							
	I																							
	I																							
	I																							

All portions of a split control field must have the same field record relation entry.

B

Figure 67. Field Record Relation (Split Control Fields)

In the input specifications, you specify the indicators that will be used to condition operations. In the calculation specifications and output-format specifications, you actually use these indicators. When conditioning operations, you must know when the indicators will be off and when they will be on.

Assigning Indicators in Columns 65–70

The following considerations apply to numeric indicators and command key indicators:

1. Indicators for plus or minus are off at the beginning of the program. They are not turned on until the condition (plus or minus) is satisfied by the field being tested on the record just read.
2. An indicator assigned to zero or blank is off at the beginning of the program. It remains off until the field being tested is zero or blank.
3. One input field may be assigned two or three field indicators. However, only the one which signals the result of the test turns on; the others are turned off.
4. If the same field indicator is assigned to fields in different record types, its status is always based on the last record type selected.
5. When different field indicators are assigned to fields in different record types, a field indicator turned on will remain on until another record of that type is read. Similarly, a field indicator assigned to more than one field within a single record type will always reflect the status of the last field defined.
6. Field indicators assigned in these columns may be SETON or SETOF in calculation specifications.

Numeric Indicators (01–99)

Use numeric indicators 01–99 when you wish 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. You usually use these same indicators to control certain calculation or output operations (see *Columns 9–17* under *Calculation Specifications* or *Columns 23–31* under *Output-Format Specifications*).

Halt Indicators (H1–H9)

Specify any halt indicator (H1–H9) in columns 65–70 when you wish to check for an error condition in your data. For example, if a field should not be zero, you can specify a halt indicator to check for that zero condition. If a zero field is found, the halt indicator turns on and the job stops after the record with the zero field has been processed.

Indicators H1–H9 cause the program to halt after the record which caused the indicator to turn on is completely processed. The operator can restart the system by pressing the start key.

COLUMNS 71–74

Columns 71–74 are not used. Leave them blank.

COLUMNS 75–80 (PROGRAM IDENTIFICATION)

See *Common Entries*.

COLUMNS 1–2 (PAGE)

See *Common Entries*.

COLUMNS 3–5 (LINES)

See *Common Entries*.

COLUMN 6 (FORM TYPE)

A C must appear in column 6.

COLUMNS 7–8 (CONTROL LEVEL)

<i>Entry</i>	<i>Explanation</i>
Blank	Calculation operation is not part of a subroutine and may only be performed for detail calculations.
L0	Calculation operation to be done at total calculation time for each program cycle.
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.
AN, OR	Establishes AND and OR relationships between lines of indicators.

Columns 7–8 may be used for four different purposes:

1. To perform total calculation operations when the appropriate control break occurs.
2. To perform calculation operations which are done only after the last record has been read.
3. To indicate that an operation is part of a subroutine.
4. To specify that certain lines of indicators are in an AND/OR relationship.

If you leave columns 7–8 blank, the operation specified on the same line is done every time a record is read, provided indicators in columns 9–17 allow it (see *Columns 9–17*).

Control Level Indicators (L0, L1–L9)

The L0 indicator is on during the entire program. You need never assign this indicator, but you may use it. The indicator is often used when no control fields have been assigned. Remember, when a control break occurs, all operations conditioned by control level indicators are done before those that are not conditioned. If you have no control field but want total calculations to be done and total output records to be written or punched, you may use the L0 indicator to condition those operations (see *Examples, Example 1*).

Use control level indicators L1–L9 to signal when certain operations are to occur. If you specify a control level indicator (L1–L9) in columns 7–8, the operation described on the same specifications line is done only when that indicator is on. Remember that a control level indicator turns on when information in a control field changes (see *Columns 59–60* under *Input Specifications*).

A control break for a certain level causes all lower control level indicators to turn on. Thus, if you used indicators L3, L2, and L1 in your program and L3 turns on, L1 and L2 will also turn on. All operations conditioned by L3, L2, and L1 will be done.

There is an exception, however. When a control level indicator used as a record identifying indicator turns on to reflect the type of record read or when a control level indicator turns on by the SETON instruction, 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 7–8 are done at total calculation time. Operations that are conditioned by control level indicators in columns 9–17 are done at detail calculation time immediately following the control break (see *Relationship Between Columns 7–8 and Columns 9–17*).

Last Record Indicator (LR)

The last record (LR) indicator automatically turns on after the last record is read and processed. You may have certain operations which are to be done only after the last record has been read. Condition these operations with the LR indicator. Place operations conditioned by LR after all calculations conditioned by L0–L9 (column 7–8) or after detail calculations if there are no total calculations. The last record causes the LR indicator and all other control level indicators specified (L1–L9) to turn on.

Subroutine Lines (SR)

An SR entry in columns 7–8 indicates that a line is part of a subroutine (see *Subroutines* under *Supplementary Information*). Subroutine lines must be specified last.

AND/OR Lines (AN, OR)

Columns 7–8 can be used to specify that lines of indicators are in an AND/OR relationship. By using the AND/OR relationship, many lines of indicators may be grouped together to condition an operation. A maximum of seven OR lines or seven AND lines or any combination thereof may be used to condition an operation.

The first line of such a group contains blanks in columns 7–8 or an L0–L9, LR, or SR entry if the group of lines is conditioned by a control level indicator or is part of a subroutine. All lines after the first line in the group must have an AN or OR entry in columns 7–8. The last line of the group contains the operation and the necessary operands. All lines in the group prior to the last line must contain blanks in the columns for Factor 1, Factor 2, Operation, Result Field, and Resulting Indicator (see *Examples, Example 2 and Example 3*).

Examples

Example 1: Figure 69 shows the format of the report printed by the job described in Figure 70. The job shows how total operations can be performed even though there is no control field (no L1–L9 indicators). The job requires:

1. A list of items sold in each district.
2. A total of all sales for each district.
3. A grand total of all sales in all districts.

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 69. Format of a Printed Report

The input records have ITEM and COST fields and a one-position record identification field. The records are grouped in ascending sequence by district; that is, the district one records as a group are followed by a blank record, and the district two records as a group are followed by a blank record (Figure 71).

There is no field that can serve as a control field, since 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 is used to tell the program that total calculations and total output operations must be done. However, no total operations may 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 7-8 because some type of control level indicator must be assigned to all total operations.

Example 2: Figure 72, insert A, shows the use of AN and OR entries to 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 will be performed.

Example 3: Figure 72, insert B, illustrates a case in which three conditions will 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.

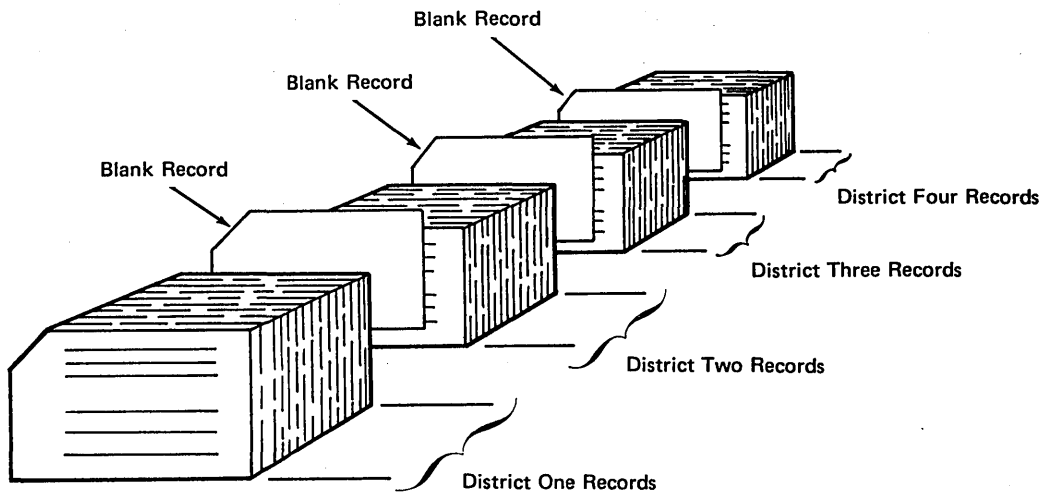


Figure 71. Blank Records Used to Signal Control Breaks

IBM International Business Machines Corporation Form X21-9093 Printed in U.S.A.

RPG CALCULATION SPECIFICATIONS

Date _____

Program _____

Programmer _____

Punching Instruction	Graphic								
	Punch								

Page

1	2
---	---

 Program Identification

75	76	77	78	79	80
----	----	----	----	----	----

Line	Form Type	Control Level (LO, LR, SR)	Indicators			Factor 1	Operation	Factor 2	Result Field	Field Length	Decimal Positions	Half Adjust (H)	Resulting Indicators			Comments
			And	And									Arithmetic	Plus	Minus	
			Not	Not	Not								High	Low	Equal	
01	C		01	02	03											
02	C	AN	04													
03	C	OR	01	02	03											
04	C	AN	05			FIELD A	SUB	FIELD B	QTY	4			23			
05	C															
06	C															
07	C															

(A)

IBM International Business Machines Corporation Form X21-9093 Printed in U.S.A.

RPG CALCULATION SPECIFICATIONS

Date _____

Program _____

Programmer _____

Punching Instruction	Graphic								
	Punch								

Page

1	2
---	---

 Program Identification

75	76	77	78	79	80
----	----	----	----	----	----

Line	Form Type	Control Level (LO, LR, SR)	Indicators			Factor 1	Operation	Factor 2	Result Field	Field Length	Decimal Positions	Half Adjust (H)	Resulting Indicators			Comments
			And	And									Arithmetic	Plus	Minus	
			Not	Not	Not								High	Low	Equal	
01	C	L4	01	02	N03											
02	C	OR	01	N02	03											
03	C	OR	N01	02	03	SUM	ADD	SUMTOT	SUMTOT	82H						
04	C															
05	C															

(B)

Figure 72. Use of AND/OR Lines for Indicators

COLUMNS 9–17 (INDICATORS)

<i>Entry</i>	<i>Explanation</i>
Blank	Operation is performed on every program cycle.
01–99	Field indicators, record identifying indicators, or resulting indicators used elsewhere in the program.
KA–KN, KP, KQ	Command key indicators previously assigned.
L1–L9	Control level indicators previously assigned.
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 9–17 to assign indicators that control the conditions under which an operation is done. You may use from one to three separate fields (9–11, 12–14, and 15–17) on each line, one for each indicator. If the indicator must not be in order to condition the operation, place an N before the appropriate indicator (columns 9, 12, 15).

AND/OR Lines (AN, OR)

By using AN or OR entries in columns 7–8, many indicators can be used to condition one operation. A maximum of seven OR lines may be used to condition an operation if no AND lines are used. A maximum of seven AND lines may be used to condition an operation if no OR lines are used. If AND or OR lines are combined, the total number of AND and OR lines used to condition an operation cannot exceed seven.

AND Relationship

All three indicators on one 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 7–8) must all be exactly as specified before the operation is done (see *Examples, Example 1*).

Field Indicators (01–99)

Use any field indicators previously specified in columns 65–70 on the Input sheet 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 *Examples, Example 2*).

Command Keys Indicators (KA–KN, KP, KQ)

Use any command key indicators previously specified in columns 54–59 of the Calculation sheet for a SET or SETOF operation. See *Operation Codes, SET or SETOF* under *Supplementary Information* for complete information on each operation.

Record Identifying Indicators (01–99)

Use any record identifying indicators previously specified in columns 19–20 on the Input sheet to condition an operation that is to be done only for a certain type of record (see *Examples, Example 3*).

Resulting Indicators (01–99)

Use any resulting indicators specified in columns 54–59 on the Calculation sheet to condition operations according to the results of previous calculation operations (see the example in *Columns 54–59*).

Control Level Indicators (L1–L9)

Use any control level indicators specified in columns 59–60 on the Input sheet or in columns 54–59 on the Calculation sheet. If control level indicators are used in these columns but not in columns 7–8, the operation is performed (at detail calculation time) on the first record of a new control group or whenever the indicator is on.

Last Record Indicator (LR)

To condition operations to be performed at end of job, use the last record (LR) indicator in columns 9–17 only if LR is turned on during calculations. If LR is off during calculations, all operations to be performed at end of job should be conditioned by LR in columns 7–8.

Matching Record Indicator (MR)

Use the matching record (MR) indicator to condition an operation that is to be done only when matching records have been found. See *Columns 61–62 under Input Specifications* for more information on matching fields.

Halt Indicators (H1–H9)

Use any halt indicators previously used in columns 65–70 on the Input sheet or in columns 54–59 on the Calculation sheet to prevent the operation from being done when a specified error condition has been found in the input data (see *Columns 19–20 under Input Specifications*) or on previous calculations. This is necessary because the record that causes the halt condition will be completely processed before your program stops. Thus, if the operation is performed even on an error condition, the results are in error. It is also possible to use a halt indicator to condition an operation that is to be done only when an error occurs.

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. See *Indicators, U1–U8 under Supplementary Information* for more information.

Overflow Indicators (OA–OG, OV)

Use any overflow indicators previously specified in columns 33–34 on the File Description sheet to condition operations that are to be done when the last line to be printed on a page has been reached. See *Overflow Indicators under Supplementary Information* 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 7–8 (total time) are done before operations that are conditioned by control level indicators in columns 9–17 (see *Examples, Example 4*).

When a control level indicator is used in columns 9–17 and columns 7–8 are not used (detail time), the operation conditioned by the indicator is done only on the record that causes control break or any higher level control break.

When a control level indicator (L1–L9) is specified in columns 7–8 (total time) and MR is specified in columns 9–17, 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 7–8 of the Calculation sheet) are done, MR then indicates the matching condition of the record just read.

Examples

Example 1: Figure 73 shows the use of control level indicators to condition calculation operations. The operation in line 02 may 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.

Example 2: Figure 74 shows the use of field indicators to condition operations. Assume the job is to find weekly earnings including overtime. The overtime field is checked to see if any overtime has been 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 (lines 03 and 04).

IBM																					
Date _____																					
Program _____																					
Programmer _____																					
Line	Form Type	Control Level (L, O, L3, LR, SR)	Indicators															Fi			
			Not	And	Not	And	Not	Not	Not	Not	Not	Not	Not	Not	Not	Not					
			3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
0 1	C									25		L1									
0 2	C	L2							I	φ	NL3										
0 3	C																				
0 4	C																				
0 5	C																				

**Figure 73. Conditioning Operations
(Control Level Indicators)**

IBM International Business Machines Corporation Form X21-9094
Printed in U.S.A.

RPG INPUT SPECIFICATIONS

Date _____
 Program _____
 Programmer _____

Punching Instruction	Graphic						
	Punch						

Page 1 2 Program Identification 75 76 77 78 79 80

Line	Form Type	Filename	Sequence Number (1-9)	Option (O)	Record Identification Codes			Field Location		Field Name	Control Level (L1-L9)	Matching Fields or Chaining Fields	Field Record Relation	Field Indicators			Sterling Sign Position
					1	2	3	From	To					Plus	Minus	Zero or Blank	
					Position	Not (N) C/Z/D Character	Position	Not (N) C/Z/D Character	Position					Not (N) C/Z/D Character	Stacker: Select P = Packed/B = Binary	Decimal Positions	
01	I	CARDS	AB	Ø1													
02	I							1	7	EMPLNO							
03	I							8	10	OVERTM							
04	I							15	20	RATE							
05	I							21	25	RATEOT							
06	I																
07	I																

IBM International Business Machines Corporation Form X21-9093
Printed in U.S.A.

RPG CALCULATION SPECIFICATIONS

Date _____
 Program _____
 Programmer _____

Punching Instruction	Graphic						
	Punch						

Page 1 2 Program Identification 75 76 77 78 79 80

Line	Form Type	Indicators			Factor 1	Operation	Factor 2	Result Field	Field Length	Decimal Positions	Half Adjust (H)	Resulting Indicators			Comments		
		And		And								Arithmetic		Plus		Minus	Zero
		Not	Not	Not								High	Low	Equal		High	Low
01	C				RATE	MULT	Ø	WAGE	62	H							
02	C	10			OVERTM	MULT	RATEOT	OVERPY	62	H							
03	C	10			WAGE	ADD	OVERPY	TOTAL	62								
04	C																

Field indicator 10 was assigned on the input specifications. It is being used here to condition calculation operations.

Figure 74. Conditioning Operations (Field indicators)

Example 3: Figure 75 shows how a record identifying indicator is used to condition an operation. When a record is read that has a T in position 1, the O1 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 O2 indicator is on. The subtract operation, since it is conditioned by O2, is then done instead of the add operation.

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 7–8 are performed before operations conditioned by control level indicators in columns 9–17. Thus, the operation in line O2 occurs before the operation in line O1. The operation in line O1 is done on the first record of the new control group indicated by 25, whereas the operation in line O2 is a total operation done for all records of the previous control group.

Example 4: Line O2 of Figure 73 shows the use of a control level indicator in columns 9–17. Assume that indicator

IBM International Business Machines Corporation Form X21-9094 Printed in U.S.A.

RPG INPUT SPECIFICATIONS

Date _____

Program _____

Programmer _____

		Record Identification Codes												Field Location		Field Name	Control Level (L1-L9)	Matching Fields or Chaining Fields	Field Record Relation	Field Indicators			Sterling Sign Position						
		1				2				3				From	To					Decimal Positions	Plus	Minus		Zero or Blank					
Line	Form Type	Filename	Sequence Number (L+N)	Option (O)	Record Identifying Indicator	Position	Not (N)	C/Z/D	Character	Position	Not (N)	C/Z/D	Character	Position	Not (N)	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	Sterling Sign Position	
0 1	I	FILE	AA	Ø1	1	CT																							
0 2	I		OR	Ø2	1	INCT												1Ø	152		SAVE								

IBM International Business Machines Corporation Form X21-9093 Printed in U.S.A.

RPG CALCULATION SPECIFICATIONS

Date _____

Program _____

Programmer _____

Line	Form Type	Control Level (LD-L9, LR, SR)	Indicators			Factor 1	Operation	Factor 2	Result Field	Field Length	Decimal Positions	Half Adjust (H)	Resulting Indicators			Comments
			Not	And	And								Arithmetic	Plus	Minus	
												High	Low	Equal		
0 1	C		Ø1			SUM	ADD	SAVE		SUM	82					
0 2	C		Ø2			SUM	SUB	SAVE		SUM	82					

Record identifying indicators O1 and O2 are assigned on the input specifications. They are used here to condition calculation operations.

Figure 75. Conditioning Operations (Resulting Indicators)

COLUMNS 18–27 (FACTOR 1) AND COLUMNS 33–42 (FACTOR 2)

Use columns 18–27 and 33–42 to name the fields or to give the actual data (literals) on which an operation is to be performed. See Figure 77 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 alphanumeric or numeric literal.
- Any subroutine, table, array name, or array element.
- Any date field names (UPDATE, UMONTH, UDAY, UYEAR).
- The special names PAGE, PAGE1, or PAGE2.

The following entry can be made for Factor 1 only:

- A label for a TAB, BEGSR, or ENDSR operation.

The following entries can be made for Factor 2 only:

- The entry *BOTH for a KEY operation.
- A label for a GOTO or EXSR operation.
- A filename for a KEY, SET, CHAIN, DEBUG, DSPLY, READ, or FORCE operation.
- A subroutine name for an EXIT operation.

An entry in Factor 1 must begin in column 18; an entry in Factor 2 must begin in column 33.

The entries you use depend upon the operation you are describing. Some operations need entries in both sets of columns, some need entries in only one, and some need no entries at all. See *Columns 28–32* for more information on operation codes. If you are naming a subroutine, see *Subroutines* under *Supplementary Information*.

Literals

A literal is the actual data used in an operation rather than the field name representing that data. A literal may be either alphameric or numeric.

Consider the following rules when using an alphameric literal (Figure 76, insert A):

1. Any combination of characters may be used in an alphameric literal. Blanks are also valid.
2. The maximum length of an alphameric literal is 8 characters.
3. Alphameric literals must be enclosed with apostrophes (').
4. An apostrophe required as part of a literal is represented by two apostrophes. For example, the literal O'CLOCK would be coded as 'O''CLOCK'.
5. Alphameric literals may not be used for arithmetic operations.

Consider the following rules when using a numeric literal (see Figure 76, insert B):

1. A numeric literal consists of any combination of the digits 0–9. A decimal point or sign may also be included.
2. The sign (+ or -), if present, must be the leftmost character. An unsigned literal is treated as a positive number.
3. The maximum total length of a numeric literal is 10 characters including signs and decimal points.
4. Blanks may not appear in a numeric literal.
5. Numeric literals must not be enclosed with apostrophes (').
6. Numeric literals are used in the same way as a numeric field.

COLUMNS 28–32 (OPERATION)

Use columns 28–32 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 have been defined which you must use to indicate the type of operation desired. Every operation code used requires certain entries on the same specification line. See Figure 77 for a summary of all possible codes and the additional entries required for

each code. For further information on the operations that can be performed, see *Operations Codes* under *Supplementary Information*.

The operations are performed in the order specified on the Calculation sheet.

Note: All operations conditioned by control level indicators in columns 7–8, must follow those that are not conditioned by control level indicators.

Type of Operation	Function of Operation	Operation Code (columns 28-32)	Control Level	Indicators	Factor 1	Factor 2	Result Field	Field Length	Decimal Position	Half Adjust	Resulting Indicators
Arithmetic Operations	Add Factor 2 to Factor 1.	ADD	O	O	R	R	R	O	O	O	O
	Clear Result Field and add Factor 2.	Z-ADD	O	O	B	R	R	O	O	O	O
	Subtract Factor 2 from Factor 1.	SUB	O	O	R	R	R	O	O	O	O
	Clear Result Field and subtract Factor 2.	Z-SUB	O	O	B	R	R	O	O	O	O
	Multiply Factor 1 by Factor 2.	MULT	O	O	R	R	R	O	O	O	O
	Divide Factor 1 by Factor 2.	DIV	O	O	R	R	R	O	O	O	O
	Move remainder of preceding division to a Result Field.	MVR	O	O	B	B	R	O	O	O	O
	Sum elements of an array and put sum in Result Field. Derive the square root of Factor 2.	XFOOT SQRT	O O	O O	B B	R R	R R	O O	O O	O O	O B
Move Operation	Move Factor 2 into Result Field, right justified.	MOVE	O	O	B	R	R	O	O	B	B
	Move Factor 2 into Result Field, left justified.	MOVEL	O	O	B	R	R	O	O	B	B
Move Zone Operations	Move zone from low-order position of Factor 2 to low-order position of Result Field.	MLLZO	O	O	B	R	R	O	O	B	B
	Move zone from high-order position of alphameric Factor 2 to high-order of alphameric Result Field.	MHHZO	O	O	B	R	R	O	B	B	B
	Move zone from low-order position of Factor 2 to high-order position of alphameric Result Field.	MLHZO	O	O	B	R	R	O	B	B	B
	Move zone from high-order position of alphameric Factor 2 to low-order position of Result Field.	MHLZO	O	O	B	R	R	O	O	B	B
Compare and Zone Testing Operations	Compare Factor 1 to Factor 2.	COMP	O	O	R	R	B	B	B	B	R
	Identify the zone in the leftmost position of an alphameric Result Field.	TESTZ	O	O	B	B	R	O	B	B	R
Bit Operations	Set on specified bits.	BITON	O	O	B	R	R	O	B	B	B
	Set off specified bits.	BITOF	O	O	B	R	R	O	B	B	B
	Test specified bits.	TESTB	O	O	B	R	R	O	B	B	R
Setting Indicators	Set one, two, or three specific indicators on.	SETON	O	O	B	B	B	B	B	B	R
	Set one, two, or three specific indicators off.	SETOF	O	O	B	B	B	B	B	B	R
Branching Within RPG II	Branch to another RPG II calculation specification line.	GOTO	O	O	B	R	B	B	B	B	B
	Identify the name in Factor 1 as a destination label to which GOTO may branch.	TAG	O	B	R	B	B	B	B	B	B
Branching to External Subroutines	Branch to user-written Assembler subroutine.	EXIT	O	O	B	R	B	B	B	B	B
	Transfer data to user-written Assembler subroutine.	RLABL	O	B	B	B	R	O	O	B	B
Lookup Operations	Table Lookup.	LOKUP	O	O	R	R	O	O	O	B	R
	Array Lookup.	LOKUP	O	O	R	R	B	B	B	B	R
Subroutine	Beginning of the subroutine.	BEGSR	*	B	R	B	B	B	B	B	B
	End of the subroutine.	ENDSR	*	B	O	B	B	B	B	B	B
	Call to execute the subroutine.	EXSR	O	O	B	R	B	B	B	B	B
Program Control of Input and Output	Pause for input data from keyboard.	KEYnn**	O	O	B	O	O	O	O	B	O
	Control of printer and ledger card output.	SETnn**	O	O	B	O	O	O	B	B	O
	Call for immediate input.	READ	O	O	B	R	B	B	B	B	***
	Forcing record to be read on next cycle.	FORCE	B	O	B	R	B	B	B	B	B
	Forcing output printing.	EXCPT	O	O	B	B	B	B	B	B	B
	A field is printed on the printer-keyboard and/or data is entered via the printer-keyboard into a field.	DSPLY	O	O	O	R	O	B	B	B	B
Debug Function	Aid in finding programming errors.	DEBUG	O	O	O	R	O	B	B	B	B
	A record is read from a disk file.	CHAIN	O	O	R	R	B	B	B	B	****

O — Optional
 R — Required
 B — Blank

* Columns 7–8 must have an SR entry for all subroutine lines.
 ** The nn entries in columns 31–32 are for field light numbers.
 *** Columns 58–59 may contain an indicator for this operation; columns 54–57 must be blank.
 **** A resulting indicator should be entered in columns 54–55; columns 56–59 must be blank.

55250A

Figure 77. Operation Codes

COLUMNS 31–32

<i>Entry</i>	<i>Explanation</i>
Blank, 0, 1–8	A combination of these entries is used to turn on field lights for SET or KEY operations.

Columns 31–32 must contain entries for field lights for all KEY operations and for SET operations in which command key indicators are specified in columns 54–59. For these columns, either one or two field lights may be specified for one operation.

If only one field light is specified, the number may be in column 31 or column 32. The column without 1–8 must be either 0 or blank (08, 80, 08, 80, will turn on field light 8).

If two field lights are specified in columns 31–32, the following rules apply:

1. For one operation the same number cannot appear in columns 31 and 32 (2 and 2 is invalid, 1 and 2 is valid).
2. The same combination of field lights 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 9–17).

See *Operation Codes, SET and KEY* under *Supplementary Information* for complete information.

COLUMNS 43–48 (RESULT FIELD)

<i>Entry</i>	<i>Explanation</i>
SPACE _n (n=1–3) SKIP _{nn} EJECT	Special instructions to control printer output and ledger card ejection when used with the SET operation.
Field name, table names, array name, or array element	These entries hold the result of, or are the object of, the operation specified in columns 28–32.
IN _{xx} (xx= any RPG II indicator)	The indicator to be transferred to an external subroutine in an RLABL operation.

SPACE_n, SKIP_{nn}, and EJECT

The SPACE_n and SKIP_{nn} entries may be made in columns 43–48 for TRACTR1 or TRACTR2. These entries provide immediate spacing and skipping when used with the SET operation. The SKIP_{nn} entry you specify must not exceed the forms length specified in line counter specifications or must not exceed 66 if no line counter specifications are supplied (nn = 01–99, A0–A9, or B0–B2). A ledger card may also be ejected in calculations by specifying EJECT in columns 43–48 using the SET operation. These entries must begin in column 43. See *Operation Codes, SET* under *Supplementary Information* for complete information.

Field Name, Table Name, Array Name, or Array Element

Use columns 43–48 to name the field, table, array, or array element that will hold the result of the operation specified in columns 28–32, or that is the field upon which an operation will be performed.

You may use the name of a field, table, array, or array element that has already been defined either in the input, extension, or calculation specifications. See *Arrays* under *Supplementary Information* for more information on Arrays.

Otherwise you may define a new field by entering a field name that has not already been used. Any field you define here will be created at the time the program is compiled. The field you name may be either numeric or alphanumeric. A field used in arithmetic operations (see *Columns 28–32*) or numeric compare, or a field edited or zero suppressed in output-format specifications must be numeric.

The result field name must begin with an alphabetic character in column 43 and contain no blanks or special characters.

- If you are entering the name of a field that has not been defined elsewhere, columns 49–52 should also contain entries.

If you are entering the name of a field that has been defined, entries in columns 49–52 are not necessary but, if specified, must agree with the previous definition of that field.

COLUMNS 49–51 (FIELD LENGTH)

<i>Entry</i>	<i>Explanation</i>
1–220	The position to which the print element is to be moved or the position where a tab stop is to be set using the SET operation.
1–256	Result Field length.

Positioning the Print Element and Setting Tab Stop Positions

Use columns 49–51 to (1) position the print element or (2) set tab stop positions using the SET operation. To position the print element, specify the starting position of the next output line. To set tab stop positions, specify the position you want the print element to move to when the operator presses the TAB key in a manual keying operation. These entries must begin in column 49. See *Operation Codes, SET* under *Supplementary Information* for complete information.

Result Field Length

Use columns 49–51 to give the length of a result field that has not been defined elsewhere. If you are naming a new field (one that has not been used before), you must consider the form your data will be in and the length it will have after the operation has been performed.

Whenever the field length is specified for a result field, you should be careful to make the result field long enough to hold the largest possible result. If the result field is too small, significant digits may be lost. For example, you may wish to add field A (eight characters long, four decimal places) to field B (ten characters long, six decimal positions). Fields A and B have four characters to the left of the decimal, but the result field, field C, must allow for more characters to the left of the decimal.

9999.0000	Field A
0001.111111	Field B
10000.111111	Field C (result field)

In this case, Field C was 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.

Numeric fields have a maximum length of 15 characters. Alphameric fields may be up to 256 characters long. You may indicate the length of a field that has been previously described either in the input specifications or in calculation specifications. However, if you do so, you must specify the same field length and number of decimal positions as was previously given to the field.

If the result field contains the name of a table or array, an entry in these columns is optional. If used, it must agree with the length described in the extension specifications.

COLUMN 52 (DECIMAL POSITIONS)

<i>Entry</i>	<i>Explanation</i>
Blank	Alphameric or numeric field described elsewhere.
0–9	Number of decimal places in a 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 left blank if the result field is alphameric. This column may be left blank if the result field is numeric but has been previously described in the input or calculations specifications. In this case, field length (column 49–51) must also be left blank.

The number of decimal positions must never be greater than the length of the field. The number may, 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.

Figure 78 shows how the contents of a result field after a multiplication operation may change according to the

Decimal Positions (column 52) and Field Length (column 49–51) specifications.

Multiplication: $98.76 \times 1.234 = 121.86984$

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



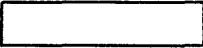
 Not permitted
 Permitted but inaccurate
 Recommended

Figure 78. Result Field Contents Based on Various Field Length and Decimal Position Specification

COLUMNS 54–59 (RESULTING INDICATORS)

<i>Entry</i>	<i>Explanation</i>
01–99	Any two-digit number.
KA–KN, KP, KQ	Any command key indicator.
H1–H9	Any halt indicator.
L1–L9	Any control level indicator.
LR	Last record indicator.
OA–OG, OV	Any overflow indicator.

Columns 54–59 are used for three different purposes:

1. To test the value of the result field after an arithmetic operation or after a CHAIN, KEY, LOKUP, COMP, READ, TESTB, or TESTZ operation. See *Operation Codes* under *Supplementary Information* for more information on each specific operation.
2. To specify which command keys may be pressed for a SET operation.
3. To specify which indicators to turn on or off using the SETON and SETOF operations.

Note: Command key indicators may be used as resulting indicators only if the operation is SET or SETOF.

Test Results

By entering an indicator in columns 54–59, you specify that the result field is to be tested after the operation specified in columns 28–32 has been performed. Normally, only indicators 01–99 and H1–H9 are used for testing. The indicator specified is turned 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 may then be used to condition following calculations or output operations (see *Example*). If the same indicator is used to test the result of more than one operation, the operation last performed determines the setting of the indicator.

Notice that three fields (columns 54–55, 56–57, and 58–59) can be used for this purpose. Each field is used to test for different conditions: columns 54–55, plus or high; columns 56–57, minus or low; columns 58–59, zero or equal. You may test for any or all conditions at the same time.

Columns 54–55 (Plus or High): An indicator in these columns when testing the result field to find:

1. If the result field in an arithmetic operation is positive.
2. If a numeric field named as a result field following a store only or display KEY operation is positive.
3. If Factor 1 is higher than Factor 2 in a compare operation.
4. If Factor 2 is higher than Factor 1 in a table array look-up operation.
5. If a CHAIN operation is not successful.
6. If each bit named in Factor 2 is off for a TESTB operation.
7. If the character tested in a TESTZ operation is one of the characters &, A-I.

Columns 56–57 (Minus or Low): Place an indicator in these columns when testing the result field to find:

1. If the result field in an arithmetic operation is negative.
2. If a numeric field named as result field following a store only or display KEY operation is negative.
3. If Factor 1 is lower than Factor 2 in a compare operation.
4. If Factor 2 is lower than Factor 1 in a table or array look-up operation.
5. If the bits names in Factor 2 are of mixed status (some bits on, some bits off) for a TESTB operation.
6. If the characters tested in a TESTZ operation is one of the characters -, J-R.

Columns 58–59 (Zero or Equal): Place an indicator in these columns when testing the result field to find:

1. If the result field in an arithmetic operation is zero.
2. If a numeric field named as result field is zero, or an alphameric field is blank following a store only or display KEY operation.
3. If Factor 1 is equal to Factor 2 in a compare operation.

4. If Factor 2 is equal to Factor 1 in a table or array look-up operation.
5. If an end-of-file condition is reached for the demand file using the READ operation.
6. If each bit named in Factor 2 is on for a TESTB operation.
7. If the character tested in a TESTZ operation is any character other than &, A-I, or -, J-R.

at that time. From 1–3 command keys may be entered for each SET operation. If one or two command keys are specified, they may appear in any of the three sets of columns. When command keys are specified, there must be at least one field light entry in columns 31–32. See *Operation Codes, SET* under *Supplementary Information* for complete information on this operation.

Allowing Command Keys to be Pressed (SET)

Columns 54–59 may contain command key indicators (KA–KN, KP, KQ) for a SET operation.

When a SET operation occurs, only the command keys in columns 54–59 for that SET operation may be pressed

Setting Indicators (SETON, SETOF)

You may enter the indicators that you want to turn on or off by the operations SETON or SETOF. See *Operation Codes, Setting Indicators* under *Supplementary Information* for more information on these operations. Any indicators to be turned on or off by the SETON or SETOF operations are specified from left to right in the three resulting indicators fields (Figure 80). Column headings in columns 54–59 have no meaning for SETON or SETOF operations.

Date _____		International Business Machines Corporation		Form X21-9093 Printed in U.S.A.										
Program _____		RPG CALCULATION SPECIFICATIONS		Page 1 2	Program Identification 75 76 77 78 79 80									
Programmer _____		Punching Instruction	Graphic											
		Punch												
Line	Form Type Control Level (LD, LR, SR)	Indicators			Factor 1	Operation	Factor 2	Result Field	Field Length	Decimal Positions Half Adjust (H)	Resulting Indicators			Comments
		And	And								Plus	Minus	Zero	
		Not	Not	Not							Compare			
											High	Low	Equal	
											1 > 2	1 < 2	1 = 2	
											Lookup			
											Table (Factor 2) is			
											High	Low	Equal	
0 1	C					SETON								Ø245
0 2	C													
0 3	C													
0 4	C					SETON								22
0 5	C													
0 6	C													
0 7	C					SETOF								45
0 8	C													
0 9	C													

Figure 80. Setting Indicators

Example

Figure 81 shows the entry of two indicators that are used to test for the different conditions in a compare operation. These indicators are used to condition the calculations which must be performed for a payroll job. Indicator 10 is turned 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 is turned on if HRSWKD is less than 40. It is also used to condition other operations. In line 03 if 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 20 is on (employee worked less than 40 hours), pay based on less than a 40 hour week is calculated.

COLUMNS 60-74 (COMMENTS)

Enter in columns 60-74 any meaningful information you wish. The comments you use should help you understand or remember what you are doing on 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*.

Line		Form Type (LD, LR, SR)	Indicators			Factor 1	Operation	Factor 2	Result Field	Field Length	Decimal Positions Half Adjust (H)	Resulting Indicators			Comments
3	4		Not	And	And							Arithmetic	Plus	Minus	
01	C				HRSWKD	COMP	40				1	0	0		
02	C		1	0	HRSWKD	SUB	40		OVERTM	3	0				
03	C		N	2	PAYRATE	MULT	40		PAY	6	2	H			
04	C		1	0	OVERTM	MULT	OVERRATE		OVRPAY	6	2	H			
05	C		1	0	OVERPAY	ADD	PAY		PAY						
06	C		2	0	PAYRATE	MULT	HRSWKD		PAY						
07	C														

Figure 81. Conditioning Operations (Resulting Indicators)

Output-format specifications describe your output records. These specifications may be divided into two general categories:

1. Record description entries (columns 7–31) which describe the output file record.
2. Field description entries (columns 23–74) which indicate the position and the format of data on the output record.

Write the specifications on the Output-Format sheet (Figure 82). The field description entries start one line lower than record description entries.

International Business Machines Corporation

Form X21-9090
Printed in U.S.A.

RPG OUTPUT - FORMAT SPECIFICATIONS

Date _____

Punching Instruction

Graphic									
Punch									

Page

1	2
---	---

Program _____

Program Identification

--	--	--	--	--	--	--	--	--	--	--	--	--

Programmer _____

Line	Form Type	Filename	Space			Skip			Output Indicators			Field Name	Edit Codes		Constant or Edit Word	Sterling Sign Position	
			Before	After	After	Before	After	After	And	And	And		Commas	Zero Balances to Print			No Sign
3																	
4																	
5																	
6																	
7																	
8																	
9																	
10																	
11																	
12																	
13																	
14																	
15																	

File Identification & Control: Line 2, Columns 15-22
Field Description: Line 4, Columns 23-74

Figure 82. Output-Format Sheet

COLUMNS 1–2 (PAGE)

See *Common Entries*.

COLUMNS 3–5 (LINE)

See *Common Entries*.

COLUMN 6 (FORM TYPE)

An O must appear in column 6.

COLUMNS 7–14 (FILENAME)

Use columns 7–14 to identify the output file you will be using. The filename must begin in column 7. Use the same filename given in the file description specifications.

When writing the specifications for output records, the filename need be specified only on the first line for the file. On subsequent lines for the same file, the filename need not be specified. However, the filename must be repeated if another output file is specified and you then wish to continue with further specifications from the first output file (Figure 83).

COLUMNS 14–16

Columns 14–16 can be used to specify AND/OR lines for output operations. For further information, see *Columns 23–31, AND and OR Lines*.

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 that is to be written. Column 15 must have an entry for every output record. This record may be printed, written on disk, written on the CRT, or punched in cards. Perhaps the clearest method of describing output files is to enter the records for each file in this order: heading, detail, total, and exception (Figure 83, insert A).

Another method is to enter all heading records for all output files, then all detail records for all output files, etc (Figure 83, insert B).

Heading Records (H)

Heading records usually contain unchanging identifying information such as column headings and page and date information.

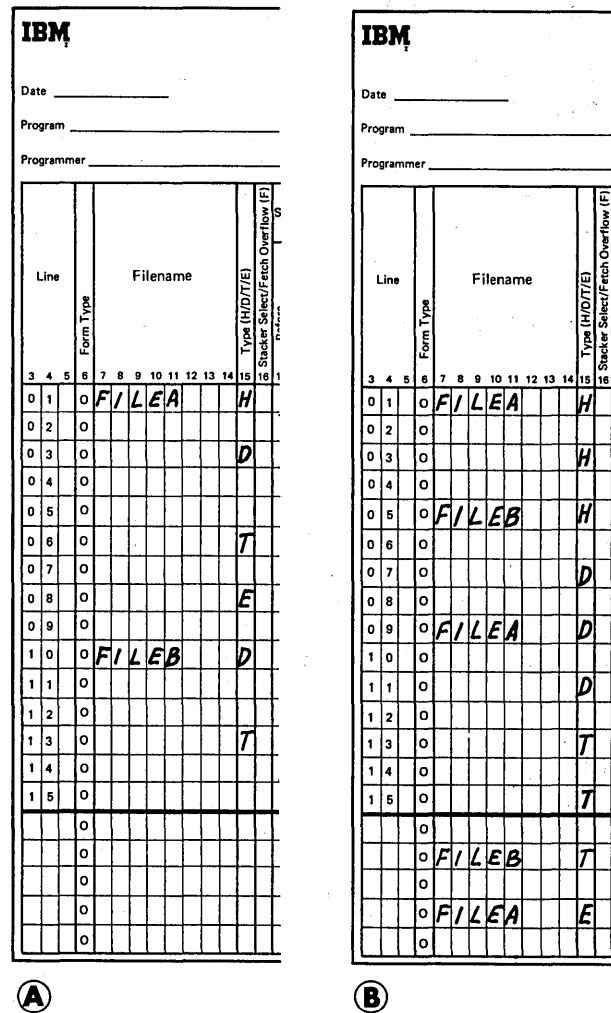


Figure 83. Order of Output Record Types

Detail Records (D)

Detail records are closely connected with input data. Most data in a detail record 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 may not be specified for primary or secondary update files. T may not be specified for a combined file.

Exception Records (E)

Exception records are written on disk, printed, or punched during calculation time. This is an unusual case and can be indicated only when the operation code EXCPT is used. E may not be specified for a combined file. See *Operation Codes* under *Supplementary Information* for further information on the EXCPT operation.

COLUMNS 16–18 (ADD A RECORD)

<i>Entry</i>	<i>Explanation</i>
ADD	Add a record.

Columns 16–18 may be used to specify that a record is to be added to an indexed file defined as an input, output, or update file. In addition to the ADD entry in columns 16–18, you must enter A in column 66 of the file description specifications for the file to which records are being added. The output device for these files must be a disk.

The ADD entry must not be specified in an OR line. An ADD entry in columns 16–18 of the previous line also applies to the record in the OR relationship.

COLUMN 16 (STACKER SELECT/FETCH OVERFLOW)

<i>Entry</i>	<i>Explanation</i>
E	Ledger card eject.
F	Fetch overflow.

Column 16 may be used to:

1. Eject a ledger card after completion of the output fields specified for a record type.
2. Indicate that the overflow routine can be used at this point for a printer file.

Ledger Card Eject (E)

A ledger card may be ejected by an E entry in column 16. Conditioning indicators may be used so that the eject occurs for only a particular record type. When eject is specified and the appropriate indicators are on or off, the fields for that record type are printed and the card is ejected after completion of the output fields specified for a record type. Consider the following rules for ledger card ejection:

1. An E may appear in column 16 on the same line as an H, D, T, or E entry in column 15, or it may appear on an OR line. An entry in column 16 of the previous line does not apply to the record in an OR line. Thus, if no E entry is made in column 16 of the OR line, the ledger will not be ejected.
2. If eject is not specified and a ledger card read is called for when a ledger is still in the machine, the ledger card in the machine is automatically ejected, but the program will halt.
3. If a new control number is specified for a ledger card and the previous ledger card has not been ejected, an automatic eject occurs, but the program will halt.
4. Indicators may be specified in columns 23–31 for the eject record.
5. Spacing may be specified in column 18 only (space after). If no entry is made for spacing, an entry of 1 is assumed. When spacing occurs, a line finder mark is placed on the card. If a zero is placed in the space after column, no line finder mark is printed, even if eject is specified.
6. It is not necessary to have any output fields specified in order to eject a ledger card.

Fetch Overflow (F)

Fetch overflow applies to the printer only. For the ledger card device, fetch overflow is automatic (see *Ledger File Processing* under *Supplementary Information*). When the fetch overflow routine is not used and overflow is specified for the file, the following occurs when the overflow line is sensed:

1. All remaining detail lines in that program cycle are printed.
2. All remaining total lines in that program cycle are printed.
3. All lines conditioned by an overflow indicator are printed.
4. Forms advance to a new page if a skip to a new page has been specified.

When the fetch overflow routine is not used and the overflow is not specified for the file, the following occurs when the overflow line is sensed:

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.

If you do not want all of the remaining detail and total lines printed on the page before overflow lines are printed and forms advance to the new page, you may cause overflow lines to be printed ahead of the usual time. This is known as fetching the overflow routine and is indicated by the entry in column 16. Overflow is fetched only if all conditions specified by the indicators in columns 23–31 are met and an overflow has occurred. See *Overflow Indicators* under *Supplementary Information* for detailed information and examples of a fetched overflow routine.

The fetched overflow routine does not automatically cause forms to advance. A skip to line 01 (new page) must also be specified on a line conditioned by the overflow indicator.

F must be entered in column 16 of each OR line if you want to fetch the overflow routine for each record in the OR relationship.

COLUMNS 17–22 (SPACING AND SKIPPING)

Columns 17–22 are used to specify spacing and line skipping for printer files and CRT files, and spacing only for console files and ledger files. Figure 84 shows the possible spacing and skipping entries for the various files.

If columns 17–22 are blank, single spacing occurs after each line is printed. Otherwise, the system performs exactly the spacing and skipping you specify. There may be as many as six spaces (three before, three after) between two lines of printing on the printer or the console.

Files	Space Before	Space After	Skip Before *	Skip After *
	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
Ledger	Not Used	0–9	Not Used	Not Used
CRT	0–3	0–3	01	Not Used
Console	0–3	0–3	Not Used	Not Used

* The skip entries you specify in columns 19–22 must not exceed the forms length specified in line counter specifications or must not exceed 66 if no line counter specifications are supplied.

Figure 84. Possible Spacing and Skipping Entries

You may save time by specifying that spacing or skipping should be done after printing. This means that the output file does not have to wait for paper movement before it can print.

You may specify different spacing and skipping on OR lines. If there are no spacing or skipping entries in the OR line, spacing and skipping is done according to the entries in the line preceding the OR line.

You may want to specify a skip to a new page at the beginning of each job to prevent OCL statements from printing on the same page as output data. If no space or skip before entry is specified on the first output line, the first output line will print on top of the last OCL statement (*// RUN*). When specifying spacing and skipping, remember that if the last output line for the job is printed on line 01 of a page and no printer movement is specified, the system begins printing OCL statements for the next job on the next line. If this is not desirable, specify the appropriate printer movement.

COLUMN 17 (SPACE BEFORE)

<i>Entry</i>	<i>Explanation</i>
0-3	Number of lines to be spaced for console files, CRT files, or printer files.

Note: A blank or 0 entry for console files results in a single space.

COLUMN 18 (SPACE AFTER)

<i>Entry</i>	<i>Explanation</i>
0-3	Number of lines to be spaced for console files, CRT files, or printer files.
0-9	Number of lines to be spaced for ledger files.

COLUMNS 19-20 (SKIP BEFORE)

<i>Entry</i>	<i>Explanation</i>
01	CRT screen is to be blanked immediately.
01-99	Skip to lines 01-99 for printer files.
A0-A9	Skip to lines 100-109 for printer files.
B0-B2	Skip to lines 110-112 for printer files.

COLUMNS 21-22 (SKIP AFTER)

<i>Entry</i>	<i>Explanation</i>
01-99	Skip to lines 01-99 for printer files.
A0-A9	Skip to lines 100-109 for printer files.
B0-B2	Skip to lines 110-112 for printer files.

Spacing and Skipping for Printer Files

Line spacing and skipping may be specified both before and after printing of a line. If both spacing and skipping are specified on the same line, they are done in this order:

1. Skip before
2. Space before
3. Skip after
4. Space after

Spacing or skipping to the overflow line or past the overflow line causes the overflow indicator to turn on. Skipping past the overflow line to a line on the next page, however, does not cause the overflow indicator to turn on.

If you want to turn on the overflow indicator to condition overflow operations when you skip to a lower line number (higher position) on the next page from a line above the overflow line, you may:

- Use a SETON operation, or
- Specify two skips (a skip to the overflow line, then to the first printing line on the next page).

This is necessary because the overflow indicator will not be turned on if the skip to a new page occurs on a non-overflow line.

Skipping before and after refers to jumping from one printing line to another without stopping at lines between. This is usually done when a new page is needed. A skip to a lower line number means advance to a new page. Skipping may also be used, however, when a great deal of space is needed between lines. The entry must be the two-digit number which indicates the number of the next line to be printed. You may indicate that skipping should be done before (columns 19–20) or after (columns 21–22) a line is printed. If you specify a skip to the same line number the forms are positioned on, no movement of the paper occurs.

Spacing for Ledger Files

The following rules apply to spacing for ledger files:

1. A space before entry must not be made in column 17.
2. A space after entry (0–9) should be made in column 18. If no entry is made, the system assumes an entry of 1.

Skipping cannot be specified for ledger files.

Spacing and Skipping for CRT Files

The following rules apply to spacing and skipping for CRT files.

1. A space before entry (0–3) can be made in column 17.
2. A space after entry (0–3) can be made in column 18. If columns 17–22 are blank, the system assumes an entry of 1.
3. A skip before to line 01 only can be specified in columns 19–20. Enter 01 in these columns whenever you wish to clear the CRT screen.
4. A skip after entry (columns 21–22) must not be specified for CRT files.

Spacing for Console Files

The following rules apply to spacing for console files:

1. A space before entry (0–3) can be made in column 17. The console will always space before printing, due to the carriage return mechanism. Therefore, a Space Before entry of blank, zero, or one will result in a single space before printing.
2. A space after entry (0–3) can be made in column 18.

Skipping cannot be specified for console files.

COLUMNS 23–31 (OUTPUT INDICATORS)

<i>Entry</i>	<i>Explanation</i>
01–99	Any resulting indicator, field indicator, or record identifying indicator previously specified.
KA–KN, KP, KQ	Any command key indicator previously specified.
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.

Use output indicators to give the conditions under which output operations are to be done. More specifically, use them to tell:

1. When you want to output a line (see *Examples, Example 1*).
2. When you want to output a field (see *Examples, Example 2*).

When you use an indicator to condition an entire line of print, place it on the line which specified the type of record (Figure 85, insert A). Place an indicator which conditions when a field is to be printed on the same line as the field name (Figure 85, insert B).

There are three separate output indicator fields (columns 23–25, 26–28, and 29–31). One indicator may be entered in each field. If these indicators are on, the output operation will be done. An N in the column (23, 26, or 29) preceding each indicator means that the output operation will be done only if the indicator is not on. No output line should be conditioned by all negative indicators (at least one of the indicators used should be positive). If all negative indicators do condition a heading or detail operation, the operation is performed at the beginning of the program cycle when 1P lines are written.

IBM International Business Machines Corporation Form X21-9090
Printed in U.S.A.

RPG OUTPUT - FORMAT SPECIFICATIONS

Date _____ Page 1 2 Program Identification 75 76 77 78 79 80

Program _____

Programmer _____

Line	Form Type	Filename	Type (H/D/T/E)			Space		Skip		Output Indicators			Field Name	End Position in Output Record	Sterling Sign Position
			Stacker Select/ Fetch Overflow (F)	Before	After	Before	After	Not	And	And	Not	Not			
0 1	O	PRINT	D			1					44				
0 2	O											INVOIC	10		
0 3	O											AMOUNT	18		
0 4	O											CUSTR	65		
0 5	O											SALSMN	85		

Edit Codes

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

Constant or Edit Word

A

IBM International Business Machines Corporation Form X21-9090
Printed in U.S.A.

RPG OUTPUT - FORMAT SPECIFICATIONS

Date _____ Page 1 2 Program Identification 75 76 77 78 79 80

Program _____

Programmer _____

Line	Form Type	Filename	Type (H/D/T/E)			Space		Skip		Output Indicators			Field Name	End Position in Output Record	Sterling Sign Position
			Stacker Select/ Fetch Overflow (F)	Before	After	Before	After	Not	And	And	Not	Not			
0 1	O	PRINT	D			1					44				
0 2	O											INVOIC	10		
0 3	O											AMOUNT	18		
0 4	O											CUSTR	65		
0 5	O										LI	SALSMN	85		

Edit Codes

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

Constant or Edit Word

B

Figure 85. Output Indicators

AND and OR Lines

If you need to use more than three indicators to condition an output operation, you may use an AND line. Enter the word AND in columns 14–16 and as many indicators as needed. The condition for all indicators in an AND relationship must be satisfied before the output operation is done. A maximum of twenty AND lines may be used for an output operation if no OR lines are used.

Output indicators may also be in an OR relationship. If one or the other condition is met, the output operation will be done. A maximum of twenty OR lines may 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 twenty.

AND and OR lines are used to condition entire output lines; they must not be used to condition fields (see *Examples, Example 3*).

The use of an L0-L9 indicator in an OR relationship with an LR indicator can result in the specified operation being done twice when LR is on. One operation will be performed during LR processing and the other at detail or total time. The following example shows how to correctly use the L0-L9 indicators in an OR relationship.

IBM		International Business Machines Corporation																																		
		RPG															OUTPUT - F																			
Date _____																																				
Program _____																	Punching Instruction Graphic																			
Programmer _____																	Punch																			
Line	Form Type	Filename	Type (H/D/T/E)	Stacker Select/Fatal Overflow (E)	Space		Skip		Output Indicators						Field Name																					
					Before	After	Before	After	And	And	And	And	And	And																						
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		
0	1	O	OUT																																	
0	2	O		OR																																
0	3	O																																		
0	4	O																																		

Command Key Indicators (KA–KN, KP, KQ)

Command key indicators may be used in columns 23–31 to condition output operations; however, any command keys entered in these columns must also be specified in columns 54–59 of the Calculation sheet for a SET or SETOF operation. See *Operation Codes* under *Supplementary Information* for complete information on each specific operation.

Overflow Indicators (OA–OG, OV)

Overflow indicators are used to condition output operations on the printer or ledger card device. The operations conditioned by the overflow indicator are done only after the overflow line (end of page) has been reached.

If you have not assigned an overflow indicator to the printer file in the file description specifications, you may not use an overflow indicator in the output-format specifications. In this case, advancing the forms to a new page is handled by the compiler, even though no overflow indicator has been assigned.

If an overflow indicator has been assigned in file description specifications and a specification line not conditioned by an overflow indicator specifies a skip to a line on a new page, the overflow indicator turns off before forms advance to a new page.

An overflow indicator may appear on either AND or OR lines. However, only one overflow indicator may be associated with one group of output indicators. That overflow indicator must also be the same indicator associated with the file on the File Description sheet.

When the overflow indicator is used in an AND relationship with a record identifying indicator, unusual results are often obtained. This is because the record type might not be the one read when overflow has occurred. Thus, the record type indicator is not on and all lines conditioned by both overflow and record type indicators do not print.

An overflow indicator cannot condition an exception line (E in column 15) but may condition fields within the exception record. An overflow indicator must not be specified for ledger card output files.

First Page Indicator (1P)

The first page (1P) indicator is usually used to allow printing on the first page. It may also be used in connection with the overflow indicator to allow printing on every page (see *Examples, Example 4*). The information printed out on the line conditioned by the 1P indicator must be constant information used as headings or fields for reserved words, such as PAGE and UDATE. The constant information is specified on the Output-Format sheet.

The 1P indicator is used only with heading or detail output lines. It cannot be used to condition total or exception output lines. The 1P indicator cannot be used in an AND relationship with control level indicators. See *Column 41*, under *Control Card Specification* for information on forms alignment of the first page.

Halt Indicators (H1–H9)

On certain error conditions, you may not want output performed. Halt indicators can be used to prevent the data that caused the error from being used (see *Examples, Example 5*).

External Indicators (U1–U8)

A file named in the output-format specifications may be conditioned by an external indicator in the file description specifications. In this case, every output record for that file must be conditioned by the same external indicator used in the file description specifications.

Examples

Example 1: Figure 85, insert A, shows the use of one indicator to condition an entire line of printing. When 44 is on, the fields named INVOIC, AMOUNT, CUSTR, and SALSMN are all printed.

Example 2: Figure 85, insert B, shows the use of a control level indicator 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.

Example 3: The use of indicators in both AND and OR lines to condition an output line is shown by Figure 86, insert A. The specifications in lines 01–04 say that the detail line is printed if either one 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.

A maximum of three indicators may be used on the Output-Format sheet to condition a field since AND and OR lines may not be used to condition an output field (Figure 86, insert B).

However, you can condition an output field with more than three indicators by using the SETON operation in calculations. For instance, indicators 10, 12, 14, 16, and 18 are to condition an output field named PAY. In calculation specifications, you can SETON indicator 20 if indicators 10, 12, and 14 are on. Then condition the output field PAY on indicators 20, 16, and 18 on the Output-Format sheet.

IBM International Business Machines Corporation Form X21-9090 Printed in U.S.A.

RPG OUTPUT - FORMAT SPECIFICATIONS

Date _____ Program _____ Programmer _____

Page 1 2 Program Identification 75 76 77 78 79 80

Line	Form Type	Filename	Type (H/D/T/E)				Space		Skip		Output Indicators			Field Name	Edit Codes	End Position in Output Record	Sterling Sign Position
			Stacker Select/Fetch Overflow (F)	Before	After	Before	After	Not	And	And	Not						
01	O	TRSACTW D									21	40	01				
02	O										16						
03	O										21	40	N01				
04	O										N16						
05	O													NAME	15		
06	O													ACCTNO	25		
07	O													ADDR	60		
08	O													BALNC	70		
09	O																
10	O																

Edit Codes					
Commas	Zero Balances to Print	No Sign	CR	-	X
Yes	Yes	1	A	J	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	

Constant or Edit Word

A

IBM International Business Machines Corporation Form X21-9090 Printed in U.S.A.

RPG OUTPUT - FORMAT SPECIFICATIONS

Date _____ Program _____ Programmer _____

Page 1 2 Program Identification 75 76 77 78 79 80

Line	Form Type	Filename	Type (H/D/T/E)				Space		Skip		Output Indicators			Field Name	Edit Codes	End Position in Output Record	Sterling Sign Position
			Stacker Select/Fetch Overflow (F)	Before	After	Before	After	Not	And	And	Not						
01	O	TRSACTW D									21	40	01				
02	O										16						
03	O										21	40	N01				
04	O										N16						
05	O													NAME	15		
06	O													ACCTNO	25		
07	O													ADDR	60		
08	O													MR L1 02BALNC	70		
09	O																
10	O																

Edit Codes					
Commas	Zero Balances to Print	No Sign	CR	-	X
Yes	Yes	1	A	J	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	

Constant or Edit Word

A maximum of three indicators may be used to condition a field.

B

Figure 86. Output Indicators in AND and OR Lines

Example 5: Figure 88 shows coding necessary to check for an error condition and to stop processing on and printing from the record in error. If FIELDDB contains all zeros, halt indicator H1 turns on (see line 03 of Figure 88, insert A). In the calculation specifications, if H1 is on, all calculations are bypassed (see line 01 of Figure 88, insert B).

On the output-format specifications, FIELD A and FIELD B are printed only if H1 is not on (see lines 04 and 05 of Figure 88, insert C). Therefore, if indicator H1 is on, fields A and B are not printed. Use this general format when you do not want information that is in error to be printed.

IBM International Business Machines Corporation Form X21-9094 Printed in U.S.A.

RPG INPUT SPECIFICATIONS

Date _____ Page 1 2 Program Identification 75 76 77 78 79 80

Program _____

Programmer _____

Punching Instruction		Graphic		Punch	
----------------------	--	---------	--	-------	--

Line	Form Type	Filename	Sequence	Number (1-N)	Option (O)	Record Identifying Indicator	Record Identification Codes			Field Location		Field Name	Control Level (L1-L9)	Matching Fields or Chaining Fields	Field Record Relation	Field Indicators			Sterling Sign Position
							1	2	3	From	To					Plus	Minus	Zero or Blank	
01	I	READER	AA			01				1	3	FIELDAL1							
02	I									4	8	FIELDDB						H1	

(A) When an error condition (zero in FIELDDB) is found, the halt indicator turns on.

IBM International Business Machines Corporation Form X21-9093 Printed in U.S.A.

RPG CALCULATION SPECIFICATIONS

Date _____ Page 1 2 Program Identification 75 76 77 78 79 80

Program _____

Programmer _____

Line	Form Type	Control Level (L0-L9, LR, SR)	Indicators			Factor 1	Operation	Factor 2	Result Field	Field Length	Decimal Positions	Half Adjust (H)	Resulting Indicators			Comments
			And	And	Not								Plus	Minus	Zero	

01	C							GOTO END									
02	C		H1					⋮									
03	C		01														
	C		01														
	C		01														
	C					END	TAG										

(B) When H1 is on, all calculations are bypassed.

IBM International Business Machines Corporation Form X21-9090 Printed in U.S.A.

RPG OUTPUT - FORMAT SPECIFICATIONS

Date _____ Page 1 2 Program Identification 75 76 77 78 79 80

Program _____

Programmer _____

Line	Form Type	Filename	Type (H/D/T/E)	Space	Skip	Output Indicators			Field Name	Edit Codes	End Position in Output Record	Constant or Edit Word	Sterling Sign Position
						And	And	Not					

Edit Codes			
Commas	Zero Balances to Print	No Sign	CR
Yes	Yes	1	A
Yes	No	2	B
No	Yes	3	C
No	No	4	D

Constant or Edit Word			
X	Y	Z	
Remove Plus Sign	Date	Zero	Suppress

01	O	PRINTER	H	0201		L1						
02	O		D	0		INH1		5	'HEADING'			
	O					FIELDA		5				
	O					FIELDDBZ		15				

(C)

Figure 88. Preventing Fields From Printing

COLUMNS 32–37 (FIELD NAME)

In columns 32–37, use one of the following to name every field that is to be written out:

1. Any field name previously used in this program.
2. The special words PAGE, PAGE1, PAGE2, *PLACE, UPDATE, UDAY, UMONTH, and UYEAR.
3. A table name, array name, or array element.

Field Names

The field names used are the same as the field names on the Input sheet (columns 53–58) or the Calculation sheet (columns 43–48). Do not use these columns if a constant is used (see *Columns 45–70*). If a field name is entered in columns 32–37, columns 7–22 must be blank.

Fields may be listed on the sheet in any order since the sequence in which they appear on the printed form is determined by the entry in columns 40–43. However, they are usually listed sequentially. If fields overlap, the last field specified is the only field completely printed.

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*).

Special Words

Page Numbering (PAGE, PAGE1, PAGE2)

PAGE is a special word which, when used, causes automatic numbering of your pages. Enter the word PAGE,

PAGE1, or PAGE2 in these columns if you wish pages to be numbered. When a PAGE field is named in these columns without being defined elsewhere, it is assumed to be a four column, numeric field with no decimal position. Leading zeros are suppressed automatically.

The page number starts with 1 unless otherwise specified, and 1 is automatically added for each new page. See *Columns 53–58* under *Input Specifications* for information concerning page numbering starting at a number other than 1.

It is possible at any point in your job to restart the page numbering sequence. To do this, set the PAGE field to zero before it is printed. One method of setting the PAGE field to zero is to use Blank After (see *Column 39*). Another way is to use an output indicator. If the status of the indicator is as specified, the PAGE field is reset to zero. Remember that 1 is added to the PAGE field before it is printed (see *Examples, Example 1*).

The three possible PAGE entries (PAGE, PAGE1, and PAGE2) may be needed for naming different output files. Do not use the same name for two different output files.

Note: A PAGE field named only in output specifications is four characters long and need not be defined elsewhere. However, a PAGE field can be defined in input or calculation specifications and may be of any length. These PAGE fields are treated exactly as if they were named only in output specifications except for the difference in field length.

Repeating Output Fields (*PLACE)

*PLACE is a special RPG II word which makes it possible to write or punch the same field in several locations on one record without having to name the field and give its end position each time the field is to be written or punched. The fields are written or punched in the same relative positions ending in the column specified by *PLACE. For example, if you wish FIELDS A,

B, and C to appear twice on one line, you can specify this in two ways:

1. Define each field and its corresponding end position each time it is to be printed (Figure 89, insert A).
2. Use the special word *PLACE (see Figure 89, insert B).

IBM International Business Machines Corporation Form X21-9090 Printed in U.S.A.

RPG OUTPUT - FORMAT SPECIFICATIONS

Date _____ Program _____ Programmer _____

Punching Instruction: Graphic _____ Punch _____

Page 1 2 Program Identification 75 76 77 78 79 80

Line	Form Type	Filename	Type (H/D/T/E)	Space		Skip		Output Indicators			Field Name	Edit Codes	Blank After (B)	End Position in Output Record	P = Packed/B = Binary	Constant or Edit Word	Sterling Sign Position
				Before	After	Before	After	Not	And	And							
0 1	O	PRINT	D														
0 2	O										FIELD A		10				
0 3	O										FIELD B		20				
0 4	O										FIELD C		30				
0 5	O										FIELD A		40				
0 6	O										FIELD B		50				
0 7	O										FIELD C		60				
0 8	O										FIELD D		75				
0 9	O																

A

IBM International Business Machines Corporation Form X21-9090 Printed in U.S.A.

RPG OUTPUT - FORMAT SPECIFICATIONS

Date _____ Program _____ Programmer _____

Punching Instruction: Graphic _____ Punch _____

Page 1 2 Program Identification 75 76 77 78 79 80

Line	Form Type	Filename	Type (H/D/T/E)	Space		Skip		Output Indicators			Field Name	Edit Codes	Blank After (B)	End Position in Output Record	P = Packed/B = Binary	Constant or Edit Word	Sterling Sign Position
				Before	After	Before	After	Not	And	And							
0 1	O	PRINT	D														
0 2	O										FIELD A		10				
0 3	O										FIELD B		20				
0 4	O										FIELD C		30				
0 5	O										*PLACE		60				
0 6	O										FIELD D		75				
0 7	O																

B

Figure 89. Printing Fields Twice on the Same Line

Both coding methods produce a line which looks like this:

Print Positions	10	20	30	40	50	60	75
Fields	FIELDA	FIELDB	FIELD C	FIELDA	FIELDB	FIELD C	FIELD D

However, it is easy to see that using the special word *PLACE saves extra coding.

When using *PLACE, all fields named for each record type (H/D/T/E) are written or punched as usual in the location specified. The entry *PLACE then causes all of these same fields to be written or punched ending at the position specified in the *PLACE statements. When using *PLACE, remember:

1. *PLACE must be specified after the field names which are to be placed in different positions in one line (see *Examples, Example 2*).
2. *PLACE causes all fields (in a record type) above the *PLACE entry to be written or punched, not just the field named on the line above *PLACE.
3. *PLACE must appear on a separate specification line for every additional time you want the field or group of fields written or punched.
4. An end position no greater than 256 must be specified for every *PLACE line. Be sure to allow enough space for all fields prior to the *PLACE to be printed again (see *Examples, Example 2*). Otherwise overlapping occurs.
5. The leftmost position of the fields to be moved by the *PLACE specification is always assumed to be position 1.
6. When *PLACE is specified for card output, the fields named above *PLACE will be repunched.
7. 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 which has an end position greater than the end position of the whole array. This field can be a one-position blank constant.
8. Additional fields or constants may be specified after *PLACE. These fields are not affected by any *PLACE above them.

Note: Attempts to use the *PLACE function for other than its defined purpose may produce unpredictable results.

Date Fields (UPDATE, UMONTH, UDAY, UYEAR)

Often you want the date to appear on your printed report, punched card, or program listing. Use special words UPDATE, UMONTH, UDAY, and UYEAR to get the date field you desire. The date fields are established at job setup time. The following rules apply to date fields:

1. UPDATE gives a six-character numeric date field in one of three formats:
 - a. Domestic (mmddyy).
 - b. United Kingdom (ddmmy).
 - c. World Trade (ddmmyy).

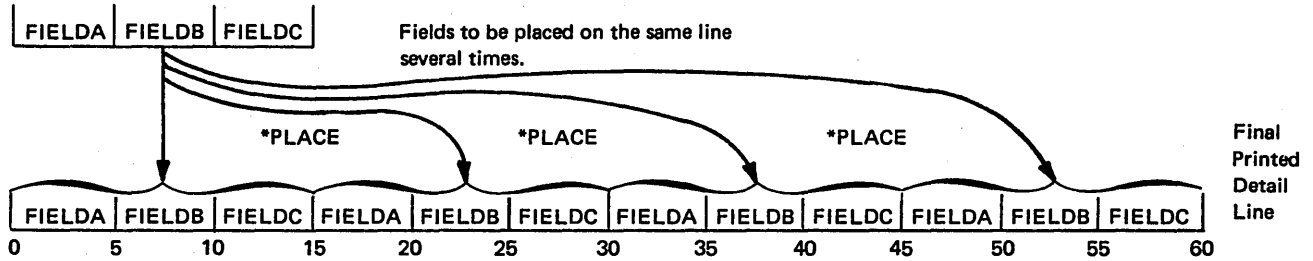
The format is specified by an entry in column 21 of the control card specifications.

2. UDAY may be used for days only, UMONTH for months only, and UYEAR for years only.
3. These fields may not be changed by any operations specified in the program. Thus, these fields are usually used only in compare and test operations.

Example 2: Figure 91 shows the use of the special word *PLACE to print the same fields several times on the same line. Fields A, B, and C are to be printed four times on one line (Figure 91, insert A). They are printed once when they are named and once for every *PLACE entry. In Figure 91, insert B, *PLACE is specified after the fields which are to be printed several times on the same line. 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 all allows room for the printing of 15 characters. This eliminates any overlapping.



A

IBM International Business Machines Corporation Form X21-9090 Printed in U.S.A.

RPG OUTPUT - FORMAT SPECIFICATIONS

Date _____ Program _____ Programmer _____

Punching Instruction: Graphic _____ Punch _____

Page 1 2 Program Identification 75 76 77 78 79 80

Line	Form Type	Filename	Type (H/D/T/E)	Space			Skip			Output Indicators			Field Name	Edit Codes	End Position in Output Record	Sterling Sign Position
				Before	Alter	After	Before	Alter	After	Not	And	And				
01	O	PRINT	H		3											
02	O															
03	O		D		1											'SUMMARY'
04	O												FIELD A		5	
05	O												FIELD B		10	
06	O												FIELD C		15	
07	O												*PLACE		30	
08	O												*PLACE		45	
09	O												*PLACE		60	
10	O		T							L2						
11	O												FIELD D		85	
12	O															

Edit Codes

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

Constant or Edit Word

B

Figure 91. *PLACE

COLUMN 38 (EDIT CODES)

Use column 38 when you want to:

1. Suppress leading zeros of a numeric field
2. Omit a sign from the low order position of a numeric field.
3. Punctuate a numeric field without setting up your own edit word.

A table summarizing the edit codes that can be used is printed above columns 45–70 on the Output-Format sheet. Each edit code punctuates differently. If you use an edit code in column 38, columns 45–70 must be blank except for the following condition. If asterisk fill or a floating dollar sign is required, enter '*' or '\$' in columns 45–47. When an edit code is used to punctuate an array, two spaces are automatically left between fields of the array to the left of each element. Only unpacked numeric fields can be edited.

Figure 92 shows the edit codes and the options they provide. Figure 93 illustrates how data looks when it is edited by edit codes. Figure 94 shows editing for date fields. Each code punctuates the field a little differently. All codes suppress leading zeros, with the following exception. One of the World Trade formats for output is a J entry in column 21 in the control card specifications. For this J entry, all zero balances and balances with zero

values to the left of the decimal comma are always written or punched with one leading zero (such as 0,00 or 0,04). If an edit code is specified on the Output-Format sheet and the edit code is to print zero balances, a zero balance field will always have a zero to the left of the decimal comma. The edit code cannot suppress this zero.

Normally, when you use an edit code in column 38, you cannot define an edit word in columns 45–70; however, there are two exceptions:

- If you want leading zeros replaced by asterisks, enter '*' in columns 45–47 of the line containing the edit code.
- If you want a dollar sign to appear before the first digit in the field (floating dollar sign), enter '\$' in columns 45–47 of the line containing the edit code.

It is also possible to have a dollar sign appear before the asterisk fill (fixed dollar sign). This is done in the following manner:

1. Place a dollar sign constant one space before the beginning of the edited field.
2. Place '*' in column 45–47 of the line containing the edit code.

Edit Code	Commas	Decimal Point	Sign For Negative Balance			Print Out On Zero Balance *			Zero Suppress
			No Sign	CR	-- (Minus)	Domestic United Kingdom	World Trade I	World Trade J	
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			--	.00 or 0	,00 or 0	0,00 or 0	Yes
K	Yes	Yes			--	Blanks	Blanks	Blanks	Yes
L		Yes			--	.00 or 0	,00 or 0	0,00 or 0	Yes
M		Yes			--	Blanks	Blanks	Blanks	Yes
X **									
Y ***									Yes
Z									Yes

* Zero balances for the World Trade format are printed or punched in two ways, depending on the entry made in column 21 of the control card specifications.

** The X code performs no editing.

*** 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 92. Edit Codes

Figure 93. Examples of Edit Code Usage

Edit Codes	Positive Number - Two Decimal Positions	Positive Number- No Decimal Positions	Negative Number * Three Decimal Positions	Negative Number * No Decimal Positions	Zero Balance - Two Decimal Positions **			Zero Balance - No Decimal Positions
					Domestic United Kindom	World Trade /	World Trade J	
Unedited	1234567	1234567	00012 }	00012 }	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 }	00012 }	000000	000000	000000	000000
Y								0/00/00
Z	1234567	1234567	120	120				

* The character } is a negative zero.

** Zero balances for the World Trade format are printed or punched in two ways, depending on the entry made in column 21 of the control card specifications.

Date	Edit Code	Domestic Format	United Kingdom Format	World Trade / Format	World Trade J Format
January 30, 1970	Y	1/30/70	30/01/70	30.01.70	30.01.70

Figure 94. Date Fields

Edit Codes	Negative Number - Two Decimal Positions - End Position Specified as 10.									
	Output Print Positions									
	3	4	5	6	7	8	9	10	11	
Unedited				0	0	4	1	K*		
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*		
Y		.	0	/	4	1	/	2		
Z					4	1	2			

* K represents a negative 2.

Figure 95. Effect of Edit Codes on End Position

Figure 95 shows the effect the different edit codes will have on the same field with a specified end position for output.

COLUMN 39 (BLANK AFTER)

Entry *Explanation*

Blank Field is not to be reset.

B Field specified in columns 32–37 is to be reset after the output operation is complete.

Use column 39 to reset a field to zeros or blanks. Numeric fields are set to zero and alphameric fields are set to blanks. This column must be blank for look-ahead and UDATE fields.

Resetting fields to zeros is useful when you are accumulating and printing totals for each control group. After finding the total for one group and printing it, you want to start accumulating totals for the next group. Before you do this, however, you want your total field to start with zeros, not with the total it had for the previous group. Column 39 will reset the total field to zero after it is printed.

Note: If Blank After (column 39) is specified for a field to be punched, printed, or written more than once, the B entry should be entered on the last line specifying output for that field.

COLUMNS 40–43 (END POSITION IN OUTPUT RECORD)

<i>Entry</i>	<i>Explanation</i>
1–4096	End position for disk.
90–214	End position for ledger.
1–220 or 1–132	End position for 220 position printer (without ledger) or end position for 132 position printer.
1–96	End position for data recorder.
1–125	End position for console.
N001– N009	End position for ledger card control number (N in column 40, and 1-9 in column 43).
1–64	End position for CRT.

Columns 40–43 apply to all output devices. These columns are used to define the end position on the output record of a field or constant. All entries in these columns must end in column 43. Enter only the number of the end position of the rightmost character in the field or constant. When writing a control number for a new ledger card, the control number must be specified on the first output lines for the new ledger card.

Repeating Output Fields (*PLACE)

When *PLACE is specified for the printer (see *Columns 32–37*), end position indicates the end position of the

last field of the group that is to be printed. Thus you must be sure you have indicated an end position that allows enough room for all specified fields to be printed. Be sure to allow enough space (as indicated by end position entries) on your output record to hold edited fields.

COLUMN 44 (PACKED OR BINARY FIELD)

<i>Entry</i>	<i>Explanation</i>
Blank	Field is unpacked numeric data, alphameric data, or to be printed or punched.
P	Field is to be written on disk in the packed decimal format.
B	Field is to be written on disk in the binary format.

Column 44 must have an entry if a numeric field (decimal number) is to be written on disk in packed decimal or binary format. Packed decimal and binary fields cannot be printed or punched; these fields can only be written on disk. Column 44 must be blank for *PLACE.

After decimal numbers have been processed, they may be left in the unpacked format. However, for efficient use of disk space, decimal numbers can be converted into packed decimal or binary format. When binary output is specified, a numeric field 1–4 bytes long (unpacked in storage) is converted into a 2-byte binary field when it is written on disk; a numeric field 5–9 bytes 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* under *Input Specifications* for a description of how data fields are represented in unpacked decimal, packed decimal, and binary formats.

Note: Although packed and binary fields require less disk storage space, the conversion routines needed to handle such data will increase the object program size.

COLUMNS 45-70 (CONSTANT OR EDIT WORD)

Use columns 45-70 to specify a constant or an edit word.

Constant

A constant is any unchanging information that is entered by a specification. Constants are usually words used for report headings or column headings.

The following rules apply to constants (Figure 96 contains examples):

1. Field name (columns 32-37) must be blank.
2. A constant must be enclosed in apostrophes. Enter the leading apostrophe in column 45.
3. 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'.
4. Numeric data may be used as a constant.
5. Up to 24 characters of constant information can be placed in one line. Additional lines may be used, but each line must be treated as a separate line of constants. The end position of each line must appear in columns 40-43.

International Business Machines Corporation Form X21-9090
Printed in U.S.A.

RPG OUTPUT - FORMAT SPECIFICATIONS

Punching Instruction	Graphic												
	Punch												

Page 1 2 Program Identification 75 76 77 78 79 80

Output Indicators		Field Name	Edit Codes Blank, Alter (B)	End Position in Output Record	Constant or Edit Word	Sterling Sign Position																																																							
d	And																																																												
Not	Not																																																												
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													
														10					'AMOUNT'																																										
														30					'SALESMAN''S TOTALS'																																										
														40					'147679'																																										
														50					'2.50'																																										
														80					'SALES ANALYSIS BY CUSTOM'																																										
														104					'ER AND SALESMAN FOR MONT'																																										
														114					'H OF MARCH'																																										

Figure 96. Examples of Output Constants

Edit Words

An edit word gives you more flexibility in punctuating a numeric field than an edit code. You specify directly whether commas, decimal points, and zero suppression are needed; if the negative sign should print; and, if the output is dollars and cents, whether you want a dollar sign and leading asterisks.

The following rules apply to edit words:

1. Column 38 (edit codes) must not be used.
2. Columns 32–37 (field name) must contain the name of a numeric field.
3. Columns 40–43 (end position in output record) must contain an entry.
4. An edit word must be enclosed in apostrophes. Enter leading apostrophe in column 45. The edit word itself must begin in column 46.
5. Any printable character is valid, but certain characters in certain positions have special uses (see *Editing Considerations* in the following text).
6. An edit word cannot be longer than 24 characters.
7. The number of replaceable characters in the edit word must be equal to the length of the field to be edited (see *Editing Considerations* in the following text).
8. All leading zeros are suppressed unless a zero or asterisk is specified in the edit word. The zero or asterisk indicates the last leading zero in the field to be replaced by a blank or asterisk.
9. Any zeros or asterisks following the leftmost zero or asterisk are treated as constants (they are not replaceable characters).

Editing Considerations

One important thing to keep in mind is that you must leave exactly enough room on the printed form for the edited word. If the field to be edited is seven characters long on the input record, consider whether seven positions allows enough space for it to print on the report. By the time the field is edited, it may contain many more characters than seven.

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 be equal to the length of the field to be edited. A replaceable character is a part of the edit word that does not require a position in the output file. The replaceable characters are:

- ∅ (if used for zero suppression)
- *
- b (blank)
- \$ (if it appears immediately to the left of zero suppress: a floating dollar sign)

A fixed dollar sign, decimal points, floating dollar sign, commas, ampersands (representing blanks), negative signs (- or CR) and constant information all require space in the output field.

Note: There are two exceptions to the rule that the number or replaceable characters in the edit word must be equal to the length of the field to be edited. The exceptions are:

1. An extra space must be left in the edit word for the floating dollar sign. This ensures a print position for the dollar sign if the output field is full.

<i>Unedited Field</i>	<i>Edit Word</i>	<i>Edited Field</i>	<i>Unedited Field Length</i>	<i>Replaceable Characters in Edit Word</i>
72432N	' <i>\$\$\$,\$\$Q.##&-</i> '	\$ 7,243.25-	6	7

2. 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 will not be zero suppressed by the leading zero, but all other specified editing will be performed.

<i>Unedited Field</i>	<i>Edit Word</i>	<i>Edited Field</i>	<i>Unedited Field Length</i>	<i>Replaceable Characters in Edit Word</i>
00746J	' <i>0###.###</i> '	007,461	6	7

Formatting Edit Words

The Printer Spacing Chart is helpful when forming edit words. Figure 97 shows how an output line can be formatted using this chart. Note that X's and zeros are used to show field positions. A zero indicates where zero suppression will stop. An X indicates any number can appear in the position. Use blanks in place of the X's when writing the edit words. Two additional X's are provided for percent profit or loss since a negative value must be recognizable.

If it is necessary to show a negative number, a sign must be included in the edit word. You may use either the minus sign (-) or the letters CR. These print only for a negative number; however, the character positions they require must be taken into consideration when entering the end position of the field on the Output-Format sheet. Figure 97 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 looks like this:

25CR

If PERCPL was positive, CR would not print and the same field would appear as:

25

You may also use a minus sign to indicate a negative balance. If you want to leave a space between the number and the negative sign, place an ampersand (&) in the edit word before the minus sign. PERCPL would then print as:

25&-

Unedited Data

Item number - 000241
 Item cost - 02000
 Selling price - 02200
 % profit or loss - 25

Printer Spacing Chart

	0									1									2									3									4												
	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
1																																																	
2																																																	
3																																																	
4																																																	
5																																																	
6																																																	
7																																																	
8																																																	
9																																																	
10																																																	
11																																																	
12																																																	

Output - Format Sheet

Indicators	Field Name	Edit Codes Blank After (B) End Position in Output Record P = Packed/B = Binary	Edit Codes						Constant or Edit Word
			Commas	Zero Balances to Print	No Sign	CR	-	X = Remove Plus Sign	
28	INUMB	7							
31	ICOST	18							
38	\$PRICE	27							
44	PERCPL	37							

Edited Output

241 \$ 20.00 \$ 22.00 25

Figure 97. Using the Printer Spacing Chart to Format Data

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	Edit Word	Example Number	Source Data	Appears in Output Record as:
	*0CR	25	0000000000 -	*****0CR
	*0CR	26	0000000000 -	*****00CR
*		27	0000135678 -	*000135678
*		28	1234567890 +	1234567890
	*1	29	0000135678 -	****135678
,	&CR&NET	30	0000135678 -	1,356.78CRNET
,	&CR&-NET	31	0000135678	1,356.78--NET
\$&0,	&NET	32	0000135678 +	\$0,001,356.78NET
,	\$0 - &NET	33	0000000005	0.05NET
,	\$0.	34	0000000005	\$.05
,	\$0. -	35	1234567890 -	\$12,345,678.90-
,	\$0. CR	36	0001356789 -	\$13,567.89CR
,	*. *CR**	37	0000135678 +	****1,356.78**
,	0. &CR*	38	00000000 -	0.00CR*
,	*. -	39	0000000000 -	*****.00-
,	\$0. - SALES	40	0000001234	\$,012.34SALES
\$&	0. CR	41	1234567890 -	\$12,345,678.90CR
,	-OLD&BALNCE	42	1234567890 -	1,234,567,890-OLDBALANCE
,	0 -OLD&BALNCE	43	0000000000 +	000000000000OLDBALANCE
,	DOLLARS CENTS	44	0000135678	1,356DOLLARS78CENTS
,	DOLLARS CENTS	45	000000 +	000000000000CENTS
,	0 DOLLARS CENTS &CR	46	000000	0DOLLARS0000000000
0	LBS. & OZ. TARE&-	47	000002 +	0LBS.0200000000
0	LBS. OZ. TARE&-	48	000002 -	0LBS.02OZ. TARE-
0	- -	49	095140036	95-14-0036
0	HRS. MINS. &0' 'CLOCK'	50	0042	0HRS.42MINS.0'CLOCK
,	0.	51	000000	0.00
,	0.	52	000000	000000
,	\$0.	53	00123456	1\$,234.56
,	0 *	54	0000000000	0*00
0,	0	55	001234	0,012,034
,	*0	56	0000001234	*****012*34
&	*0	57	013579	***130,579
-	&LATER	58	093066	9-30-66LATER
&	&&&LATER	59	093066	93066LATER
/	/	60	100166	10/01/66
,	0. -	61	000000015 -	000000015-
,	0. -	62	000000005	0.05

(B)

Figure 99. Examples of Edit Words (Part 2 of 2)

The letters and numbers under the heading "Example Number" in Figure 99, inserts A and B, refer to the letters and numbers in the following text:

A. Normal method of editing an amount field. Decimal point appears between dollars and cents; commas off-set appears 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. Since zero suppression

occurs through the unit-dollars position (0 in the edit word just left of the decimal point), blanks replace leading zeros and constants until a significant digit is encountered or until the specified zero is encountered. Thus, the decimal point and data to its right always appear in the edited data. Notice that, since zero suppression occurs through the position of the 0 in the edit word, that 0 is replaced by a blank when no significant digit appears in the data field.

- B. Normal method of punctuating a quantity field. Leading zeros and constants are replaced by blanks through the position of the zero suppression 0 (the next-to-last position in the edit word). 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.
- C. Normal editing of an amount field. Because the zero suppression 0 appears in the ten-dollar position of the edit word, leading zeros and constants are retained starting with the unit-dollars position. Because the dollar sign is placed just left of the zero suppression 0, it becomes 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. Notice that 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 since a zero is specified to the left of it.
- D. Similar to example C, except that (1) zero suppression is allowed up to the decimal point, (2) CR is used to indicate a negative value, and (3) two asterisks are printed 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, it would appear in the output record as \$.0000**. Note the extra position in the leftmost portion of the edit word to allow for the dollar sign.
- E. Similar to example D, except that (1) no symbol is used to indicate a negative value and (2) the edit word includes a fixed dollar sign. Because the dollar sign is placed in the extreme left position of the edit 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 to the edited data field.
- F. This example shows that 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 becomes 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.
- G. By not specifying a zero or asterisk, zero suppression can occur throughout the field; thus, edited data begins with the first significant digit. If the unedited data field only contained zeros, the entire edit word, except the minus, would be replaced by blanks when the field was edited.
- H. Zero suppression can occur throughout the field. The symbol CR will appear in the edited data field when the field contains a minus zero.
- I. This example shows the use of asterisk fill. Asterisks replace all positions in the edit word to the left of the first significant digit. If the asterisk were in the rightmost position of the edit word, the entire edited field would contain asterisks when the data was all zero.
- 1, 2, and 3. No edit word. The data in the output record has the same format as the unedited data. Notice that the low-order position of the output field is printed as an alphabetic character (J-R) if the source data field is negative.
- 4, 5, and 6. A blank edit word. All leading zeros are blanked and any sign in the low-order position of the unedited field is removed when the data is edited. Negative values are not identified.
7. The effect is the same as shown in examples 4, 5, and 6.
8. 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.
- 9 and 10. An ampersand appears as a blank in the edited data. The symbol CR appears in the edited data if the field is negative; it is replaced by blanks if the field is positive. The constant NET always appears in the edited data field.
11. An ampersand appears as a blank in the edited data. A minus sign, instead of CR, indicates negative values.
- 12 and 13. NET CR indicates when the edited data field is negative. Therefore, when the edited field is positive, NET CR appears as blanks.
14. The constant PROFIT appears in the edited data field. Negative values are not identified.

- 15 and 16. Similar to example 11, except that a fixed dollar sign is shown. An extra position is added to the edit word to allow for the dollar sign.
17. When the dollar sign appears to the immediate left of the zero suppression zero, it becomes a floating dollar sign, even when the dollar sign is entered in the leftmost position of the edit word.
- 18 and 19. The floating dollar sign is shown for different numbers of leading zeros. Note the extra position in the high-order portion of the edit word to allow for the dollar sign.
- 20 and 21. Even if there is no 0 in the edit word, the minus sign will appear in the edited field when the contents of the data field are minus zero. The constant TOTAL will always appear in the edited field.
22. This example shows how some zeros can appear in the edited field when the entire field is zero. Zero suppression occurs through the position of the 0 in the edit word. This leaves two positions in which zeros can appear in the edited field.
23. Similar to example 22, except that this example uses a floating dollar sign to replace the last suppressed zero.
24. Because the dollar sign is adjacent to the 0 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.
25. Because the asterisk appears in the low-order position of the edit word, asterisks will 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.
26. This example shows asterisk protection to a certain position; thereafter, any additional leading zeros appear in the edited field.
- 27 and 28. This example shows asterisk protection and zero suppression for a single position. Note that the asterisk is replaced by a significant digit in the position. Negative values are not identified.
29. Asterisk protection and zero suppression for an entire field. Asterisks are replaced by significant digits.
- 30 and 31. A method of editing an amount field. Punctuation and zeros to the left of the first significant digit are blanked. The decimal point is also lost when there are fewer than three significant digits. The CR symbol is printed for an all-zero negative field; the constants NET or -NET always appear in the edited field.
32. The ampersand, which appears in the edited field as a space, makes it possible to keep the dollar sign fixed while limiting zero suppression to the minimum one position. All punctuation is retained regardless of leading zeros because the 0 in the edit word is placed to the left of the first comma.
- 33 – 36. Standard methods for placing the floating dollar sign 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 compensate for the floating dollar sign.
37. Asterisk protection and zero suppression to the decimal point. The decimal point is retained regardless of the number of leading zeros. Note that 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.
38. 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.
39. Asterisk protection and zero suppression 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.
40. This example shows that 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 applies if there is a number of leading zeros. In the case of a comma, this has an awkward-looking effect; in the case of a decimal point, it is a normal approach (see example 34).

41. This example shows how to insert a space 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.
42. Normal punctuation of a quantity field. In this example, all leading zeros, including the units position, are suppressed (compare with example 43).
43. Normal method of showing a single zero in the edited data field when the data field contains only zeros.
- 44 – 48. 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 zero appear in the edited data. Examples 46–47 show how more edit word constants, other than the CR or minus, can be blanked on a positive field. Examples 46–48 also show the effect that the position of the zero suppression 0 has on constants. In example 47 an ampersand placed after the first constant provides a space following that constant in the edited data.
49. Possible method for editing a social security number field. A hyphen (-) is used within the edit word. In the example shown, the initial zero is suppressed. However, if you want the initial zero to appear in the edited data, you must leave an extra position in the edit word. See the note under *Editing Considerations* for a discussion of this exception.
50. This example shows the use of constants in the edit word. In this example the constant is an apostrophe.
- 51 and 52. 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.
53. This example shows that a dollar sign separated from the zero suppression zero, even if only by a comma, is not a floating dollar sign, but a constant.
- 54 – 57. 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 56 and 57 also show that asterisk protection replaces not only blanks but also other constants to the left of the first significant digit.
- 58 – 60. 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 59 shows the use of an ampersand to retain a blank space in the edited data.
61. This example shows what happens to the decimal point when no zero suppression 0 is specified for a field which has fewer than three significant digits. This applies if the field is more than three digits long.
62. This example shows how to retain the decimal point in a data field which has fewer than three significant digits. This applies if the field is more than three digits long.

COLUMNS 71–74

Columns 71–74 are not used. Leave them blank.

COLUMNS 75–80 (PROGRAM IDENTIFICATION)

See *Common Entries*.

This section further explains topics which were introduced, but not fully explained, in the preceding sections. The discussion of each topic is complete.

INDICATORS

Indicators are used to signal when certain conditions occur or do not occur. After you have assigned an indicator (on one of the specification sheets) to signal a certain condition, the indicator assigned should be associated with that one condition throughout the entire program.

Many times you want operations to be performed only when certain conditions occur. Because the indicator associated with the condition tells whether or not the condition has occurred, you may use the indicator to signal whether or not the operation should be done. In this way, indicators condition operations.

The status (on or off) of an indicator assigned on a specification line is determined by the results of processing the instruction on that specification line. If the condition has been satisfied, the indicator turns on; if it has not, the indicator turns off.

Usually indicators are set on or off by the conditions in the program itself. However, you may also set certain indicators by the SETON, SETOF, and SET operations. At the start of each program all indicators are off except the 1P indicator, L0 indicator, and any external indicators which have been set on. All indicators which you may use are shown in Figure 100. A summary of all indicators is shown in Figure 101.

Note: Only record identifying indicators and control level indicators (L1–L9) are turned off at the beginning of each program cycle. All other indicators are left unchanged.

Indicators	File Description Specifications		Input Specifications			Calculation Specifications			Output-Format Specifications
	Overflow (33–34)	File Conditioning (71–72)	*Record Identifying (19–20)	*Field Record Relation (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***
MR				X**			X		X
OA–OG,OV	X						X	X	X****
L0						X			X
L1–L9			X	X**		X	X	X	X
LR			X			X	X	X	X
U1–U8		*****X		X			X		X
KA–KN, KP, KQ							X	X	X

Note: X denotes the indicators that may be used.

- * Not valid on look-ahead fields.
- ** When field named is not a match field or a control field.
- *** Only for detail or heading lines.
- **** Cannot condition an exception line, but may condition fields within the exception record.
- ***** Not valid for table input files.

Figure 100. Valid Indicators

Indicators	Where Located	Where Normally Used as Conditioning Indicators	Normally Turned On		Normally Turned Off	
			By	When	By	When
Record Identifying Indicator	Input Sheet Cols. 19-20	<ul style="list-style-type: none"> { Input: Field 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	<ul style="list-style-type: none"> { Calculation: Indicators (cols. 9-17) { Output: Output Indicators (cols. 23-31) 	Data field with plus or minus balance	Before detail-time calculations	Data field without a plus or minus balance	Before detail-time calculations.
Zero/Blank	Cols. 69-70		Data field with zero or blank balance	Initially, upon Blank-After, and before detail-time calculations when field is zero or blank	Data field without a zero or blank balance	
Control Level (L1-L9)	Input Sheet Cols. 59-60	<ul style="list-style-type: none"> { 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	<ul style="list-style-type: none"> { 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	<ul style="list-style-type: none"> { Calculation Indicators (cols. 9-17) { Output: Output Indicators (cols. 23-31) 			Failure to satisfy the assigned condition when the specifications in the line are executed	Immediately
Arth. Ops. <ul style="list-style-type: none"> { Plus { Minus { Zero 			Plus result Minus result Field contents Zero*	Immediately when the specified condition is met upon execution of the operation		
COMP <ul style="list-style-type: none"> { High { Low { Equal 			Factor 1 > Factor 2 Factor 1 < Factor 2 Factor 1 = Factor 2			

Figure 101. A Summary of Indicators (Part 1 of 2)

Indicators	Where Located	Where Normally Used as Conditioning Indicators	Normally Turned On		Normally Turned Off		
			By	When	By	When	
Location of Specification Sheets	TESTZ Plus Minus Blank			Presence of a C zone Presence of a D zone Any zone other than a C or D zone			
	LOKUP High Low Equal			Factor 1 < Factor 2 Factor 1 > Factor 2 Factor 1 = Factor 2			
	KEY Plus Minus Zero Blank			Plus result Minus result Field contents Zero or blank	Immediately when the specified condition is met after the field is keyed.	Failure to satisfy the assigned condition when the field is keyed	Immediately
Indicators	Where Normally Specified to be Turned On or Off		Turned On by Program Itself		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-output time		
	L0 (Level Zero)	Nowhere	Initially and after each detail-output time		Never		
	LR (Last Record Total)	Nowhere	Before total time following last data record (after /*B)		After detail-output time (unless LR terminated job)		
	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-output time, halts system thereafter		When system is restarted after halt		
	01-99 (General)	Field and Resulting Indicators	Never		Never		
	KA-KN, KP, KQ	Resulting Indicators	Never		Never		

* Note: Zero or blank indicators for arithmetic and TESTZ operations are on initially and following Blank-After.

Figure 101. A Summary of Indicators (Part 2 of 2)

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

You may assign any of the numbers 01–99 to indicate such things as:

1. The type of record read (see *Columns 19–20* under *Input Specifications*).
2. The status (plus, minus, zero/blank) of an input field (see *Columns 65–70* under *Input Specifications*).
3. The results of a calculation operation (see *Columns 54–59* under *Calculation Specifications*). See *Examples, Example 1* and *Example 2*.

Any of these indicators which you have assigned in those columns or used in SETON, SETOF, or SET operations may also be used to:

1. Establish field record relations (see *Columns 63–64* under *Input Specifications*).
2. Condition calculation operations (see *Columns 9–17* under *Calculation Specifications*).
3. Condition output operations (see *Columns 23–31* under *Output-Format 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 in the last operation performed (see *Examples, Example 3*).

If any indicator 01–99 is set on or off by the operation codes 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 a condition from the operation performed.

Good Programming Practice: When assigning two or more numeric indicators (01–99), try to use indicators that are close to each other numerically. The system requires less core storage to check the status of indicators 02, 03, and 04 than it does to check 02, 15, and 22.

Examples

Example 1: Figure 102, insert A, shows that resulting indicator 10 has been assigned to signal when a minus condition occurs. Indicator 10 turns on if the result is negative after the subtraction operation has been performed. It then remains on (or off depending upon the result) until the same operation is performed again. It is always set to reflect the result of the subtraction operation each time it is done.

IBM																																																										International Business Machines Corporation																																																									
		RPG CALCULATION SPECIFICATIONS																																																						Page 1 2																																																											
Date																																																										Punching Instruction	Graphic																																																								
Program																																																										Punch																																																									
Programmer																																																																																																																			
Line	Form Type	Control Level (LO, LR, SR)	Indicators									Factor 1	Operation	Factor 2	Result Field	Field Length	Resulting Indicators																																																																																																		
			Not	And	And	Not	Not	Not	Not	Not	Not						Not	Not	Not	Not	Not	Not	Not																																																																																												
01	C										SALES	SUB	RETRNS	NETSLS	92			10																																																																																																	
02	C																																																																																																																		
03	C										SALES	SUB	RETRNS	NETSLS	92			10																																																																																																	
04	C																																																																																																																		
05	C										SALES	SUB	RETRNS	NETSLS	92			10																																																																																																	
06	C										NETSLS	ADD	TOTSLS	TOTSLS	102																																																																																																				
07	C										PROFIT	SUB	RTRPRF	NETPRF	82			10																																																																																																	
08	C																																																																																																																		

A
B
C

Figure 102. Indicators 01–99

Example 2: Figure 102, insert B, shows the same operation as insert A. However, this operation is conditioned by indicator 01. The operation is done only when indicator 01 is on. Resulting indicator 10 is set on only when the result of the operation is negative.

Example 3: Figure 102, insert C, shows the use of the same indicator (10) in two lines. The status of this indicator reflects the result of each operation. For instance, indicator 10 turns on after the operation in line 05 has been done if the result of the operation is negative. However, if the result of the operation in line 07 is positive or zero, indicator 10 turns off. It is then reset only when the operation in line 05 is done again.

KA—KN, KP, KQ (Command Key Indicators)

Command key indicators may be assigned to indicate:

1. What command keys may be pressed by the operator for a SET operation. (see *Operation Codes, Set* for further information).

Any of the indicators which have been used in a SET or SETOF operation may then also be used to:

1. Condition calculation operations in columns 9—17 of the Calculation sheet.
2. Condition output operations in columns 23—31 of the Output-Format sheet.

The command key indicators used as conditioning indicators in calculations (columns 9—17) and in output-operations (columns 23—31) are turned on and off in the following manner:

- The indicators are turned on by pressing the appropriate command key for a SET operation.
- The indicators are turned off when the SET operation in which they are specified is performed or when a SETOF operation is performed.

H1—H9 (Halt Indicators)

You may use any halt indicator to:

1. Cause the program to stop after finding an unacceptable condition.
2. Condition calculation or output operations that are not to be performed when such an unacceptable condition has occurred. This is necessary because all calculation and detail output operations are still performed for the record that caused the unacceptable condition before processing stops.
3. Establish field record relations (see *Columns 63—64* under *Input Specifications*).

Using the same indicator to test for two or more error conditions is not usually good practice. For example, Figure 103, insert A, shows the use of H1 in two different specification lines. If the result of the calculation operation in line 01 is negative, H1 turns on. This is an error condition. Processing continues, however, until this program cycle is completed. Thus, the operation in line 03 is done. If the result of this subtraction operation is positive, H1 turns off. The program does not stop because H1 is not on, even though an error condition has been found in line 01.

The use of two different halt indicators as shown in Figure 103, insert B, does not allow a situation like the one just described to occur.

Any halt indicator assigned to test for zero or blank is off at the beginning of the program.

Note: If a halt indicator stops processing, it is turned off when the system is restarted. If more than one halt indicator turns on during a program cycle, each halt

indicator must be considered separately. Every time the program is restarted, only one halt indicator is bypassed.

1P (First Page Indicator)

Use the first page indicator to condition those lines which are to be printed on only the first page. These lines are usually heading lines. Data is provided for lines conditioned by the 1P indicator by constants entered in columns 45–70 of the Output-Format sheet.

All lines conditioned by the 1P indicator are printed out even before the first record from input file is processed. Therefore, do not condition output fields which are based upon data from input records by the 1P indicator. You get meaningless output if you do.

Calculation operations cannot be conditioned by the 1P indicator either. This indicator is on at the beginning of the program and turns off after the detail output has been performed on the first page output.

RPG CALCULATION SPECIFICATIONS

Date _____
Program _____
Programmer _____

Punching Instruction	Graphic						
	Punch						

Page

Program Identification

Line	Form Type	Control Level (LD, LR, SR)	Indicators			Factor 1	Operation	Factor 2	Result Field	Field Length	Decimal Positions	Half Adjust (H)	Resulting Indicators			Comments
			And	And	Not								Plus	Minus	Zero	
0 1	C					GROSS	SUB	DEDCTN	NET	62					H1	
0 2	C					HRS	ADD	TOTHR	TOTHR	51						
0 3	C					BALNCE	SUB	GROSS	NEWBAL	102					H1	
0 4	C															
0 5	C															

A This operation is not conditioned. It will always be done even when the halt indicator is on to signal an error condition.

RPG CALCULATION SPECIFICATIONS

Date _____
Program _____
Programmer _____

Punching Instruction	Graphic						
	Punch						

Page

Program Identification

Line	Form Type	Control Level (LD, LR, SR)	Indicators			Factor 1	Operation	Factor 2	Result Field	Field Length	Decimal Positions	Half Adjust (H)	Resulting Indicators			Comments
			And	And	Not								Plus	Minus	Zero	
0 1	C					GROSS	SUB	DEDCTN	NET	62					H1	
0 2	C					HRS	ADD	TOTHR	TOTHR	51						
0 3	C		NH1			BALNCE	SUB	GROSS	NEWBAL	102					H2	
0 4	C															
0 5	C															

B If H1 turns on as a result of the operation in line 01, this operation is not performed.

Figure 103. One Halt Indicator Testing for Two Conditions

MR (Matching Record Indicator)

Use the MR indicator to condition calculation and output operations which are to be done only when records match. The MR indicator turns on when a primary file record matches any secondary file record on the basis of the matching fields indicated by M1–M9. The matching record 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.

When a control level indicator (L1–L9) is specified in columns 7–8 (total time) and MR is specified in columns 9–17, 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 7–8 of the Calculation sheet) are done, MR then indicates the matching condition of the record just read.

If record types for which no matching fields have been specified are read, they are processed immediately as if they belong to the same match group as the record previously processed. MR is always off for these types.

OA–OG, OV (Overflow Indicators)

Overflow indicators are used for printer files and ledger files. Use them primarily to condition the printing of heading lines. If you intend to use an overflow indicator to condition output lines, you must assign an overflow indicator to the printer or ledger file on the File Description sheet (columns 33–34). This same indicator must then be used to condition all lines that are to be written only when overflow occurs.

If the destination of a space/skip or print operation falls within the form overflow area, the overflow indicator is turned on and remains on until all overflow lines are printed. However, if a skip is specified that advances the form past the overflow line to the first line or past the first line on a new page, the overflow indicator does not turn on. Certainly, you do not want the overflow indicator on to signal a need for a new page when you just skipped to a new page.

If an overflow indicator is used as a conditioning indicator, it indicates that output is to be performed at overflow time. This applies regardless of whether or not the line conditioned by the indicator is in an AND or OR relationship with other indicators.

When an overflow indicator is used, a form skip specification usually will be made on the first line conditioned by an overflow indicator. Otherwise, forms do not advance. Remember, they advance automatically if you do not use overflow indicators.

The overflow indicator may be set by the SETON or SETOF operation code. After all total records have been written, however, the indicator is set as it normally is in accord with the overflow line. See *Overflow Indicators* in this section for further information.

L1–L9 (Control Level Indicators)

Control level indicators are used to signal when a change in a control field has occurred.

Because they turn on when the information in a control field changes, they may be used to condition operations (such as finding totals) that are to be performed only when all records having the same information in the control field have been read. They may also be used 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 may be used in three different types of specifications: input, calculation, and output-format.

Input Specifications

If a control level indicator is entered in columns 59–60 of this sheet, the field described in columns 53–58 is declared to be a control field. This means that the field on each card read is matched against the same field on the previous card. If the information is not the same, the control level indicator turns on. All lower level indicators turn on when a higher level indicator turns on. For example, if L8 turns on, L1–L7 also turn on.

When a control level indicator is used on the Input sheet in columns 63–64 (field record relation), the data from the field named in columns 53–58 is accepted and used only when the control level indicator is on.

If record types without a control field are read, they are treated as if they belong to the same control group as the preceding record. No control level indicator is set for them. Control level indicators may also be used to establish field record relations (see *Columns 63–64* under *Input Specifications*).

Calculation Specifications

When a control level indicator is entered in columns 7–8 of this sheet (total calculations), it conditions the operation so that it is done only when a control field changes. If any control level indicator appears in columns 9–17 (detail calculations), the operation is done only on the first record of a new control group.

A control level indicator may be turned on or off by operation codes SETON and SETOF. However, these operations do not cause all control level indicators lower than the one specified to turn on or off. For example, when L2 is set on, L1 does not automatically turn on.

Output-Format Specifications

Control level indicators entered in columns 23–31 of this sheet specify when output records are to be written:

1. If the control level indicator is entered along with a T in column 15 and no overflow indicator is used, the record is written only after the last record of a control group has been processed.
2. If the indicator is entered along with an H or D in column 15 and no overflow indicator is used, the record is written only after the first record of the new control group has been processed.
3. If the control level indicator is entered along with an overflow indicator, the record is written after the overflow line has been sensed (provided a control break has also occurred).

LO Indicator

The LO indicator is never assigned, but it is always automatically on. Thus, it can be used to condition certain calculation or output operations. LO is used as a conditioning indicator in the same way as the other control level indicators. However, it is used only when L1–L9 cannot be assigned because the input data records have no field available which can serve as a control field.

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 disk, console, or data recorder files, the LR indicator is normally turned on when a /* record is detected. No record

identifying indicators can be on while last record processing is performed for these files. When LR is turned on by a /* record, all other control level indicators (L1–L9) used also turn on automatically.

For ledger or keyboard files, the LR indicator must be turned on at the appropriate time in calculation specifications. Record identifying indicators may be on while last record processing is performed for these files. When LR is turned on in detail calculations, all other control level indicators used are automatically turned on at the beginning of the next cycle.

Do not specify an L0-L9 indicator in an OR relationship with an LR indicator. If you do, the specified operations will be done twice at LR time.

In System/3 Disk RPG II, all total lines conditioned by LR will be performed last. The job ends after all total records have been written. The LR indicator cannot be turned off by a SETOF operation.

U1–U8 (External Indicators)

Indicators U1–U8 are external indicators. This means they are set prior to processing by an operation control language (OCL) statement. Their setting cannot be changed during processing. Thus, the program has no control over them.

You may use these indicators as file conditioning indicators. They tell whether or not a certain file is to be used for a job. For example, you may have a job which one time requires the use of two output (or input) files and another time the use of only one. Instead of writing two different programs (one using one file, the other two), you can condition a file (in columns 71–72 of the file description specifications) by an external indicator. When the indicator is on, the file is used; when it is off, the file is not used.

If a file is conditioned by an external indicator, all output data handled by the file must also be conditioned by the same indicator. Any calculation operations which would not be done when the file is not in use should also be conditioned by the same indicator.

In addition to using these indicators as file conditioning indicators, you may use them:

1. To condition calculation operations.
2. To condition output operations.
3. As field record relation indicators (columns 63–64 of input specifications).

LEDGER FILE PROCESSING

For systems with a ledger card device, all pertinent data regarding an account, an item, or an employee can be printed on a ledger card in addition to being stored on disk. The ledger card is valuable in that it provides a visual historical record which you can reference easily (Figure 104).

The desired data fields are printed on the ledger card according to output specifications you supply. Data printed on the ledger is not machine readable. The ledger card device reads or senses only the control number field and the line finder marks which are recorded as dashes in the right margin of the ledger card.

Line finder marks are used to indicate the last line printed on a ledger card. This allows the ledger card device to automatically position the ledger card once it has been inserted. A line finder mark is printed on the ledger card each time the ledger is spaced. If spacing is suppressed by a 0 space after output specifications, no line finder mark is made; printing can be done on the same line again. Each time a ledger card is spaced, the ledger card device checks to determine whether or not the ledger is full. See *Ledger Card Overflow* for information on how the full-card condition is handled.

The control number field indicates the control number of the ledger card. The first line of the control number field has a dash in each column if the ledger card has a control number. The second line of the control number field is used for parity check. The system determines the total number of dashes in each of the two columns of the control number field. If the total number of dashes in a column is even, a dash is placed in the second line of the column; if the total number of dashes is odd, the second line of the column is left blank. Thus, the number of dashes in each column will always be odd.

The remainder of the dashes in the control number field represent the control number. The dashes can be interpreted by reading two lines at a time. Each dash in this two-line block has a numeric value:

Upper lefthand dash = 8	8	4
	↓	↓
Upper righthand dash = 4	↓	↓
	—	—
Lower lefthand dash = 2	↑	↑
	2	1
Lower righthand dash = 1		

The numeric value of the dashes in each block determines the number represented (one of the digits from 0–9). For example:

—	}	Represents 9
—		
—	}	Represents 7
—		
—	}	Represents 0
—		

The first block of dashes (third and fourth lines) represents the first number in the control number, the second block represents the second number in the control number, and so on. Thus, if the control number for a ledger card is 583, the dashes on the ledger card should look like this:

Line Finder Marks	↓	—	—	← Control Number Field
		—	—	← Indicates control number present
		—	—	← Parity Check
		—	—	} 5
		—	—	
		—	—	} 8
		—	—	
		—	—	} 3
		—	—	

Notice the parity check on the second line. A dash was entered on this line in the right column so the total number of dashes would be odd (5). The number of dashes in the center column was already odd (3) so the second line was left blank.

See *Types of Ledger Files, Output Ledger Files* for information on how control numbers are created.

Note: Do not interrupt your program for an inquiry request while a ledger card is being processed. Inquiries should be made only after a ledger card has been ejected and before another ledger card is inserted.

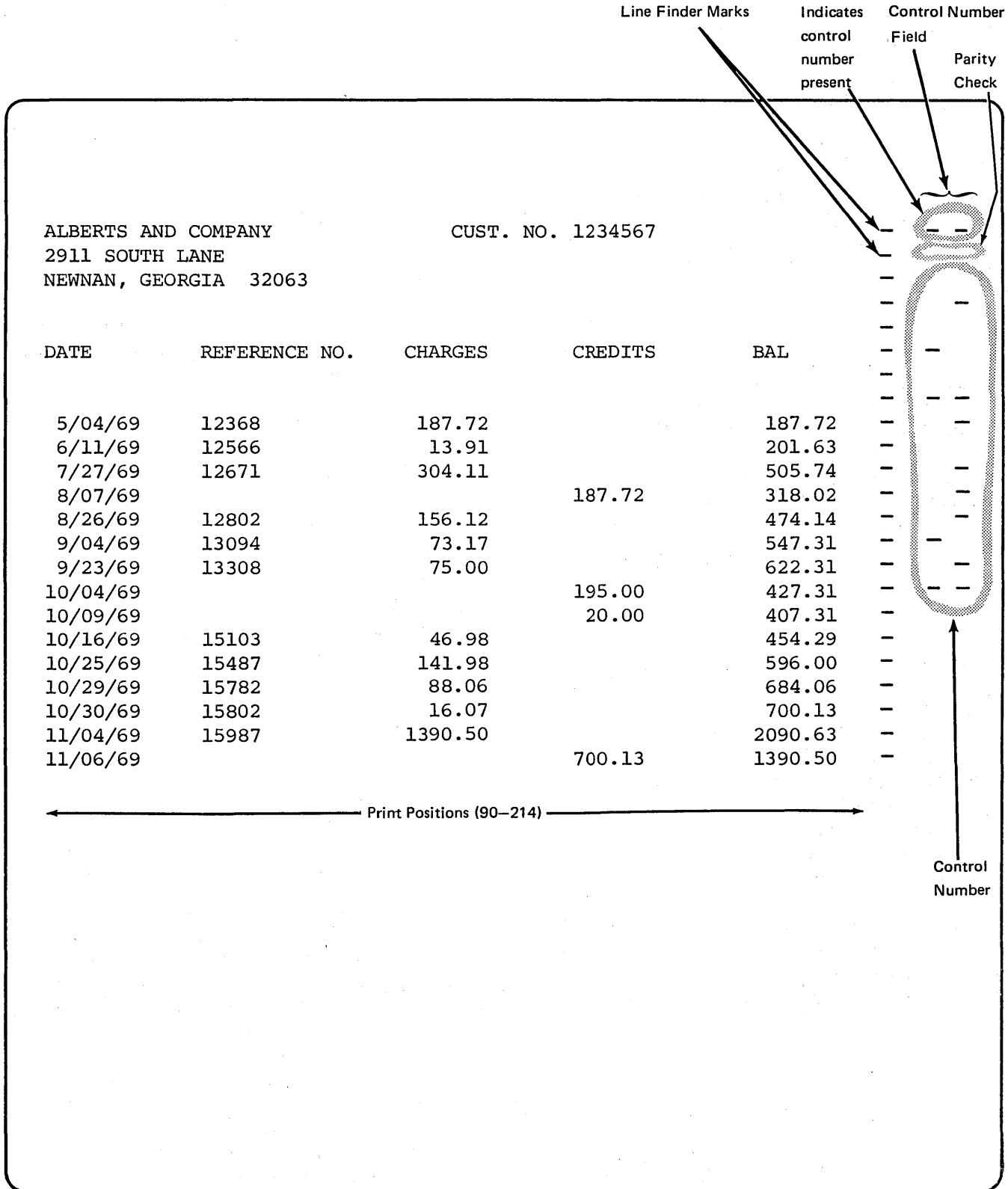


Figure 104. Example of the IBM System/3 Ledger Card

Spacing on the Ledger Card

An entry must not be made in column 17 (space before) of the Output-Format sheet when processing ledger files. An entry (0–9) should be made in column 18 (space after) of the Output-Format sheet. If no entry is made in column 18, however, an entry of 1 is assumed. Skipping (columns 19–22 of the Output-Format sheet) cannot be specified for ledger files.

Printing on the Ledger Card

Whenever possible, you should specify beginning and ending print positions for ledger files so that most printing is done on the right side of the ledger card. The minimum print position allowed is 83; the maximum print position allowed is 214. This requires less print time because the print element does not have to move as far across the ledger card to record line finder marks each time printing or spacing occurs.

Ejecting the Ledger Card

You must specify ledger card eject for output and combined files when you are ready to process a different ledger card. This would usually be necessary after all output fields have been written on a particular ledger card. (Ledger card eject is indicated by an E in column 16 of the Output-Format sheet.) When eject is specified and the proper indicators (columns 23–31 of the Output-Format sheet) are on, the fields for that record type are printed and the card is ejected immediately.

Even when no output is written to a record type in a ledger file you must specify eject. Eject must also be specified as a part of your error handling procedures. This allows the program to go to the next record after errors such as record-not-found after a chain operation or unsatisfactory results in a compare operation. For instance, if the operator keys a control number field and inserts the wrong ledger card, the normal procedure is to turn on a halt indicator to signal this error. Then, conditioned by the appropriate halt indicator, ledger card eject is specified in output operations.

When specifying ledger card eject, consider:

1. If eject is not specified and a ledger card read is called for, an error halt occurs and the ledger card is automatically ejected. The operator must press the start key to continue processing.

2. You need not specify fetch overflow (F in column 16 of the Output-Format sheet) for ledger files. Fetch overflow is performed automatically for ledger files.
3. When ledger card eject is specified, an E may appear in column 16 on the same line as an H, D, T, or E entry in column 15, or it may appear on an OR line. The eject applies only to the specification line it is on.
4. If a new control number is specified for a ledger card and the previous ledger card has not been ejected, an automatic eject occurs, but the program will halt. A blank card must be inserted and the start key must be pressed to continue processing.
5. It is not necessary to have any output fields specified in order to eject a ledger card.
6. Eject is not required for overflow lines.

Types of Ledger Files

On the File Description sheet (column 16), you can specify two types of ledger files: combined or output. The chart in Figure 105 shows the possible types of ledger files and the operations performed for each type. You can specify only one ledger card file in a program.

Combined Ledger File

A combined ledger file contains ledger cards that a program uses in both input and output operations. A program reads the control number field from ledger cards in a combined file (input) and then writes data fields on the ledger cards in the file (output). Both input and output operations must be specified for each record in a combined ledger file. If no output is to be written to a particular record type, you must at least code the output specifications needed to eject that record type (E in column 16). See *Examples, Example 1* for use of a combined ledger file in an RPG II program.

A combined ledger file can also be used to create new ledger cards. To create a new card, code the output specifications the same as you would for an output ledger file (see *Output Ledger Files* in this section). When you are creating new ledger cards, matching record logic for the ledger card device is not allowed. If matching record logic is needed, you must code your own.

Ledger Card Files	
<u>File Type</u>	<u>Operations Performed</u>
Combined Files (primary, secondary, or demand)	<ul style="list-style-type: none"> ● Control number area is read to ensure that it is blank. ● Control number field is encoded on the ledger card. ● Control number field is read. ● Data fields may be printed. ● Card is ejected by output-format specifications.
Output Files	<ul style="list-style-type: none"> ● Control number area is read to ensure that it is blank. ● Control number field is encoded on the ledger card. ● Data fields may be printed on the ledger card. ● Card is ejected by output-format specifications.

Figure 105. Possible Ledger File Types

Output Ledger Files

When an output ledger file is described in a program, it indicates that the ledger file is being created. To create a ledger file, you must place a control number on each ledger card in the file using output specifications. You may also provide for the printing of heading lines or data fields on the ledger if this is a part of your program.

The control number serves as a record identifying field which differentiates one ledger card type from another or one ledger file from another. It is a one- to nine-digit field located on the upper right side of the ledger card. The control number usually contains a control field and a numeric identification code which you have assigned. The control number must be specified on the first lines in a record when a ledger file is being created.

When creating a ledger file, you must provide the appropriate control number by making the following specifications on the Output-Format sheet:

1. Leave columns 7–22 blank.
2. In columns 23–31, enter any valid indicators if you wish to condition the encoding of a control number when more than one control number is specified.
3. In columns 32–37, enter the control field to be used as all or part of the control number. If this data field makes up part of the control number, additional lines are required to complete the control number.
4. Leave column 38 blank.
5. Leave column 39 blank unless you want to blank the field after printing.
6. In columns 40–43, enter the end position of the control number. This entry may be from N001 to N009. When you specify the location of the control number, enter the constant N in column 40; enter the number of the end position (001–009) in columns 41–43. If the control number that you specify is less than nine positions long, the actual number should be placed in the high-order positions of the area set aside for the control number on the ledger card. For example, if a four-position control field and a one-position code are to be used, they should be placed in positions 1–5 of the ledger card (005 in columns 41–43). Positions 6–9 should be left blank. This allows the control number to be printed faster.

two customer number fields are compared. Again, assume that the two fields are equal so that indicator 03 is off. The operator keys the DESC and QUANTY fields in the first transaction record for this customer. He does not press command key KB since another transaction record is to be keyed for this customer, nor does he press KC since this is not the end of the program. The operator does enter the PRICE field. On the Output-Format sheet, the fields in the first transaction record are printed on the ledger. Note that the ledger is not ejected (command key KB is not on).

command key KB to indicate that he is completing the last transaction for this customer; he also presses command key KC to indicate that this is the last CUSTOMER record to be updated. The operator keys the PRICE field. Conditioned by indicator KC, the last record indicator is turned on to signal the end of this program. On the Output-Format sheet, the fields just keyed are printed and the ledger card is ejected.

On the next program cycle, the GOTO operation (line 02 of the Calculation sheet) is performed because indicator 02 is on and indicator KB is off. The program branches immediately to line 11 of the Calculation sheet where the operator can key another transaction for the same customer. The operator enters the DESC and QUANTY fields when prompted by field lights 2 and 3. He presses

If an equal condition is not found after the compare operation in this program, indicator 03 is turned on. A SET operation conditioned by 03 ejects the ledger card and allows the operator to press command key KA. If KA is not pressed, the program branches to the point in the program where the ledger card read is repeated. If KA is pressed, the program branches directly to the end of the calculation operations.

IBM

Date _____

Program _____

Programmer _____

International Business Machines Corporation

RPG OUTPUT - FORMAT SPECIFICATIONS

Form X21-9090
Printed in U.S.A.

Page 1 2

Program Identification 75 76 77 78 79 80

Line	Form Type	Filename	Type (H/D/T/E)				Space				Skip			Output Indicators			Field Name	Edit Codes	End Position in Output Record	Sterling Sign Position	
			Blank	Print	Print	Print	Before	After	Before	After	Not	Not	Not	And	And	And					
01	O	CUSTOMER																01			
02	O	ORE																02			
03	O																	03			
04	O																	04			
05	O																	05			
06	O																	06			
07	O																	07			
08	O																	08			
09	O																	09			

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

Constant or Edit Word

Figure 107. Combined Ledger Files (Part 3 of 3)

Output-Format Sheet: Use the overflow indicator assigned on the File Description sheet to condition those lines you want printed on each ledger card. Usually these lines are total or heading lines which must be printed at the top of each new ledger card (Figure 110).

2. Fetch overflow (F in column 16 of the Output-Format sheet) need not be specified for a ledger file; it is handled automatically.
3. Space Before entries cannot be made in column 17.
4. Skipping (entry in columns 19–22 of the Output-Format sheet) cannot be specified for a ledger file.

Assigning Overflow Indicators

When using the overflow indicator to condition overflow printing on a ledger card, consider the following:

1. Overflow indicators may be turned on and off by the operation codes SETON and SETOF.

Date _____		RPG OUTPUT - FORMAT SPECIFICATIONS		Form X21-9090 Printed in U.S.A.														
Program _____		Punching Instruction		Page <input type="text" value="1"/> <input type="text" value="2"/>														
Programmer _____		Graphic		Program Identification <input type="text" value="75"/> <input type="text" value="76"/> <input type="text" value="77"/> <input type="text" value="78"/> <input type="text" value="79"/> <input type="text" value="80"/>														
Programmer _____		Punch																
Line	Form Type	Filename	Type H/D/T/E	Stacker Select/Fetch Overflow (F)	Space		Skip			Output Indicators			Field Name	Edit Codes	End Position in Output Record	Constant or Edit Word	Sterling Sign Position	
					Before	After	Before	After	Not	And	And	Not						
01	O	LEDGER	H	1						OF	03							
02	O												CUSTNA		159			
03	O												60		201	'20 %'		
04	O												W60		201	'10 %'		
05	O		H	1						OF	03							
06	O												CUSTAD		159			
07	O		H	3						OF	03							
08	O												CTYSTA		159			
09	O												ZIP		168			
10	O		H	1						OF	03	15						
11	O														145	'DATE'		
12	O														162	'INVOICE NO.'		
13	O														179	'INVOICE AMT.'		
14	O														197	'AMOUNT DUE'		
15	O																	
16	O																	

Figure 110. Printing Overflow Lines on the Ledger Card

LOOK AHEAD

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 RPG II look ahead feature enables you to use information from records that follow the one being processed. The fields containing the information are called look ahead fields.

Look Ahead Fields

The look ahead feature can be used only with input, update, or combined files. Chained files, demand files, or files that contain header/trailer records may not use the look ahead feature. Look ahead fields cannot be specified for ledger card or keyboard files. To use the look ahead feature, you must describe the look ahead fields and reference them as you do normal fields. You can describe any or all of the fields in a record as look ahead fields. The description applies to all records in the file, regardless of their type. All record types should contain the same look ahead fields. (The specifications for describing the fields are given later.)

Look ahead fields always apply to the next record in the file, provided the file is not a combined or update file. Thus, if you wish to use information both before and after the record is selected for processing, you must describe the field twice, once as a look ahead field and once as a normal field.

For combined and update files the look ahead fields apply to the next record in the file only if the current record was not read from that combined or update file. Therefore, when you are reading from only one file and the file is a combined or update file, look ahead fields always apply to the current record.

Figure 111 shows the processing of three records from two input files, one primary and one secondary. The first record from each file is read (see Figure 111, insert A). In Figure 111, insert B record P1 is selected for processing; in Figure 111, insert C, record P2; and in Figure 111, insert D, record S1. The records available for look ahead during the

processing of these records are:

<i>Record Being Processed</i>	<i>Records Available for Look Ahead</i>
P1	P2 and S1
P2	P3 and S1
S1	P3 and S2

In general, when the record being processed is from an input file, the next record in the input file is available as are the records which were read, but not selected, from the other files.

Figure 112 shows the same files as Figure 111 with one exception: The primary file is an update file. The records available for look ahead during the processing of the three records are:

<i>Records Being Processed</i>	<i>Records Available for Look Ahead</i>
U1	U1 and S1
U2	U2 and S1
S1	U3 and S2

In general, when the record being processed is from a combined or update file, only the records which were read, but not selected, from the other files are available for look ahead. The next record from the combined or update file is not read until after the current record has been processed. Therefore, the next record from the combined or update file is not available for look ahead.

After the last record from a file has been processed, every look ahead field for the file is automatically filled with 9's. For example, a field three record positions long contains 999. The 9's remain in the fields until the job ends. Note also that blank after (B in column 39 of the Output-Format sheet) cannot be used with look ahead fields.

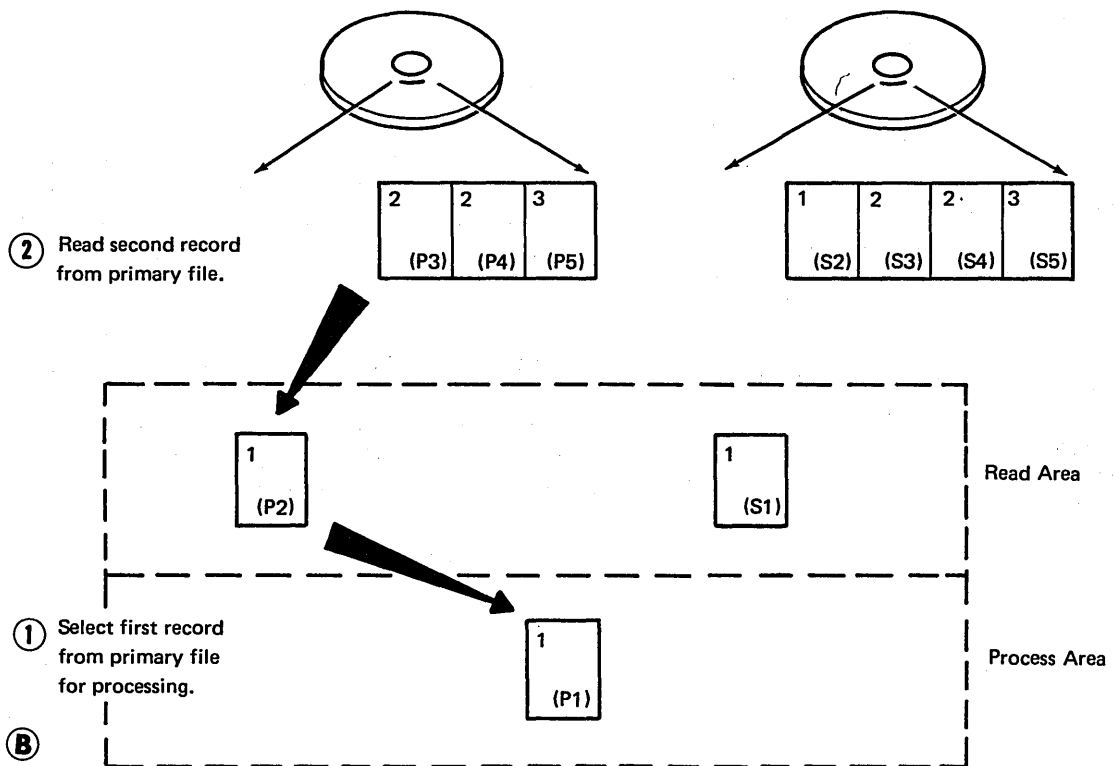
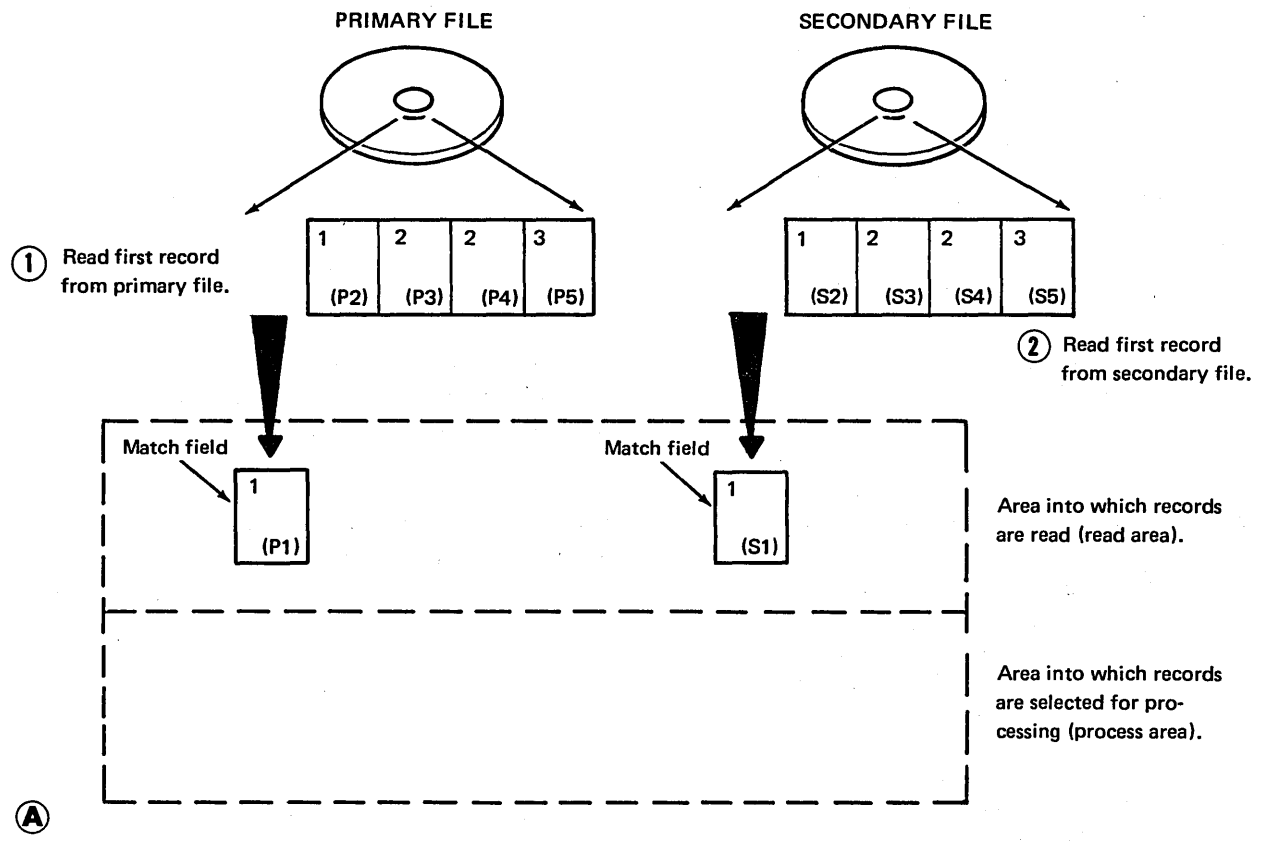


Figure 111. Available Records: Two Input Files (Part 1 of 2)

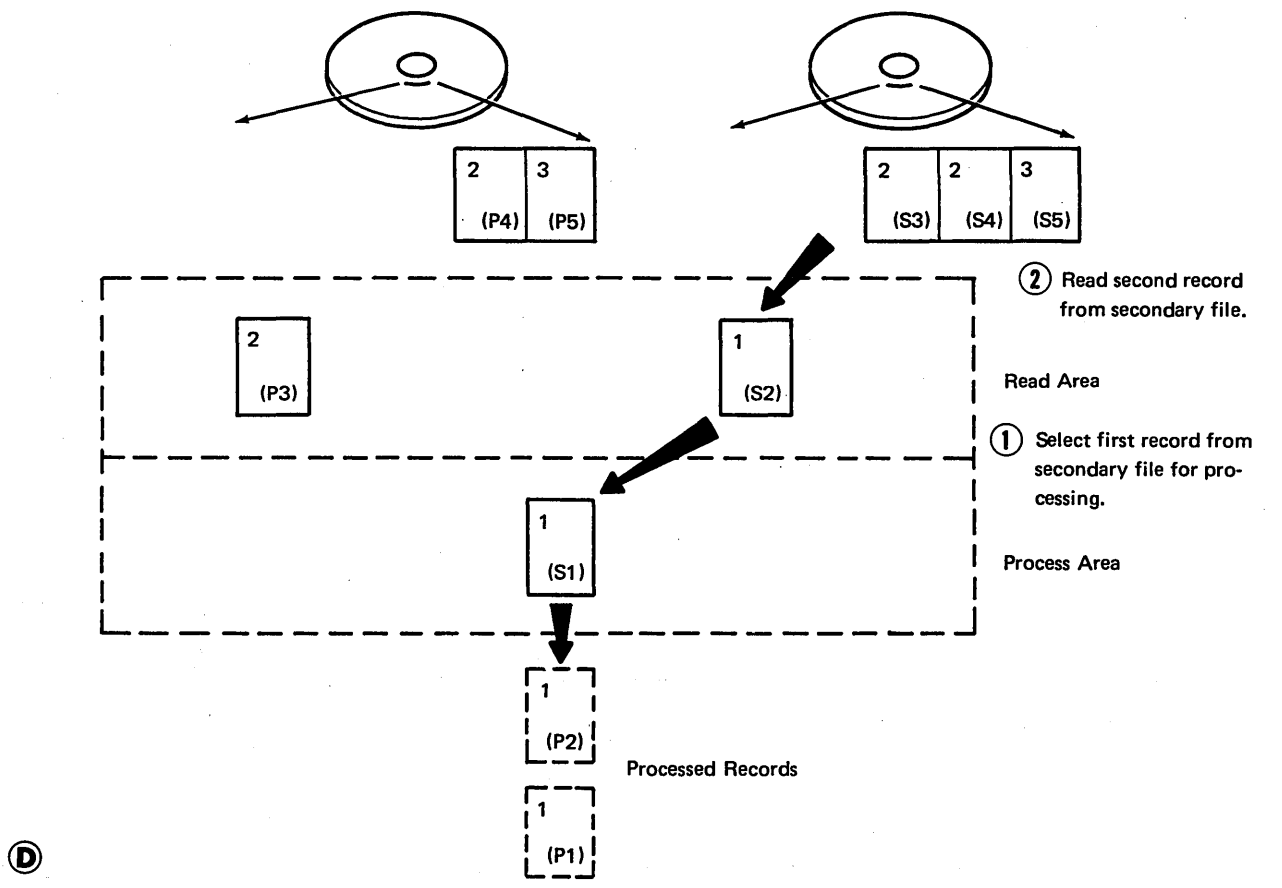
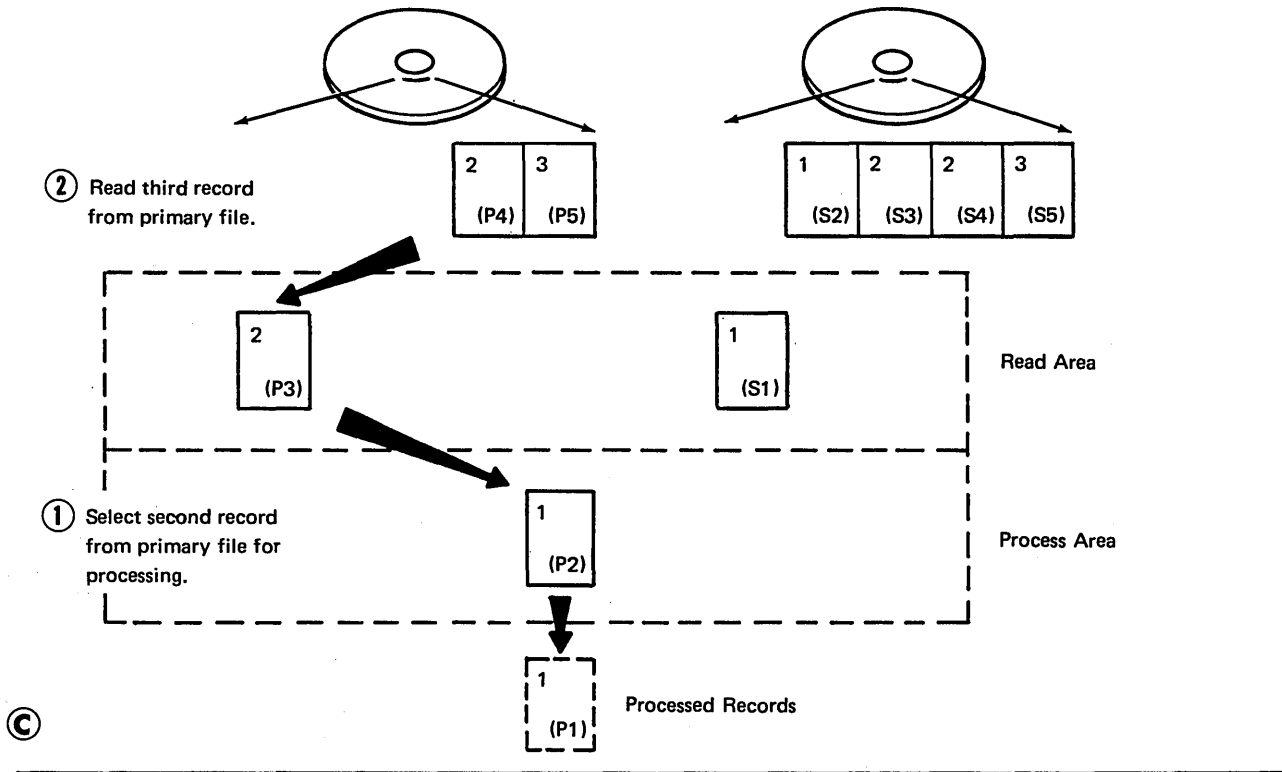


Figure 111. Available Records: Two Input Files (Part 2 of 2)

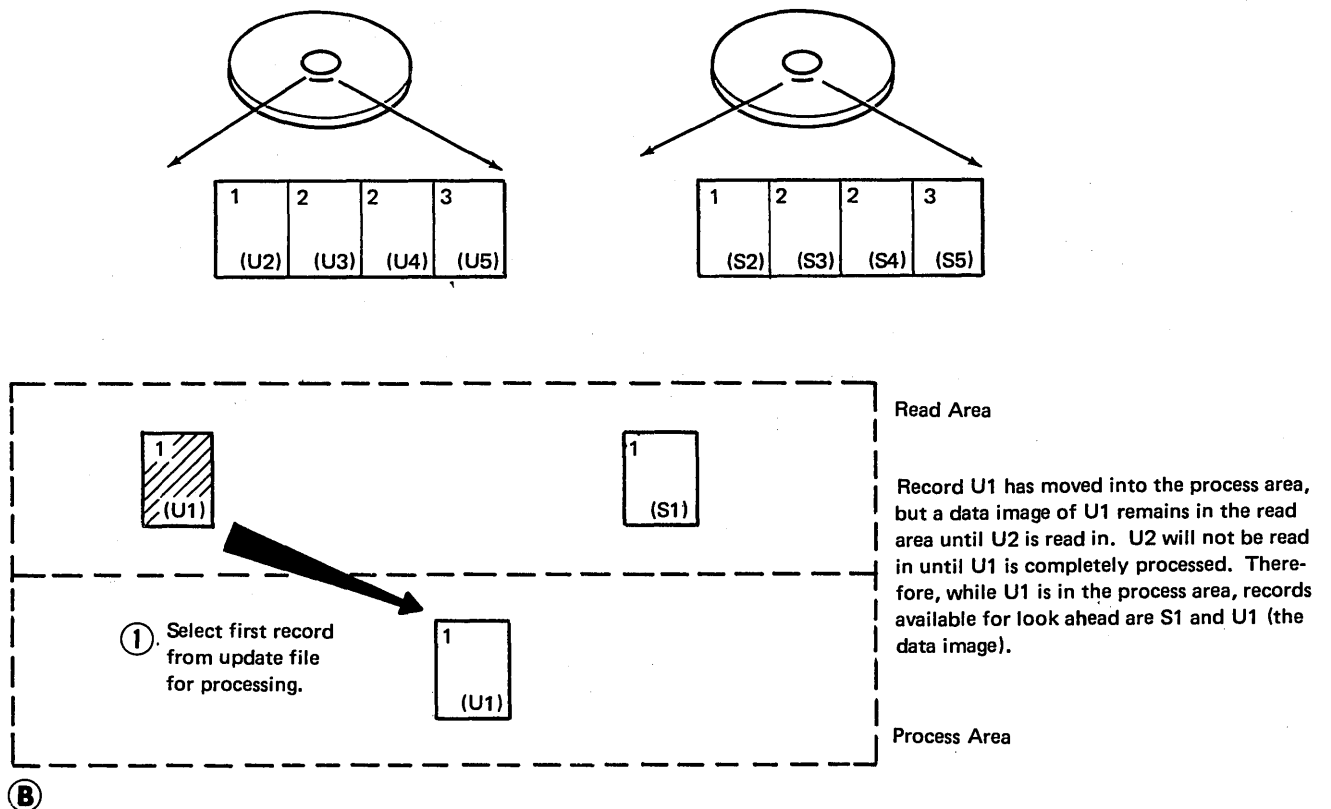
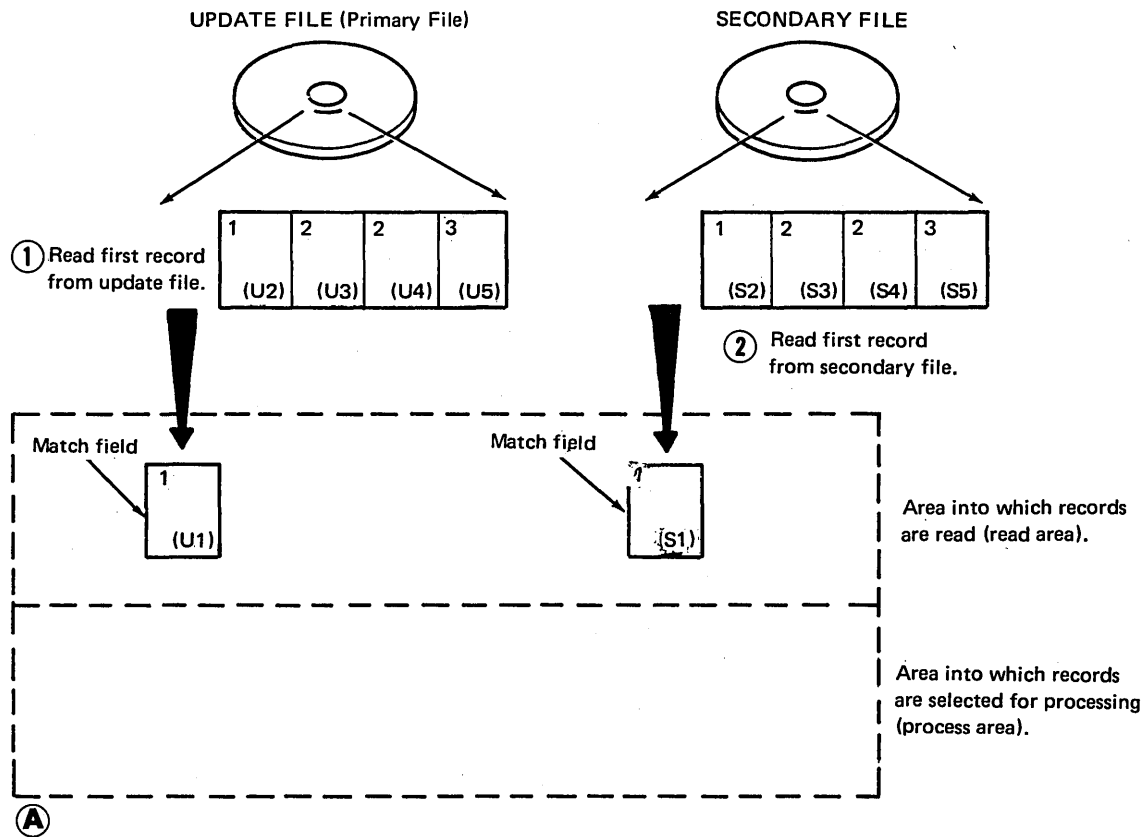


Figure 112. Available Records: One Input File, One Update File (Part 1 of 3)

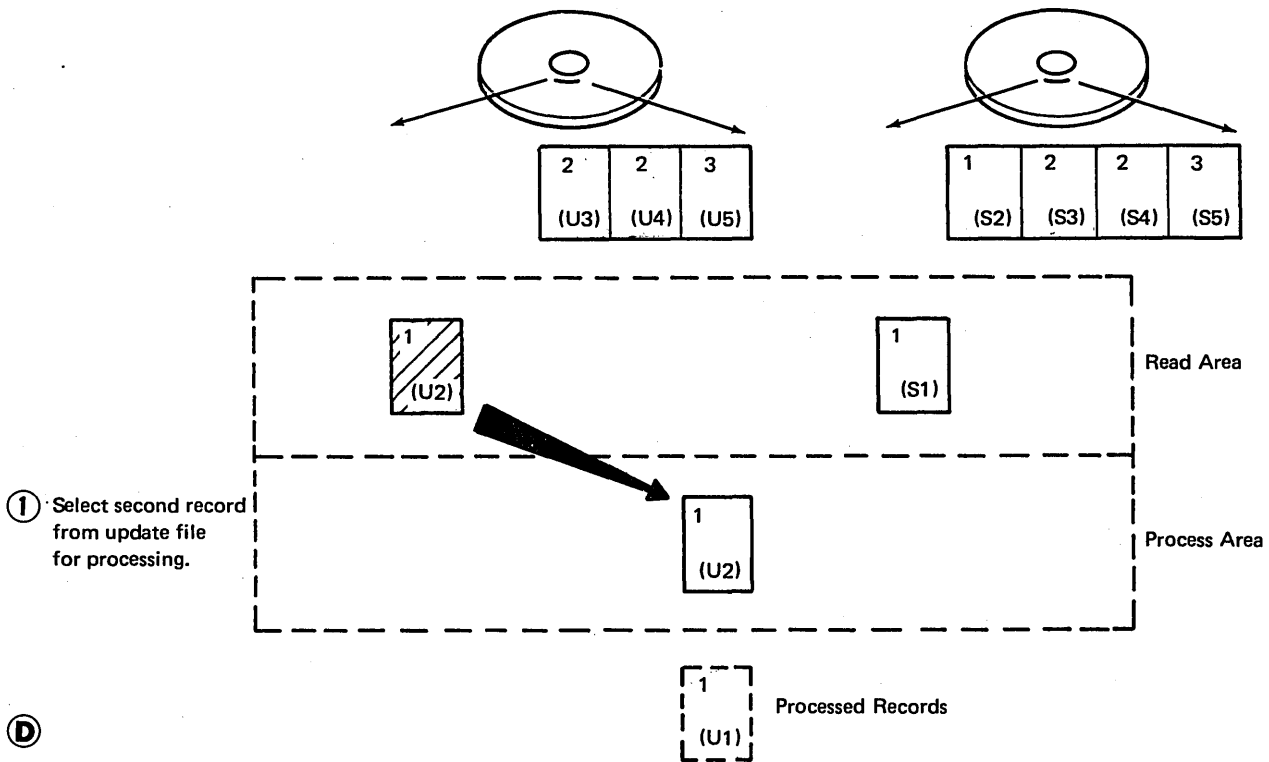
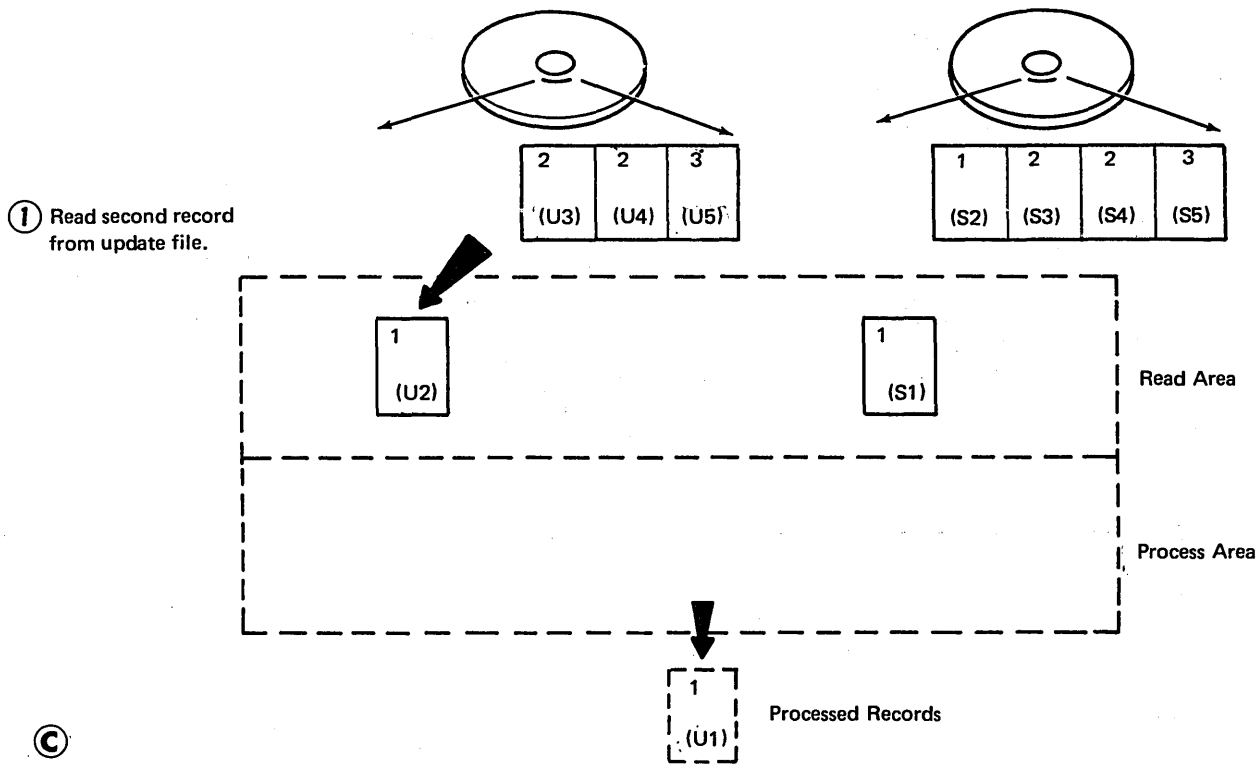


Figure 112. Available Records: One Input File, One Update File (Part 2 of 3)

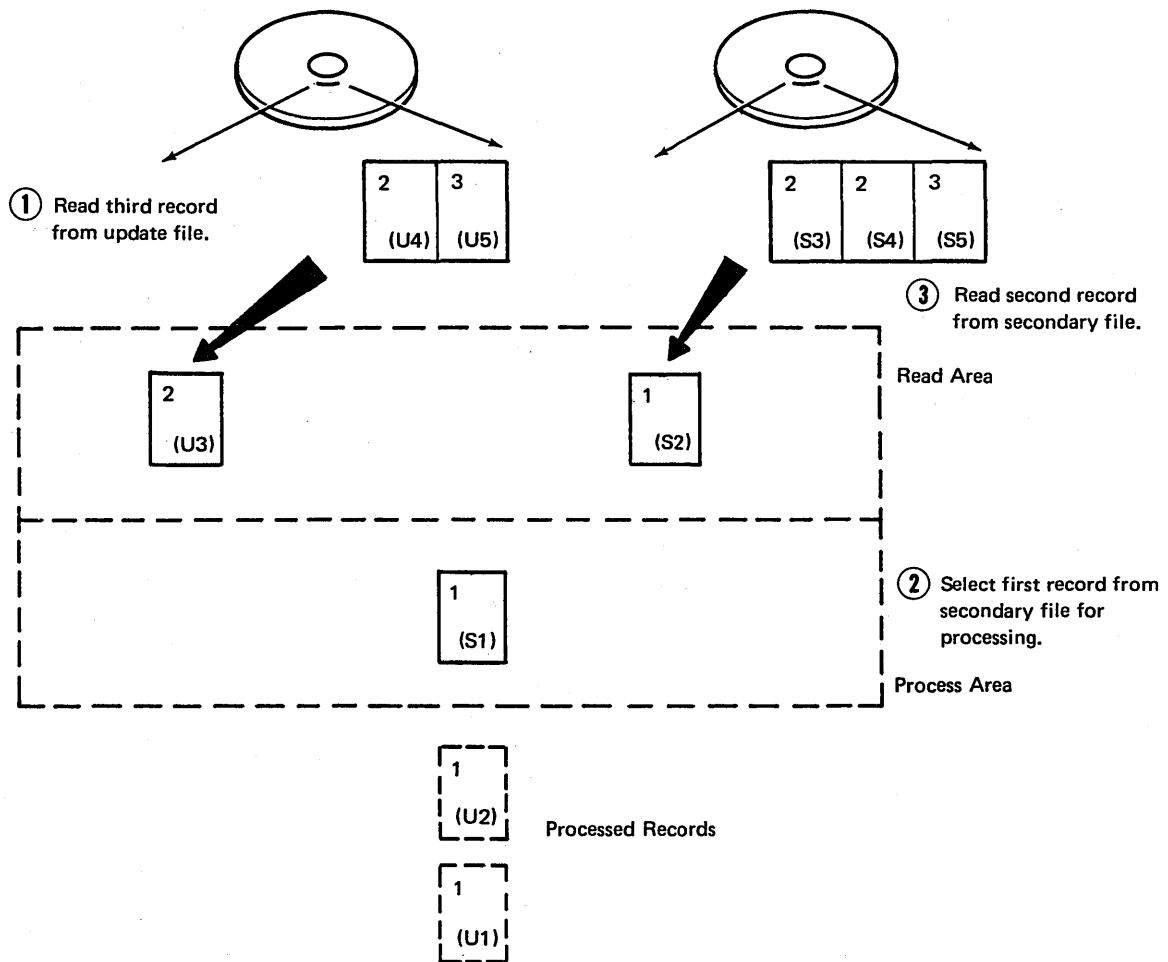


Figure 112. Available Records: One Input File, One Update File (Part 3 of 3)

IBM International Business Machines Corporation Form X21-9094
Printed in U.S.A.

RPG INPUT SPECIFICATIONS

Date _____
 Program _____
 Programmer _____

Punching Instruction	Graphic						
	Punch						

Page

1	2
---	---

 Program Identification

75	76	77	78	79	80
----	----	----	----	----	----

Line	Form Type	Filename	Sequence Number (L/N)	Option (O)	Record Identifying Indicator or	Record Identification Codes									Field Location		Field Name	Control Level (L1-L9)	Matching Fields or Chaining Fields	Field Record Relation	Field Indicators			Sterling Sign Position	
						1			2			3			From	To					Plus	Minus	Zero or Blank		
						Position	Not (N)	C/Z/D Character	Position	Not (N)	C/Z/D Character	Position	Not (N)	C/Z/D Character											
01	I	PRIMARY	AA		Φ1																				
02	I													1	1Φ	NAME1									
03	I													11	14	MATCH		M1							
04	I	SECONDARY	AB		Φ2																				
05	I													1	1Φ	NAME1									
06	I													11	14	MATCH		M1							
07	I		AC		**																				
08	I													1	1Φ	NXTNAM									
09	I																								
10	I																								
11	I																								
12	I																								

Look ahead field (field from secondary file records needed in primary file records).

Ⓑ

IBM International Business Machines Corporation Form X21-9090
Printed in U.S.A.

RPG OUTPUT - FORMAT SPECIFICATIONS

Date _____
 Program _____
 Programmer _____

Punching Instruction	Graphic						
	Punch						

Page

1	2
---	---

 Program Identification

75	76	77	78	79	80
----	----	----	----	----	----

Line	Form Type	Filename	Type (H/D/E)	Space	Skip	Output Indicators						Field Name	Edit Codes	Blank After (B)	End Position in Output Record	P = Packed/B = Binary	Sterling Sign Position								
						Before	After	Before	After	Not	And							And	Not						
																				Not	And	And	Not		
01	O	PRIMARY	D																						
02	O																								
03	O		D																						
04	O																								
05	O																								
06	O																								
07	O																								
08	O																								

Edit Codes

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

Place the look ahead field from secondary records into positions 31-40 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 113. Look Ahead Fields (Part 2 of 2)

MULTIFILE PROCESSING

Multifile processing applies to programs that read records from a primary file and one or more secondary files. It is the name given to the methods by which the programs select records for processing. The method used depends upon whether or not match fields are used in the records.

No Match Fields

When no match fields are used, records are selected from one file at a time. When the records from one file have all been processed, the records from next file are selected. The files are chosen in this order:

1. Primary file.
2. Secondary files in the order in which they are described in the file description specifications.

Match Fields

When match fields are used, records are selected according to the contents of the match fields. One record is read from every file, and the match fields in the records are compared. If the records are in ascending order, the record with the lowest match field is selected for processing. If the records are in descending order, the record with the highest match field is selected.

When a record is selected from a file, the next record from the file is read. At the beginning of the next program cycle, the new record is compared with the other records in the read area which are waiting for selection and one is selected.

Records without match fields can be included in the files. Such records are selected 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 files is:

1. Primary file.
2. Secondary files in the order in which they are described in the file description specifications.

When the primary record matches one or more of the secondary records, the MR (matching record) indicator is turned on. The indicator can be used to condition calculations or output 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.

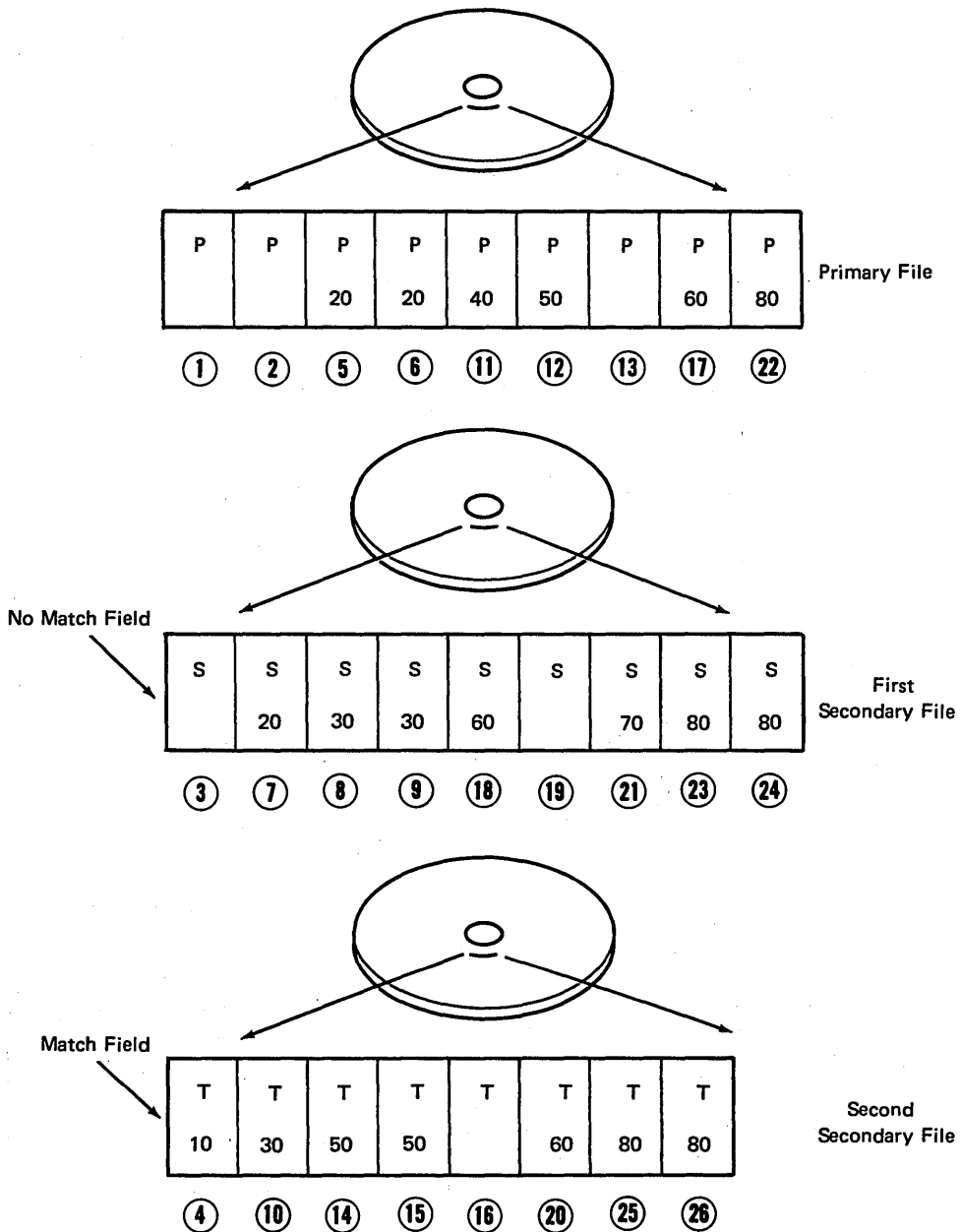
Example

Figures 114–116 show the order of record selection from three files using match fields. In Figure 114, the File Description sheet shows a primary file, a secondary file, and a second secondary file.

On the Input sheet, two record types are described for each file: one type has a match field assigned match value M1, in the other type the field in the same position is not used as a match field.

Figure 115, part 1, shows all the records from the three files. The circled numbers represent the order of selection for the 26 records shown.

Figure 115, part 2, provides a summary of the 26 cycles required to process the records shown in Figure 115, part 1. The summary tells what file is being processed in each cycle, the indicators that are on for that cycle, and a brief explanation of why a particular record was selected.



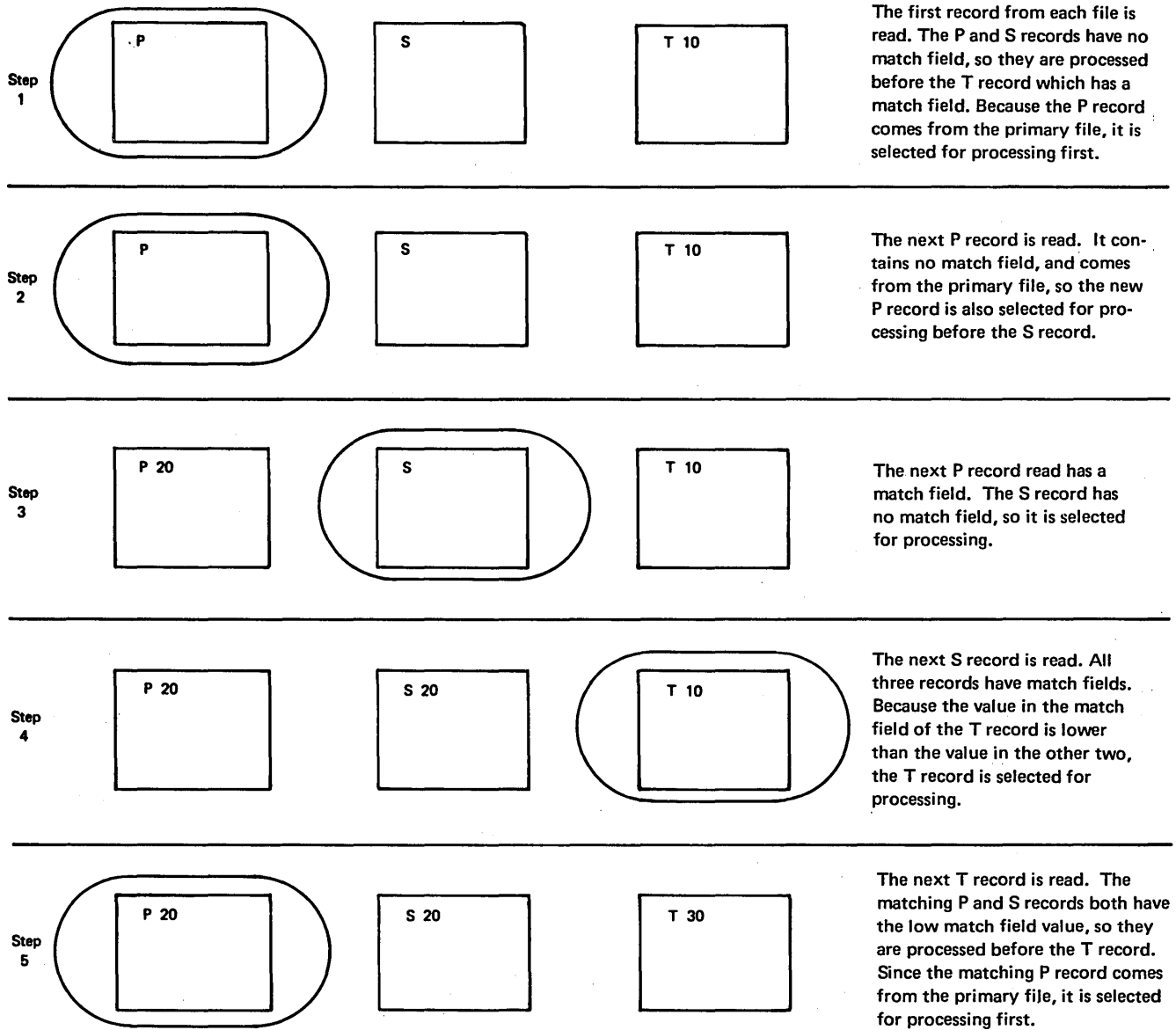
The records from the three disk files are selected in the order indicated by the circled numbers.

Figure 115. Normal Record Selection from Three Disk Files (Part 1 of 2)

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 115. Normal Record Selection from Three Disk Files (Part 2 of 2)

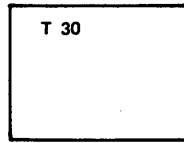
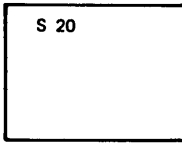
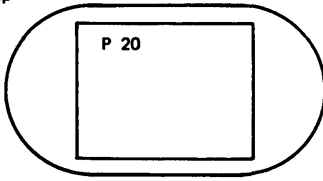
In Figure 116 a more thorough explanation is given for the selection of the first ten records from the disk files shown in Figure 115, part 1. The reasons for selection of these ten records cover most of the situations involved in multifile processing.



 ART: 550o7.2

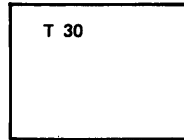
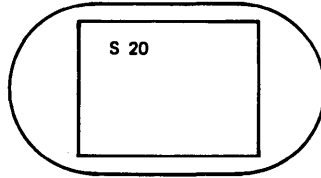
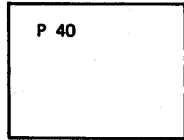
Figure 116. Normal Record Selection from Three Disk Files (Part 1 of 2)

Step
6



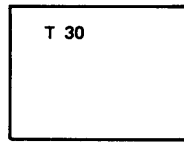
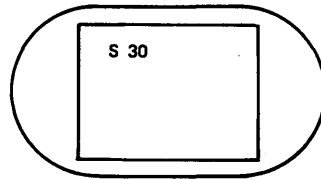
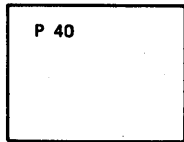
The next P record is read. Because it contains the same match field and comes from the primary file, the new P record is selected instead of the S record.

Step
7



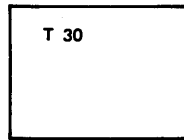
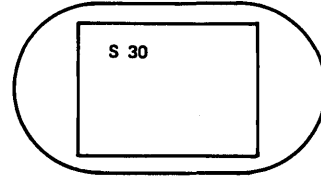
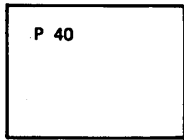
The next P record is read. The value of the match field in the S record is the lowest of the three, so the S record is selected for processing.

Step
8



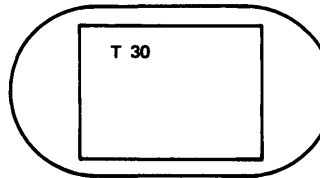
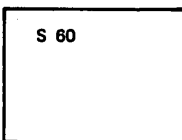
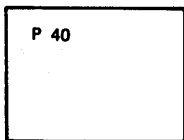
The next S record is read. Because the S and T records match and have the lowest match field, they are selected before the P record. Because the S record comes from the first secondary file, it is selected for processing before the T record.

Step
9



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.

Step
10



The next S record is read. The T record contains the lowest match field value, and is selected for processing.

ART: 55007.3

Figure 116. Normal Record Selection from Three Disk Files (Part 2 of 2)

OPERATION CODES

You are able to perform many different types of operations on your data using the RPG II language. Special codes have been set up which indicate the operation to be performed. Usually these are just abbreviations of the name of the operation. You must use these codes to specify the operation to be performed.

Operations may be divided into 12 categories; all codes in each category are explained in this section. Examples are also given for many codes. Figure 117 provides a summary of the operation codes.

It also shows what other specifications need to be used with each code.

Type of Operation	Function of Operation	Operation Code (columns 28-32)	Control Level	Indicators	Factor 1	Factor 2	Result Field	Field Length	Decimal Position	Half Adjust	Resulting Indicators
Arithmetic Operations	Add Factor 2 to Factor 1.	ADD	O	O	R	R	R	O	O	O	O
	Clear Result Field and add Factor 2.	Z-ADD	O	O	B	R	R	O	O	O	O
	Subtract Factor 2 from Factor 1.	SUB	O	O	R	R	R	O	O	O	O
	Clear Result Field and subtract Factor 2.	Z-SUB	O	O	B	R	R	O	O	O	O
	Multiply Factor 1 by Factor 2.	MULT	O	O	R	R	R	O	O	O	O
	Divide Factor 1 by Factor 2.	DIV	O	O	R	R	R	O	O	O	O
	Move remainder of preceding division to a Result Field.	MVR	O	O	B	B	R	O	O	O	O
	Sum elements of an array and put sum in Result Field.	XFOOT	O	O	B	R	R	O	O	O	O
Derive the square root of Factor 2.	SQRT	O	O	B	R	R	O	O	O	B	
Move Operation	Move Factor 2 into Result Field, right justified.	MOVE	O	O	B	R	R	O	O	B	B
	Move Factor 2 into Result Field, left justified.	MOVEL	O	O	B	R	R	O	O	B	B
Move Zone Operations	Move zone from low-order position of Factor 2 to low-order position of Result Field.	MLLZO	O	O	B	R	R	O	O	B	B
	Move zone from high-order position of alphameric Factor 2 to high-order of alphameric Result Field.	MHHZO	O	O	B	R	R	O	B	B	B
	Move zone from low-order position of Factor 2 to high-order position of alphameric Result Field.	MLHZO	O	O	B	R	R	O	B	B	B
	Move zone from high-order position of alphameric Factor 2 to low-order position of Result Field.	MHLZO	O	O	B	R	R	O	O	B	B
Compare and Zone Testing Operations	Compare Factor 1 to Factor 2.	COMP	O	O	R	R	B	B	B	B	R
	Identify the zone in the leftmost position of an alphameric Result Field.	TESTZ	O	O	B	B	R	O	B	B	R
Bit Operations	Set on specified bits.	BITON	O	O	B	R	R	O	B	B	B
	Set off specified bits.	BITOF	O	O	B	R	R	O	B	B	B
	Test specified bits.	TESTB	O	O	B	R	R	O	B	B	R
Setting Indicators	Set one, two, or three specific indicators on.	SETON	O	O	B	B	B	B	B	B	R
	Set one, two, or three specific indicators off.	SETOF	O	O	B	B	B	B	B	B	R
Branching Within RPG II	Branch to another RPG II calculation specification line.	GOTO	O	O	B	R	B	B	B	B	B
	Identify the name in Factor 1 as a destination label to which GOTO may branch.	TAG	O	B	R	B	B	B	B	B	B
Branching to External Subroutines	Branch to user-written Assembler subroutine.	EXIT	O	O	B	R	B	B	B	B	B
	Transfer data to user-written Assembler subroutine.	RLABL	O	B	B	B	R	O	O	B	B
Lookup Operations	Table Lookup.	LOKUP	O	O	R	R	O	O	O	B	R
	Array Lookup.	LOKUP	O	O	R	R	B	B	B	B	R
Subroutine	Beginning of the subroutine.	BEGSR	*	B	R	B	B	B	B	B	B
	End of the subroutine.	ENDSR	*	B	O	B	B	B	B	B	B
	Call to execute the subroutine.	EXSR	O	O	B	R	B	B	B	B	B
Program Control of Input and Output	Pause for input data from keyboard.	KEYnn**	O	O	B	O	O	O	O	B	O
	Control of printer and ledger card output.	SETnn**	O	O	B	O	O	O	B	B	O
	Call for immediate input.	READ	O	O	B	R	B	B	B	B	***
	Forcing record to be read on next cycle.	FORCE	B	O	B	R	B	B	B	B	B
	Forcing output printing.	EXCPT	O	O	B	B	B	B	B	B	B
	A field is printed on the printer-keyboard and/or data is entered via the printer-keyboard into a field.	DSPLY	O	O	O	R	O	B	B	B	B
A record is read from a disk file.	CHAIN	O	O	R	R	B	B	B	B	****	
Debug Function	Aid in finding programming errors.	DEBUG	O	O	O	R	O	B	B	B	B

O — Optional
 R — Required
 B — Blank

* Columns 7—8 must have an SR entry for all subroutine lines.
 ** The nn entries in columns 31—32 are for field light numbers.
 *** Columns 58—59 may contain an indicator for this operation; columns 54—57 must be blank.
 **** A resulting indicator should be entered in columns 54—55; columns 56—59 must be blank.

55250A

Figure 117. Operation Codes

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 may occur, the position of the decimal point in the Result Field is not affected. For arithmetic operations in which all three fields are used:

1. Factor 1, Factor 2, and the Result Field may all be different fields.
2. Factor 1, Factor 2, and the Result Field may all be the same field.
3. Factor 1 and Factor 2 may be the same field but different from the Result Field.
4. Either Factor 1 or Factor 2 may be the same as the Result Field.

The length of any field involved in an arithmetic operation cannot exceed 15 characters. If the result exceeds 15 characters, characters may be dropped from either or both ends depending on the location of the decimal point. The results of all operations are signed (+, -). Any data placed in the result field replaces the data that was there previously.

Add (ADD)

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.

Zero and Add (Z-ADD)

Factor 2 is added to a field of zeros. The sum is placed in the Result Field. Factor 1 is not used.

Subtract (SUB)

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.

Note: Subtracting two fields which are the same is a method of setting the result field to zero.

Zero and Subtract (Z-SUB)

Factor 2 is subtracted from a field of zeros. The difference is placed in the Result Field. This actually places the negative of Factor 2 in the Result Field. This operation can be used to change the sign of a field. Factor 1 is not used.

Multiply (MULT)

Factor 1 is multiplied by Factor 2. The product is then placed in the Result Field. Factor 1 and Factor 2 are not changed. You must be sure the Result Field is large enough to hold the product. In order to determine the minimum result field length, use this rule: Result Field length equals the length of Factor 1 plus the length of Factor 2 plus one.

Divide (DIV)

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 will be 0. Factor 2 cannot be 0.

If it is, the job stops immediately. You may continue processing, however, by pressing the start key. When processing is continued, the result and remainder are set to zero.

Any remainder resulting from the divide operation is lost unless the move remainder 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).

Move Remainder (MVR)

This operation moves the remainder from the previous divide operation to a separate field named under 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 118 shows the use of the move remainder operation.

Square Root (SQRT)

This operation derives the square root of the field named in Factor 2. The square root of Factor 2 is placed in the Result Field. Do not use Factor 1.

A whole array can be used in a SQRT operation if Factor 2 and Result Field contain array names.

The number of decimal places in the Result Field may be 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 will halt. You may continue processing by pressing the start key. When processing is continued, the Result Field is set to zero.

Crossfoot (XFOOT)

This operation is used only on numeric arrays. It adds all the elements of the array together and puts the sum into a separate field specified as the result field. Factor 1 is not used. Factor 2 contains the name of the array.

Move Operations

Move operations move part or all of Factor 2 to the Result Field. Factor 2 remains unchanged, but the result field is changed.

Factor 1 is not used in any move operations. It must always be blank. No resulting indicators may be used. Numeric fields may be changed to alphameric fields and alphameric fields may be changed to numeric fields by the move operations. To change a numeric field to an alphameric field, place the name of the numeric field in the Factor 2 columns and use an alphameric result field. To change an alphameric field to a numeric field, place the name of the alphameric field in the Factor 2 columns and use a numeric result field.

When move operations are specified to move data into numeric fields, decimal positions are ignored. For example, if the data 1.00 is moved into a numeric field with one decimal position, the result is 10.0.

IBM

Date _____

Program _____

Programmer _____

International Business Machines Corporation

RPG CALCULATION SPECIFICATIONS

Form X21-9093
Printed in U.S.A.

Page 1 2 Program Identification 75 76 77 78 79 80

Line	Form Type	Indicators										Factor 1	Operation	Factor 2	Result Field	Field Length	Decimal Positions Half Adjust (H)	Resulting Indicators			Comments
		Control Level (L, U, L, R, S, P)			Not	And	And	Plus	Minus	Zero	Compare							High	Low	Equal	
		1	2	3																	
		4	5	6																	
01	C												FIELDA	DIV	FIELDB	SAVE	6				
02	C												FIELDA	MVR	STORE	6					
03	C																				
04	C																				
05	C																				

Figure 118. Move Remainder Operation

Move (MOVE)

This operation causes characters from factor 2 to be moved to the rightmost positions in the Result Field. Moving starts with the rightmost characters.

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.

An alphameric field or constant may be changed into a numeric field by moving it into a numeric field. When this

is specified, 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. However, the zone portion of the rightmost alphameric character is converted to a corresponding sign and is moved to the rightmost position of the numeric field where it becomes the sign of the field. A numeric field may also be changed into an alphameric field by moving it into an alphameric field. The MOVE operation is summarized in Figure 119.

Factor 2 Shorter Than Result Field				
	<u>Factor 2</u>		<u>Result Field</u>	
a. Alphameric	P H 4 S N	Before MOVE Operation	1 2 3 4 5 6 7 8 4 ⁺	Alphameric
	P H 4 S N	After Move Operation	1 2 3 4 P H 4 S N	
b. Alphameric	P H 4 S N	Before MOVE Operation	1 2 3 4 5 6 7 8 4 ⁺	Numeric
	P H 4 S N	After MOVE Operation	1 2 3 4 7 8 4 2 5 ⁻	
c. Numeric	1 2 7 8 4 2 5	Before MOVE Operation	1 2 3 4 5 6 7 8 9	Numeric
	1 2 7 8 4 2 5	After MOVE Operation	1 2 1 2 7 8 4 2 5	
d. Numeric	1 2 7 8 4 2 5	Before MOVE Operation	A C F G P H 4 S N	Alphameric
	1 2 7 8 4 2 5	After MOVE Operation	A C 1 2 7 8 4 2 5	
Factor 2 Longer Than Result Field				
	<u>Factor 2</u>		<u>Result Field</u>	
a. Alphameric	A C E G P H 4 S N	Before MOVE Operation	5 6 7 8 4	Alphameric
	A C E G P H 4 S N	After MOVE Operation	P H 4 S N	
b. Alphameric	A C E G P H 4 S N	Before MOVE Operation	5 6 7 8 4 ⁺	Numeric
	A C E G P H 4 S N	After MOVE Operation	7 8 4 2 5 ⁻	
c. Numeric	1 2 7 8 4 2 5	Before MOVE Operation	5 6 7 4 8	Numeric
	1 2 7 8 4 2 5	After MOVE Operation	7 8 4 2 5	
d. Numeric	1 2 7 8 4 2 5	Before MOVE Operation	P H 4 S N	Alphameric
	1 2 7 8 4 2 5	After MOVE Operation	7 8 4 2 5	
Factor 2 and Result Field Same Length				
	<u>Factor 2</u>		<u>Result Field</u>	
a. Alphameric	P H 4 S N	Before MOVE Operation	5 6 7 8 4	Alphameric
	P H 4 S N	After MOVE Operation	P H 4 S N	
b. Alphameric	P H 4 S N	Before MOVE Operation	5 6 7 8 4	Numeric
	P H 4 S N	After MOVE Operation	7 8 4 2 5 ⁻	
c. Numeric	7 8 4 2 5 ⁻	Before MOVE Operation	A L T 5 F	Numeric
	7 8 4 2 5 ⁻	After MOVE Operation	7 8 4 2 5 ⁻	
d. Numeric	7 8 4 2 5 ⁻	Before MOVE Operation	A L T 5 F	Alphameric
	7 8 4 2 5 ⁻	After MOVE Operation	7 8 4 2 N	

⁺
 4 = letter D
⁻
 5 = letter N

Figure 119. MOVE Operations

Move Left (MOVE L)

This operation causes characters from Factor 2 to be moved to the leftmost positions in the Result Field. Moving begins with the leftmost characters.

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 excess rightmost characters in the Result Field are unchanged. In this case the sign of a numeric field is not changed either.

An alphameric field or constant may be changed into a numeric field by moving it into a numeric Result Field. When this is specified, the digit portion of each character is converted to its corresponding numeric character and then moved into the Result Field.

Blanks are transferred as zeros. If the rightmost character is moved, the zone is also converted and used as the sign of the field. When the rightmost character is not transferred, the zone is, nevertheless, still transferred and used as the sign of the Result Field.

A numeric field may also be changed into an alphameric field by moving it 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 rules for MOVE L transfers are as follows (See also Figure 120).

1. Factor 2 is the same length as the Result Field.
 - a. Factor 2 and Result Field numeric: the sign is moved with the rightmost position.
 - b. Factor 2 numeric, Result Field alphameric; the sign is moved with the rightmost position.
 - c. Factor 2 alphameric, Result Field numeric: if the zone from the rightmost position of Factor 2 is a D (minus zone), a minus zone will be moved into the rightmost position of the Result Field. If the zone from the rightmost position of Factor 2 is not a D, a positive zone will be moved into the rightmost position of the Result Field. Digit portions are converted to their corresponding numeric characters.
 - d. Factor 2 and Result Field alphameric: all characters are moved.
2. Factor 2 is longer than the Result Field.
 - a. Factor 2 and Result Field numeric: the sign from the rightmost position of Factor 2 is moved into the rightmost position of the Result Field.
 - b. Factor 2 numeric, Result Field alphameric: the Result Field contains only numeric characters.
 - c. Factor 2 alphameric, Result Field numeric: if the zone from the rightmost position of Factor 2 is a D (minus zone), a minus zone will be moved into the rightmost position of the Result Field. If the zone from the rightmost position of Factor 2 is not a D, a positive zone will be moved into the rightmost position of the Result Field. Other Result Field positions contain only numeric characters.
 - d. Factor 2 and Result Field alphameric: only the number of characters needed to fill the Result Field are moved.
3. Factor 2 is shorter than the Result Field.
 - a. Factor 2 either numeric or alphameric, Result Field numeric: digit portion of Factor 2 replaces the contents of the leftmost positions in the Result Field. The sign in the rightmost position of the Result Field is not changed.
 - b. Factor 2 either numeric or alphameric, Result Field alphameric: 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.

	Factor 2	Factor 2 and Result Field Same Length	Result Field	
a. Numeric	7, 8, 4, 2, 5	Before MOVE _L Operation	5, 6, 7, 8, 4	Numeric
	7, 8, 4, 2, 5	After MOVE _L Operation	7, 8, 4, 2, 5	
b. Numeric	7, 8, 4, 2, 5	Before MOVE _L	A, K, T, 4, D	Alphameric
	7, 8, 4, 2, 5 (5=letter N)	After MOVE _L	7, 8, 4, 2, N	
c. Alphameric	P, H, 4, S, N	Before MOVE _L	5, 6, 7, 8, 4	Numeric
	P, H, 4, S, N	After MOVE _L	7, 8, 4, 2, 5	
d. Alphameric	P, H, 4, S, N	Before MOVE _L	A, K, T, 4, D	Alphameric
	P, H, 4, S, N	After MOVE _L	P, H, 4, S, N	
	Factor 2	Factor 2 Longer Than Result Field	Result Field	
a. Numeric	0, 0, 0, 0, 0, 8, 4, 2, 5	Before MOVE _L Operation	5, 6, 7, 8, 4	Numeric
	0, 0, 0, 0, 0, 8, 4, 2, 5	After MOVE _L Operation	0, 0, 0, 0, 0	
b. Numeric	9, 0, 3, 1, 7, 8, 4, 2, 5	Before MOVE _L	A, K, T, 4, D	Alphameric
	9, 0, 3, 1, 7, 8, 4, 2, 5	After MOVE _L	9, 0, 3, 1, 7	
c. Alphameric	B, R, W, C, X, H, 4, S, N	Before MOVE _L	5, 6, 7, 8, 4	Numeric
	B, R, W, C, X, H, 4, S, N	After MOVE _L	2, 9, 6, 3, 7	
d. Alphameric	B, R, W, C, X, H, 4, S, N	Before MOVE _L	A, K, T, 4, D	Alphameric
	B, R, W, C, X, H, 4, S, N	After MOVE _L	B, R, W, C, X	
	Factor 2	Factor 2 Shorter Than Result Field	Result Field	
a. Numeric	7, 8, 4, 2, 5	Before MOVE _L Operation	1, 3, 0, 9, 4, 3, 2, 1, 0	Numeric
	7, 8, 4, 2, 5	After MOVE _L Operation	7, 8, 4, 2, 5, 3, 2, 1, 0	
Alphameric	C, P, T, 5, N	Before MOVE _L	1, 3, 0, 9, 4, 3, 2, 1, 0	Numeric
	C, P, T, 5, N	After MOVE _L	3, 7, 3, 5, 5, 3, 2, 1, 0	
b. Numeric	7, 8, 4, 2, 5	Before MOVE _L	B, R, W, C, X, H, 4, S, A	Alphameric
	7, 8, 4, 2, 5	After MOVE _L	7, 8, 4, 2, N, H, 4, S, A	
Alphameric	C, P, T, 5, N	Before MOVE _L	B, R, W, C, X, H, 4, S, A	Alphameric
	C, P, T, 5, N	After MOVE _L	C, P, T, 5, N, H, 4, S, A	

The arrow ↓ between numbers indicates a decimal point.

Figure 120. MOVE_L Operations

Move Zone Operations

These operations are used only to move the zone portion of a character. There are four varieties of the move zone operation (Figure 121).

Using a minus (-) sign in a move zone operation will not yield a negative character in the result field, since minus is represented by a X'60' internally, and a D zone is required for a negative character. Characters J-R have D zone representations, and can be used to obtain a negative value (J = X'D1', . . . , R = X'D9').

Note: Whenever the word high is used, the field involved must be alphameric; whenever low is used, the field involved may be either alphameric or numeric.

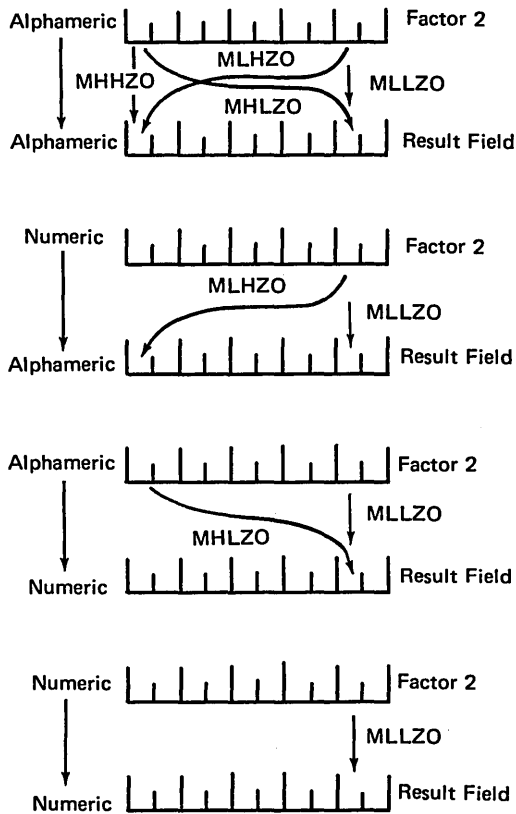


Figure 121. Function of Move Zone Operations

Move High to High Zone (MHHZO)

This 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.

Move High to Low Zone (MHLZO)

This operation moves the zone from the leftmost position of Factor 2 to the rightmost position of the Result Field. Factor 2 can be only alphameric. The Result Field may be either alphameric or numeric.

Move Low to Low Zone (MLLZO)

This 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 may be either alphameric or numeric.

Move Low to High Zone (MLHZO)

This 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 can only be alphameric.

Compare and Testing Operations

These operations test fields for certain conditions. The result of the test is shown by the resulting indicators assigned in columns 54-59. No fields are changed by these operations.

Compare (COMP)

This operation causes Factor 1 to be compared with Factor 2. As a result of the compare, indicators are turned 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 either be both alphameric or both numeric.

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 (Figure 122). The maximum field length for alphameric fields which are to be compared is 256 characters.

If the fields which are to be compared are numeric they are aligned according to the decimal point. Any missing digits are filled in with zeros (Figure 123). The maximum field length for numeric fields which are to be compared is 15 digits.

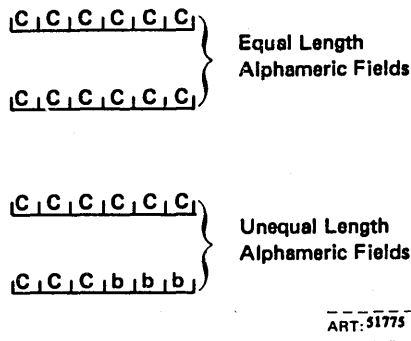


Figure 122. Comparison of Alphameric Fields

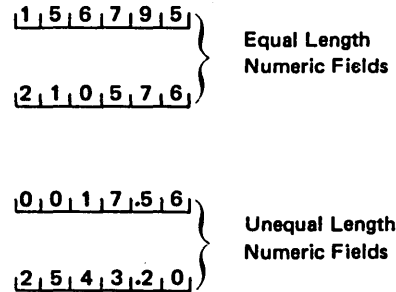


Figure 123. Comparison of Numeric Fields

If an alternate collating sequence is defined, alphameric fields are compared according to the sequence.

Figure 124 shows some specifications for compare operations.

Line 01: The contents of the field SLS69 (1969 sales) are compared with the contents of SLS70. If 1969 sales exceed 1970 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.

Line 03: The alphameric constant OCTOBER is compared against 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 after the compare operation.

Line 05: The contents of the field named GRSPAY (which must be defined as numeric) is 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.

Line 07: The contents of the field NETPAY (which must be defined as numeric) is decimal-aligned with numeric constant 0 and then compared to it. If NETPAY is greater than zero, indicator H1 remains off after the compare operation. If NETPAY is zero or negative, indicator H1 turns on.

IBM		International Business Machines Corporation		Form X21-6093 Printed in U.S.A.							
Date _____		RPG CALCULATION SPECIFICATIONS		Page 1 2							
Program _____		Punching Instruction		Program Identification							
Programmer _____		Graphic		75 76 77 78 79 80							
Punch											
Line	Form Type	Control Level (C, D, L, R, SR)	Indicators	Factor 1	Operation	Factor 2	Result Field	Field Length	Decimal Positions	Resulting Indicators	Comments
			And And							Arithmetic	
			Not Not Not							Plus Minus Zero	
										Compare	
										High Low Equal	
										1 > 2 1 < 2 1 = 2	
										Lookup	
										Table (Factor 2) is	
										High Low Equal	
										64 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74	
01	C			SLS69	COMP	SLS70				212630	
02	C										
03	C			'OCTOBER'	COMP	MONTH				131315	
04	C										
05	C			GRSPAY	COMP	1250.00				040504	
06	C										
07	C			NETPAY	COMP	0				H1H1	
08	C										
09	C										

Figure 124. Compare Operations

Test Zone (TESTZ)

This operation tests the zone of the leftmost character in the Result Field. The result field must be alphameric since this operation can be done only on alphameric characters. Resulting indicators are used to determine the results of the test. The zone portion of characters &, A–I causes the plus indicator to turn on. The zone portion of the characters - (minus), J–R causes the minus indicator to turn on. All other characters, when tested, cause the zero indicator (columns 58–59) to turn on. Factor 1 and Factor 2 are not used in this operation.

Bit Operations

Three operation codes BITON, BITOF, and TESTB, are provided to set and test individual bits. The individual bits can be used as switches in a program, thus allowing a saving of storage for binary type switches.

Note: When using the BITON, BITOF, and TESTB operations, any field named in Factor 2 or Result Field must be a 1-position alphameric field. A field is considered alphameric if there are no entries in the Decimal Positions columns on the Input or Calculation sheet.

Set Bit On (BITON)

This operation code causes bits identified in Factor 2, to turn on (set to 1) in a field named as the Result Field.

Factor 2 is always used as a source of bits for the Result Field. The bits to be turned on may be entered as literals in Factor 2, or may be contained in a field named in Factor 2. If literal values are entered in Factor 2, the bits to be turned on are identified by the numbers 0–7 (0 is the first bit number). The bit numbers must be enclosed by apostrophes, and the entry must begin in column 33. From one to eight bits may be identified in Factor 2 for a BITON operation. If a field name is entered in Factor 2, the field must be a 1-position alphameric field. The bits that are on in the Factor 2 field will be turned on in the Result Field. The field in Factor 2 may be an array element if each element in the array is a 1-position alphameric element. When Factor 2 is a field name or array element, apostrophes are not required, and the entry begins in column 33.

The Result Field is the field in which the bits are turned on. The field named here must be 1-position alphameric field. The Result Field may be an array element, if each element in the array is a 1-position alphameric element. See Figure 125 for a summary of the BITON operation.

The operation code BITON must appear in columns 28–32. Conditioning indicators may be used in columns 7–17, and any entry under Field Length must be 1.

Factor 1, Decimal Positions, Half-Adjust, and the Resulting Indicator columns must be blank.

Set Bit Off (BITOF)

This operation code causes bits identified in Factor 2, to turn off (set to 0) in a field named under Result Field.

Factor 2 is always used as a source of bits for the Result Field. The bits to be turned off can be entered as literals in Factor 2, or may be contained in a field named in Factor 2. If literal values are entered in Factor 2, the bits to be turned off are identified by the numbers 0–7 (0 is the first bit number). The bit numbers must be enclosed by apostrophes, and the entry must begin in column 33. From one to eight bits may be specified in Factor 2 for a BITON operation. If a field name is entered in Factor 2, the field must be a 1-position alphameric field. The bits that are on in the Factor 2 field will be turned off in the Result Field. The field in Factor 2 may be an array element if each element in the array is a 1-position alphameric element. When Factor 2 is a field name or array element, apostrophes are not required, and the entry begins in column 33.

The Result Field is the field in which the bits are turned off. The field named here must be a 1-position alphameric field. The Result Field may be an array element if each element in the array is a 1-position alphameric element. See Figure 126 for a summary of the BITOF operation.

The operation code BITOF must appear in columns 28–32. Conditioning indicators may be used in columns 7–17, and any entry under Field Length must be 1.

Factor 1, Decimal Positions, Half-Adjust, and the Resulting Indicator columns must be blank.

RPG CALCULATION SPECIFICATIONS

Date _____
Program _____
Programmer _____

Punching Instruction	Graphic								
	Punch								

Page 1 2

Program Identification '75 76 77 78 79 80

Line	Form Type	Control Level (L, O, L, R, S, F)	Indicators						Factor 1	Operation	Factor 2	Result Field	Field Length	Decimal Positions Half Adjust (H)	Resulting Indicators			Comments		
			Not	And	And	And	And	And							Plus	Minus	Zero			
3	c	*																		
4	c	*	THE FOLLOWING BITOF OPERATION WILL SET BIT 5 OFF IN THE FIELD																	
5	c	*	NAMED BITSW. THE FIELD IS DEFINED IN THE SAME LINE WITH A FIELD																	
6	c	*	LENGTH OF 1.																	
7	c																			
8	c																			
9	c																			
10	c																			
11	c																			
12	c																			
13	c																			
14	c																			
15	c																			
16	c																			
17	c																			
18	c																			
19	c																			
20	c																			
21	c																			
22	c																			
23	c																			
24	c																			
25	c																			
26	c																			
27	c																			
28	c																			
29	c																			
30	c																			
31	c																			
32	c																			
33	c																			
34	c																			
35	c																			
36	c																			
37	c																			
38	c																			
39	c																			
40	c																			
41	c																			
42	c																			
43	c																			
44	c																			
45	c																			
46	c																			
47	c																			
48	c																			
49	c																			
50	c																			
51	c																			
52	c																			
53	c																			
54	c																			
55	c																			
56	c																			
57	c																			
58	c																			
59	c																			
60	c																			
61	c																			
62	c																			
63	c																			
64	c																			
65	c																			
66	c																			
67	c																			
68	c																			
69	c																			
70	c																			
71	c																			
72	c																			
73	c																			
74	c																			
75	c																			
76	c																			
77	c																			
78	c																			
79	c																			
80	c																			

Figure 126. Summary of BITOF Operations

Test Bit (TESTB)

This operation code causes bits identified in Factor 2, to be compared with corresponding bits in a field named as the Result Field. The condition of the Result Field bits is known by the use of resulting indicators in columns 54–59.

Factor 2 is always used as a source of bits for the comparison with the Result Field. The bits to be tested may be entered as literals in Factor 2, or may be contained in a field named in Factor 2. If literal values are entered, the bits to be compared are identified by the numbers 0–7 (0 is the first bit number). The bit numbers must be enclosed by apostrophes, and the entry must begin in column 33. From one to eight bits may be identified in Factor 2 for a TESTB operation. If a field name is entered in Factor 2, the field must be a 1-position alphameric field. The bits that are on in the Factor 2 field will be compared with corresponding bits in the Result Field. The condition of the Result Field bits is known by the use of resulting indicators in columns 54–59. The field in Factor 2 may be an array element if each element in the array is a 1-position alphameric element. When Factor 2 is a field name or array element, apostrophes are not required, and the entry begins in column 33.

The Result Field is the field which corresponding bits are compared from the bits specified in Factor 2. The result field must be a 1-position alphameric field. This field may be an array element if each element in the array is a 1-position alphameric field. See Figure 127 for a summary of the TESTB operation.

Indicators may be assigned in columns 54–59 to reflect the status of the Result Field bits. At least one indicator should be assigned, and as many as three can be assigned for one operation. Two indicators may be the same for a TESTB operation, but not three. A resulting indicator has the following meanings for these columns:

Columns 54–55: An indicator in these columns turns on if each literal 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 literal 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 literal bit specified in Factor 2, or each bit that is on in the Factor 2 field is on in the Result Field.

The operation code TESTB must appear in columns 28–32. Conditioning indicators may be used in columns 7–17, and any entry under Field Length must be 1. At least one resulting indicator should be assigned in columns 54–59. As many as three resulting indicators may be assigned, but not more than two can be the same.

Factor 1, Decimal Positions, and the Half-Adjust columns must be blank.

RPG CALCULATION SPECIFICATIONS

Date _____

Program _____

Programmer _____

Punching Instruction	Graphic					
	Punch					

Page

1	2
---	---

Program Identification

75	76	77	78	79	80
----	----	----	----	----	----

Line	Form Type	Control Level (LD, LR, SR)	Indicators						Factor 1	Operation	Factor 2	Result Field	Field Length	Decimal Positions	Resulting Indicators			Comments		
			Not	And	And	Not	Not	Not							Plus	Minus	Zero			
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																				

Figure 127. Summary of TESTB Operations

Setting Indicators

These operation codes are used to turn indicators off or on. Any indicator to be turned on or off is specified in columns 54–59. The headings in the Resulting Indicators field (Plus or High, Minus or Low, Zero or Equal) have no meaning in these operations. When setting indicators, remember:

1. The following indicators cannot be turned on by the SETON operation: 1P, MR, LO, U1–U8, KA–KN, KP, KQ.
2. The following indicators may not be turned off by the SETOF operation: 1P, MR, LO, LR, U1–U8.
3. If the LR indicator is turned on by a SETON operation which is conditioned with a control level indicator (columns 7–8 of the Calculation sheet), 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 7–8), processing stops after the next total output operation is completed.
4. If the halt indicators (H1–H9) are set on and not turned off before the detail output operations are complete, the system stops. Processing may be continued by pressing the start key on the Processing Unit once for every halt indicator that is on.
5. Setting on L1–L9 or setting off L1–L9 does not automatically set any lower control level indicators.
6. Indicators L1–L9 and the record identifying indicators are always turned off after detail output operations are completed regardless of the previous set on or set off operation.
7. Whenever a new record is read, record identifying indicators (01–99) 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.
8. If a numeric indicator (01–99) is SETON and not changed in other calculations, it will remain on until SETOF.

Set On (SETON)

This operation causes any indicators in columns 54–59 to be turned on.

Set Off (SETOF)

This operation causes any indicators in columns 54–59 to be turned off.

Branching Within RPG II

Operations are normally performed in the order that they appear on the Calculation sheet. There may be times, however, when you do not want the operations performed in the order they are specified. For example, you may wish to:

1. Skip several operations when certain conditions occur.
2. Perform certain operations for several, but not all, record types.
3. Perform several operations over and over again.

Go To (GOTO)

This operation allows you to skip instructions by specifying some other instruction to go to. You can branch to an earlier line or to a later specification line. You can branch from a detail calculation line to another detail calculation line, and you can branch from a total calculation line to another total calculation line. You cannot branch from a detail calculation line to a total calculation line or vice versa. Neither can you branch from calculations conditioned by L0–L9 to calculations conditioned by LR or vice versa. (A total calculation line is defined as one which is conditioned by a control level indicator in columns 7–8 of the Calculation sheet.)

Factor 2 must contain the name of the point to which you wish to go (identified by the label on a TAG statement). The name in Factor 2 is called a label. The label can be from 1–6 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 it is not conditioned, the operation is always done. See *Examples* under *Branching Operations* for use of the GOTO operations.

Tag (TAG)

The operation code names the point to which you are branching in the GOTO operation. Factor 1 contains this

label. The name must begin in column 18. The same label cannot be used for more than one TAG instruction.

Factor 2 and the result field are not used. No indicators may be entered in columns 9–17 for a TAG instruction. Control level indicators must be used, however, if branching is to occur only when the information in a control field has changed. See *Examples* under *Branching Operations* for use of the TAG operation.

Examples

Example 1: Figure 128 shows how TAG and GOTO may be used to skip operations on certain conditions.

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 10–12 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 06–12. Nothing is done in line 09 since TAG only gives a name. These same operations are performed again and again until 15 does turn on.
5. When 15 is on, the branch to RTN2 is not taken. The TESTZ operation is then performed. If this operation causes 20 to turn on, a branch is taken to line 17 (GOTO END). If 20 is not on, the operation in line 16 is done.

IBM						International Business Machines Corporation																														Form X21-9093 Printed in U.S.A.	
RPG CALCULATION SPECIFICATIONS						Punching Instruction						Page		Program Identification																							
Date _____						Graphic						1 2		75 76 77 78 79 80																							
Program _____						Punch																															
Programmer _____																																					
Line	Form Type	Control Level (LO-LB, LF, SF)	Indicators			Factor 1	Operation	Factor 2	Result Field	Field Length	Decimal Positions	Resulting Indicators			Comments																						
			Not	And	And							Arithmetic	Plus	Minus		Zero																					
			Not	And	And						High	Low	Equal																								
											1 > 2	1 < 2	1 = 2																								
											Lookup																										
											Table (Factor 2) is																										
											High	Low	Equal																								
											54	55	56	57																							
											58	59	60	61																							
											62	63	64	65																							
											66	67	68	69																							
											70	71	72	73																							
											74	75	76	77																							
01	C					FIELDA	SUB	FIELDB	FIELDDB	52																											
02	C		10				GOTO	RTN1																													
03	C					FIELDB	MULT	4	SAVE	82																											
04	C						GOTO	RTN1																													
05	C					RTN2	TAG																														
06	C																																				
07	C																																				
08	C																																				
09	C					RTN1	TAG																														
10	C																																				
11	C																																				
12	C												15																								
13	C		W15				GOTO	RTN2																													
14	C						TESTZ		FIELD C		20	20																									
15	C		20				GOTO	END																													
	C						MHLZ0	FIELD C	FIELD D																												
	C					END	TAG																														

Figure 128. Using GOTO and TAG (Skipping Operations)

EXCPT line is printed out. One is added to COUNT in order to keep track of how many times the line has been 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 8 is subtracted from the COUNT field so that it will be zero for the next cycle.

Branching to External Subroutines

Linkage from RPG II to an assembler language subroutine is accomplished through the EXIT and RLABL operations. Control cannot be transferred from one user assembler subroutine to another user assembler subroutine.

Exit to an External Subroutine (EXIT)

The EXIT operation code is used to designate a point in the calculation specifications where control is to be transferred from RPG II to a user-written assembler language subroutine. (A description of how to code assembler subroutines to be referenced in an EXIT operation is available in the *IBM System/3 Disk System Basic Assembler Program Reference Manual*, SC21-7509.)

The operation code EXIT must be entered in columns 28–32. Factor 2 contains the name of the subroutine that will receive control. The subroutine name must be in the form SUBRxx, where x is any alphabetic character. The EXIT operation can be conditioned by indicators (columns 7–8 and columns 9–17). If not conditioned, the EXIT operation is always done at detail calculation time.

The following columns must be left blank for an EXIT operation: Columns 18–27 (Factor 1), Columns 43–48 (Result Field), Columns 49–51 (Field Length), Column 52 (Decimal Positions), Column 53 (Half Adjust), Columns 54–59 (Resulting Indicators).

RPG II Label (RLABL)

The RLABL operation allows the subroutine specified in an EXIT operation to reference a field, table, array, or indicator defined in the RPG II program. RLABL operations must be specified immediately after the EXIT operation which refers to the subroutine using the field, table, array, or indicator in the RLABL statement. All external subroutines will be a part of the root segment and will not be put into overlays. (A description of how to code assembler subroutines to be referenced in EXIT and RLABL operations is available in the *IBM System/3 Disk System Basic Assembler Reference Manual*, SC21-7509.)

The operation code RLABL must be entered in columns 28–32. Result Field must contain an indicator or the name of a field, table, or an array previously defined in RPG II specifications. When an indicator is specified in an RLABL operation, you must use the form INxx, where xx is the indicator to be transferred to the subroutine. For example, if the MR indicator is to be transferred to a subroutine, you must enter INMR in the Result Field of the RLABL operation.

Only RLABL operations specifying a field or a table or array name can have entries under Field Length (columns 49–51) and Decimal Positions (column 52).

The following columns must be left blank in an RLABL operation: Columns 7–8 (Control Level), Columns 9–17 (Indicators), Columns 18–27 (Factor 1), Columns 33–42 (Factor 2), Column 53 (Half Adjust), Columns 54–59 (Resulting Indicators).

Coding Examples

Figure 130 shows an example of the linkage when the subroutine refers to a field, table, or array defined in the RPG II program. Figure 131 shows an example of the linkage when the subroutine references indicators defined in the RPG II program.

International Business Machines Corporation

RPG CALCULATION SPECIFICATIONS

Punching Instruction	Graphic								
	Punch								

Factor 1	Operation	Factor 2	Result Field	Field Length	Decimal Positions	Half Adjust (H)	Res Ind																																		
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			
							EXIT																																		
							RLABL																																		

Field, Table, or Array Name

Figure 130. Field, Table, or Array Entry

RPG CALCULATION SPECIFICATIONS

Punching Instruction	Graphic									
	Punch									

Factor 1	Operation	Factor 2	Result Field	Field Length	Decimal Positions	Half Adjust (H)	Plur	High	Tab	High
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	E		
	EXIT	SUBRXX								
	RLABL		IN35							

Indicator

Figure 131. Indicator Entry

Look-Up Operations (LOKUP)

Look-up operations are used when searching through a table or an array to find a specific element. The LOKUP operation code causes a search to be made for a particular item in a table or array. The table or array is named in Factor 2. Factor 1 is the search word (data for which you wish to find a match in the table or array named). Factor 1, the search word, may be:

1. An alphameric or numeric constant.
2. A field name.
3. An array element.
4. A table name.

Remember that 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 in connection with LOKUP. They are used to first indicate the type of search desired and then to reflect the result of the search. A resulting indicator assigned to equal (columns 58–59) instructs the program to search for an entry in the table or array equal to the search word. The indicator turns on only if such an entry is found. If there are several entries identical to the search word, the first one that is encountered is selected.

An indicator assigned to Low (columns 56–57) 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 causes the indicator assigned to Low to turn on.

The indicator assigned to High (columns 54–55) instructs the program to find the entry that is nearest to, yet higher in sequence than, the search word. The first higher entry found causes the indicator assigned to High to turn on. In all cases the resulting indicator turns on only if the search is successful.

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 Equal and Low. The program searches for an entry that satisfied either condition with Equal given precedence; that is, if no Equal entry can be 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 using the LOKUP operation, remember:

1. The search word and each table or array item must have the same length and the same format (alphameric or numeric).
2. You may search on High, Low, High and Equal, or Low and Equal only if your table or array is in sequence.
3. No resulting indicator turns on if the entry searched for is not found.

LOKUP with One Table

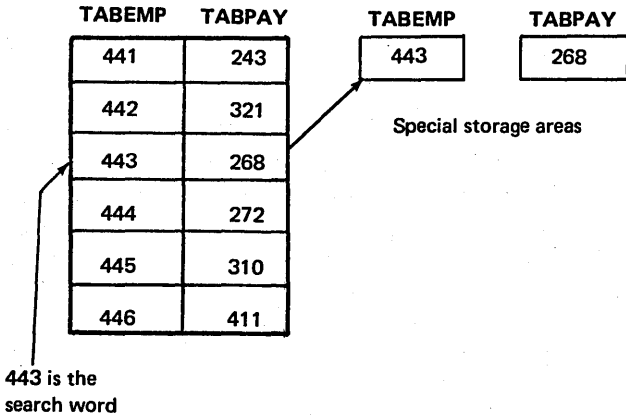
When searching a single table, Factor 1, Factor 2, and at least one resulting indicator must be specified. Conditioning indicators (specified in columns 7–17) may also be used.

Whenever a table item is found that satisfies the type of search being made (Equal, High, Low), a copy of that table item is placed in a special storage area. Every time a search is successful, the newly found table item is placed in this area, destroying what was there before. If the search is not successful, no table item is placed in the storage area. Instead the area keeps the contents it had before the unsuccessful search.

Resulting indicators are always set to reflect the result of the search. If the indicator is on, showing a successful search, you know that a copy of the item 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. When the search condition (High, Low, Equal) is satisfied, the corresponding data items from both tables are placed in their respective special storage areas and are made available for use.



Factor 1 must be the search word and Factor 2 must name the table to be searched. The result field must name the related table from which data is made available for use. Resulting indicators must also be used. Conditioning indicators (specified in columns 7-17) may be specified if needed.

The two tables involved should be the same length. If the table that is searched is longer than its related table, it is possible to satisfy the search condition. However, there will not be a data table item in the second table which corresponds to the item found in the search table. Unpredictable results may occur.

Note: Using a table name in an operation (other than LOKUP) before a successful LOKUP occurs may lead to unpredictable results, since you do not know what is in the special storage area referenced by the table name.

In Figure 132, insert A, the related tables TABEMP and TABPAY have been read into storage. Assume that an input record is read with 443 in the EMPNUM field. With 443 as

A

Line		Form Type	Control Level	Indicators			Factor 1	Operation	Factor 2	Result Field	Field Length	Decimal Positions	Resulting Indicators	Comments	
		(LO, LR, SR)	Not	And	And						Half Adjust (H)	Arithmetic			
				Not	Not	Not						Plus Minus Zero			
												Compare			
												High Low Equal			
												1 > 2 1 < 2 1 = 2			
												Lookup			
												Table (Factor 2) is			
												High Low Equal			
01	C						EMPNUM	LOKUPTABEMP	TABPAY				09		
02	C														
03	C	*					THE ABOVE OPERATION WILL SEARCH 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, 09 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 WILL MULTIPLY 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														

B

Figure 132. LOKUP with Related Tables

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. Insert B of Figure 132 shows the coding needed to perform the LOKUP operation described in insert A. It also shows how to reference the contents of the special storage area after a successful LOKUP operation.

Referencing the Table Item Found in a LOKUP Operation

Whenever a table name is used in an operation other than LOKUP, the table name really refers to the data placed in the special table storage area by the last successful search. Thus, by specifying the table name in this fashion, you can use data items from a table in calculation operations.

If the table is used as Factor 1 in a LOKUP operation, the contents of the special table storage area is used as the search word. In this way a data item from a table can itself become a search word.

The table may also be used as the result field in operations other than the LOKUP operation. In this case the contents of the special table storage area is changed by the calculation operation. The corresponding table item in the table in core storage is also changed. This is a way in which you can modify the contents of the table by calculation operations (see Figure 133).

Example

Figures 134 through 137 show the use of the LOKUP operation. Figure 134, insert A, shows the contents of four tables: table A, table B, table C, and table D (loaded at compile time). Each table has five entries.

Figure 134, insert B, shows the extension specifications for these tables. Table A and Table B are described separately and are, therefore, entered separately. Tables C and D are related tables and are entered in alternating format on the table input records. Figure 135 shows the order in which the table input records are loaded into the system at compile time.

Figure 136 shows 15 different LOKUP operations using these four tables. The results of these operations are shown in Figure 137. Figure 137 tells if the entry searched for was found, and if so, what indicator is on to indicate the result of the search and what item was taken from a related table (when one is used).

IBM			International Business Machines Corporation			Form X21-9093 Printed in U.S.A.									
RPG CALCULATION SPECIFICATIONS			Date _____			Page 1 2			Program Identification 75 76 77 78 79 80						
Punching Instruction			Graphic			Punch			Programmer _____						
Line	Form Type	Control Level (LC, LR, SR)	Indicators			Factor 1	Operation	Factor 2	Result Field	Field Length	Decimal Positions	Resulting Indicators			Comments
			And	And	Not							Plus	Minus	Zero	
			Not	Not	Not						High	Low	Equal		
			1 > 2	1 < 2	1 = 2						Table (Factor 2) is	High	Low	Equal	
											Half Adjust (H)	58	59	60	
01	C					ARGMNT	LOKUP	TABLEA							20 SEARCH FOR =
02	C		20			TABLEA	MULT	1.5	TABLEA						IF ELEMENT IS
03	C	*													FOUND, MULTIPLY
04	C	*													BY 1.5
05	C														

Figure 133. Referencing the Table Item Found in a LOKUP Operation

	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

Ⓐ

International Business Machines Corporation

RPG EXTENSION AND LINE COUNTER SPECIFICATIONS

Punching Instruction	Graphic								
	Punch								

Page 1 2

Extension Specifications

To Filename	Table or Array Name	Number of Entries Per Record	Number of Entries Per Table or Array	Length of Entry	P = Packed/B = Binary Decimal Positions Sequence (A/D)	Table or Array Name (Alternating Format)	Length of Entry	P = Packed/B = Binary Decimal Positions Sequence (A/D)
	TABLEA	2	5	2	A			
	TABLEB	5	5	4	2			
	TABLEC	1	5	3		DTABLED	1	φ

Ⓑ

Figure 134. Table Look-Up (Tables Used)

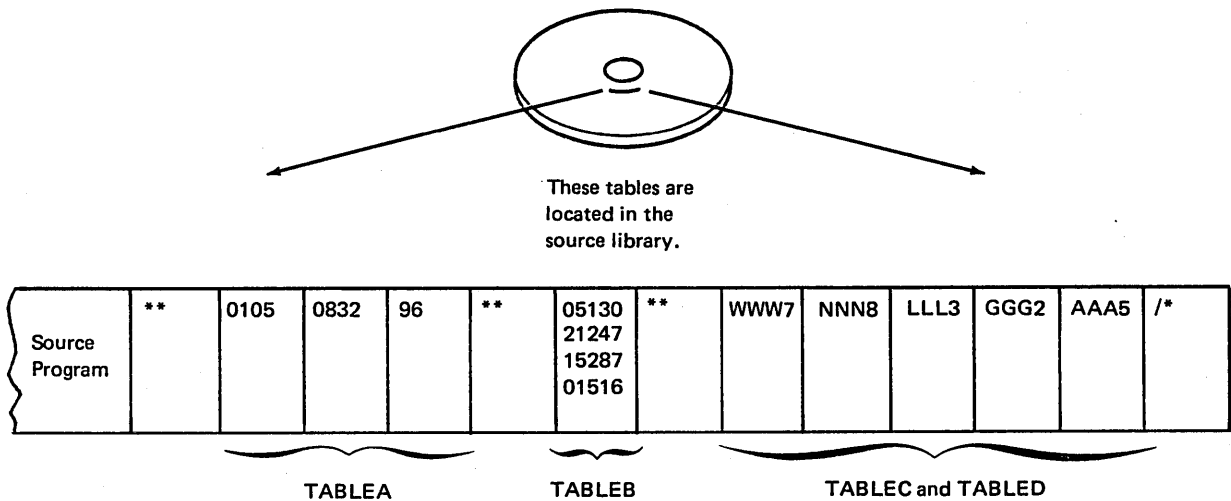


Figure 135. Order in Which Tables are Loaded

RPG CALCULATION SPECIFICATIONS

Date _____
 Program _____
 Programmer _____

Punching Instruction	Graphic						
	Punch						

Page 1 2

Program Identification 75 76 77 78 79 80

Line	Form Type	Control Level (LD, LR, SR)	Indicators			Factor 1	Operation	Factor 2	Result Field	Field Length	Decimal Positions Half Adjust (H)	Resulting Indicators			Comments
			Not	And	And							Arithmetic			
												Plus	Minus	Zero	
01	C				'08'	LOKUPTABLEA	TABLEB					01			
02	C				'08'	LOKUPTABLEA	TABLEB					02			
03	C				'08'	LOKUPTABLEA	TABLEB					03			
04	C				'08'	LOKUPTABLEA	TABLEB					04	05		
05	C				'08'	LOKUPTABLEA	TABLEB					06	07		
06	C				'00'	LOKUPTABLEA	TABLEB					08			
07	C				'97'	LOKUPTABLEA	TABLEB					09			
08	C				'09'	LOKUPTABLEA	TABLEC					10	11		
09	C				'09'	LOKUPTABLEA	TABLEC					12	13		
10	C				'LLL'	LOKUPTABLEC	TABLED					14			
11	C				'JJJ'	LOKUPTABLEC	TABLED					15			
12	C				'JJJ'	LOKUPTABLEC	TABLED					16			
13	C				'JJJ'	LOKUPTABLEC	TABLED					17			
14	C				2	LOKUPTABLED	TABLED					18			
15	C				6	LOKUPTABLED	TABLED					19			

Figure 136. Table LOKUP Operation

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	47.15
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 137. Results of LOKUP Operation

Using the LOKUP Operation with Arrays

The LOKUP specifications for arrays are the same as for tables except that if Factor 2 is an array, the result field cannot be used. In addition if the desired item is found, it is not moved to a special holding area since these holding areas are only associated with tables. Instead, the indicators will reflect only that the desired item is in the array; the programmer does not have ready access to this item.

Example: Figure 138 shows two LOKUP operations performed with an array. MANNOS, a 2100-element array of employee numbers, is read in at execution time from file ARRFIL with six 10-position elements per record; the array elements are in ascending order. Line 01 of the Calculation sheet shows a LOKUP of array MANNOS to find the element nearest to but higher in sequence than the search word 100336. If this desired element is found in the array, indicator 20 turns on and the GOTO in line 02 is performed. Notice that the result of this LOKUP indicates only whether or not the desired element exists in the array.

Line 05 of the Calculation sheet shows essentially the same LOKUP operation – indicator 20 will turn on when the first element higher in sequence than 100336 is found. Note, however, that 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 will begin at the first element of MANNOS. If the desired element is found, the number of this element (not its contents) is placed in the field INX. In this way, the actual element which 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 would be reset to 1. Refer to *Starting the Search at a Particular Array Item* under *Look-up Operations (LOKUP)* for more information on indexing an array in a LOKUP operation.

IBM
International Business Machines Corporation
Form X21-9091
Printed in U.S.A.

RPG EXTENSION AND LINE COUNTER SPECIFICATIONS

Date _____

Program _____

Programmer _____

Punching Instruction	Graphic						
Punch							

Page 1 2

Program Identification

--	--	--	--	--	--	--	--	--

Extension Specifications

Line	Form Type	Record Sequence of the Chaining File																To Filename	Table or Array Name	Number of Entries Per Record	Number of Entries Per Table or Array	Length of Entry	P = Packed/Binary Decimal Positions Sequence (A/D)	Table or Array Name (Alternating Format)	Length of Entry	P = Packed/Binary Decimal Positions Sequence (A/D)	Comments	
		Number of the Chaining Field																										
		From Filename																										
01	E	ARBFIL																	MANNOS	102100	6	0A						
02	E																											
03	E																											

IBM
International Business Machines Corporation
Form X21-9093
Printed in U.S.A.

RPG CALCULATION SPECIFICATIONS

Date _____

Program _____

Programmer _____

Punching Instruction	Graphic						
Punch							

Page 1 2

Program Identification

--	--	--	--	--	--	--	--	--

Calculation Specifications

Line	Form Type	Control Level (CO, LP, LR, SR)	Indicators						Factor 1	Operation	Factor 2	Result Field	Field Length	Decimal Positions	Resulting Indicators			Comments				
			And			Not									Plus	Minus	Zero		Compare	High 1 > 2	Low 1 < 2	Equal 1 = 2
			Not	Not	Not	Not	Not	Not														
01	C							100336	LOKUPMANNOS							20						
02	C							20	GOTO NEXT													
03	C																					
04	C								Z-ADDI	INX		40										
05	C							100336	LOKUPMANNOS, INX				20									
06	C			N					GOTO END													
07	C								MOVE MANNOS, INXSAVE			60										
08	C							NEXT	TAG													
09	C																					
10	C								Calculation Operations													
11	C																					
12	C																					
13	C																					
14	C																					

Figure 138. LOKUP with an Array

Subroutine Operations

These operation codes are only used for subroutines. See *Subroutines* under *Supplementary Information* for information on subroutines. All subroutine lines must be written in specification lines following all operations conditioned by control level indicators specified in columns 7–8. Subroutine lines are always identified by an SR in columns 7–8.

Begin Subroutine (BEGSR)

This operation code serves as the beginning point of the subroutine. Factor 1 must contain the name of the subroutine.

End Subroutine (ENDSR)

This operation code must be the last statement of the subroutine. It serves to define the end of the subroutine. Factor 1 may contain a name. This name then serves as a point to which you can branch by a GOTO statement within the subroutine. The ENDSR operation ends the subroutine and automatically causes a branch back to the next statement after the EXSR operation.

Note: Neither BEGSR operations nor ENDSR operations can have conditioning indicators (columns 7–8, 9–17).

Execute Subroutine (EXSR)

This operation causes operations in the subroutine to be performed. EXSR may appear anywhere in the program. Whenever it appears, the subroutine is executed. After operations in the subroutine are done, the operation in the line following the EXSR operation is performed.

This operation may be conditioned by any indicators, meaning the subroutine is executed only when all conditions are satisfied. Factor 2 must contain the name of the subroutine that is to be executed. This same name must appear on a BEGSR instruction. A subroutine must not call itself.

Programmed Control of Input and Output

The normal RPG II processing cycle is as follows:

1. A record is read.
2. Calculations are performed.
3. Records are written.

See *General RPG II Program Logic* under *Introduction* for a brief description of the program cycle.

The normal program cycle can be altered to allow input and output operations during calculations. The following operations provide this capability:

- Exception (EXCPT)
- Read (READ)
- Force (FORCE)
- Display (DSPLY)
- Chain (CHAIN)
- Key (KEY)
- Set (SET)

Exception (EXCPT)

This operation allows records to be written at the time calculations are being done. Use this primarily when you wish to have a variable number of similar or identical records (either detail or total) written in one program cycle. (Remember that normally only the exact number of records specified in the output-format specifications are written on a file in one program cycle.) For example, you might use EXCPT 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.

When the EXCPT operation is used, EXCPT is entered in columns 28–32, and columns 7–17 may have entries. All other columns must be blank. The line or lines which are to be written out during calculation time are indicated by an E in column 15 of the Output-Format sheet. Exception lines may not be used in a combined file.

When writing output lines at exception time, the system does not print blanks that may be at the end of the output line. To separate output written at this time from any response the operator keys during a KEY operation, specify separating characters or delimiters (such as /, -, or *) at the end of the exception output line rather than specifying blanks. In KEY operations, you can also separate the exception output from the operator response by positioning the print element using calculation specifications.

Figure 140 shows the use of the EXCPT operation to produce a variable number of records on a printer file having the same information. 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 a GOTO operation avoids 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-Format sheet is printed. The field CONSEC is used to keep track of the number of records printed. Each time an exception record is printed, 1 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 DOAGAIN. 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 have been 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.

IBM International Business Machines Corporation Form X21-9094 Printed in U.S.A.

RPG INPUT SPECIFICATIONS

Date _____

Program _____

Programmer _____

Punching Instruction	Graphic								
	Punch								

Page 1 2 Program Identification 75 76 77 78 79 80

Line	Form Type	Filename	Sequence Number (LN) Option (O)	Record Identifying Indicator	Record Identification Codes									Field Location		Field Name	Control Level (L1-L9)	Matching Fields or Chaining Fields	Field Record Relation	Field Indicators			Sterling Sign Position
					1			2			3			From	To					Plus	Minus	Zero or Blank	
					Position	Not (N) C/Z/D Character	Character	Position	Not (N) C/Z/D Character	Character	Position	Not (N) C/Z/D Character	Character	Character	Character					Decimal Positions	Control Level (L1-L9)	Plus	
01	I	INPUT	AB	01											1	2	NAME						
02	I														2	1	COUNT						

IBM International Business Machines Corporation Form X21-9093 Printed in U.S.A.

RPG CALCULATION SPECIFICATIONS

Date _____

Program _____

Programmer _____

Punching Instruction	Graphic								
	Punch								

Page 1 2 Program Identification 75 76 77 78 79 80

Line	Form Type	Control Level (LD-L9, LR, SR)	Indicators			Factor 1	Operation	Factor 2	Result Field	Field Length	Decimal Positions	Half Adjust (H)	Resulting Indicators			Comments
			And	And	And								Plus	Minus	Zero	
			Not	Not	Not								High	Low	Equal	
01	C					COUNT	COMP	0							02 COUNT = 0?	
02	C		02				GOTO	END							YES-SKIP TO END	
03	C					DOAGIN	TAG									
04	C						EXCPT									
05	C					1	ADD	CONSEC	CONSEC	3						
06	C					COUNT	COMP	CONSEC						02 EQUAL?		
07	C		N2	02			GOTO	DOAGIN						NO-DOAGIN		
08	C					CONSEC	SUB	CONSEC	CONSEC					YES-SET INDEX=0		
09	C					END	TAG									

IBM International Business Machines Corporation Form X21-9090 Printed in U.S.A.

RPG OUTPUT - FORMAT SPECIFICATIONS

Date _____

Program _____

Programmer _____

Punching Instruction	Graphic								
	Punch								

Page 1 2 Program Identification 75 76 77 78 79 80

Line	Form Type	Filename	Type (M/D/T/E)	Space Before	Skip After	Output Indicators			Field Name	Edit Codes	Constant or Edit Word	Sterling Sign
						And	And	And				
						Not	Not	Not				
01	O	OUTPUT	E						NAME	2		

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

Figure 140. EXCPT Operation (Producing a Variable Number of Identical Records)

Read (READ)

The read operation is used to call 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 28–32. Factor 2 contains the name of the file from which a record should be read immediately. An indicator may be used in columns 58–59. An indicator specified in these columns will turn on when an end of file condition is reached for the demand file (except for ledger card files), or on each READ operation after an end of file condition is reached. If columns 58–59 are left blank, a halt will occur on an end of file condition and on subsequent read operations after the end of file condition is reached. Indicators may be specified in columns 7–17.

The following columns must remain blank for a READ operation: Columns 18–27 (Factor 1), columns 43–48 (Result Field), columns 49–51 (Field Length), column 52 (Decimal Positions), column 53 (Half Adjust), and columns 54–55 and 56–57 (Resulting Indicators).

The following files may 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.
- Ledger card file specified as combined file.
- Direct files processed consecutively as input or update files.
- Data recorder files specified as input files.

Remember these points when using the READ operation:

1. All demand files, except those assigned to the KEYBOARD, must be processed by the Read operation.
2. Control levels, matching fields, and look ahead fields are not allowed with demand files.
3. Numeric sequence testing on the Input sheet is not allowed for demand files.
4. Columns 63–64 (Field Record Relation) on the Input sheet may not use the MR indicator for demand files.

Note: When a program is doing multiple reads from one or more demand files during the same RPG II cycle, the record identifying indicators assigned to the file or files will remain on throughout the cycle if the previous READ operations were executed successfully.

Force (FORCE)

FORCE statements enable you to select the file from which the next record is to be taken for processing. They apply to primary or secondary input, update, or combined files. The FORCE operation cannot be used to read from files assigned to the KEYBOARD.

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.

Display (DSPLY)

The display operation allows either or both of the following:

1. A field, table element, literal, or array element is printed on the printer during program execution without a program halt.
2. A field, table element, literal, or array element is printed on the printer and the program halts, allowing that field to be changed.

See Figure 142 for coding possibilities and results. Also see Figure 143 under CHAIN operation for an example using the display operation.

In order to use the display operation, a console file must be defined with a D entry in column 15 of the file description specifications. When the display operation is used, DSPLY must appear in columns 28–32, and the filename of the console device under Factor 2. Indicators in columns 7–17 may be specified. Half-adjust and Resulting Indicators (columns 53–59) must be left blank. No input or output specifications are required for this operation. However, the File Description sheet must have entries in columns 7–14, 15, 19, 24–27, and 40–46 (columns 71–72 are optional).

If data is to be printed but not changed, enter a field name, a table name, an array name plus an index, or a literal in Factor 1, and the filename of the console device in Factor 2. Result Field must be blank in this case. The data in Factor 1 will be printed, but not changed, and the program will continue.

IBM		International Business Machines Corporation																																								Form X21-9093 Printed in U.S.A.	
Date _____		Punching Instruction		Graphic		Page		Program Identification		75 76 77 78 79 80																																	
Program _____		Punch				1 2																																					
Programmer _____																																											
Line	Form Type	Control Level (L, O, L, R, S, R)	Indicators			Factor 1	Operation	Factor 2	Result Field	Field Length	Decimal Positions	Resulting Indicators	Comments																														
			Not	And	And						Half Adjust (H)	Arithmetic																															
												Plus Minus Zero																															
												Compare																															
												High Low Equal																															
												1 > 2 1 < 2 1 = 2																															
												Lookup																															
												Table (Factor 2) is																															
												High Low Equal																															
												54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74																															
01	C												Field A is printed.																														
02	C												Field A does not change.																														
03	C		10			FIELD A	DSPLYCONSLOUT						Program does not halt.																														
04	C																																										
05	C																																										
06	C																																										
07	C	L3	20				DSPLYCONSLOUT	FIELD A					Field A is printed.																														
08	C												Program halts.																														
09	C												Data can be entered in Field A.																														
10	C																																										
11	C																																										
12	C		40			FIELD A	DSPLYCONSLOUT	FIELD B					Field A is printed.																														
13	C												Field A does not change.																														
14	C												Field B is printed.																														
15	C												Program halts.																														
	C												Data can be entered in Field B.																														
	C																																										
	C																																										
	C																																										
	C																																										

Note: Factor 1 cannot be the name of an array.

Figure 142. Display Operation

If data is to be changed during program execution, enter the field name, table name, or array name plus index under Result Field, and the filename of the console device under Factor 2. This causes the data to be printed, and the program halts. The operator can now enter data into the field, table element, or array element via the keyboard. There are several points to remember when this is done:

1. If a field is displayed, but no keying is done (start key is depressed), the field remains unchanged.
2. If a field is displayed and keying is done, the positions of the new field that do not contain characters are blanked or zeroed out. Alphameric fields are blanked out, and numeric fields are zeroed out.
3. Alphameric fields keyed will be left justified.
4. Numeric data need not be keyed with leading zeros. It will be right justified.
5. To key a negative field, the field is keyed followed by a minus sign. The length of the field does not need to accommodate the minus sign.

Chain (CHAIN)

The chain operation causes a record to be read from a disk file during calculations. This operation allows one record to be read in when the operation code CHAIN appears in columns 28–32 of the Calculation sheet.

Indicators in columns 7–17 may be used, but Result Field, Field Length, Decimal Position, and Half-Adjust (columns 45–53) must be blank. If the chained file is conditioned in the file description specifications by an external indicator, the CHAIN statement should be conditioned by that same external indicator.

Columns 54–55 should contain an entry. If the record is not found, the indicator specified in these columns will turn on. No update is permitted to a chained update file when the specified record is not found; however, addition to a file is allowed. Columns 56–59 must always be blank for chain operations:

If an indicator is not specified in columns 54–55 and the record is not found, the program will halt. Processing can be continued by pressing the start key. If LR processing has already been initiated, the bypass and begin new cycle option will not be allowed. If the controlled cancel option is taken, files will be closed but the rest of LR processing will not occur.

When chaining to a file with packed record keys, the entry in Factor 1 of the CHAIN operation must have a packed length which 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 following chart shows the packed equivalents for unpacked fields up to 15 bytes long:

Unpacked 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

The chain operation is used for two purposes:

1. Random processing of an indexed, sequential, or direct file.
2. Loading a direct file.

Note: When chaining to one or more files during the same RPG II cycle, record identifying indicators assigned to the chained file or files will remain on throughout the cycle if the previous chain operations were executed successfully. When chaining to the same file more than once during an RPG II cycle, only the last record processed will be updated during output time unless an exception output is associated with each chain operation.

Random Processing: In order to read a record from a sequential or direct file, the record must be identified by relative record number. To read a record from an indexed file, a record key is used for identification. The relative record number or key can be contained in a field specified for that purpose.

The chain operation requires the operation code CHAIN in columns 28–32 of the Calculation sheet. Factor 1 entries must be a relative record number or key, or the name of a numeric field that contain a relative record number or key. Factor 2 must contain the name of the file from which the record will be read. This file is called the file that is chained to or the chained file. It is this file that must be defined with a C entry in column 16 of the File Description sheet (see *Example*).

Direct File Load: A direct file load is defined by specifying the disk file to be loaded as a chained output file on the File Description sheet. In the calculation specifications, Factor 1 must contain a relative record number, columns 28–32 must contain the operation code CHAIN, and Factor 2 must contain the name of the direct disk file to be loaded. The relative record number of the input record defines the record position for each record in the direct disk file. The relative number can be all or part of a field in the input records. Such fields are used for record identification of the input records, as well as for the disk records after the disk file is loaded.

When a direct file is loaded, you must define the record length and number of records in your file. The system then clears the disk space required for the file 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. The blanks at that record position are read in, 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 direct file is loaded, the space reserved for that record in the disk file remains blank (until the proper record is read in later).

Once the direct file is loaded, records are inserted or changed in the file by defining the direct file as an update file processed consecutively or by the chain operation. (Remember that any file defined as a chained output file will be cleared entirely to blanks before any records are processed.)

You may have to allow for *synonyms* when you load a direct file. Synonyms are two or more records with the same relative record number. If you will have synonyms, you can load the file in one of two ways, using multiple passes:

1. Define the disk file as a direct file and clear it to blanks in your first job (by defining it as a chained output file). Once the file has been cleared, one or more subsequent jobs can be run using the update function to read record locations and check for synonyms while loading the file.
2. Load the direct file with records without synonyms, then run another job using the update function to identify synonyms and load them into the file.

Note: The insertion of records in direct disk files is very different from record addition to sequential or indexed files. For sequential disk files, the new record is added in 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 inserted in 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.

Example: Figure 143 shows the coding necessary to chain to and update an indexed file, MASTINV. The CARDIN file consists of cards sorted by item number, each card representing some quantity ordered. Item number is used as a control field. When all the quantities for one item number are added, a control break will occur. At this point in calculations, the master record for that item number must be found and updated. ITEMNO is a field containing the item number of the cards presently being worked on. The chain operation uses ITEMNO to find the master record for that item number. If it is not found, a display operation prints out the item number of the cards. Note that indicator (20) turns on when the records are not found.

specifying tab stop positions). Tab stop positions must be specified in ascending sequence (from the lowest print position to the highest print position). When the operator begins a manual keying operation, he must first key a data field. If other fields with tab sets are to be keyed in this manual operation, the operator must press the TAB key before entering another field. When the TAB key is pressed, the print element moves to the right until it reaches the first tab stop position specified for this KEY operation. The operator, then, keys the next field, and presses the TAB key again if still other fields are to be keyed. If fields are to be printed on more than one line, the operator must press the carriage return key to begin each new line. The operator continues to key fields and press the TAB key alternately until all fields have been entered. Once a manual keying operation is begun, the system operates in manual mode until the operator presses the start key.

Note: When keying numeric fields in display or manual modes you must be sure that there is enough space on the print line to print the field you specified and the appropriate sign (minus sign for a negative field, blank for a positive field). To do this, position the print element at the desired position using the SET operation code.

Bypassing a KEY Operation: When the KEY operation causes a pause, the operator has the option of going to the next calculation operation without keying any data. If this is necessary, the operator simply presses the start key when the program is at the KEY operation. This causes any data in the KEY operation result field to be changed to zero or blank. After each KEY operation (regardless of whether data is entered or not), the operator must press the start key before the next calculation operation can be done. (See *Operation Codes, Set* for the special situation which allows the SET and KEY operations to be combined with only one depression of the start key.)

Note: The keying operations described in this section apply to entering data from the device KEYBOARD, not to entering data from the device CONSOLE. In order to perform the operations described here, you must first specify a keyboard file on the File Description sheet.

Calculation Specifications Used for KEY Operations

The following specifications are used to describe all KEY operations. The calculation specifications required for a KEY operation vary depending upon whether you want keying done in store only, display, or manual mode (see Figure 144 for a summary of these specifications).

Columns 7-8: Enter any valid conditioning indicator. Leave these columns blank if the KEY 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. Leave these columns blank if the KEY operation is to be performed on every program cycle. Command Key indicators may be used here if they have been specified in a SET or SETOF operation.

Columns 18-27: Leave these columns blank.

Columns 28-30: Enter the operation code KEY.

Columns 31-32: Enter the numbers of the field lights you want turned on to prompt the operator to perform a particular keying operation. Field lights must be used in all KEY operations to cause a pause during which the operator may enter data. You may specify one or two field lights in a KEY operation. Valid entries are blank, 0, or 1-8. Only entries 1-8 turn on field lights. If only one field light is specified, it can be entered in either column 31 or column 32. You may enter a 0 in the column which has no entry, or you may leave it blank. If two field lights are specified, the same field light cannot be specified in both of these columns. The same combination of field lights should not be used more than once in the program.

Valid field light entries:		
Column 31	Column 32	Meaning
6	0	Field light 6 only
6	Blank	Field light 6 only
Blank	6	Field light 6 only
0	6	Field light 6 only
5	6	Field lights 5 and 6
Invalid field light entries:		
5	5	

Figure 144. Summary of Calculation Specifications for KEY Operations

KEY Operations	Control Level (L0-L9, LR, SR)	Indicators															Factor 1	Operation	Factor 2	Result Field	Field Length	Decimal Positions Half Adjust (H)	Resulting Indicators			Comments																																									
		And					And					Arithmetic																																																							
		Not					Not					Plus	Minus	Zero																																																					
		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
Store Only	<ul style="list-style-type: none"> Blank L0-L9, LR, SR 																	KEY		Name of field to be keyed	<ul style="list-style-type: none"> 1-128 1-15 	0-9	<ul style="list-style-type: none"> Blank 01-99, H1-H9, LR, L1-L9, OA-OG, OV 	Any meaningful comments																																											
Display	<ul style="list-style-type: none"> Blank L0-L9, LR, SR 																	KEY	Name of the printer file	Name of field to be keyed	<ul style="list-style-type: none"> 1-128 1-15 	0-9	<ul style="list-style-type: none"> Blank 01-99, H1-H9, LR, L1-L9, OA-OG, OV 	Any meaningful comments																																											
Manual	<ul style="list-style-type: none"> Blank L0-L9, LR, SR 																	KEY	<ul style="list-style-type: none"> Name of the printer file *BOTH 						Any meaningful comments																																										

● The shaded columns must be left blank.

Columns 33–42: Leave these columns blank for fields keyed in store only mode. For display or manual keying operations, enter the name of the printer file on which the fields will be printed. For manual operations, you may enter *BOTH in these columns. This entry causes the system to automatically perform spacing operations on both print units each time the operator presses the carriage return key. The operator is not required to print fields on both TRACTR1 and TRACTR2 when *BOTH is entered in these columns. Printing may be done on either or both print units.

Columns 43–48: Leave these columns blank for fields keyed in manual mode. For store only or display keying operations, enter the name of the field to be keyed. A field name can be from one to six characters long and must begin in column 43. The first character must be alphabetic; the remaining characters can be any combination of alphabetic and numeric characters (special characters are not allowed). Blanks must not appear between characters in the field name.

Columns 49–51: Leave these columns blank for fields keyed in manual mode. For keying operations in store only or display mode, enter the length of the keyed field if that field has not been defined previously. The maximum length for a numeric field is 15; the maximum length for an alphameric field is 128.

Column 52: Leave this column blank for all fields keyed in manual mode and for alphameric fields keyed in store only or display mode. For numeric fields keyed in store only or display mode, enter the number of decimal positions (0–9) in the keyed field if that field has not been defined previously.

Column 53: Leave this column blank.

Columns 54–59: Leave these columns blank for all fields keyed in manual mode. Use these columns to test the condition of numeric fields keyed in store only or display mode. You may test for three conditions: plus (columns 54–55), minus (columns 56–57), zero or blank (columns 58–59).

Columns 54–55 (Plus): Any valid indicator entered here is turned on if the numeric field named in columns 43–48 is greater than zero.

Columns 56–57 (Minus): Any valid indicator entered here is turned on if the numeric field named in columns 43–48 is less than zero.

Columns 58–59 (Zero or Blank): Any valid indicator entered here is turned on if a numeric field named in columns 43–48 is all zeros or if an alphameric field is all blanks. Leave columns 54–57 blank for alphameric fields keyed in store only or display mode.

Columns 60–74: Enter any meaningful comments you wish.

Figure 145 shows the different keying operations that may be performed in store only, display, or manual mode.

The KEY operation is normally used in conjunction with the SET operation (see *Operation Codes, SET* for further information on this topic).

Using KEY and SET Operations in Subroutines

Sometimes it is necessary to write a program performing the same KEY operations at several different points in the program. Instead of writing these KEY operations and related SET operations every time they are needed, it is easier and less time consuming if they are written just once in a subroutine. Then, you can refer to the subroutine each time it is needed (see *Subroutines* for information on specifying and using subroutines).

Set (SET)

The SET operation is used to control certain printer and ledger card output operations during calculations. This operation code allows you to specify one, or several combinations, of the following functions:

1. Spacing on the printer forms.
2. Skipping to a particular line on the printer forms.
3. Ejecting a ledger card.
4. Positioning the print element.
5. Setting tab stop positions.
6. Allowing command keys to be pressed.



RPG CALCULATION SPECIFICATIONS

Date _____
 Program _____
 Programmer _____

Punching Instruction	Graphic						
	Punch						

Page

1	2
---	---

Program Identification

75	76	77	78	79	80
----	----	----	----	----	----

Line	Form Type Control Level (LD, LR, SR)	Indicators						Factor 1	Operation	Factor 2	Result Field	Field Length	Decimal Positions Half Adjust (H)	Resulting Indicators Arithmetic Plus Minus Zero Compare High Low Equal 1 > 2 1 < 2 1 = 2 Lookup Table (Factor 2) is High Low Equal 54 55 56 57 58 59	Comments
		Not	And	Not	And	Not	And								
3								28	STORE	ONLY	KEYING	OPERATIONS			*****
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															

Figure 145. Possible Store Only, Display, and Manual Keying Operations

You may combine as many of the functions of the SET operation as possible, except for setting tab stop positions. Because SPACE, SKIP, and EJECT are entered in the same columns, only one of these functions can be performed in a SET operation.

Spacing and Skipping on the Printer Forms

These functions of the SET operation cause immediate spacing or skipping to a specific line on the printer forms. The space and skip functions apply to any fields keyed in display or manual mode. Spacing may be 1–3 lines; skipping may be to line 01–99, A0–A9 (100–109), B0–B2 (110–112). You would normally specify spacing and skipping on the Calculation sheet to establish the starting position of the next output line.

In display keying operations, fields may be printed again later in the program using output specifications. If this is done, spacing and skipping for this output operation is specified as usual on the Output-Format sheet.

Ledger Card Eject

The eject function of the SET operation causes an immediate ledger card eject. Ledger card eject is specified on the Calculation sheet only as a means of handling errors that you would normally anticipate when processing ledger files. By specifying eject on the Calculation sheet, you allow the program to go immediately to the next record after errors such as a record-not-found condition in a chain operation or unsatisfactory results in a compare operation. For instance, if the operator keys the appropriate control number field and inserts a ledger card, the normal procedure is to compare the keyed control number with the control number from the ledger. If the operator has inserted the wrong ledger (a not equal condition is found), an indicator is turned on to signal this error. Then, conditioned by the appropriate indicator, the ledger card is ejected immediately. At this point, the program can read another ledger, or it can perform whatever calculations follow.

It is possible to specify ledger card eject as a part of output specifications (see *Column 16, Ledger Card Eject* under *Output-Format Specifications* and *Ledger File Processing* for a description of this output operation.

Positioning the Print Element

This function of the SET operation causes the print element to move immediately to the specified position. This positioning function applies to any fields keyed in display or manual mode. Positioning the print element is normally specified on the Calculation sheet to establish the starting print position for these fields.

In display keying operations, fields may be printed again later in the program according to output specifications.

Setting Tab Stop Positions

This function of the SET operation allows you to set tab stop positions for data keyed in manual mode. Tab stop positions can only be specified immediately before a manual key operation. These tab stop positions must be specified in ascending sequence from the lowest print position to the highest print position (01–220). When the operator presses the TAB key, the print element moves to the right until it reaches the first tab stop position specified for that manual keying operation.

You may specify as many tab stop positions as necessary; however, only one tab stop position can be entered on a specification line. Tab stop positions remain in effect until the operator presses the start key.

Allowing Command Keys to be Pressed

This function of 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 is turned on. These command key indicators may 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 using the SETOF operation code. When the program is at a particular specification line, you may give the operator the option of pressing 1–3 command keys.

Calculation Specifications Used for SET Operations

The following specifications are used to describe all SET operations. The calculation specifications required for a SET operation vary depending upon which function, or combination of functions, you wish to perform (see Figure 146 for a summary of these specifications).

Figure 146. Summary of Calculation Specifications for SET Operations

Line	Form Type	Control Level (L, O, L, R, SR)	Indicators						Factor 1	Operation	Factor 2	Result Field	Field Length	Decimal Positions Half/Adjust (H)	Resulting Indicators		
			Not	And	And	Not	And	Not							Plus	Minus	Zero
3																	
4									SET			SPACE	n				
5									SET			SPACE	n				
6									SET			SPACE	n				
7									SET			SPACE	n				
8									SET			SKIP	nn				
9									SET			SKIP	nn				
10									SET			SKIP	nn				
11									SET			SKIP	nn				
12									SET			SKIP	nn				
13									SET			EJECT					
14									SET			EJECT					
15									SET								
16									SET								
17									SET								
18									SET								
19									SET								
20									SET								
21									SET								
22									SET								
23									SET								
24									SET								
25									SET								
26									SET								
27									SET								
28									SET								
29									SET								
30									SET								
31									SET								
32									SET								
33									SET								
34									SET								
35									SET								
36									SET								
37									SET								
38									SET								
39									SET								
40									SET								
41									SET								
42									SET								
43									SET								
44									SET								
45									SET								
46									SET								
47									SET								
48									SET								
49									SET								
50									SET								
51									SET								
52									SET								
53									SET								
54									SET								
55									SET								
56									SET								
57									SET								
58									SET								
59									SET								

These indicators may be used with any of the SET operations.

Unshaded columns may contain blank, 0, or 1-8.

Unshaded columns must contain the printer filename.

Unshaded columns may contain 1-220.

Unshaded columns may contain KA-KN, KP, KQ.

Note: The shaded columns must be blank for that function of the SET operation. The unshaded columns require one of the entries listed below the particular set of columns.

Columns 7–8: Enter any valid conditioning indicator. 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 except setting tab stop positions. Leave these columns blank if the SET operation is to be performed on every program cycle.

Columns 18–27: Leave these columns blank.

Columns 28–30: Enter the operation code SET.

Columns 31–32: These columns must contain an entry for a SET operation in which command key indicators are specified in columns 54–59. Field lights cause a pause during which the operator may press the command keys entered in columns 54–59. You may specify one or two field lights in a SET operation. Valid entries are blank, 0, or 1–8. Only entries 1–8 turn on field lights. If only one field light is specified, it can be entered in either column 31 or column 32. You may enter a 0 in the column which has no entry, or you may leave it blank. If two field lights are specified, the same field light cannot be used in both of these columns. The same combination of field lights should not be used more than once in the program.

Valid field light entries:		
Column 31	Column 32	Meaning
6	0	Field light 6 only
6	Blank	Field light 6 only
Blank	6	Field light 6 only
0	6	Field light 6 only
5	6	Field lights 5 and 6
Invalid field light entries:		
5	5	

Columns 33–42: Leave these columns blank when the only function performed by the SET operation is (1) ejecting a ledger card, (2) setting tab stop positions, or (3) allowing command keys to be pressed. For spacing, skipping, or positioning the print element, enter the name of the printer file on which this function will be done.

Columns 43–48: Leave these columns blank when the only function performed by the SET operation is (1) positioning the print element, (2) setting tab stop positions, or (3) allowing command keys to be pressed.

- Enter SPACEn for immediate spacing on the printer forms. The letter n may be 1–3.
- Enter SKIPnn for immediate skipping on the printer forms. The letters nn may be 01–99, A0–A9, B0–B2.
- Enter EJECT for an immediate ledger card ejection.

Columns 49–51: Use these columns to (1) position the print element or (2) set tab stop positions.

- When you wish to position the print element, enter the starting position of the next field. Valid entries are 1–220 for the 220-position printer without ledger, 1–214 for the 220-position printer with ledger, or 1–132 for the 132-position printer.
- When you wish to set tab stop positions, enter the position to which you want the print element to move when the operator presses the TAB key. You can specify only one tab stop position on a specification line. Valid entries are 1–220 for the 220-position printer without ledger, 1–214 for the 220-position printer with ledger, 1–132 for the 132-position printer. Columns 31–32 must be blank when tab stop positions are specified. Tab stop positions can only be specified immediately before a manual key operation. See *Operation Codes, Key* for a description of manual keying operations.

Column 52: Leave this column blank.

Column 53: Leave this column blank.

Columns 54–59: Enter the command keys (KA–KN, KP, KQ) that the operator is allowed to press when the program has paused at this specification line. You may specify 1–3 command keys; if only one or two command keys are specified, they may be entered in any of the three sets of columns. When the operator presses a command key entered in these columns, that command key indicator remains on until it is used again in A SET operation or until it is turned off using the SETOF operation code. A halt occurs if the operator presses a command key other than those specified in columns 54–59 of a SET operation. Field lights must be specified in columns 31–32 of a SET operation when command key indicators are entered in columns 54–59.

Columns 60-74: Enter any meaningful comments you wish.

Figure 147 shows the coding needed to perform the SET operation. Figure 148 shows the possible combinations of these functions.

IBM		International Business Machines Corporation																														Form X21-9093 Printed in U.S.A.			
Date _____		RPG CALCULATION SPECIFICATIONS																														Page 1 2		75 76 77 78 79 80	
Program _____		Punching Instruction																														Graphic		Program Identification	
Programmer _____		Punch																																	
Line	Form Type	Control Level (LD, LB, SR)	Indicators				Factor 1	Operation	Factor 2	Result Field	Field Length	Decimal Positions	Half Adjust (H)	Resulting Indicators		Comments																			
			Not	And	And									Arithmetic																					
			Not	Not	Not									Plus	Minus	Zero																			
														Compare																					
														High	Low	Equal																			
														1 > 2	1 < 2	1 = 2																			
														Lookup																					
														Table (Factor 2) is																					
														High	Low	Equal																			
														54	55	56	57																		
														58	59	60	61																		
														62	63	64	65																		
														66	67	68	69																		
														70	71	72	73																		
														74																					
3	c	*																																	
4	c																																		
5	c																																		
6	c																																		
7	c																																		
8	c																																		
9	c																																		
10	c																																		
11	c																																		
12	c																																		
13	c																																		
14	c																																		
15	c																																		
16	c																																		
17	c																																		
18	c																																		
19	c																																		
20	c																																		
21	c																																		
22	c																																		
23	c																																		
24	c																																		
25	c																																		
26	c																																		
27	c																																		
28	c																																		
29	c																																		
30	c																																		
31	c																																		
32	c																																		
33	c																																		
34	c																																		
35	c																																		
36	c																																		
37	c																																		
38	c																																		
39	c																																		
40	c																																		
41	c																																		
42	c																																		
43	c																																		
44	c																																		
45	c																																		
46	c																																		
47	c																																		
48	c																																		
49	c																																		
50	c																																		
51	c																																		
52	c																																		
53	c																																		
54	c																																		
55	c																																		
56	c																																		
57	c																																		
58	c																																		
59	c																																		
60	c																																		
61	c																																		
62	c																																		
63	c																																		
64	c																																		
65	c																																		
66	c																																		
67	c																																		
68	c																																		
69	c																																		
70	c																																		
71	c																																		
72	c																																		
73	c																																		
74	c																																		

Figure 147. Different Functions Performed by the Operation

RPG CALCULATION SPECIFICATIONS

Date _____
Program _____
Programmer _____

Punching Instruction	Graphic								
	Punch								

Page 1 2

Program Identification 75 76 77 78 79 80

Line	Form Type	Control Level (LD, LR, SR)	Indicators			Factor 1	Operation	Factor 2	Result Field	Field Length	Decimal Positions	Half Adjust (H)	Resulting Indicators		Comments
			Not	And	And								Plus	Minus	
C	*														THE FOLLOWING COMBINATIONS OF FUNCTIONS ARE VALID IN SET OPERATIONS.
C	*														SPACING AND POSITIONING THE PRINT ELEMENT
C															SET PRINTER SPACE3 52
C	*														SPACING AND ALLOWING COMMAND KEYS TO BE PRESSED
C															SET P1 PRINTER SPACE2 KDKE
C	*														SPACING, POSITIONING THE PRINT ELEMENT, AND ALLOWING COMMAND KEYS TO BE PRESSED
C															SET P2 PRINTER SPACE1 52 KFKG
C	*														SKIPPING AND POSITIONING THE PRINT ELEMENT
C															SET PRINTER SKIP P1 6 P
C	*														SKIPPING, AND ALLOWING COMMAND KEYS TO BE PRESSED
C															SET P3 PRINTER SKIP P2 KHKI
C	*														SKIPPING, POSITIONING THE PRINT ELEMENT, AND ALLOWING COMMAND KEYS TO BE PRESSED
C															SET P4 PRINTER SKIP P3 6 P KA
C	*														EJECTING A LEDGER AND POSITIONING THE PRINT ELEMENT
C															SET PRINTER EJECT 75
C	*														EJECTING A LEDGER AND ALLOWING COMMAND KEYS TO BE PRESSED
C															SET 5 EJECT KB
C	*														EJECTING A LEDGER, POSITIONING THE PRINT ELEMENT, AND ALLOWING COMMAND KEYS TO BE PRESSED
C															SET P5 PRINTER EJECT 75 KCKE
C	*														POSITIONING THE PRINT ELEMENT AND ALLOWING COMMAND KEYS
C															SET P6 PRINTER 8 P KD KQ
C	*														* * * * *

Figure 148. Possible Combinations of Functions Performed by the SET Operation

Special Combinations of the SET and KEY Operations

Normally, the operator must press the start 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 54–59 of a SET operation) and key a field (specified in a KEY operation) with only one depression of the start key. 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 7–17). Indicators for both operations must be specified in the same order.
- The SET and KEY operations turn on the same field lights.

If both operations are to be performed, the operator must press the command keys before keying the field (Figure 149).

Line		Form Type Control Level (LD, LR, SR)	Indicators			Factor 1	Operation	Factor 2	Result Field	Field Length	Decimal Positions Half Adjust (H)	Resulting Indicators			Comments
3	4		7	8	9							Plus	Minus	Zero	
01	C														
02	C		*												
03	C		*												
04	C		*												
05	C		*												
06	C		*												
07	C														
08	C						SET68						KAKBKC		
09	C						KEY68		PRICE	50					
10	C														
11	C														
12	C														
13	C														
14	C														
15	C														

Figure 149. Special Combination of SET and KEY Operations

Using KEY and SET Operations in Subroutines

Sometimes it is necessary to write a program performing the same KEY operations at several different points in the program. Instead of writing these KEY operations and related SET operations every time they are needed, it is easier and less time consuming if they are written once in a subroutine. Then, you can refer to the subroutine each time it is needed. See *Subroutines* for information on specifying and using subroutines.

Debug Operation

The debug operation is an RPG II function that you may use to help you find errors in a program which is not working properly. This code causes either one or two records to be printed. They contain information which is helpful for finding programming errors.

Debug (DEBUG)

The DEBUG operation code may be placed at any point or at several points in the calculation operations. Whenever it is encountered, either one or two records are printed depending upon the specifications entered. One record contains a list of all indicators which are on at the time the DEBUG code was encountered. The other shows the contents of any one field.

Note: The decimal point is not printed.

Specifications

Factor 1 is optional; It may contain a literal of 1-8 characters which will identify the particular DEBUG operation. The name entered here is printed on record 1. Factor 2 must contain the name of the output file on which the records are written. The same output filename must appear in Factor 2 for all DEBUG statements in a program. The result field may be a field or array whose contents you wish to appear on record 2. Any valid indicator may be used in columns 7-17. Columns 49-59 must be blank.

The operation code produces results only if the proper entry (1 in column 15) has been made in the control card specifications. If the control card specifications entry has

not been made, the operation code DEBUG and the indicators conditioning it are treated as a comment. See *Column 15* under *Control Card Specifications* for more information.

Records Printed for Debug

Record 1 is required. It is printed in the following format:

<i>Print Positions</i>	<i>Information</i>
1-8	DEBUG-
9-16	Literal entered in Factor 1 (optional)
17	Blank
18-32	The words INDICATORS ON-
33-any position (depending on length of field)	The names of all indicators which are on, each separated by a blank; the word NONE, if no indicators are on. More than one record may be needed.

Record 2 is optional and is printed only when there is a Result Field. The record is printed in the following format:

<i>Print Positions</i>	<i>Information</i>
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 record may be needed to contain the array.

OVERFLOW INDICATORS

Overflow occurs when a record is printed on the overflow line or when a space or skip instruction causes the overflow line to move past the print element. When overflow occurs, the overflow indicator you specified on the File Description sheet is turned on. This indicator remains on for the rest of the program cycle; it is turned off after all lines conditioned by an overflow indicator have been printed. See *Ledger File Processing, Ledger Card Overflow* for detailed information and examples of RPG II overflow logic for ledger files.

According to RPG II logic, there are three times in the program cycle when an overflow indicator can be turned on: (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 RPG II program checks to see if the overflow indicator is on. This is right after all total records have been printed.

If overflow occurs when a total record is printed, the appropriate overflow indicator turns on, and the remainder of the total lines are printed. A test is then made to determine whether or not the overflow indicator is on. If it is, all total, detail, and heading lines conditioned by the overflow indicator are printed.

If overflow occurs when a detail record is printed, the appropriate overflow indicator turns on, and the remainder of the detail lines are printed. The next record is read. If this record causes a control break, total calculations and total output operations are performed, followed by overflow output.

When the printer has reached the end of a printed page, the RPG II language allows you to do one of four things:

1. Advance to the top of the next page and continue printing.
2. Ignore the fact that the end of the page has been reached and continue printing.
3. Print special lines at the bottom of the page and at the top of the new page.
4. Alter the basic RPG II overflow logic by fetching the overflow routine at detail time, total time, or during calculations using exception output.

Automatic Page Formatting

In the first case, RPG II will automatically advance to the next page and continue printing once overflow has occurred. Line counter specifications may be entered, but they are optional.

Do not specify an overflow indicator on either the File Description sheet or the Output-Format sheet.

When RPG II handles overflow automatically, printing is set to begin on line 06 once the operator has positioned the first page. This means that detail lines begin on line 06 for all pages except the first. When RPG II handles overflow, printing ends on line 60 or on the defined overflow line.

Output-Format Sheet:

- Use the overflow indicator assigned on the File Description sheet to condition those lines you want printed on each page. Usually these lines are total lines which must be printed at the bottom of each page or heading lines which must be printed at the top of each new page (Figure 152).
- The normal sequence is:
 1. Print page totals
 2. Skip to new page
 3. Print heading lines

In the case where you have specified an overflow indicator and are using it to condition output lines, the following steps occur when the overflow line (end of page) has been sensed:

1. Detail lines are printed (if that part of the program cycle has not already been 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 specification on a heading line or total line.
5. Heading and detail lines are printed if conditioned by overflow indicators.

IBM			International B									
			RPG OUTPUT - F									
Date _____			<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="border: 1px solid black; padding: 2px;">Punching Instruction</td> <td style="border: 1px solid black; padding: 2px;">Graphic Punch</td> </tr> </table>			Punching Instruction	Graphic Punch					
Punching Instruction	Graphic Punch											
Program _____												
Programmer _____												
Line	Form Type	Filename	Type (M/D/T/E)	Space		Skip		Output Indicators			Field Name	
				Before	After	Before	After	Not	And	And		Not
3												
0 1	O	PRINT	H			3	φ	6	OA			
0 2	O											
0 3	O											
0 4	O											

Figure 152. Page Formatting (Output Specifications)

Using Overflow Indicators and Control Level Indicators

Often you want each page to contain information from only one control group. (Information from one group may require several printed pages, however.) You might also wish each page to have headings identifying the type of information on the page. For these cases you need to use both the control level indicators and the overflow indicators. Together they condition when both headings and group information are to be printed.

A new page should advance either when the overflow line has been reached (the overflow indicator you assigned is on) or when there is a change in a control field (control level indicator is on). You must specify that each indicator causes a new page to be advanced by specifying a skip to the first printing line on a page. If the control level has changed and the overflow condition has occurred at the same time, it is possible to duplicate an output line (one called for by the overflow indicator, the other by the control level indicator). A blank page can appear in your report as a result.

Figure 153 shows 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 01) 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. This is true because the first record always causes a control break (L2 turns on), if control fields are specified on the record.

Figure 154 shows 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.

Fetching the Overflow Routine

When the overflow line is reached during detail time, the same sequence of events always takes place. These were described previously. Briefly, remaining detail lines, total lines, and overflow total lines (total lines conditioned by the overflow indicator) are printed on the page even after overflow has occurred. Therefore, you must leave enough space between the overflow line and the actual end of page for all these lines to print.

However, you can run into problems when you do this. If a different number of detail or total lines can be printed depending upon when overflow occurs, you may find that you have allowed too little or too much space below the overflow line. Assume that your job involves printing two detail records or four total records in a program cycle. Sometimes, both detail and total records may be printed in the same program cycle. In this case, the overflow indicator can be turned on (1) when any one of the detail records is printed or (2) when any one of the total records is printed.

IBM International Business Machines Corporation Form X21-9080 Printed in U.S.A.

RPG OUTPUT - FORMAT SPECIFICATIONS

Date _____ Program _____ Programmer _____

Punching Instruction: Graphic, Punch

Page 1 2 Program Identification 75 76 77 78 79 80

Line	Form Type	Filename	Type (H/D/T/E)	Space		Skip		Output Indicators			Field Name	Edit Codes	Blank After (B)	End Position in Output Record	P = Percent/B = Binary	Edit Codes						Sterling Sign Position
				Before	After	Before	After	Not	Not	Not						Commas	Zero Balances to Print	No Sign	CR	-	X	
01	O	PRINT	H		306			0ANL2									1	A	J	Y		
02	O		OR					L2									2	B	K			
03	O																3	C	L			
04	O																4	D	M			
05	O																					
08	O																					
07	O																					

Constant or Edit Word

8	'DATE'
12	'ACCOUNT'
28	'NAME'
46	'BALANCE'

Figure 153. Printing Headings on Every Page

IBM International Business Machines Corporation Form X21-9080 Printed in U.S.A.

RPG OUTPUT - FORMAT SPECIFICATIONS

Date _____ Program _____ Programmer _____

Punching Instruction		Graphic				
		Punch				

Page 1 2 Program Identification

Line	Form Type	Filename	Type (H/D/T/E)	Space	Skip	Output Indicators						Field Name	Edit Codes	End Position in Output Record	Starting Sign Position
						Before	After	Not	And	And	And				
01	O	PRINT	D	306			O	A	N	L	Z				
02	O		O						L	Z					
03	O														
04	O														
05	O														

Edit Codes

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

Constant or Edit Word

Figure 154. Printing Fields on Every Page

If overflow occurs when the first detail record is printed, the second detail line and all total lines (provided a control break has occurred) will print before forms advance. Printing will be done on the perforation and run over onto the next page unless you specified the overflow line high enough on the page.

On the other hand, suppose overflow was caused by the third total record instead of the first detail record. Only one more total line would print before forms advanced. In this instance, you may find that you have allowed so much room between the overflow line and the end of the page that only half a page is actually used.

To prevent printing over the perforation and at the same time use as much of each page as possible, you can fetch the overflow routine. Fetch overflow specifications allow you to alter the basic RPG II overflow logic. You can cause forms to advance when total or detail records are printed instead of waiting for the usual time. To fetch the overflow routine, enter an F in column 16 of the Output-Format sheet for any detail or total record.

The fetch overflow specification causes the computer to check to see if the overflow indicator is on before it prints total or detail records. The test is made each time an F is entered in column 16 of the Output-Format sheet. If the overflow indicator is on when the test is made, all operations conditioned by the overflow indicator are performed immediately.

When the overflow routine is fetched (F in column 16 of the Output-Format sheet), 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 06 has been specified in a line conditioned on an overflow indicator.
3. Heading lines and detail lines conditioned by the overflow indicator are printed.
4. The line that fetched overflow is printed.
5. Any detail and total lines left to be printed for that program cycle are printed.

You should fetch the overflow routine (F in column 16) only when you feel that (1) printing a particular line would cause overflow and (2) if it did, there would not be enough space left on the page to print the remaining detail and total output lines and heading lines conditioned by the overflow indicator.

When more than one printer file is used, fetch applies only to the overflow lines associated with the file in which the record that specified fetch is defined.

Note: Fetch overflow cannot be specified when an overflow indicator is specified in columns 23-31 on the same specification line. If this condition does occur, fetch overflow is not performed.

To determine when to fetch the overflow routine (F in column 16 of the Output-Format sheet), you must study all possible overflow situations. By counting lines and spaces, you can calculate what will happen if overflow occurs on each detail and total line.

Figure 155 shows the use of the fetch specification. Total lines 03 and 09 can fetch the overflow routine. They do this, however, only if the overflow line has been sensed prior to the printing of one of these lines. When it is time for the operation in line 03 to be performed, a test is made to see if the internal overflow indicator is on. If it is, the overflow routine is fetched causing these operations to be performed:

1. All total lines conditioned by the overflow indicator are printed (if a control break has occurred).
2. Forms are advanced provided a skip to 06 has been specified in a line conditioned by the overflow indicator.
3. Heading lines conditioned by the overflow indicator are printed (line 01).
4. An internal overflow indicator is turned off, so that overflow processing will not be repeated until another page overflows.

Note: The external overflow indicator is not turned off at this time.

5. The record specified on line 03 is printed.

Another test is made to see if the internal overflow indicator is on because of the fetch specification (F in column 16) in line 09. If line 03 caused the form to advance, the internal overflow indicator would not be on at this time. The total record in line 09 would print normally.

However, if the records in lines 03, 05, or 07 were printed on the overflow line, the internal overflow indicator would be on. The specification in line 09 would then fetch the overflow routine.

Overflow Printing with EXCPT Operation Code

Overflow indicators cannot condition an exception line, but can condition fields within an exception line. The use of the EXCPT operation code with exception lines (E in column 15 of the Output-Format sheet) causes only exception lines to be printed during calculation time. If the overflow line is sensed when an exception line is printed, the overflow indicator turns on as usual, but overflow processing does not occur until another exception line conditioned to print (and with fetch overflow specified) is encountered.

The actual overflow output lines (totals and/or headings) must be coded as H, D, or T types. The use of fetch overflow will cause the H, D, or T overflow output lines to be printed if the overflow indicator is on. The overflow output lines are printed prior to the printing of the line on which fetch overflow is specified. The user may also force overflow by issuing a SETON of the appropriate overflow indicator prior to the EXCPT operation code, provided fetch overflow has been specified.

Assigning Overflow Indicators

When using the overflow indicator to condition overflow printing, consider the following:

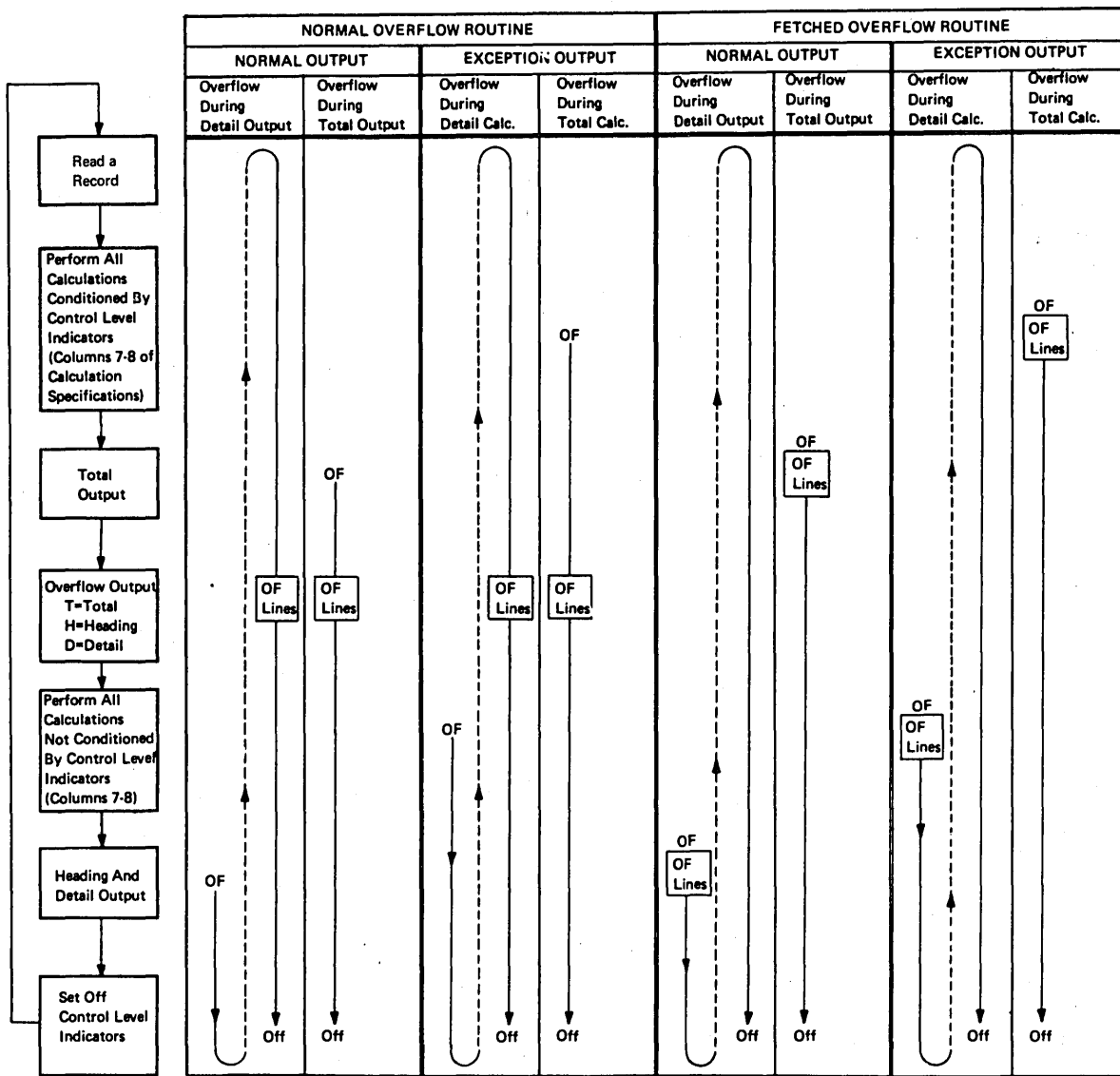
1. Overflow indicators may be turned on and off by the operation codes SETON and SETOF.
2. Spacing past the overflow line causes the overflow indicator to turn on.
3. Skipping past the overflow line to any line on the same page turns the overflow indicator on.
4. Skipping past the overflow line to any line on the new page does not turn the overflow indicator on.
5. A skip to a new page specified on a line not conditioned by an overflow indicator causes the overflow indicator to turn off.

IBM			RPG OUTPUT -																																						
Date _____			Punching Instruction															Graphic Punch																							
Program _____			Programmer _____																																						
Line	Form Type	Filename	Type (H/D/T/E)																Space		Skip		Output Indicators						Field Name												
			Stacker	Select	Fetch	Overflow	(F)	Before	After	Before	After	Not	Not	Not	And	And	And	And	And	And																					
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								
01	O	PRINTER	H																																						
02	O																																								
03	O												F		I																										
04	O																																								
05	O												T		I																										
06	O																																								
07	O												T		I																										
08	O																																								
09	O												F		I																										
10	O																																								
11	O												T		I																										
12	O																																								
13	O												T		I																										
14	O																																								
15	O												T		I																										

Figure 155. Uses of Fetch

Figure 156 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 lefthand portion of the graph shows when the indicators are on or off in relation to the general program cycle. For example, if, during normal output, a detail line is printed

on the line number specified as the overflow line, the overflow indicator turns on. It remains on until the end of the next program cycle. The solid black lines indicate that the indicator is on. The dashes are used to show a connection between the end of one cycle and the start of the next.



ART: 55008

Figure 156. Overflow Printing: Setting of the Overflow Indicator

RPG II OBJECT PROGRAM LOGIC (DETAILED)

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 7–8 of the Calculation sheet or a T in column 15 of the Output-Format sheet). 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 *General RPG II Program Logic* under *Introduction* for a general description of the logic flow.

The specific steps taken in one program cycle are shown in Figure 157. The item numbers in the following description refer to the numbers in the figure. A program cycle begins with step 1 and continues through step 26.

1. All data files to be used by the RPG II object program are *opened*; that is, they are prepared to be processed by the object program. Pre-execution time tables and arrays are loaded before the first program cycle.
2. The object program performs all output conditioned by the 1P 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 performs a test to determine if 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 turns on. Otherwise, the indicator turns 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.
- 5A. 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.
- 5B. If the operator desires 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, a branch is taken to step 35.
6. All record identifying indicators and indicators 1P, L1–L9, and H1–H9 are turned off.
7. The program tests to see if the LR indicator is on. If it is, the program branches to step 27.
8. The program tests to see if the keyboard is the primary file. If it is, 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, it is read only on the first cycle. After that, records with look-ahead fields are identified only.
10. The program performs a test to determine if 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, the program performs a test to determine if the input records are in the sequence specified for them on the Input Specifications sheet. If the sequence is incorrect, the program branches to step 33. The program also branches to step 33 if non-sequential input records are specified and the record cannot be identified.
12. If end-of-job conditions have been met, a branch is taken to step 27. All files for which an E has been specified in column 17 of the File Description Specifications sheet must be at end of file.
13. When multiple input files are used, it is necessary to select the next record to process. A branch to step 28 is made.
14. If there is only one input file, no record selection is needed. A test is made to determine if sequence checking has been requested. If so, a branch is taken to step 31.

15. The record identifying indicator specified for the current record type turns 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 performs a test to determine if a control break has occurred (the contents of the control field are not equal to the contents of a previously stored 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 turned on. All lower level indicators are also turned on.
18. If this is the first program cycle, the program bypasses all total calculation and output operations and branches to step 21.
19. All calculations conditioned by control level indicators (columns 7–8 of calculation specifications) are performed and resulting indicators are turned 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, 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 turned on.
20. All total output that is not conditioned by an overflow indicator is performed. The program performs a test to determine if an overflow condition has occurred. If an overflow condition has occurred at any time during this cycle, the overflow indicator turns 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 performs a test to determine if the last record indicator (LR) is on. If the indicator is on, the program branches to step 38.
22. The program performs a test to determine if 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 (no N preceding the indicator) overflow indicator are performed. File translation, if specified, is done for overflow output.
24. The MR indicator turns on if this is a multifile job and the record to be processed is a matching record. Otherwise, the MR indicator turns off.
25. Field indicators are turned on or off as specified. Data from the last record read and from specified look ahead fields is made available for processing.
26. Any calculations not conditioned by control level indicators (columns 7–8 of the calculation specifications) are performed, and resulting indicators are turned on or off as specified. File translation, if specified, is done for exception output, 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 turned on. Processing continues with step 3.
27. The last record indicator (LR) and all control level indicators (L1–L9) are turned on and processing continues with step 19.
28. If a file has been forced, the next record in that file is selected for processing and a branch is taken to step 15.
29. If a record with no matching fields is found in a normal input file which is not at end of file, it is selected.
30. When matching fields are specified, the normal file with the highest priority matching record filed 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 is stopped because a file with matching fields is out of sequence. The operator's option, indicated in step 34, is to bypass (read the next record from the same file) or cancel the job.
33. The execution of the program is stopped because of a record type sequence error or an unidentified record.

Figure 157. Detailed RPG II Object Program Cycle (Part 1 of 2)

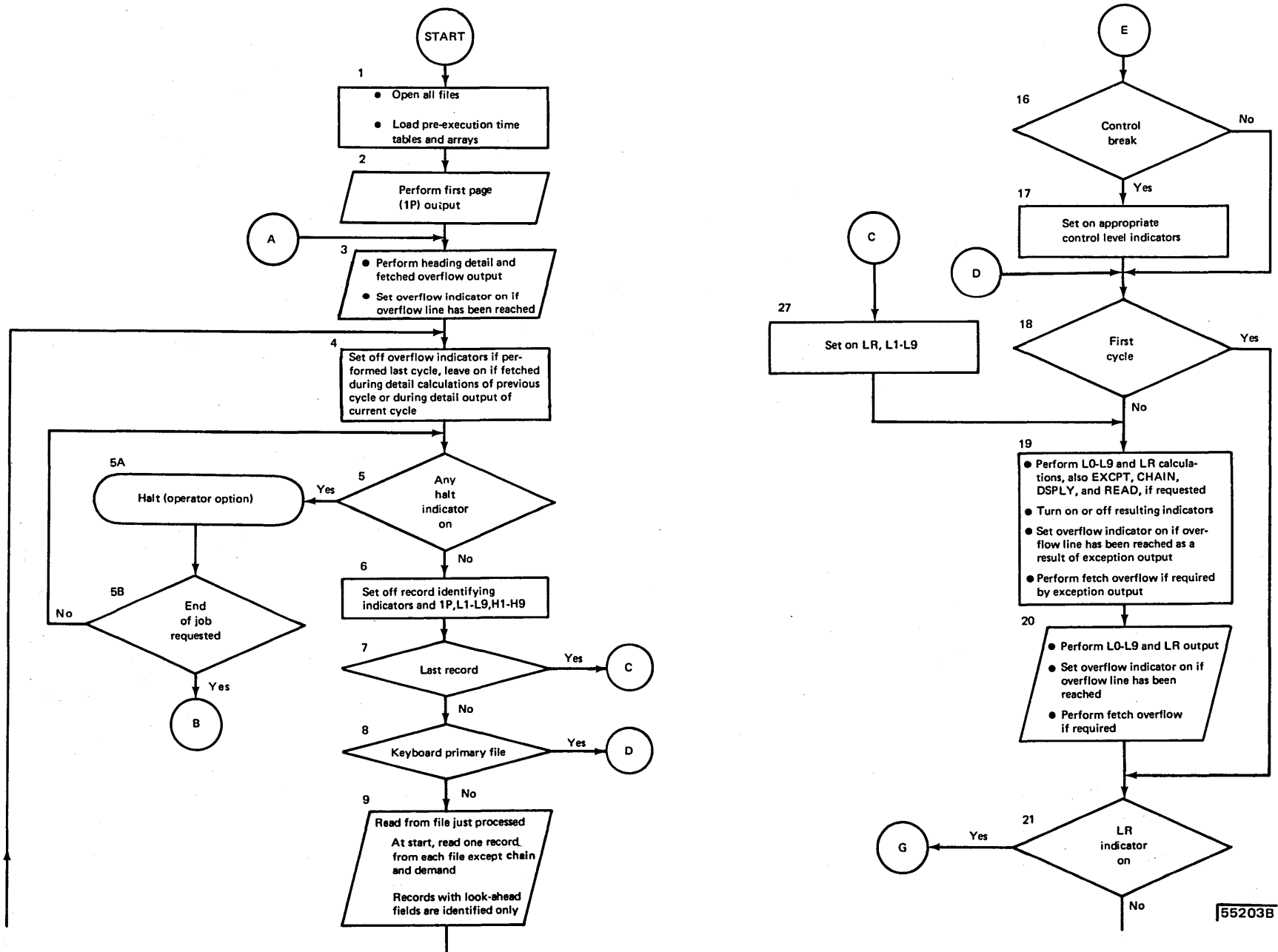
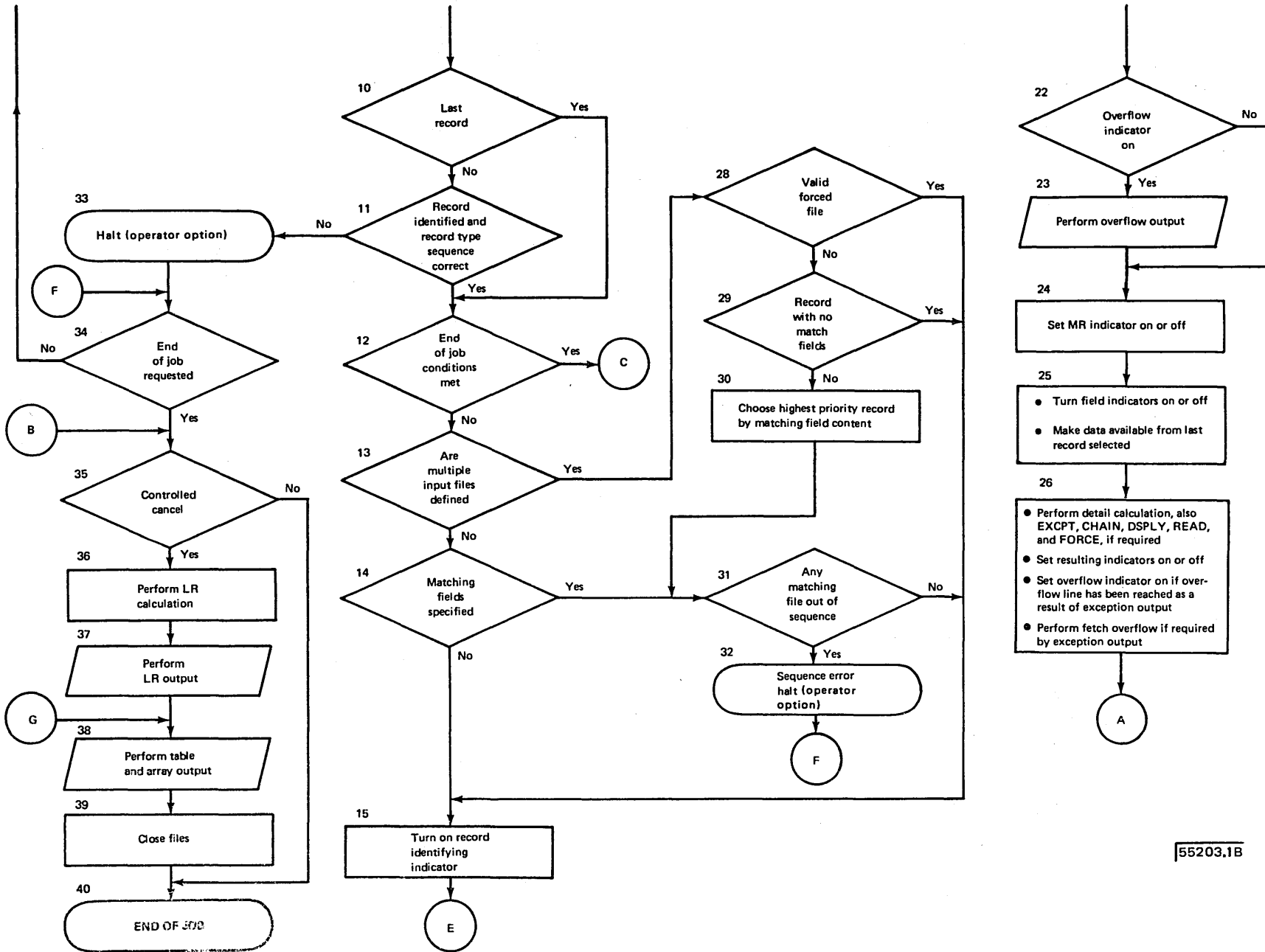


Figure 157. Detailed RPG II Object Program Cycle (Part 2 of 2)



55203.1B

34. Step 34 tests the operator's decision either to bypass the record which causes 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 36–40 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 out any tables or arrays for which a To Filename is specified on the Extension sheet. Output tables or arrays are translated, if necessary.
39. All files used by the program are closed (final termination functions are done.)
40. End of job occurs.

SUBROUTINES

A subroutine is a routine that is part of another main routine. A routine is something done over and over again. A program can be called a routine because the instructions in a program are done again and again (the program cycle). A subroutine is a group of instructions in that main routine (program) which may be done several times in one program cycle.

Sometimes it is necessary to write a program which at several points does the same operations. Instead of having to write these instructions every time they are needed, it is easier and less time consuming if they can be written just once and then referred to each time they are needed. You can do this by writing a subroutine which then consists of all those operations you have to do at several points in your program.

Coding Subroutines

Subroutines are coded and used on the Calculation sheet. They are entered after all other calculation operations. Every subroutine must have a name, but no two subroutines used in the same program may have the same name. A subroutine name cannot be used as the label of a TAG or ENDSR operation.

Enter the name of the subroutine in Factor 1. On the same line enter the operation code BEGSR (line 10 of Figure 158). The subroutine name can be 1–6 characters long and must begin in column 18 with an alphabetic character. The

remaining characters can be any combination of alphabetic or numeric characters (no special characters). Blanks may not appear between characters in the name.

IBM International Business Machines Corporation
RPG CALCULATION SPECIFICATIONS

Date _____
 Program _____
 Programmer _____

Punching Instruction	Graphic								
	Punch								

Line	Form Type (LO, LR, SR)	Indicators						Factor 1	Operation	Factor 2	Result Field	Field Length	Decimal Positions Max 4, Min 1, 1st
		And			And								
		Not	Not	Not	Not	Not	Not						
01	C												
02	C												
03	C												
04	C												
05	C												
06	C												
07	C												
08	C												
09	C												
10	C	SR					SUBRTN	BEGSR					
11	C	SR											
12	C	SR											
13	C	SR											
14	C	SR											
15	C	SR											
16	C	SR											
17	C	SR						ENDSR					

Calculation operations

Calculation operations in the subroutine

Figure 158. Subroutine Lines (SR)

Each specification line within the subroutine except AN/OR lines must have SR in columns 7–8 to identify it as a subroutine line (see Figure 158). The last statement of the subroutine is indicated by the operation code ENDSR

(see line 17 of Figure 158). Factor 1 of the ENDSR statement may contain a name. This name indicates the point to which a GOTO within the subroutine can branch (Figure 159).

IBM International Business Machines Corporation
RPG CALCULATION SPECIFICATION

Date _____
 Program _____
 Programmer _____

Punching Instruction	Graphic				
	Punch				

Line	Form Type	Control Level (LO, LR, SR)	Indicators						Factor 1	Operation	Factor 2	Result Fi																																											
			And		And																																																		
			Not	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												
0	1	C																																																					
0	2	C																																																					
0	3	C																																																					
0	4	C																																																					
0	5	C																																																					
0	6	C																																																					
0	7	C																																																					
0	8	C																																																					
0	9	C	SR																																																				
1	0	C	SR																																																				
1	1	C	SR																																																				
1	2	C	SR	I Ø																																																			
1	3	C	SR																																																				
1	4	C	SR																																																				
1	5	C																																																					
		C																																																					
		C																																																					
		C																																																					
		C																																																					
		C																																																					
		C																																																					

Calculation operations

Calculation operations

Factor 1 of the ENDSR statement contains a name to which the GOTO statement in the subroutine can branch.

Figure 159. Subroutines (ENDSR)

The subroutine, even though specified last on the Calculation sheet, may be performed at any point in the calculation operations. Whenever the subroutine is to be used, enter the operation code EXSR (execute subroutine). The

name of the subroutine to be used must also be entered in Factor 2 (lines 04 and 08 of Figure 160). Using the EXSR operation is known as calling a subroutine.

IBM International Business Machines Corporation
RPG CALCULATION SPECIFICATION

Date _____
 Program _____
 Programmer _____

Punching Instruction	Graphic								
	Punch								

Line	Form Type	Control Level (LD, LB, LR, SR)	Indicators			Factor 1	Operation	Factor 2	Result Field
			Not	And	And				
0 1	C								
0 2	C								
0 3	C								
0 4	C						EXSR SUBRTN		
0 5	C								
0 6	C								
0 7	C								
0 8	C	L2					EXSR SUBRTN		
0 9	C								
1 0	C								
1 1	C	SR			SUBRTN	BEGSR			
1 2	C	SR	01						
1 3	C	SR	20NHZNOI						
1 4	C	SR							
1 5	C	SR				ENDSR			
	C								
	C								
	C								
	C								

Calculation operations (lines 04, 08)
 Calculation operations within the subroutine (lines 11-15)

Figure 160. Subroutines (EXSR)

When you use a GOTO statement in a subroutine, you may only branch to another statement in that same subroutine. Branching (GOTO) to a statement in another subroutine or outside of a subroutine causes an error condition. You cannot use a GOTO from outside the subroutine to a statement within the subroutine either. Figure 162 shows the correct use of GOTO and TAG within a subroutine.

Tables are used during the execution of a program much like a shipping clerk would use a rate table for obtaining freight rates. The clerk might scan the table for the desired city, then select the corresponding rate. Tables are referenced by searching the table, one item at a time, for a specific item of data with a unique identifier.

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. Both tables and arrays are described on the Extension sheet. Important differences exist, however, in defining and processing tables and arrays.

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. This is done by indexing to a specific item in the array. Also, an entire array can be processed sequentially by using the array name only once in certain calculation operations.

IBM

Date _____

Program _____

Programmer _____

International Business Machines Corporation

RPG CALCULATION SPECIFICATIONS

Form X21-9093
Printed in U.S.A.

Page 1 2

Program Identification 75 76 77 78 79 80

Line	Form Type	Control Level (L, O, L, R, S, F)	Indicators			Factor 1	Operation	Factor 2	Result Field	Field Length	Decimal Positions Half Adjust (H)	Resulting Indicators			Comments					
			Not	And	And							Arithmetic	Plus	Minus		Zero				
												Compare								
												High	Low	Equal						
												1 > 2	1 < 2	1 = 2						
												Lookup								
												Table (Factor 2) is								
												High	Low	Equal						
												54	55	56	57	58	59			
0 1	C																			
0 2	C																			
0 3	C																			
0 4	C																			
0 5	C																			
0 6	C	SR				A	BEGSR													
0 7	C	SR				START	TAG													
0 8	C	SR					}													
0 9	C	SR	01				GOTO END													
1 0	C	SR					}													
1 1	C	SR																		
1 2	C	SR					GOTO START													
1 3	C	SR				END	TAG													
1 4	C	SR					ENDSR													
1 5	C																			

Figure 162. GOTO and TAG Within a Subroutine

Several terms are used to describe tables and arrays:

- *Compile time tables and arrays* are compiled with the source program and become a permanent part of the object program. A compile time table or array can be permanently changed only by recompiling the source program with the revised table or array.
- *Pre-execution 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. They are loaded after actual execution of your RPG II program has begun (read in as input data or created during calculations in your program). An execution time array is also described in extension specifications. Tables cannot be specified for execution time.
- *Related tables and arrays* are tables and arrays that are used together. The items in each table or array are called corresponding items; each item in the second table or array gives additional information about its corresponding item in the first.

You can specify related tables or related arrays, but you cannot enter specifications to relate tables with arrays or vice versa.

In Figure 163, table A (TABA) and table B (TABB) are related. An item in table A gives a part number, 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 may have different characteristics. Related tables and arrays do not have to have the same number of entries.

- *Short tables and arrays* are those in which not all of the entries contain data. The unused parts of numeric tables and arrays are filled with zeros; the unused parts of alphameric tables and arrays are filled with blanks. You usually create short tables or arrays when you have only a few table or array items available when building the table, but know that more items will soon 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)

345126
38A473
39K143
40B125
41C023
42D893
43K832
44H111
45P673
46C732

TABB
(unit cost)

373
498
1297
93
3998
87
349
679
898
47587

A Related Tables

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 1–60.

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 1–50.

B The two tables are described as separate table files.

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 1–110.

C The two tables are described in alternating format.

Figure 163. Related Tables

Rules for Creating Table or Array Input Records

Table and array data must be recorded according to certain rules:

1. The first table or array entry for each input record must begin in position 1.
2. An entire record need not be filled with entries. If it is not, blanks or comments can be included after the entries (Figure 164).
3. Each input record except the last must have the same number of entries. You may want to place just one entry on each record or as many entries as the record can hold.
4. Each entry must be 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, corresponding items must be on the same input record; together they cannot exceed the maximum record length for the device.
5. 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.
6. The number of table and/or array names used in a program must be no more than 60.

Defining Tables and Arrays

All tables and arrays are described on the Extension sheet. One line is used to describe each set of table or array input records.

If only one table or array is described, columns 11-45 are used. If alternating tables or arrays are described on one set of input records, the second table or array is described in columns 46-57 of the same line as the first table or array. If pre-execution time tables and arrays are being described, entries in columns 11-18 and 27-45 are required. Columns 19-26 are used if the table or array is to be written or punched at the end of the job. Columns 11-18 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 pre-execution time tables and arrays can be mixed. Remember, the sequence in which tables and arrays are specified on the Extension sheet determines the order in which they must be loaded at the start of the job.

Figure 165 shows the extension specifications required for each type of array.

- Line 1 specifies a compile time array, ARRAYC. This array has a total of eight elements (three elements per record). Each element has a length of 12 positions, including 4 decimal places.

48 K 16343J 64044H 12648A 47349K 34650B 1 25 51 C 0 4 3 52 D 89 3 73 K 33 274 H 1 2 1

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

85 P 67396 C 79598 F 37199 K 704

TABLE OF PART NUMBERS

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 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120

The table entries in this disk record end at position 84. The remaining record positions may remain blank or may be used for comments as in this example.

Figure 164. Table Input Record with Comments

IBM International Business Machines Corporation Form X21-9091 Printed in U.S.A.

RPG EXTENSION AND LINE COUNTER SPECIFICATIONS

Date _____ Program _____ Programmer _____

Punching Instruction	Graphic						
	Punch						

Page 1 2 Program Identification 75 76 77 78 79 80

Extension Specifications

Line	Form Type	Record Sequence of the Chaining File		To Filename	Table or Array Name	Number of Entries Per Record	Number of Entries Per Table or Array	Length of Entry	P = Packed/B = Binary Decimal Positions Sequence (A/D)	Table or Array Name (Alternating Format)	Length of Entry	P = Packed/B = Binary Decimal Positions Sequence (A/D)	Comments
		From Filename	Number of the Chaining Field										
01	E				ARRAYC	3		8	12				
02	E	CARDINP			ARRAYE	12	250	5		A			
03	E				ARRAYI		10	10					
04	E												
05	E												

Figure 165. Extension Specifications for the Three Types of Arrays

- Line 2 specifies a pre-execution time array, ARRAYE, to be read from file CARDINP. ARRAYE has 250 alphameric elements (12 elements per record); each element is 5 positions long and is equal to or higher than the previous element in collating sequence.
- Line 3 specifies an execution time array, ARRAYI, to be read from input records. ARRAYI has 10 numeric elements each 10 positions long.

Any of these specifications may include entries in columns 19-26 (to define a filename of a file to which the array would be output at end of job) and in columns 46-57 (to define an alternating array).

Loading Tables and Arrays

Tables and arrays can be loaded at compilation time or pre-execution time. Arrays can also be loaded at execution time.

Compilation Time Tables and Arrays

Tables and arrays loaded at compilation time are compiled along with the RPG II source program and they become a part of that program. Rules for loading tables and arrays at compile time are as follows:

- Compile time tables must be placed in the source library following the RPG II source program. (Refer to *IBM System/3 Model 6, Operation Control Language and Disk Utility Programs Reference Manual, GC21-7516*, for information on system control programs.)
- A record with **b in positions 1-3 must be placed before each table entered.
- A record with / * in positions 1 and 2 must be placed at the end of the last compile time table.
- Tables are loaded in the same order as they are described on the Extension sheet.
- A compile time array must have entries in columns 33-35 of the Extension sheet and must not have entries in columns 11-18 of the Extension sheet.
- Tables and arrays must not be in packed or binary format.

Figure 166 shows the arrangement of the RPG II source program when compile time tables are loaded.

Pre-Execution Time Tables and Arrays

Pre-execution time tables and arrays are not part of your source program. They are used by the object program like any other data file.

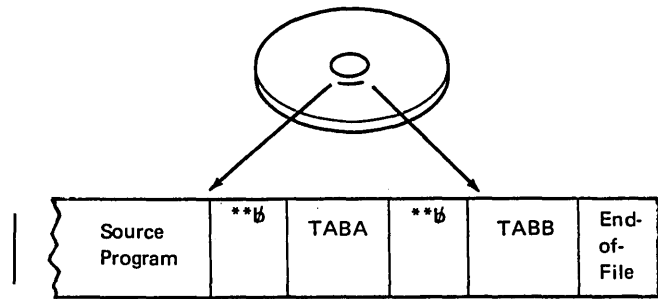
Pre-execution time tables or arrays can be loaded from the console, from the data recorder, or from disk. If a pre-execution time table or array is entered from the console, the operator must key each table or array entry (including zeros or blanks needed to pad table or array entries). The operator must also key a /* record to indicate the end of each table or array. If a pre-execution time table or array is entered from disk, the table or array file must have been created earlier. In this case, OCL statements are used just prior to program execution to call the table or array file from disk.

The following rules apply to pre-execution time tables or arrays loaded from the console, from the data recorder, or from disk:

1. The table or array must be entered before any data files.
2. A record with /* in positions 1 and 2 must be placed at the end of each pre-execution time table or array.
3. If two or more tables or arrays are to be loaded, they must be entered in the same order as they are described on the Extension sheet.

Figure 167 shows the arrangement of the input data when pre-execution time tables or arrays are loaded.

Note: When processing compile time tables and arrays or pre-execution time tables and arrays, you must not enter **b in positions 1-3 of your input records. Since **b in positions 1-3 indicates the end of a table or array, this entry would cause the system to stop processing the table or array.



These tables are located in the source library.

b = blank

Figure 166. Arrangement of the RPG II Source Program for Compile-time Tables

Execution Time Arrays

If you are loading an array from information in input records, you must describe that information in your 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, pre-execution time, 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 high or low LOKUP is used.

Array Information In One Record

If all of the array information is in one record, it can occupy consecutive positions in the record or be scattered throughout the record.

If the array elements are consecutive on the input record, they may be loaded with a single input specification. Figure 168 shows an array, INPARR, of six elements (12 positions each) being loaded from a single record from the file ARRFIL.

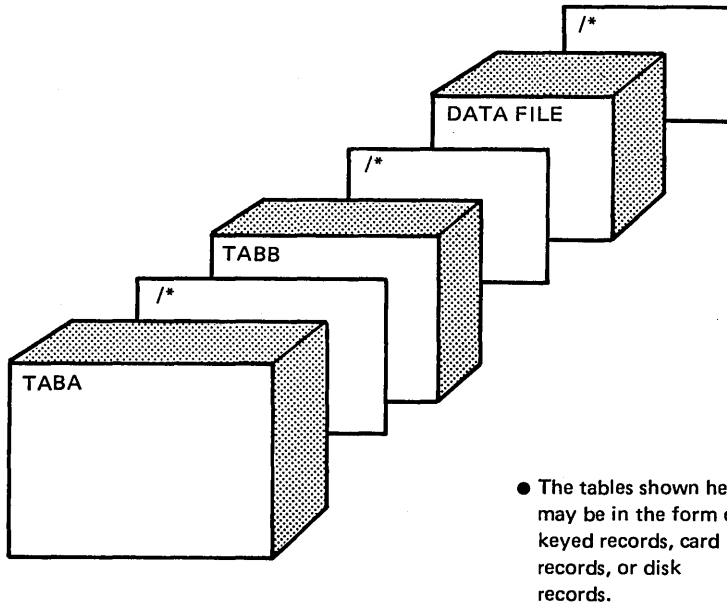


Figure 167. Arrangement of Input for Execution-Time Tables

IBM International Business Machines Corporation Form X21-9091 Printed in U.S.A.

RPG EXTENSION AND LINE COUNTER SPECIFICATIONS

Date _____ Program _____ Programmer _____

Punching Instruction: Graphic Punch

Page 1 2 Program Identification 75 76 77 78 79 80

Extension Specifications

Line	Form Type	Record Sequence of the Chaining File		To Filename	Table or Array Name	Number of Entries Per Record	Number of Entries Per Table or Array	Length of Entry	P = Packed/B = Binary Decimal Positions Sequence (A/D)	Table or Array Name (Alternating Format)	Length of Entry	P = Packed/B = Binary Decimal Positions Sequence (A/D)	Comments
		From Filename	Number of the Chaining Field										
0 1	m				IMPARR			6 12					
0 2	m												
0 3	m												

IBM International Business Machines Corporation Form X21-9094 Printed in U.S.A.

RPG INPUT SPECIFICATIONS

Date _____ Program _____ Programmer _____

Punching Instruction: Graphic Punch

Page 1 2 Program Identification 75 76 77 78 79 80

Input Specifications

Line	Form Type	Filename	Sequence	Number (1-N) Option (O)	Record Identifying Indicator or **	Record Identification Codes									Field Location		Field Name	Control Level (L1-L9)	Matching Fields or Chaining Fields	Field Record Relation	Field Indicators			Sterling Sign Position
						1			2			3			From	To					Plus	Minus	Zero or Blank	
						Position	Not (N) C/Z/D Character	Position	Not (N) C/Z/D Character	Position	Not (N) C/Z/D Character	Sticker Select: P = Packed/B = Binary												
0 1	I	ARRFILE	AA		01																			
0 2	I												1	72										
0 3	I																							
0 4	I																							

Figure 168. Defining an Execution Time Array with Consecutive Elements

If the array elements are scattered throughout the record, they may be defined and loaded one at a time, one to a specification line. In Figure 169, an array, *ARRX*, of six elements with 12 positions each, is loaded from a single record from file *ARRFILE*; a blank column appears between each two elements.

The following input specifications are required for loading an array from a single input record:

<i>Column</i>	<i>Entry</i>
6	I
7-42	Blank
43	P (packed), B (binary), or blank (unpacked).
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 sheet.
59-62	Blank
63-64	Field Record Relation indicator. See <i>Columns 63-64</i> under <i>Input Specifications</i> for information on this entry.
65-70	Blank
71-74	Blank

IBM International Business Machines Corporation Form X21-9091 Printed in U.S.A.

RPG EXTENSION AND LINE COUNTER SPECIFICATIONS

Date _____ Program _____ Programmer _____

Punching Instruction	Graphic									
	Punch									

Page 1 2 Program Identification 75 76 77 78 79 80

Extension Specifications

Line	Form Type	Record Sequence of the Chaining File		To Filename	Table or Array Name	Number of Entries Per Record	Number of Entries Per Table or Array	Length of Entry	P = Packed/B = Binary Decimal Positions Sequence (A/D)	Table or Array Name (Alternating Format)	Length of Entry	P = Packed/B = Binary Decimal Positions Sequence (A/D)	Comments
		Number of the Chaining Field	From Filename										
0 1	I				ARRX			6	12				
0 2	I												
0 3	I												

IBM International Business Machines Corporation Form X21-9094 Printed in U.S.A.

RPG INPUT SPECIFICATIONS

Date _____ Program _____ Programmer _____

Punching Instruction	Graphic									
	Punch									

Page 1 2 Program Identification 75 76 77 78 79 80

Input Specifications

Line	Form Type	Filename	Sequence Number (1-N) Option (O)	Record Identifying Indicator or **	Record Identification Codes									Field Location		Field Name	Decimal Positions	Control Level (L1-L9) Matching Fields or Chaining Fields	Field Record Relation	Field Indicators			Sterling Sign Position		
					1			2			3			From	To					Plus	Minus	Zero or Blank			
					Position	Not (N) C/Z/D	Character	Position	Not (N) C/Z/D	Character	Position	Not (N) C/Z/D	Character												
0 1	I	ARRFILE	AA	01																					
0 2	I													1	12	ARRX, 1									
0 3	I													14	25	ARRX, 2									
0 4	I													27	38	ARRX, 3									
0 5	I													40	51	ARRX, 4									
0 6	I													53	64	ARRX, 5									
0 7	I													66	77	ARRX, 6									
0 8	I																								
0 9	I																								

Figure 169. Defining an Execution Time Array with Scattered Elements

Array Information in More Than One Record

If the array information is in two or more records, there are many methods that may be used to load the array into the system. The method you use is primarily based on the size of the array and whether the array entries are consecutive in the input records. Figure 170 shows the array that could result by loading array information from certain input records. Each record identified by a 1 or 3 in column 1 contains twelve 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 section. Keep in mind that the RPG II program processes one record at a time. You cannot process the entire array until all of the records containing the array information have been read and the information has been moved into the array fields. It may, therefore, be necessary to suppress calculation and output operations until the entire array has been read into the system.

Searching Tables and Arrays

Tables and arrays can be searched using the LOKUP operation code. See *Look-up Operations* under *Operation Codes* in this section for a description of how to use LOKUP.

Using Arrays

Arrays can be used in input, output, or calculation specifications (see *Examples*). 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.

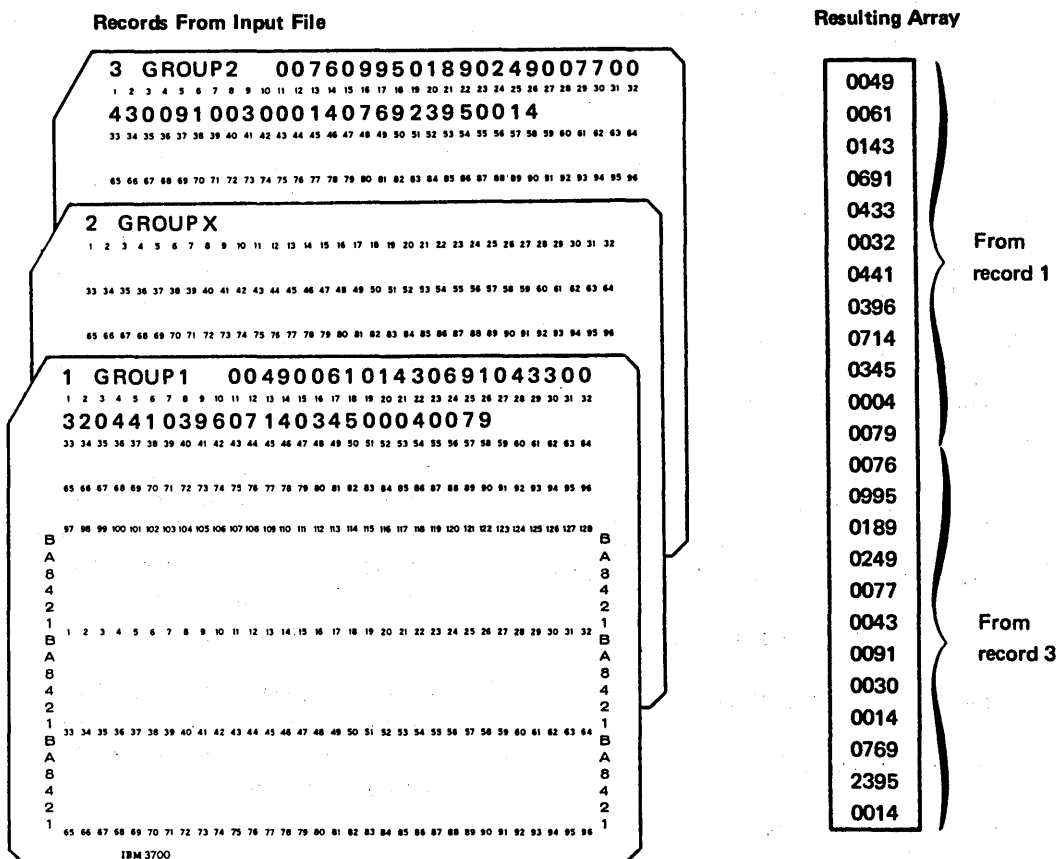


Figure 170. Loading An Array from Input Records

Array Name and Index

The array name is specified beginning in column 27 or column 46 of the Extension sheet and must be a valid RPG II name.

The length of the array name depends on how the array is being used. The array name can be from 1-6 characters long. The array name is used by itself only when referencing the entire array.

If individual elements of the array are to be referenced, the array name will require an index. An index may be a numeric field with zero decimal positions or a literal. The array name and index must be separated by a comma. The array name with comma and index will never occupy fewer than three character positions. The total length of an array name with comma and index entry is limited to either six or ten positions. It is limited to six positions for input specifications, output specifications, or the Result Field of calculation specifications or ten positions for Factor 1 or Factor 2 of calculation specifications. The index must not be zero, negative, or greater than the number of elements in the array.

Some examples of array names with and without index are:

Valid

ARAYOL

B

AR,1 (the first element of array AR)

X,YY2 (where YY2 is a field name)

Invalid

BALANCE (array name has more than six characters)

6TOTAL (first character not alphabetic)

TOTAL- (name contains special character)

CR TOT (name contains blank)

A1,A1 (array is used as index)

BAL,XX1 (name including comma has more than six characters. This name is valid for Factor 1 and Factor 2 of the calculation specifications only.)

Referencing an Array in Calculations

You can reference an entire array or individual elements in an array using calculation specifications. Process individual elements like normal fields. Remember, if an array field is to be used as a result field, the array name with comma and index entry cannot exceed six characters (see *Examples, Example 8*).

To reference an entire array use only the array name. You may use it in Factor 1, Factor 2, or the Result Field. The following operations may be used with an array name: ADD, Z-ADD, SUB, Z-SUB, MULT, DIV, SQRT, MOVE, MOVEL, MLLZO, MLHZO, MHLZO, MHHZO, DEBUG, XFOOT, and LOKUP. Except when using XFOOT and LOKUP, Factor 1 and Factor 2 cannot be an array name unless the Result Field is also an array name.

There are also several operations that can be used with an array element only, but not the array name alone. These operations are COMP, DSPLY, TESTZ, BITON, BITOF, TESTB, KEY, and MVR.

The following rules apply when using array names without an index in calculations:

1. When the factors and the result field are arrays with the same number of elements, the operation is performed using the first element from every array, then the second element from every array, etc., 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.
2. When one of the factors is a field or constant, and the other factor is an array, and the result field is an array, the operation is performed once for every element in the shorter array. The same field or constant is used in all of the operations. Except for XFOOT, the result field must always be an array.
3. 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 or constant is used in all of the operations.
4. Resulting indicators (Columns 54-59) cannot be used due to the number of operations being performed. Exceptions are XFOOT and LOKUP which allow resulting indicators.

IBM		International Business Machines Corporation		Form X21-9093 Printed in U.S.A.								
RPG CALCULATION SPECIFICATIONS												
Date _____	Punching Instruction					Page 1 2	Program Identification					
Program _____	Graphic						75	76	77	78	79	80
Programmer _____	Punch											

Line	Form Type	Control Level (LO, LR, SR)	Indicators			Factor 1	Operation	Factor 2	Result Field	Field Length	Decimal Positions	Resulting Indicators			Comments
			Not	And	And							Plus	Minus	Zero	
01	C										High	Low	Equal		
02	C		01			000	LOKUP	TABA	TABB		1>2	1<2	1=2	35	
03	C		35	01			MOVE	NEWA	TABA		Compare				
04	C		35	01			MOVE	NEWB	TABB		Lookup				
05	C										Table (Factor 2) is				
06	C										High	Low	Equal		
07	C										53	54	55	56	
08	C										57	58	59	60	

Figure 172. Adding Entries to Short Tables

Table and Array Output

Entire tables and arrays can be written out under control of RPG II only at the end of the job (LR is on). To indicate entire table or array output, specify in columns 19-26 of the Extension sheet, the name of the output file to be used. This specification causes the program to write out the entire table or array with all modifications.

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 32-37 of the Output sheet must contain the same array name used on the Extension sheet.
- Columns 40-43 of the Output sheet must contain the record position where the last field of the array is to end.

If an output record is to contain only certain fields from a table or array, describe the fields in the same way as you do normal fields, using either an array name with an index or a table name as the field name.

Editing Entire Arrays

When editing an entire array, any editing you specify applies equally to all fields in the array. If you require different editing for various elements, reference them individually.

When you specify an edit code for an entire array (column 38), note that two blanks are automatically inserted to the left of every field in the array. When you specify an edit word instead, the blanks are not inserted. The edit word must specify all the blanks you want inserted.

Example of Using Tables

A payroll job requires two related tables. TABNUM is the search table containing employee numbers. TABRAT is the related table containing employee salary rates (see Figure 173, insert A). After an employee's rate has been found, rate is multiplied by the number of hours worked. The result is the amount earned.

Table input records are organized in alternating format. There are 8 entries in each record. Each table has 500 entries. Therefore, 63 records are required. The first 62 contain characters in positions 1-72 (five-position element in TABA, plus four-position element in TABB, times 8 entries per record). The last record has only 4 entries and contains characters in positions 1-36.

Figure 173, inserts B and C, shows the RPG II specifications needed to do the job. The following paragraphs explain the entries made.

File Description Sheet

The file containing input records is called TIMECARD. It is an input file (I in column 15) and a primary file (P in column 16). When this file reaches end-of-file, processing is to end (E in column 17). This file is read in from the data recorder.

The file containing the related tables is called RATETABL. It is an input table file (I in column 15, T in column 16). The file is to be read in from disk prior to execution time. An E is required in column 39 to show that additional information about the file is on the Extension sheet.

Extension Sheet

The extension specifications complete the definition of the file RATETABL. The table searched is called TABNUM (columns 27-32). There are eight entries in each record (columns 33-35) and 500 entries in the table (columns 36-39).

Each table entry is five positions long (columns 40-42) and contains no decimal positions (column 44). The table is organized in ascending sequence (column 45). The alternating table is called TABRAT (columns 46-51). Each entry is four positions long (columns 52-54) and contains two decimal positions (column 56).

Input Sheet

The input file (TIMECARD) is assigned a sequence of AA (columns 15-16). Record identifying indicator 01 is turned on whenever an input record is present for processing. No record identification codes are specified, 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 1-5 of the input record. The number of hours worked by the employee (HRSWKD) is in positions 42-44 of the input record.

Calculation Sheet

On line 01, Factor 1 contains the search word EMPNUM (employee number). LOKUP in columns 28-32 causes the look-up operations to be performed. Factor 2 contains the name of the table to be searched (TABNUM). The result field contains the name of the related table (TABRAT).

Line 01 causes the employee number (EMPNUM) to be used as the search word for the data contained in TABNUM. Indicator 03 is turned on when the program finds an entry in the search table that is equal to the search word.

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 EARNNS, which is five positions long with two decimal positions. The result is half-adjusted.

When an equal entry is not found in TABNUM (indicator 03 is not on), line 03 is performed. The literal 000.00 is then moved to the field EARNNS, specifying that the employee does not have an entry in the table.

TABNUM	TABRAT
12345	407
12346	593
12347	369
12348	390
12349	1379

A

Figure 173. Related Tables Used in Payroll Job (Part 1 of 3)

RPG CALCULATION SPECIFICATIONS

Date _____

Program _____

Programmer _____

Punching Instruction	Graphic						
	Punch						

Page

1	2
---	---

Program Identification

75	76	77	78	79	80
----	----	----	----	----	----

Line	Form Type	Control Level (LO, L9, LR, SR)	Indicators			Factor 1	Operation	Factor 2	Result Field	Field Length	Decimal Positions	Half Adjust (H)	Resulting Indicators			Comments	
			And	And	Not								Arithmetic	Plus	Minus		Zero
01	C		01			EMPNUM	LOOKUP	TABNUM	TABRAT								
02	C		03			TABRAT	MULT	HRSWKD	EARN\$	52H							
03	C		N03				MOVE	000.00	EARN\$								
04	C																
05	C																



Figure 173. Related Tables Used in Payroll Job (Part 3 of 3)

Example 2: In Figure 175, we see a method whereby eighteen 5-position elements of array AR1 are loaded with only 2 specification lines. On succeeding lines of the Input sheet, 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 would overlay the fields read in from the first record.

The method illustrated in Example 2 works well for small arrays.

Example 3: The specifications in Figure 176 perform the function of tabulating three levels of totals. The fields FIELD A, FIELD B, FIELD C, and FIELD D are added, as they are read from input records, 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, and L3D. In addition, as control breaks occur, L1, L2, and L3 total output is performed; total fields are zeros after they are written on the output device.

IBM

International Business Machines Corporation

RPG INPUT SPECIFICATIONS

Form X21-9094
Printed in U.S.A.

Date _____

Program _____

Programmer _____

Punching Instruction	Graphic				
Punch					

Page 1 2

Program Identification 75 76 77 78 79 80

Line	Form Type	Filename	Sequence Number (1-N)	Option (O)	Record Identifying Indicator	Record Identification Codes									Field Location		Field Name	Control Level (L1-L9)	Matching Fields or Chaining Fields	Field Record Relation	Field Indicators			Sterling Sign Position	
						1			2			3			From	To					Decimal Positions	Plus	Minus		Zero or Blank
						Position	Not (N)	C/Z/D Character	Position	Not (N)	C/Z/D Character	Position	Not (N)	C/Z/D Character											
01	I	FILE1	AA		03	80																			
02	I													1	90	AR1									
03	I		BB		04	80																			
04	I													1	50	AR1, 19									
05	I													6	10	AR1, 20									
06	I																								
07	I																								
08	I																								
09	I																								
10	I																								
11	I																								

More array elements

Figure 175. Building an Array Using Fixed Indexes

Figure 177 shows the same functions being performed using arrays. Note the reduction in coding required to specify the functions. For example, line 5 of the Calculation sheet performs the same function as lines 5 through 8 of the Calculation sheet of Figure 176. Similarly, the output specifications are reduced from 15 lines to 6. (Notice, however, that the method using arrays results in only two positions between array elements.)

Example 4: This example illustrates the use of three arrays defined as follows. (Refer to Figure 178.)

Array Name	Number of Fields	Field Length
ARA	4	5
ARB	5	10
ARC	6	4

Array ARA is contained in the input records corresponding to indicator 01, ARB in the records corresponding to 02, and ARC in both types of records. Array ARC and the first field of array ARA are to be included together in an output record as are arrays ARC and a field (identified by field X1) of array ARB. Every field in array ARC is edited according to the edit word 0 \textasciitilde \textasciitilde &CR (where \textasciitilde represents a blank).

Assume that the contents of the arrays in the first two input records are:

Record	Array	Array Contents
1	ARA	12345678901234567890
	ARC	01234567890123456789876N (note that N equals minus 5)
2	ARB	JOHN \textasciitilde DOE \textasciitilde JOE \textasciitilde SMITH \textasciitilde LEE \textasciitilde MARX \textasciitilde JIM \textasciitilde KNOTS \textasciitilde TIM \textasciitilde TYLER \textasciitilde
	ARC	(The same as in record 1)

In the first output record, the location and contents of the arrays are (\textasciitilde represents a blank):

Array	Location	Contents
ARA (first field)	85-89	12345
ARC	37-84	\textasciitilde 1.23 \textasciitilde \textasciitilde 45.67 \textasciitilde \textasciitilde 89.01 \textasciitilde \textasciitilde 23.45 \textasciitilde \textasciitilde 67.89 \textasciitilde \textasciitilde 87.65 \textasciitilde CR

For the second output record assume that the contents of field X1 is 4. The locations and contents of the arrays are:

Array	Location	Contents
ARB (fourth field)	91-100	JIM \textasciitilde KNOTS \textasciitilde
ARC	37-84	\textasciitilde 1.23 \textasciitilde \textasciitilde 45.67 \textasciitilde \textasciitilde 89.01 \textasciitilde \textasciitilde 23.45 \textasciitilde \textasciitilde 67.89 \textasciitilde \textasciitilde 87.65 \textasciitilde CR

IBM International Business Machines Corporation Form X21-9091
Printed in U.S.A.

RPG EXTENSION AND LINE COUNTER SPECIFICATIONS

Date _____

Program _____

Programmer _____

Punching Instruction: Graphic _____ PUNCH _____

Page 1 2

Program Identification: 75 76 77 78 79 80

Extension Specifications

Line	Form Type	Record Sequence of the Chaining File		To Filename	Table or Array Name	Number of Entries Per Record	Number of Entries Per Table or Array	Length of Entry	P = Packed/B = Binary Decimal Positions Sequence (A/D)	Table or Array Name (Alternating Format)	Length of Entry	P = Packed/B = Binary Decimal Positions Sequence (A/D)	Comments
		Number of the Chaining Field	From Filename										
01	E				ARA			4	5				
02	E				ARB			5	10				
03	E				ARC			6	4	2			
04	E												

IBM International Business Machines Corporation Form X21-9094
Printed in U.S.A.

RPG INPUT SPECIFICATIONS

Date _____

Program _____

Programmer _____

Punching Instruction: Graphic _____ PUNCH _____

Page 1 2

Program Identification: 75 76 77 78 79 80

Record Identification Codes

Line	Form Type	Filename	Sequence Number (LN) Option (O)	Record Identifying Indicator	Record Identification Codes									Field Location		Field Name	Control Level (L-L9)	Matching Fields or Chaining Fields	Field Record Relation	Field Indicators			Sterling Sign Position
					1	2	3	From	To	Plus	Minus	Zero or Blank											
01	I	IN	AA	01	01	01	01																
02	I		OR	02	02	02	02																
03	I												51	74	2	ARC							
04	I												1	20	0	ARA					01		
05	I												1	50	0	ARB					02		

IBM International Business Machines Corporation Form X21-9090
Printed in U.S.A.

RPG OUTPUT - FORMAT SPECIFICATIONS

Date _____

Program _____

Programmer _____

Punching Instruction: Graphic _____ PUNCH _____

Page 1 2

Program Identification: 75 76 77 78 79 80

Output Indicators

Line	Form Type	Filename	Type (H/D/T/E)	Space	Skip	Output Indicators			Field Name	Edit Codes	Blank After (B)	End Position in Output Record	P = Packed/B = Binary	Sterling Sign Position
						And	And	And						
01	O	OUT	D		I									
02	O							ARC			84		'0 . &CR'	
03	O						01	ARA, I	Z		89			
04	O						02	ARB, X1			95			
05	O													

Constant or Edit Word

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

Figure 178. Using Arrays to Format Field Output

Example 5: Figure 179 shows a method of writing short arrays on the output device. The contents of one element of a 22-element array, AR2, is written to the output file ARFILE each time the specification in line 3 of the Calculation sheet is performed.

IBM			International Business Machines Corporation			Form X21-9093 Printed in U.S.A.							
RPG CALCULATION SPECIFICATIONS			Date _____			Page 1 2			Program Identification 75 76 77 78 79 80				
Punching Instruction			Graphic			Punch			Programmer _____				
Line	Form Type	Control Level (LO-LP, LR, SR)	Indicators			Factor 1	Operation	Factor 2	Result Field	Field Length	Decimal Positions	Resulting Indicators	Comments
0 1	C	LR					Z-ADD	1	IN	20		Arithmetic Plus Minus Zero	
0 2	C	LR				DUMP	TAG					Compare High Low Equal 1 > 2 1 < 2 1 = 2	
0 3	C	LR					EXCPT					Lookup Table (Factor 2) is	
0 4	C	LR				IN	ADD	1	IN			High Low Equal	
0 5	C	LR				IN	COMP	22				Table (Factor 2) is	
0 6	C	LRN50					GOTO	DUMP				High Low Equal	

IBM			International Business Machines Corporation			Form X21-9090 Printed in U.S.A.						
RPG OUTPUT - FORMAT SPECIFICATIONS			Date _____			Page 1 2			Program Identification 75 76 77 78 79 80			
Punching Instruction			Graphic			Punch			Programmer _____			
Line	Form Type	Filename	Type (H/D/T/E)	Space	Skip	Output Indicators			Field Name	End Position in Output Record	Constant or Edit Word	Starting Sign Position
0 1	O	ARFILE	E						LR			
0 2	O								AR2, IN	20		

Edit Codes

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

Constant or Edit Word

Figure 179. Printing One Array Element Per Line

This appendix contains two complete RPG II sample programs: SAMPL1 and SAMPL2. These sample programs are designed to illustrate some of the functions of RPG II. Although they are relatively simple programs, they are complete jobs which can be run on any IBM System/3 Model 6. Also included in this appendix are the specifications sheets for three complete RPG II programs: EXMPL1, EXMPL2, and EXMPL3.

SAMPLE PROGRAM 1 (SAMPL1)

SAMPL1 is used to load an indexed disk file which consists of 100 data records created in calculation operations. (Each record contains two fields: COUNT and RECNBR.) The program only requires you to enter a blank data record at the beginning of the job and a /* record (positions one and two) at the end of the job. SAMPL1 should be followed by SAMPL2 which prints the indexed file loaded in SAMPL1 to verify that the file was loaded properly. Figure 181 shows the specification sheets required for SAMPL1.

Specifications

The control card specification (Figure 181, insert A) must be supplied for every job. It is the first record in the source program.

All files used in SAMPL1 are first described on the File Description sheet (Figure 181, insert A). The primary input file, INPUT, is assigned to the CONSOLE. Both the blank data record and the /* record are entered from INPUT. The E in column 17 ensures that the program will not end until after the /* record has been read from INPUT. At the end of the job, the indexed output file, DISKOUT, consists of 128-position records with a six-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 the job.

The input file, INPUT, is further described on the Input sheet (Figure 181, insert A). The INPUT record contains a one-position blank field called NODATA (blank in position 1). When an input record containing a blank in position one is read, record identifying indicator 01 is turned on.

On the first program cycle, the result fields, COUNT and RECNBR, are set to zero when the 01 indicator is on (Figure 181, insert B). The record number field (RECNBR) is used to keep track of the number of records written to DISKOUT. A unique key field is accumulated in the COUNT field for each record written to DISKOUT.

Conditioned by indicator 01, a literal of 5 is added into COUNT to establish a key field for the first disk record, and 1 is added into RECNBR to indicate that this is the first record being written to DISKOUT. The COUNT field is then compared to 505 to determine if the 100 records with keys in increments of 5 have been written. Since this is the first program cycle, a not equal condition exists and indicator 02 is not turned on.

When indicator 01 is on and 02 is not on, the EXCPT operation is performed and the record just created is written to DISKOUT as exception output (E in column 15 of the Output sheet). After the output record is written (Figure 181, insert B), a branch is taken to REPEAT. Another disk record is created by updating the COUNT and RECNBR fields, and COUNT is compared to 505. If these fields are equal, indicator 02 is turned on and all other calculation and output operations are bypassed. If not, the same operations are done again. Whenever COUNT equals 505, the 100 records have been written to DISKOUT and end-of-job processing can begin.

A /* record must be entered from the INPUT file in order to turn on the LR indicator.

Conditioned by LR, 1 is subtracted from RECNBR so that the field indicates the actual number of records that have been loaded. In output specifications, LR causes a message to be written to the OUTPUT file indicating that the job is finished and describing the file loaded.

IBM International Business Machines Corporation Form X21-9093 Printed in U.S.A.

RPG CALCULATION SPECIFICATIONS

Date _____ Program **SAMPLE PROGRAM #1** Programmer _____

Punching Instruction: Graphic _____ Punch _____

Page **03** Program Identification **SAMPL1**

Line	Form Type	Control Level (LO, LR, SR)	Indicators			Factor 1	Operation	Factor 2	Result Field	Field Length	Decimal Positions Half Adjust (H)	Resulting Indicators			Comments
			And	And	Not							Arithmetic	Plus	Minus	
01	C														
02	C		01				Z-ADD		COUNT	60					
03	C		01				Z-ADD		RECNR	30					
04	C					REPEAT	TAG								
05	C		01			COUNT	ADD	5	COUNT						
06	C		01			RECNR	ADD	1	RECNR						
07	C		01			COUNT	COMP	505					02		
08	C		01N02				EXCPT								
09	C		01N02				GOTO	REPEAT							
10	C	LR				RECNR	SUB	1	RECNR						
11	C														

IBM International Business Machines Corporation Form X21-9090 Printed in U.S.A.

RPG OUTPUT - FORMAT SPECIFICATIONS

Date _____ Program **SAMPLE PROGRAM #1** Programmer _____

Punching Instruction: Graphic _____ Punch _____

Page **04** Program Identification **SAMPL1**

Line	Form Type	Filename	Space			Output Indicators			Field Name	Edit Codes	End Position in Output Record	Constant or Edit Word	Sterling Sign Position
			Before	After	After	And	And	Not					
01	O	OUTPUT	T	2	01			LR					
02	O									20	'SAMPLE PROGRAM 1 HAS'		
03	O									27	'LOADED'		
04	O							RECNRZ		31			
05	O									39	'RECORDS'		
06	O									61	'INTO AN INDEXED FILE'		
07	O		T	2				LR					
08	O									21	'KEYS ARE IN ASCENDING'		
09	O									42	'SEQUENCE STARTING AT'		
10	O									64	'000005 AND INCREASING'		
11	O									84	'IN INCREMENTS OF 5'		
12	O		T		01			LR					
13	O									21	'SAMPLE PROGRAM 2 WILL'		
14	O									44	'PRINT FROM THE INDEXED'		
15	O									65	'FILE TO SHOW THAT IT'		
16	O									86	'WAS PROPERLY LOADED'		
17	O	DISKOUT	E			01N02							
18	O							COUNT		6			
19	O									94	'RECORD NUMBER'		
20	O							RECNR		128			

Figure 181. Sample Program 1 (SAMPL1) (Part 2 of 2)

SAMPLE PROGRAM 2 (SAMPL2)

SAMPL2 must be preceded by Sample Program 1 (SAMPL1). SAMPL2 reads the indexed file created by SAMPL1 and prints out fields from each record read. Thus, SAMPL2 allows you to verify that SAMPL1 loaded the indexed file properly. The program specifications required for SAMPL2 are shown in Figure 182.

Specifications

The control card specification (Figure 182, insert A) must be present in every job. It is the first record in the source program.

The files used in SAMPL2 are described on the File Description sheet (Figure 182, insert A). The indexed file (DISKOUT) loaded in SAMPL1 is defined as the primary input file for SAMPL2. The E in column 17 ensures that SAMPL2 will not end until end-of-file has been reached on DISKOUT. The records read from DISKOUT will be printed on the output file, OUTPUT. An overflow indicator is specified in columns 33-34 so that subsequent operations can be conditioned on overflow.

The primary input file, DISKOUT, is further described on the Input sheet (Figure 182, insert B). DISKOUT records are 128 positions long identified by a 0 in position one. When an input record containing a 0 in position one is read, indicator 01 is turned on.

In calculation specifications (Figure 182, insert B), 1 is added to COUNT when indicator 01 is on. The COUNT field is used to keep track of the number of records read from the DISKOUT file.

In output specifications (Figure 182, insert B), 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 is turned on.

Conditioned by LR, a total line is printed indicating how many records have been read from DISKOUT. If the number printed (COUNT) is 100, SAMPL1 and SAMPL2 were executed properly.

IBM

International Business Machines Corporation

Form X21-6093
Printed in U.S.A.

RPG CALCULATION SPECIFICATIONS

Date _____
Program SAMPLE PROGRAM 2

Punching Instruction	Graphic								
	Punch								

Page 03

Program Identification SAMPL 2

Line	Form Type	Control Level (L, P, L, R, S, F)	Indicators			Factor 1	Operation	Factor 2	Result Field	Field Length	Decimal Positions Half Adjust (H)	Resulting Indicators			Comments
			Not	And	And							Plus	Minus	Zero	
01	C														
02	C			01		COUNT	ADD	1	COUNT	30					
03	C														
04	C														
05	C														
06	C														
07	C														
08	C														
09	C														
10	C														
11	C														

IBM

International Business Machines Corporation

Form X21-6090
Printed in U.S.A.

RPG OUTPUT - FORMAT SPECIFICATIONS

Date _____
Program SAMPLE PROGRAM 2

Punching Instruction	Graphic								
	Punch								

Page 04

Program Identification SAMPL 2

Line	Form Type	Filename	Space			Skip			Output Indicators			Field Name	Edit Codes Blank After (B) End Position in Output Record P = Packed/B = Binary	Sterling Sign Position	
			Before	After	Before	After	Not	And	And						
01	O	OUTPUT													
02	O														
03	O														
04	O														
05	O														
06	O														
07	O														
08	O														
09	O														
10	O														
11	O														
12	O														
13	O														
14	O														
15	O														

Figure 182. Sample Program 2 (SAMPL2) (Part 2 of 2)

EXAMPLE PROGRAMS

Example Program 1

This example contains specifications sheets for three complete RPG II programs: EXMPL1, EXMPL2, and EXMPL3. The programs are designed to be run in sequence.

EXMPL1 loads master records into an indexed file and creates a consecutive file of transactions. The transaction file will be processed against the master file in EXMPL2. EXMPL2 should follow EXMPL1. Figure 183 shows the completed specifications sheets for EXMPL1.

IBM		International Business Machines Corporation																																								Form X21-0092	
		RPG CONTROL CARD AND FILE DESCRIPTION SPECIFICATIONS																																								Printed in U.S.A.	
Date																					Page 01		Program Identification EXMPL1																				
Program	EXAMPLE PROGRAM #1																																										
Programmer																																											

Line	Form Type	Core Size to Compile	Object Output Listing Options	Core Size to Execute	Debug	MFCM Stacking Sequence	Input-Shillings	Input-Pence	Output-Shillings	Output-Pence	Inverted Print	360/20 2501 Buffer	Number Of Print Positions	Alternate Collating Sequence	Model 20	Model 20	Work Tapes	Overlay Open	Overlay Printer	Binary Search	Type Error	2162 Checking	Inquiry	Read/Write/Compare	Keyboard Output	Sign Handling	IP Form Position	Indicator Setting	File Translation	Punch MFCU Zeros	Nonprint Characters	Table Load Halt	Shared I/O	Field Print	Formatted Core Dump	RPG to RPG II Conversion	
01	H																																				

Line	Form Type	Filename	File Type			Mode of Processing						Device	Symbolic Device	Label S/N/E/M	Name of Label Exit	Extent Exit for DAM		File Addition/Unordered																				
			File Designation	Sequence	File Format	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					Core Index	Number of Tracks for Cylinder Overflow	Number of Extents	Tape Rewind	File Condition U1-UB																		
			I/O/U/C/D	P/S/C/R/T/D	E	AD	F/V/S/M/D	Block Length	Record Length	L/R	A/P/I/K	I/D/T or 2																										
02	F	XX																																				
03	F	X																																				
04	F	X THIS PROGRAM																																				
05	F	X																																				
06	F	X 1. LOADS MASTER RECORDS INTO AN INDEXED																																				
07	F	X FILE.																																				
08	F	X																																				
09	F	X 2. CREATES A CONSECUTIVE FILE OF																																				
10	F	X TRANSACTIONS.																																				
11	F	X																																				
12	F	X 3. COUNTS THE TOTAL NUMBER OF RECORDS LOADED																																				
13	F	X INTO EACH FILE. THIS TOTAL IS PRINTED AT																																				
14	F	X THE END OF JOB.																																				
15	F	X																																				
16	F	XX																																				
17	F	CARDIN IPE F 96 96 DATA96																																				
18	F	MASTER 0 F 52 26 0SA1 2 DISK																																			01	
19	F	TRANS 0 F 30 10 DISK																																		01		
20	F	TRACTR1 0 F 1 78 TRACTR1																																				

Figure 183. Specifications for EXMPL1 (Part 1 of 3)

IBM	International Business Machines Corporation	Form X21-9090 Printed in U.S.A.
RPG	OUTPUT - FORMAT SPECIFICATIONS	1 2
Date _____	Punching Instruction	Page 04
Program EXAMPLE PROGRAM #1	Graphic	75 76 77 78 79 80
Programmer _____	Punch	Program Identification EXMPL1

Line	Form Type	Filename	Type (M/D/T/E)	Space			Skip			Output Indicators			Field Name	Edit Codes	End Position in Output Record	Starting Sign Position
				Stacker Select/Fetch Overflow (F)	Before	After	Before	After	Not	And	And	Not				
01	O	MASTER	D										VALUEC	01	26	
02	O												VALUEB		22	
03	O												VALUEA		18	
04	O												DESC		14	
05	O												KEY		6	
06	O												ID		1	
07	O															
08	O	TRANS	D										AMT	02	10	
09	O												KEY		6	
10	O												ID		1	
11	O															
12	O	TRACTR1	T		204					LR					24	'TRANSACTIONS LOADED'
13	O												TOTTRNZ		4	
14	O															
15	O		T							01 LR						
16	O														19	'MASTERS LOADED'
17	O												TOTMASZ		4	

Figure 183. Specifications for EXMPL1 (Part 3 of 3)

Control Card Specifications

This card must be present in every job. It is the first card in the source deck.

File Description Specifications

These specifications describe the files in the program. The input card file, CARDIN, is read from DATA96. An E in column 17 indicates that the program will end when the last data record in the input file has been processed. The indexed output file, MASTER, will consist of 26-position records with a 5-position key field starting in the second record position. MASTER is a single volume file (01 in columns 68-69). A consecutive output file, TRANS,

with a 10-position record length is also specified on the File Description Sheet. TRANS is also a single volume file (01 in columns 68-69). A printer output file, TRACTR1, with a record length of 78 is also defined on the File Description Sheet.

Input Specifications

There are two types of records in the input card file, CARDIN: master and transaction. A character M in position 1 of the input records will turn on record identifying indicator 01, indicating a master record. A character A, B, or C in position 1 of the input records will turn on record identifying indicator 02, indicating a transaction record. No sequence checking will occur for either type of record (AA and AB in columns 15-16).

Calculation Specifications

The field names TOTMAS is incremented by one when record identifying indicator 01 is on. This maintains a running total of the master records which have been read from CARDIN and transferred to disk. The field TOTTRN is incremented by one when record identifying indicator 02 is on, maintaining a running total of the transaction records which have been read from CARDIN and transferred to disk.

Output-Format 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 (TRACTR1).

The detail records for MASTER are conditioned by record identifying indicator 01. The detail records for TRANS are conditioned by record identifying indicator 02.

Both total lines for TRACTR1 are printed when the last record identifying indicator is turned on (LR in columns 23-25). 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. The printer skips to line 1 of the next page after it is printed.

Example Program 2

EXMPL2 must be preceded by EXMPL1. EXMPL2 reads from the transaction file, TRANS, created by EXMPL1 and accumulates totals for A, B, and C records. EXMPL2 also retrieves matching master records for transaction records and prints an error message if a matching master record is not found. Figure 184 shows the completed specifications sheets for EXMPL2.

Control Card Specifications

This card must be present in every job. It is the first card in the source deck.

File Description Specifications

The input file for EXMPL2, TRANS (the output transaction file for EXMPL1), is read from disk. An E in column 17 indicates that the program will end when the last data record in the input file has been processed. TRANS is a single volume file (01 in columns 68-69). The output file, TRACTR1, will consist of 72-position records. An overflow indicator (OF in columns 33-34) is being used to condition printing of records in the file. The indexed file, MASTER, is described as 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. It is a single volume file on disk.

Input Specifications

There are two types of files specified on the Input Sheet: transaction and master. A character A, B, or C in position 1 of the input records will turn on record identifying indicator 01, 02, or 03, indicating a transaction record type A, B, or C respectively. A character M in position 1 of the update records will turn on record identifying indicator 04, indicating an update record. No sequence checking will occur for either type (AA and AB in columns 15-16).

IBM		International Business Machines Corporation		Form X21-9090 Printed in U.S.A.	
RPG		OUTPUT - FORMAT SPECIFICATIONS		Page 1 2	
Date		Punching Instruction	Graphic	Program Identification	75 76 77 78 79 80
Program	EXAMPLE PROGRAM #2	Punch		EXMPL2	
Programmer					

Line	Form Type	Filename	Space			Skip			Output Indicators			Field Name	Edit Codes	Blank After (B)	End Position in Output Record	P = Packed/B = Binary	Edit Codes						Sterling Sign Position
			Before	After	Not	Before	After	Not	And	And	And						Commas	Zero Balances to Print	No Sign	CR	-	X	
01	O																						
02	O																						
03	O											KEY											
04	O		D	I					01N10			VALUEAJ			28								
05	O											DESC			16								
06	O											KEY			5								
07	O																						
08	O		D	I					02N10			VALUEBJ			38								
09	O											DESC			16								
10	O											KEY			5								
11	O																						
12	O		D	I					03N10			VALUECJ			48								
13	O											DESC			16								
14	O											KEY			5								
15	O																						
16	O	MASTER	D						04N10			VALUEC	B		26								
17	O											VALUEB	B		22								
18	O											VALUEA	B		18								
19	O																						

● Figure 184. Specifications for EXMPL2 (Part 4 of 4)

Calculation Specifications

When indicator 01, 02, or 03 is on, two operations will occur:

1. A matching master record is retrieved for a transaction record (lines 01, 02, and 03 on the Calculation Sheet).
2. The AMT field of the transaction cards is added to the appropriate value (VALUEA, VALUEB, or VALUEC) on the master card depending on the type of card (record identifying indicator 01, 02, or 03).

If no matching record is found, indicator 10 will be turned on.

Output-Format Specifications

Eight printer output lines are described in these specifications. Four header lines conditioned by the first page indicator (1P in columns 23-25) or an overflow indicator (OF in columns 23-25) are printed. They will be 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 (line 20 on page 04 and lines 01-03 on page 05). 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 detail record is written on disk for the indexed update file, MASTER. It is conditioned by two indicators — the record identifying indicator 04 and not 10 which is the record identifying indicator for no matching master record, a match between the master and transaction record.

IBM International Business Machines Corporation Form X21-9090 Printed in U.S.A. 75 76 77 78 79 80

RPG OUTPUT - FORMAT SPECIFICATIONS

Date _____ Page **04** Program Identification **EXMPL3**

Program **EXAMPLE PROGRAM #3**

Punching Instruction: Graphic, Punch

Line	Form Type	Filename	Type (H/D/T/E)				Space		Skip		Output Indicators			Field Name	Edit Codes	End Position in Output Record	Constant or Edit Word	Sterling Sign Position
			Shrink	Stretch	Each	Overflow (F)	Before	After	Before	After	Not	Not	Not					
01	O	TRACTR1	H										1P		39	'IBM SYSTEM/3'		
02	O		OR										OF					
03	O																	
04	O		H										1P					
05	O		OR										OF					
06	O																	
07	O																	
08	O																	
09	O																	
10	O																	
11	O		H										1P					
12	O		OR										OF					
13	O																	
14	O		H										1P					
15	O		OR										OF					
16	O																	
17	O																	
18	O																	
19	O		D										01					
20	O												22					

IBM International Business Machines Corporation Form X21-9090 Printed in U.S.A. 75 76 77 78 79 80

RPG OUTPUT - FORMAT SPECIFICATIONS

Date _____ Page **05** Program Identification **EXMPL3**

Program **EXAMPLE PROGRAM #3**

Punching Instruction: Graphic, Punch

Line	Form Type	Filename	Type (H/D/T/E)				Space		Skip		Output Indicators			Field Name	Edit Codes	End Position in Output Record	Constant or Edit Word	Sterling Sign Position
			Shrink	Stretch	Each	Overflow (F)	Before	After	Before	After	Not	Not	Not					
01	O																	
02	O																	
03	O																	
04	O																	
05	O																	
06	O																	
07	O		T										LR					
08	O																	
09	O																	
10	O																	
11	O																	
12	O																	
13	O																	

Figure 185. Specifications for EXMPL3 (Part 3 of 3)

This appendix lists the RPG II and BSCA error messages for the IBM System/3 Model 6. For each error message, this appendix includes an explanation of the message, a description of any action the system takes, and suggested responses you can give to restart the system or to avoid the message when the job is run again. For information on other types of messages, see the *IBM System/3 Model 6 Operator's Guide*, GC21-7501.

MESSAGE FORMAT

Each message is preceded by an identification code. This code consists of four parts (Figure 186):

1. Program identification RG (for RPG II).
2. Message number.
3. Significance code:

W (Warning) — Warning that an abnormal condition exists. Corrective action is required only if condition is unintentional.

T (Terminal) — An error condition exists that requires corrective action before the system can continue executing the program.

4. Type of specification containing the error. (This part of the code does not appear in every message.) RPG II error messages are identified by a specification type of H, F, E, L, I, C, or O. BSCA messages are identified by a specification type of T, for Telecommunications Specifications. (A detailed explanation of the telecommunications specifications used to run BSCA programs is available in the *IBM System/3 RPG II Telecommunications Programming Reference Manual*, SC21-7507.)

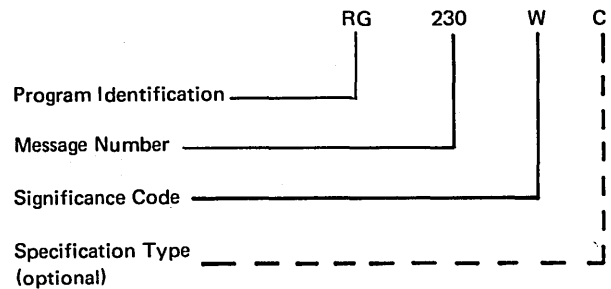


Figure 186. Message Format

RG001—NO SOURCE

Code: T—Terminal
 Explanation: You did not supply a source program for this job.
 System Action: The job is terminated.
 User Response: You must supply a source program and resubmit the job.

RG002—INVALID OBJECT OUTPUT ENTRY IN COL 10, ASSUME BLANK

Code: W—Warning
 Specification Type: H
 Explanation: The entry in-column 10 of your header line is not C, D, P, or blank.
 System Action: Blank is assumed and the object program is temporarily written in the object library.
 User Response: If this assumption was wrong, make the proper entry (C, D, or P) in column 10 and resubmit the job.

**RG003—INVALID LISTING OPTION IN COL 11,
ASSUME BLANK**

Code: W—Warning
Specification Type: H
Explanation: The entry in column 11 is neither B nor blank.
System Action: Blank is assumed. Therefore, a source program listing and the object program are produced.
User Response: If this assumption was wrong, make the proper entry in column 11 and resubmit the job.

RG004—INVALID OR BLANK STORAGE SIZE TO EXECUTE ENTRY IN COL 12-14, ASSUME SYSTEM SIZE

Code: W—Warning
Specification Type: H
Explanation: Columns 12-14 are blank or they contain an entry which is greater than 013.
System Action: The size of your system is assumed.
User Response: If this assumption was wrong, make the proper entry in columns 12-14 and resubmit the job.

RG005—INVALID DEBUG CODE IN COL 15, ASSUME BLANK

Code: W—Warning
Specification Type: H
Explanation: The entry in column 15 is neither 1 nor blank.
System Action: Blank is assumed.
User Response: If this assumption was wrong, make the proper entry in column 15 and resubmit the job.

RG006—INVALID ENTRY IN COL 16, ASSUME BLANK

Code: W—Warning
Specification Type: H
Explanation: This column is not used; it must be left blank.
System Action: Blank is assumed.
User Response: To avoid this message the next time the job is run, leave column 16 blank.

RG008—INVALID ENTRY IN COL 37 AND/OR COLUMNS 52-54. ASSUME BLANKS.

Code: W—Warning
Specification Type: H
Explanation: The entry in column 37 of your header line is not I, B, or blank.
System Action: Blank is assumed.
User Response: If this assumption was wrong, make the proper entry (I or B) in column 37 and resubmit the job.

RG009—INVALID ENTRY IN COLUMN 49, ASSUME BLANK

Code: W—Warning
Specification Type: H
Explanation: The field mode entry in column 49 of your header line is neither 1 nor blank.
System Action: Blank is assumed.
User Response: If this assumption was wrong, make the proper entry in column 49 and resubmit the job.

RG010—INVALID ENTRY IN COLUMN 45, ASSUME BLANK

Code: W—Warning
Specification Type: H
Explanation: Column 45 is not used, it must be left blank.
System Action: Blank is assumed.
User Response: To avoid this message when this job is run again, leave column 45 blank.

RG011—INVALID STERLING ENTRY OR ENTRIES IN COL 17-20, ASSUME NO STERLING

Code: W—Warning
Specification Type: H
Explanation: These columns are not used; they must be left blank.
System Action: Blanks are assumed.
User Response: To avoid this message the next time this job is run, leave columns 17-20 blank.

RG012—INVALID INVERTED PRINT ENTRY IN COL 21, ASSUME BLANK

Code: W—Warning
Specification Type: H
Explanation: The entry in column 21 of your header line is not I, D, J, or blank.
System Action: Blank is assumed; the job continues.
User Response: If this assumption was wrong, make the proper entry in column 21 and resubmit the job.

RG013—INVALID ENTRIES IN COL 22-25, ASSUME BLANKS

Code: W—Warning
Specification Type: H
Explanation: These columns are not used; they must be left blank.
System Action: Blanks are assumed.
User Response: To avoid this message the next time the job is run, leave columns 22-25 blank.

RG014—INVALID ALTERNATE COLLATING SEQUENCE ENTRY IN COL 26, ASSUME S

Code: W—Warning
Specification Type: H
Explanation: The entry in column 26 of your header line is neither blank nor S.
System Action: The entry is assumed to be S. The S entry alters the normal collating sequence.
User Response: If this assumption was wrong, make the proper entry in column 26 and resubmit the job.

RG015—INVALID ENTRIES IN COL 27-36 AND/OR 38-40, ASSUME BLANKS

Code: W—Warning
Specification Type: H
Explanation: These columns are not used; they must be left blank.
System Action: Blanks are assumed.
User Response: To avoid this message the next time the job is run, leave columns 27-36 and 38-40 blank.

RG016—INVALID 1P REPEAT ENTRY IN COL 41, ASSUME 1

Code: W—Warning
Specification Type: H
Explanation: Column 41 of your header line is neither 1 nor blank.
System Action: 1 is assumed; the job continues.
User Response: If this assumption was wrong, make the proper entry in column 41 and resubmit the job.

RG017—INVALID ENTRY IN COL 42, ASSUME BLANK

Code: W—Warning
Specification Type: H
Explanation: This column is not used; it must be left blank.
System Action: Blank is assumed; the job continues.
User Response: To avoid this message the next time the job is run, leave column 42 blank.

RG018—INVALID FILE TRANSLATION ENTRY IN COL 43, ASSUME F

Code: W—Warning
Specification Type: H
Explanation: The entry in column 43 of your header line is neither F nor blank.
System Action: F is assumed; the job continues.
User Response: If this assumption was wrong, make the proper entry in column 43 and resubmit the job.

RG019—INVALID ZERO SUPPRESS ENTRY IN COL 44, ASSUME 1

Code: W—Warning
Specification Type: H
Explanation: The entry in column 44 of your header line is neither 1 nor blank.
System Action: 1 is assumed and the job continues.
User Response: If this assumption was wrong, make the proper entry in column 44 and resubmit the job.

RG021—INVALID ENTRIES IN COL 46-47, 50-51, OR 55-74, ASSUME BLANKS

Code: W—Warning
Specification Type: H
Explanation: These columns are not used; they must be left blank.
System Action: Blanks are assumed.
User Response: To avoid this message the next time this job is run, leave columns 46-47, 50-51, and 54-74 blank.

RG022—INVALID ENTRY IN COL 6 OR SPEC TYPE OUT OF SEQUENCE

Code: T—Terminal
Specification Type: F, E, L, T, I, C, or O
Explanation: The entry in column 6 must be F, E, L, T, I, C, or O and the specifications must be in the proper sequence.
System Action: The job is terminated and the entire specification line is ignored.
User Response: Check to see which specifications contain an invalid entry in column 6 or are out of the sequence required in the source program. (Valid entries for column 6 are H, F, E, L, T, I, C, or O, and records must be in that order.) Resubmit the job.

RG023—INVALID FILENAME IN COL 7-14

Code: T—Terminal
Specification Type: F, I, L, T, or O
Explanation: Filename specified in columns 7-14 was invalid.
System Action: The job is terminated and the entire specification line is ignored.
User Response: Check your source specifications to determine which have an invalid filename. Make the proper entry and resubmit the job.

RG024—FILENAME PREVIOUSLY DEFINED IN COL 7-14

Code: T—Terminal
Specification Type: F
Explanation: This filename is not unique.
System Action: The job is terminated and the entire specification line is ignored.
User Response: Assign a unique name to the file. Resubmit the job.

RG025—INVALID DEVICE NAME IN COL 40-46, ASSUME DISK

Code: T—Terminal
Specification Type: F
Explanation: The entry in columns 40-46 is not a valid device name.
System Action: DISK is assumed, but the job is terminated.
User Response: Enter the proper device name in columns 40-46 of the File Description sheet and resubmit the job.

RG026—INVALID FILE TYPE ENTRY IN COL 15, ASSUME DEFAULT FOR DEVICE

Code: T—Terminal
Specification Type: F
Explanation: The file type entry in column 15 is not I, O, C, U, or D.
System Action: O is assumed for files assigned to TRACTR1 and TRACTR2; I is assumed for files assigned to CONSOLE or DATA96; U is assumed for files assigned to DISK. The job is terminated.
User Response: Enter the proper file type in column 15 and resubmit the job.

RG028—FILE DESIGNATION IN COL 16 IS INVALID FOR EITHER FILE TYPE OR DEVICE. ASSUME SECONDARY.

Code: W—Warning
Specification Type: F
Explanation: The entry in column 16 is not valid for an input, combined or update file.
System Action: S is assumed and the job continues.
User Response: If this assumption was wrong, make the proper entry in column 16 and resubmit the job.

**RG030—FILE DESIGNATION ENTRY IN COL 16
INVALID FOR OUTPUT OR DISPLAY FILE, ASSUME
BLANK**

Code: W—Warning
Specification Type: F
Explanation: Column 16 must be blank for out-
put files and display files (O or D
in column 15).
System Action: Blank is assumed.
User Response: To avoid this message the next
time this job is run, make a blank
entry in column 16.

**RG032—NO PRIMARY FILE SPECIFIED IN COL 16,
ASSUME FIRST SECONDARY FILE AS PRIMARY**

Code: W—Warning
Specification Type: F
Explanation: No primary file was specified (P in
column 16) of your file description
specifications.
System Action: The first secondary file is assumed
to be the primary file.
User Response: If this assumption was wrong, make
the proper entry in column 16 and
resubmit the job.

**RG033—NO PRIMARY OR SECONDARY FILE
SPECIFIED IN COL 16 OR NO FILE DESCRIPTION
SPEC FOUND**

Code: T—Terminal
Specification Type: F
Explanation: No primary or secondary file was
specified (P or S in column 16 of
the file description specifications)
or no file description specifications
were supplied.
System Action: The job is terminated.
User Response: Supply file description specifications
or define an input file and resubmit
the job.

**RG034—MULTIPLE PRIMARY FILES DEFINED IN
COL 16, ASSUME SECONDARY**

Code: W—Warning
Specification Type: F
Explanation: More than one primary file (P in
column 16) was defined in your
file description specifications.
System Action: All primary files except the first
one are assumed to be secondary.
User Response: If this assumption was wrong, make
the proper entry (S in column 16).
Resubmit the job.

**RG036—INVALID END OF FILE ENTRY IN COL 17,
ASSUME E FOR INPUT FILE TYPE WITHOUT RAN-
DOM PROCESSING**

Code: W—Warning
Specification Type: F
Explanation: The entry in column 17 of your
file description specifications is
neither E nor blank.
System Action: E is assumed for input files not
processed randomly; blank is as-
sumed for all other files.
User Response: If this assumption was wrong, make
the proper end-of-file entry in
column 17. Resubmit the job.

RG037—INVALID ENTRY IN COL 19, ASSUME F

Code: W—Warning
Specification Type: F
Explanation: The entry in column 19 of your
file description specifications is not
F.
System Action: F is assumed.
User Response: To avoid this message the next
time this job is run, enter F in
column 19.

RG038—END OF FILE ENTRY IN COL 17 INVALID FOR FILE TYPE

Code: W—Warning
Specification Type: F
Explanation: Column 17 must be blank for output, demand, table, and display files.
System Action: Blank is assumed.
User Response: To avoid this message the next time this job is run, leave column 17 blank.

RG039—INVALID SEQUENCE ENTRY IN COL 18, ASSUME PREVIOUS ENTRY

Code: W—Warning
Specification Type: F
Explanation: The entry in column 18 is not A, D, or blank.
System Action: The entry in column 18 from the previous specification line is assumed.
User Response: If this assumption was wrong, make the proper entry in column 18.

RG040—ENTRY IN COL 18 INVALID FOR TYPE OF FILE OR MODE OF PROCESSING, ASSUME BLANK

Code: W—Warning
Specification Type: F
Explanation: Column 18 must be blank for demand files, output files, record address files, display files, and for any files processed randomly.
System Action: Blank is assumed.
User Response: To avoid this message the next time this job is run, leave column 18 blank.

RG041—INVALID RECORD LENGTH ENTRY IN COL 24–27, ASSUME DEFAULT FOR DEVICE

Code: W—Warning
Specification Type: F
Explanation: Incorrect record length was specified in columns 24–27.
System Action: The maximum record length for the device is assumed, except DISK which is assumed to be 256.
User Response: If this assumption was wrong, make the proper record length entry and resubmit the job.

RG042—INVALID ENTRIES IN COL 20–23, ASSUME RECORD LENGTH

Code: W—Warning
Specification Type: F
Explanation: The entry in columns 20–23 is neither equal to nor a multiple of the record length specified in columns 24–27.
System Action: The record length is assumed.
User Response: If this assumption was wrong, make the proper block length a multiple of the record length and resubmit the job.

RG043—DUAL I/O ENTRY IN COL 32 INVALID FOR TYPE OF FILE OR MODE OF PROCESSING, ASSUME BLANK

Code: W—Warning
Specification Type: F
Explanation: Dual I/O (1–9 in column 32) cannot be specified for combined, demand, table, and update files, or for any file processed randomly. Neither can dual I/O be specified if shared I/O has been specified (column 48 of control card specifications).
System Action: Blank is assumed.
User Response: If this assumption was wrong, make the proper dual I/O entry and resubmit the job.

RG044—INVALID ENTRY IN COL 32, ASSUME BLANK

Code: W—Warning
Specification Type: F
Explanation: The entry in column 32 was not 1–9, I, T, or blank.
System Action: Blank is assumed.
User Response: If the assumption was wrong, make the proper entry and resubmit the job.

**RG045—OVERFLOW INDICATOR IN COL 33-34
PREVIOUSLY ASSIGNED, ASSUME BLANK**

Code: T—Terminal
Specification Type: F
Explanation: The same overflow indicator was assigned to more than one file.
System Action: Blank is assumed, but the job is terminated.
User Response: Assign different overflow indicators to each file being described.

**RG046—INVALID OVERFLOW INDICATOR IN COL
33-34, ASSUME BLANK**

Code: T—Terminal
Specification Type: F
Explanation: The entry in columns 33-34 was not OA-OG, or OV.
System Action: Blank is assumed, but the job is terminated.
User Response: Enter OA-OG or OV in columns 33-34 if you want to specify overflow for this file, if not, leave columns 33-34 blank. Resubmit the job.

**RG047—OVERFLOW INDICATOR IN COL 33-34
INVALID FOR DEVICE, ASSUME BLANK**

Code: W—Warning
Specification Type: F
Explanation: The overflow indicator in columns 33-34 was not assigned to a printer file or a ledger card device file.
System Action: Blank is assumed.
User Response: To avoid this message the next time this job is run, assign overflow indicators to printer files or combined ledger card files only.

**RG048—INVALID OR BLANK EXTENSION CODE
ENTRY IN COL 39 FOR TABLE OR RECORD ADDRESS
FILE, ASSUME E**

Code: W—Warning
Specification Type: F
Explanation: The extension code in column 39 was not E for a table or record address file.
System Action: E is assumed and the job continues.
User Response: To avoid this message the next time this job is run, enter E in column 39.

RG049—INVALID EXTENSION CODE IN COL 39

Code: W—Warning
Specification Type: F
Explanation: The entry in column 39 is neither L nor blank for output files assigned to the printer.
System Action: L is assumed and the job continues.
User Response: If this assumption was wrong, make the entry in column 39 blank. Resubmit the job.

**RG051—EXTENSION CODE ENTRY IN COL 39
INVALID WITH DEVICE OR P, S, C, OR D IN COL 16,
ASSUME BLANK**

Code: W—Warning
Specification Type: F
Explanation: Column 39 can only be used with table, record address or printer files.
System Action: Blank is assumed and the job continues.
User Response: If this assumption was wrong, make the proper entry in column 39 and resubmit the job.

**RG052—DEVICE IN COL 40-46 PREVIOUSLY
ASSIGNED TO OUTPUT OR NON-TABLE INPUT FILE**

Code: T—Terminal
Specification Type: F
Explanation: The device name in columns 40-46 was assigned to more than one output or non-table input file.
System Action: The job is terminated and the entire specification line is ignored. This condition may cause other errors to be generated.
User Response: Make the device name entry in columns 40-46 unique for each output or non-table input file (except those assigned to disk and console). Resubmit the job.

**RG053—INVALID ENTRIES IN COL 47-52, ASSUME
BLANKS.**

Code: W—Warning
Specification Type: F
Explanation: These columns are not used; they must be left blank.
System Action: Blanks are assumed and the job continues.
User Response: To avoid this message the next time this job is run, leave columns 47-52 blank.

**RG055—ENTRIES IN COL 71-72 INVALID FOR TABLE
FILE, ASSUME BLANK**

Code: W—Warning
Specification Type: F
Explanation: Columns 71-72 must be left blank for table files, since table files cannot be conditioned by U1-U8.
System Action: Blanks are assumed and the job continues.
User Response: To avoid this message the next time the job is run, leave columns 71-72 blank for table files.

**RG057—INVALID FILE CONDITIONING ENTRIES IN
COL 71-72**

Code: T—Terminal
Specification Type: F
Explanation: Columns 71-72 of your file description specification are not blank nor do they contain one of the external indicators (U1-U8).
System Action: The job is terminated.
User Response: Leave columns 71-72 blank or enter one of the external indicators (U1-U8). Resubmit the job.

**RG058—INVALID ENTRIES IN COLS 67 AND/OR 73-
74, ASSUME BLANK**

Code: W—Warning
Specification Type: F
Explanation: Columns 67 and 73-74 are not used; they must be left blank.
System Action: Blanks are assumed and the job continues.
User Response: To avoid this message when the job is run again, leave columns 67 and 73-74 blank.

**RG060—INVALID ENTRY IN COLUMN 48, ASSUME
BLANK**

Code: W—Warning
Specification Type: H
Explanation: The shared I/O entry in column 48 is neither I nor blank.
System Action: Blank is assumed.
User Response: If this assumption was wrong, make the proper entry in column 48 and resubmit the job.

**RG061—INVALID ENTRIES IN COL 7-10, ASSUME
BLANK**

Code: W—Warning
Specification Type: E
Explanation: Columns 7-10 are not used; they must be left blank.
System Action: Blanks are assumed and the job continues.
User Response: To avoid this message when the job is run again, leave columns 7-10 blank.

RG062—INVALID OR UNDEFINED FROM FILENAME ENTRY IN COL 11-18

Code: T—Terminal
Specification Type: E
Explanation: The From Filename in columns 11-18 of your extension specifications is invalid or has not been previously defined in file description specifications. (The From Filename must start in column 11.)
System Action: The job is terminated.
User Response: Make the proper From Filename entry in columns 11-18. If columns 11-18 already contain a valid entry, check to make sure that the file was defined in your file description specifications. Resubmit the job.

RG063—TYPE OF FILE INVALID FOR FROM FILENAME ENTRY IN COL 11-18

Code: T—Terminal
Specification Type: E
Explanation: The From Filename entry does not refer to a table or record address input file.
System Action: The job is terminated.
User Response: Make sure the entry in columns 11-18 refers to a table or record address input file. Resubmit the job.

RG064—INVALID OR UNDEFINED TO FILENAME IN COL 19-26

Code: T—Terminal
Specification Type: E
Explanation: The To Filename in columns 19-26 of your extension specifications is invalid or has not been defined in file description specifications. (The To Filename must start in column 19.)
System Action: The job is terminated.
User Response: Make the proper To Filename entry in columns 19-26. If columns 19-26 already contain a valid entry, check to make sure the filename has been previously defined in your file description specifications. Resubmit the job.

RG065—TYPE OF FILE INVALID OR INCORRECT FOR TO FILENAME ENTRY IN COL 19-26

Code: T—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.
System Action: The job is terminated.
User Response: Be sure the entry in columns 19-26 refers to an output file or to a file processed by a record address file. Resubmit the job.

RG067—INVALID TABLE OR ARRAY NAME IN COL 27-32

Code: T—Terminal
Specification Type: E
Explanation: The table or array name in columns 27-32 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.
System Action: The job is terminated.
User Response: Make the proper table or array name entry in columns 27-32 and resubmit the job.

RG068—INVALID OR MISSING NUMBER OF ENTRIES PER RECORD ENTRY IN COL 33-35, ASSUME 08

Code: T—Terminal
Specification Type: E
Explanation: The entry in columns 33-35 is missing on a specification line which has a From Filename in columns 11-18, or it is not a one to three-digit number (1-999).
System Action: 08 is assumed, but the job is terminated.
User Response: Define the number of entries per record. To do so, make a numeric entry (1-999) in columns 33-35. Resubmit the job.

RG070—INVALID OR MISSING NUMBER OF ENTRIES PER TABLE OR ARRAY IN COL 36-39, ASSUME 05

Code: T—Terminal
Specification Type: E
Explanation: The entry in columns 36-39 is missing or it is not a one to four-digit number (1-9999).
System Action: 05 is assumed, but the job is terminated.
User Response: Define the maximum number of entries per table or array. To do so, make a numeric entry (1-9999) in columns 36-39. Resubmit the job.

RG071—NO. OF ENTRIES PER RECORD IN COL 33-35 EXCEEDS NO. OF ENTRIES PER TABLE/ARRAY IN COLUMNS 36-39

Code: T—Terminal
Specification Type: E
Explanation: Number of entries per record specified is greater than the number of entries per table or array specified.
System Action: The job is terminated.
User Response: Make the proper entries in columns 33-35 and columns 36-39. The number of entries per record (columns 33-35) can be equal to or less than the number of entries per table or array (columns 36-39). Resubmit the job.

RG072—INVALID OR MISSING LENGTH OF ENTRY IN COL 40-42 OR 52-54, ASSUME 05

Code: T—Terminal
Specification Type: E
Explanation: 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).
System Action: 05 is assumed, but the job is terminated.
User Response: Enter a one to three-digit number in columns 40-42 or 52-54 to define length of table or array entries (1-15 or 1-256). Resubmit the job.

RG073—LENGTH SPECIFIED FOR EACH TABLE/ARRAY RECORD IN COL 33-35 AND COL 40-42 OR 52-54 EXCEEDS RECORD LENGTH

Code: T—Terminal
Specification Type: E
Explanation: 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 file description specifications.
System Action: The job is terminated.
User Response: Make the necessary changes so that the table record length does not exceed the record length in file description specifications. Resubmit the job.

RG074—INVALID ENTRY IN COL 43 OR 55, ASSUME BLANK

Code: W—Warning
Specification Type: E
Explanation: The entry in column 43 or column 55 of your extension specifications is not P, B, or blank.
System Action: Blank is assumed.
User Response: Make the entry in column 43 or column 55 P, B, or blank. Resubmit the job.

RG075—PACKED OR BINARY VALID ONLY FOR PRE-EXECUTION TIME TABLE OR ARRAY, ASSUME BLANK

Code: W—Warning
Specification Type: E
Explanation: Packed or binary format can only be specified (column 43 or column 55) for pre-execution time tables or arrays.
System Action: Blank is assumed.
User Response: To avoid this message the next time this job is run, leave column 43 and column 55 blank for compile time tables or arrays and for execution time arrays.

RG076—INVALID DECIMAL POSITION ENTRY IN COL 44 OR 56, ASSUME 0

Code: T—Terminal
Specification Type: E
Explanation: Decimal position entry in column 44 or column 56 is not a number 0–9 or blank.
System Action: Zero is assumed, but the job is terminated.
User Response: Make the proper decimal position entry (0–9, blank) in columns 44 and 56. Resubmit the job.

RG077—INVALID SEQUENCE ENTRY IN COL 45 OR 57, ASSUME BLANK

Code: T—Terminal
Specification Type: E
Explanation: Sequence entry in column 45 or column 57 is not A, D, or blank.
System Action: Blank is assumed, but the job is terminated.
User Response: Make the proper sequence entry (A, D, or blank) in column 45 or 57 and resubmit the job.

RG079—INVALID ALTERNATE TABLE/ARRAY NAME IN COL 46–51

Code: T—Terminal
Specification Type: E
Explanation: The table or array name in columns 46–51 was not specified properly. The table or array name must start in column 46. A table name must begin with TAB.
System Action: The job is terminated.
User Response: Enter the proper table or array name in columns 46–51 and resubmit the job.

RG080—ALTERNATE TABLE/ARRAY NAME IN COL 46–51 AND/OR 27–32 MISSING FOR ENTRIES IN COLUMNS 33–45 AND/OR 52–57, ASSUME COL 33–57 AND/OR 46–57 BLANK

Code: T—Terminal
Specification Type: E
Explanation: Columns 52–57 contain entries describing an alternating table or array, but no alternating table or array name was specified in columns 46–51 or no table or array name was specified in columns 27–32.
System Action: The job is terminated.
User Response: Make a valid table or array name entry in columns 27–32 and in columns 46–51 if an alternating table or array is described. Resubmit the job.

RG082—LENGTH OF TABLE/ARRAY ENTRY IN COL 40–42 OR 52–54 FOR ALPHA FIELDS EXCEEDS 256. ASSUME 256 FOR NON-COMPILE TIME TABLE/ARRAY, OTHERWISE ASSUME 96

Code: T—Terminal
Specification Type: E
Explanation: Length of table or array entry specified in columns 40–42 or 52–54 is too large.
System Action: 256 is assumed for non-compile time tables or arrays; 96 is assumed for compile time tables or arrays. The job is terminated.
User Response: Enter 256 or less for the length of table or array entry specifications in columns 40–42 or 52–54.

RG083—LENGTH OF TABLE/ARRAY ENTRY IN COL 40–42 OR 52–54 FOR NUMERIC FIELD EXCEEDS 15, ASSUME 15

Code: T—Terminal
Specification Type: E
Explanation: Length of numeric table or array entry specified in columns 40–42 or 52–54 is too large.
System Action: 15 is assumed, but the job is terminated.
User Response: Enter 15 or less for the length of a numeric table or array entry in columns 40–42 and/or 52–54.

RG084—FILE AND RECORD TYPE ENTRIES IN COL 7-42 AND FIELD TYPE ENTRIES IN COL 43-74 ON SAME LINE, ASSUME 7-42 BLANK

Code: T—Terminal
Specification Type: I
Explanation: Field type entries (columns 43-74) are not specified one line lower than file and record type entries (columns 7-42).
System Action: File and record type entries (columns 7-42) are assumed to be blank and the job is terminated.
User Response: Specify the field type entries (columns 43-74) one line lower than the file and record type entries (columns 7-42). Resubmit the job.

RG085—INVALID, MISSING OR UNDEFINED FILE NAME

Code: T—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 your file description specifications.
System Action: The job is terminated.
User Response: Make the proper filename entry. Also be sure that the filename has been previously defined in file description specifications. Resubmit the job.

RG086—FILENAME IN COL 7-14 DOES NOT REFER TO PRINTER FILE

Code: T—Terminal
Specification Type: L
Explanation: Filename in your line counter specifications does not refer to a printer file.
System Action: The job is terminated.
User Response: Place the proper filename entry in columns 7-14. The filename specified must refer to a printer file. Resubmit the job.

RG087—FORM LENGTH ENTRY IN COL 15-17 INVALID OR GREATER THAN 112

Code: T—Terminal
Specification Type: L
Explanation: The form length entry in columns 15-17 is not properly specified, or is too large.
System Action: The job is terminated.
User Response: Enter 112 or less for the form length entry in columns 15-17. Resubmit the job.

RG088—INVALID OR MISSING FL ENTRY IN COL 18-19, ASSUME FL

Code: W—Warning
Specification Type: L
Explanation: Columns 18-19 were left blank or the entry specified is not FL.
System Action: FL is assumed.
User Response: To avoid this message when this job is run again, enter FL in columns 18-19.

RG089—OVERFLOW LINE ENTRY IN COL 20-22 INVALID OR GREATER THAN 112

Code: T—Terminal
Specification Type: L
Explanation: The overflow line entry in columns 20-22 is invalid or a number greater than 112.
System Action: The job is terminated.
User Response: Columns 20-22 must be a number from 1-112.

RG090—INVALID OR MISSING OL ENTRY IN COL 23-24, ASSUME OL

Code: W—Warning
Specification Type: L
Explanation: Columns 23-24 were left blank or the entry specified is not OL.
System Action: OL is assumed.
User Response: To avoid the message when this job is run again, enter OL in columns 23-24.

**RG091—OVERFLOW LINE IN COL 20-22 EXCEEDS
FORM LENGTH IN COL 15-17, ASSUME FORM LENGTH**

Code: T—Terminal
Specification Type: L
Explanation: Overflow line specified is too large.
System Action: Form length is assumed, but the job is terminated.
User Response: Make the overflow line entry (columns 20-22) equal to or less than the form length entry (columns 15-17).

**RG092—INVALID OR UNDEFINED FILENAME IN COL
7-14**

Code: T—Terminal
Specification Type: L, I, T
Explanation: The filename entry is not specified properly, or it was not previously defined in your file description specifications.
System Action: The job is terminated.
User Response: Make the proper filename entry in columns 7-14. Also, make sure the filename has been previously defined in file description specifications. Resubmit the job.

**RG093—FILE AND RECORD TYPE ENTRIES IN COL
7-42 AND FIELD TYPE ENTRIES IN COL 43-74 ON
SAME LINE, ASSUME 43-74 BLANK**

Code: T—Terminal
Specification Type: I
Explanation: Field description entries (columns 43-74) are not specified one line lower than file and record identification entries (columns 7-42).
System Action: Field type entries (columns 43-74) are assumed to be blank and the job is terminated.
User Response: Specify the field type entries (columns 43-74) one line lower than the file and record type entries (columns 7-42). Resubmit the job.

**RG094—FILE AND RECORD TYPE DESCRIPTION
MUST PRECEDE THIS SPECIFICATION**

Code: T—Terminal
Specification Type: I
Explanation: File and record type entries in columns 7-42 do not precede the related field description entries in columns 43-74.
System Action: The job is terminated.
User Response: Enter the file and record type entries in columns 7-42 of the specification line immediately preceding the related field description entries in columns 43-74. Resubmit the job.

RG095—AND OR OR LINE OUT OF ORDER

Code: T—Terminal
Specification Type: I, C
Explanation: AND or OR line does not follow the proper file or record type entries or is on the first line of calculation specifications. (The system may have dropped your file and record type specifications because of other errors in your program.)
System Action: The job is terminated.
User Response: Make sure that the AND or OR line follows the proper file and record type entries and is not the first line in calculation specifications. Resubmit the job.

**RG096—AND LINE FOLLOWS LINE WITH NO RECORD
IDENTIFICATION CODES**

Code: T—Terminal
Specification Type: I
Explanation: The specification line which precedes your AND line does not contain record identification codes.
System Action: The job is terminated.
User Response: Make the proper record identification entries in the line preceding the AND line. Resubmit the job.

RG097—NO FIELDS DESCRIBED FOR THIS OR PREVIOUS RECORD

Code: W—Warning
Specification Type: I
Explanation: No field description entries were specified for this or the previous record.
System Action: No action is taken.
User Response: Make sure that all fields to be used from input records are described.

RG098—INVALID SEQUENCE ENTRY IN COL 15-16, ASSUME ALPHABETIC SEQUENCE ENTRY

Code: W—Warning
Specification Type: I
Explanation: The sequence entry in columns 15-16 is neither a two-digit number nor a two-character alphabetic entry.
System Action: A two-character alphabetic entry is assumed.
User Response: If this assumption was wrong, make the proper sequence entry and re-submit the job.

RG101—NUMERIC SEQUENCE ENTRY IN COL 15-16 NOT IN ASCENDING ORDER OR THE FIRST IS NOT 01, ASSUME PREVIOUS NUMERIC SEQUENCE OR 01 IF FIRST NUMERIC RECORD

Code: W—Warning
Specification Type: I
Explanation: Either the first numeric sequence entry is not 01 or your numeric sequence entries are not in ascending order.
System Action: If this is the first numeric sequence entry, 01 is assumed; otherwise, the numeric sequence entry from the previous specification line is assumed.
User Response: If this assumption was wrong, specify the numeric sequence entries in columns 15-16 in ascending order starting with 01, and resubmit the job.

RG102—INVALID NUMBER ENTRY IN COL 17 FOR NUMERIC SEQUENCE, ASSUME N

Code: W—Warning
Specification Type: I
Explanation: The number entry in column 17 is neither 1 nor N.
System Action: N is assumed.
User Response: If this assumption was wrong, make the proper number entry in column 17 and resubmit the job.

RG103—INVALID OPTION ENTRY IN COL 18 FOR NUMERIC SEQUENCE, ASSUME O

Code: W—Warning
Specification Type: I
Explanation: The option entry is neither O nor blank.
System Action: O is assumed.
User Response: If this assumption was wrong, leave column 18 blank and resubmit the job.

RG104—NUMBER/OPTION ENTRIES IN COL 17-18 INVALID WITH ALPHAMERIC SEQUENCE ENTRIES

Code: T—Terminal
Specification Type: I
Explanation: Columns 17 and 18 must be blank when columns 15-16 contain an alphabetic sequence entry.
System Action: The job is terminated.
User Response: Make columns 17 and 18 blank when columns 15-16 contain an alphabetic entry. Resubmit the job.

RG105—NUMBER/OPTION ENTRIES IN COL 17-18 INVALID FOR AND OR OR LINE, ASSUME BLANK

Code: W—Warning
Specification Type: I
Explanation: Columns 17 and 18 must be blank in an AND or OR line.
System Action: Blanks are assumed.
User Response: To avoid the message when this job is run again, leave columns 17 and 18 of an AND or OR line blank.

RG106—INVALID POSITION ENTRY FOR RECORD ID CODES IN COL 21-24, 28-31, OR 35-38, OR TO POSITION COL 48-51, ASSUME 1

Code: T—Terminal
Specification Type: I
Explanation: The position entry for record ID codes or the To position for a field exceeds the record length.
System Action: One is assumed; the job is terminated.
User Response: Make the proper position entry for record ID codes or To position for a field and resubmit the job.

RG107—INVALID NOT ENTRY IN COL 25, 32, OR 39, ASSUME N

Code: W—Warning
Specification Type: I
Explanation: The entry in column 25, 32, or 39 is not N or blank.
System Action: N is assumed.
User Response: If this assumption was wrong, leave column 25, 32, or 39 blank and re-submit the job.

RG108—INVALID C/Z/D ENTRY IN COL 26, 33, OR 40, ASSUME C

Code: W—Warning
Specification Type: I
Explanation: The entry in column 26, 33, or 40 is not C, Z, or D.
System Action: C is assumed.
User Response: If this assumption was wrong, make the proper entry in column 26, 33, or 40 and resubmit the job.

RG109—INVALID STACKER SELECT ENTRY IN COL 42 OR NOT ALLOWED WITH DEVICE, ASSUME BLANK

Code: W—Warning
Specification Type: I
Explanation: Column 42 is not used; it must be left blank.
System Action: Blank is assumed.
User Response: To avoid this message the next time this job is run, leave column 42 blank.

RG111—INVALID ENTRY IN COL 43, ASSUME BLANK

Code: W—Warning
Specification Type: I
Explanation: The entry in column 43 is not P, B, or blank.
System Action: Blank is assumed.
User Response: If the assumption was wrong, make the proper entry in column 43 and resubmit the job.

RG112—INVALID OR BLANK FROM OR TO ENTRY IN COL 44-51, ASSUME 1 FOR BOTH ENTRIES

Code: T—Terminal
Specification Type: I
Explanation: Columns 44-47 and/or 48-51 do not contain an entry from 1 to 4096.
System Action: 1 is assumed for columns 44-47 or columns 48-51, or for both; but the job is terminated.
User Response: Make the proper From or To entry in columns 44-47 and/or 48-51 and resubmit the job.

RG113—FROM ENTRY IN COL 44-47 EXCEEDS TO ENTRY IN COL 48-51, ASSUME TO ENTRY EQUAL TO FROM ENTRY

Code: T—Terminal
Specification Type: I
Explanation: From entry specified in columns 44-47 is larger than the To entry specified in columns 48-51.
System Action: To entry is assumed to be equal to the From entry, but the job is terminated.
User Response: Make the From entry (columns 44-47) equal to or less than the To entry (columns 48-51). Resubmit the job.

RG114—LENGTH OF NUMERIC FIELD IN COL 44-51 EXCEEDS 15, ASSUME 15

Code: T—Terminal
Specification Type: I
Explanation: Length specified in columns 44-51 for numeric field is too large.
System Action: Length of 15 is assumed, but the job is terminated.
User Response: Make the length (columns 44-51) 15 or less. Resubmit the job.

RG115—ALPHAMERIC FIELD SPECIFIED AS PACKED OR BINARY, ASSUME NUMERIC FIELD

Code: T—Terminal
Specification Type: I
Explanation: Column 43 must be blank for alphameric fields.
System Action: The field is assumed to be numeric, but the job is terminated.
User Response: Leave column 43 blank for alphameric fields or make an entry (0-9) in column 52 for numeric fields, and resubmit the job.

RG116—INVALID DECIMAL POSITION ENTRY IN COL 52; ASSUME 0

Code: T—Terminal
Specification Type: I
Explanation: Decimal position entry in column 52 is not 0-9 or blank.
System Action: Zero is assumed, but the job is terminated.
User Response: Make the proper decimal position entry in column 52 and resubmit the job.

RG117—DECIMAL POSITION ENTRY IN COL 52 INVALID FOR ARRAY; ASSUME BLANK

Code: W—Warning
Specification Type: I
Explanation: No decimal position entry can be specified in column 52 for an array.
System Action: Blank is assumed.
User Response: Leave column 52 blank for an array. Decimal positions for arrays must be specified in your extension specifications. Resubmit the job.

RG118—FIELD NAME IN COL 53-58 MISSING OR INVALID

Code: T—Terminal
Specification Type: I
Explanation: The field name entry in columns 53-58 is missing or is not specified properly.
System Action: The job is terminated.
User Response: Make a valid field name entry starting in column 53. Resubmit the job.

RG119—INVALID CONTROL LEVEL INDICATOR IN COL 59-60; ASSUME BLANK

Code: T—Terminal
Specification Type: I
Explanation: The control level entry in columns 59-60 is neither L1-L9 nor blank.
System Action: Blank is assumed, but the job is terminated.
User Response: Make the proper control level entry in columns 59-60 and resubmit the job.

RG120—INVALID MATCHING FIELD ENTRY IN COL 61-62; ASSUME M1

Code: T—Terminal
Specification Type: I
Explanation: The matching field entry in columns 61-62 is not M1-M9 or blank.
System Action: M1 is assumed, but the job is terminated.
User Response: Make the proper matching fields entry in columns 61-62 and resubmit the job.

RG121—FROM FILE CANNOT HAVE AN E IN COLUMN 17 OF FILE DESCRIPTION SPECIFICATION WHEN TO FILE IS A DEMAND FILE

Code: T—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 which is used to process a demand file.
System Action: Job is terminated.
User Response: Leave column 17 blank and resubmit the job.

RG 122—FIELD WAS PREVIOUSLY DEFINED WITH DIFFERENT LENGTH OR DECIMAL POSITIONS, ASSUME FIRST DEFINITION—OR FIELD IS NOW DEFINED AS A LOOK AHEAD FIELD

Code: W—Warning
Specification Type: I or C
Explanation: A conflicting length or number of decimal positions has been detected, or field is now used as a look ahead field.
System Action: The length or decimal positions of the first entry are assumed, or second field is a look ahead field.
User Response: If the assumption was wrong, define the field correctly and resubmit the job.

RG123—INVALID ENTRY IN COL 7-8

Code: T—Terminal
Specification Type: C
Explanation: The control level entry in columns 7-8 is not AN, OR, L0-L9, LR, SR, or blank.
System Action: The job is terminated.
User Response: Make the proper control level entry in columns 7-8 and resubmit the job.

RG124—INVALID NOT ENTRY IN COL 9, 12, OR 15; ASSUME N

Code: W—Warning
Specification Type: C
Explanation: The entry in column 9, 12, or 15 is not N or blank.
System Action: N is assumed.
User Response: If this assumption was wrong, leave column 9, 12, or 15 blank and resubmit the job.

RG125—INVALID FIELD NAME OR CONSTANT FOR FACTOR 1 IN COL 18-27

Code: T—Terminal
Specification Type: C
Explanation: The field name or constant in columns 18-27 is not specified properly. Both must begin in column 18.
System Action: The job is terminated.
User Response: Make the proper field name or constant entry in columns 18-27. Resubmit the job.

RG126—LENGTH OF TABLE/ARRAY EXCEEDS MAXIMUM CORE STORAGE

Code: T—Terminal
Specification Type: E
Explanation: The number of entries per TABLE/ARRAY (columns 36-39) multiplied by the length of entry (columns 40-42) exceeds maximum storage.
System Action: Job is terminated.
User Response: Reduce the number of entries or the length of the entries.

RG127—ENTRY IN COL 49-51 INVALID WITH NO RESULT FIELD, ASSUME 49-51 BLANK

Code: W—Warning
Specification Type: C
Explanation: This calculation specification contains a field length entry (columns 49-51) but no result field entry (columns 43-48).
System Action: Blank in columns 49-51 is assumed.
User Response: If a result field is being described, place the proper entry in columns 43-48 and resubmit the job.

RG128—INVALID OPERATION CODE ENTRY IN COL 28-32

Code: T—Terminal
Specification Type: C
Explanation: Operation code is not specified properly.
System Action: The job is terminated.
User Response: Enter the proper RPG II operation code in columns 28-32, and resubmit the job.

RG129—FACTOR 2 FIELD NAME IN COL 33-42 EXCEEDS SIX CHARACTERS

Code: T—Terminal
Specification Type: C
Explanation: The field name or label specified in Factor 2 is too large.
System Action: The job is terminated.
User Response: Make the field name or label in Factor 2 (columns 33-42) six characters or less. Resubmit the job.

RG130—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 ADDRROUT FILE.

Code: T—Terminal
Specification Type: E
Explanation: The file types specified on the Extension specification are not used properly.
System Action: Job is terminated.
User Response: Make To file a limits file if From file is a record address file, or make To file a random access file if From file is an ADDRROUT file. Resubmit job.

RG131—FACTOR 2 IN COL 33-42 INVALID

Code: T—Terminal
Specification Type: C
Explanation: The field name or constant in columns 33-42 is not specified properly. Entry must start in column 33.
System Action: The job is terminated.
User Response: Make the proper field name or constant entry in columns 33-42. Resubmit the job.

RG132—FACTOR 2 MUST BE A FILENAME

Code: T—Terminal
Specification Type: C
Explanation: For this operation the entry in Factor 2 must be a filename.
System Action: The job is terminated.
User Response: Make the proper filename entry in Factor 2 (columns 33-42) for this operation. Resubmit the job.

RG133—NUMERIC FIELD LENGTH EXCEEDS 15; ASSUME 15

Code: T—Terminal
Specification Type: C
Explanation: Length specified in columns 49-51 for numeric field is too large.
System Action: Length of 15 is assumed, but the job is terminated.
User Response: Make the length (columns 49-51) 15 or less. Resubmit the job.

RG134—ALPHAMERIC FIELD LENGTH EXCEEDS 256; ASSUME 256

Code: T—Terminal
Specification Type: C
Explanation: Length specified in columns 49-51 for an alphameric field is too large.
System Action: Length of 256 is assumed, but the job is terminated.
User Response: Make the length (columns 49-51) 256 or less. Resubmit the job.

RG135—INVALID RESULT FIELD ENTRY IN COL 43-53

Code: T—Terminal
Specification Type: C
Explanation: The result field entry in columns 43-53 is not specified properly.
System Action: The job is terminated.
User Response: Make the proper result field entries, and resubmit the job.

RG137—INVALID RESULT FIELD LENGTH IN COL 49-51; ASSUME 15 FOR NUMERIC OR 256 FOR ALPHAMERIC FIELD

Code: T—Terminal
Specification Type: C
Explanation: The field length entry in columns 49-51 is not specified properly.
System Action: 15 is assumed for numeric fields; 256 is assumed for alphameric fields. The job is terminated.
User Response: Enter 15 or less in columns 49-51 for numeric fields, 256 or less for alphameric fields. Resubmit the job.

**RG138—DECIMAL POSITION ENTRY IN COL 52
INVALID WITH NO FIELD LENGTH ENTRY IN COL
49-51; ASSUME BLANK**

Code: T—Terminal
Specification Type: C
Explanation: Column 52 must be blank when columns 49-51 are blank.
System Action: Blank in column 52 is assumed, but the job is terminated.
User Response: Leave column 52 (decimal position) blank when columns 49-51 (field length) are blank. Resubmit the job.

**RG139—INVALID DECIMAL POSITION ENTRY IN COL
52; ASSUME 0**

Code: T—Terminal
Specification Type: C
Explanation: The decimal position entry is not a number from 0 to 9 or blank.
System Action: Zero is assumed, but the job is terminated.
User Response: Make the proper decimal position entry (0-9 or blank) in column 52 and resubmit the job.

**RG140—INVALID HALF ADJUST ENTRY IN COL 53;
ASSUME H**

Code: W—Warning
Specification Type: C
Explanation: The half adjust entry in column 53 is neither H nor blank.
System Action: H is assumed.
User Response: If this assumption was wrong, leave column 53 blank and resubmit the job.

**RG141—DEBUG CALCULATION OPERATION USED,
BUT DEBUG OPTION NOT SPECIFIED IN THE
CONTROL CARD**

Code: W—Warning
Specification Type: C
Explanation: The DEBUG operation code was used in your calculation specifications, but you had not specified the DEBUG option (1 in column 15) in your control card specifications.
System Action: DEBUG operations are not executed.
User Response: Specify the DEBUG option (1 in column 15) in your control card specifications if you have DEBUG statements to be executed, and resubmit the job.

**RG142—FILE AND RECORD IDENTIFICATION ENTRIES
IN COL 7-31 AND FIELD DESCRIPTION ENTRIES IN
COL 32-74 ON SAME LINE**

Code: T—Terminal
Specification Type: O
Explanation: Your field description entries in columns 23-74 are not specified one line lower than the file and record identification entries in columns 7-31.
System Action: Blanks are assumed for columns 7-31 and the job is terminated.
User Response: Place the field description entries (columns 32-74) one line lower than the file and record identification entries (columns 7-31). Resubmit the job.

RG143—INVALID LINE TYPE ENTRY IN COL 15

Code: T—Terminal
Specification Type: O
Explanation: The line type entry in column 15 is not H, D, T, or E.
System Action: H is assumed; the job is terminated.
User Response: Make the proper line type entry in column 15 and resubmit this job.

RG144—AND OR OR LINE NOT PRECEDED BY RECORD IDENTIFICATION

Code: T—Terminal
Specification Type: O
Explanation: An AND or OR line is not preceded by record identification entries in columns 15–31.
System Action: The job is terminated.
User Response: Make sure that record identification entries in columns 15–31 precede any AND or OR lines. Resubmit the job.

RG145—INVALID ENTRIES IN COL 17–22 FOR AND LINE, ASSUME BLANK

Code: W—Warning
Specification Type: O
Explanation: Columns 17–22 of an AND line contain space/skip entries; they should be blank.
System Action: Blanks are assumed.
User Response: To avoid this message when the job is run again, remove all space/skip entries (columns 17–22) from an AND line.

RG146—INVALID FILENAME OR ENTRY IN COL 15 MISSING ON FIRST OUTPUT SPECIFICATION

Code: T—Terminal
Specification Type: O
Explanation: Either columns 7–14 contain an invalid filename or no line type entry was specified in column 15 of the specification line.
System Action: The job is terminated.
User Response: Check to make sure the proper filename entry is made in columns 7–14 and that the proper line type entry is made in column 15. Resubmit the job.

RG147—INVALID NOT ENTRY IN COL 23, 26, OR 29; ASSUME N

Code: W—Warning
Specification Type: O
Explanation: The entry in column 23, 26, or 29 is neither N nor blank.
System Action: N is assumed.
User Response: If this assumption was wrong, make the proper entry in column 23, 26, or 29 and resubmit the job.

RG148—INVALID FIELD NAME IN COL 32–37

Code: T—Terminal
Specification Type: O
Explanation: The field name entry in columns 32–37 is not specified properly or was not defined previously in input or calculation specifications.
System Action: The job is terminated.
User Response: Make the proper field name entry starting in column 32 and resubmit the job.

RG149—INVALID OR MISSING CONSTANT

Code: T—Terminal
Specification Type: O
Explanation: The constant in columns 45–70 is not specified properly.
System Action: The job is terminated.
User Response: Make the proper entry in columns 45–70 and resubmit the job.

RG150—INVALID BLANK AFTER ENTRY IN COL 39; ASSUME BLANK

Code: T—Terminal
Specification Type: O
Explanation: The blank after entry in column 39 is neither B nor blank.
System Action: Blank is assumed, but the job is terminated.
User Response: Make the proper entry in column 39 and resubmit the job.

RG151—MISSING OR INCORRECTLY SPECIFIED END POSITION IN COL 40-43; ASSUME END POSITION 1

Code: T—Terminal
Specification Type: O
Explanation: The end positions entry in columns 40-43 is either missing or is not specified properly.
System Action: The job is terminated.
User Response: Make the proper numeric entry in columns 40-43; the entry must end in column 43. Resubmit the job.

RG152—INVALID ENTRY IN COL 44; ASSUME BLANK

Code: W—Warning
Specification Type: O
Explanation: The entry in column 44 is not P, B, or blank.
System Action: Blank is assumed.
User Response: If the assumption was wrong, make the proper entry in column 44 and resubmit the job.

RG154—ENTRIES IN COL 7-22 INVALID FOR A FIELD DESCRIPTION SPECIFICATION, ASSUME BLANK

Code: T—Terminal
Specification Type: O
Explanation: The file and record identification entries in columns 7-22 are not specified one line above the first related field description entries.
System Action: The job is terminated.
User Response: Place your file and record identification entries (columns 7-22) one line above the field description entries (columns 32-74). Resubmit the job.

RG155—INVALID STERLING SIGN POSITION ENTRY IN COL 71-74; ASSUME BLANK

Code: T—Terminal
Specification Type: I or O
Explanation: Sterling is not allowed on this system.
System Action: Blank is assumed; the job is terminated.
User Response: Replace sterling entry in columns 71-75 with blanks and resubmit the job.

RG158—TABLE NAME INVALID FOR A FIELD NAME ENTRY IN COL 53-58

Code: T—Terminal
Specification Type: I
Explanation: The field name entry in columns 53-58 refers to a table.
System Action: The job is terminated.
User Response: Place the proper field name entry in columns 53-58; the entry must not be a table name. Resubmit the job.

RG159—MISSING RECORD IDENTIFYING INDICATOR IN COL 19-20

Code: W—Warning
Specification Type: I
Explanation: No record identifying indicator is specified in columns 19-20.
System Action: No action taken.
User Response: Check your input specifications to determine whether or not a record identifying indicator should be entered in columns 19-20. If so, make the proper entry and resubmit the job.

RG160—FILE NAMED IN COL 7-14 NOT SPECIFIED AS AN INPUT, COMBINED, UPDATE-PRIMARY, SECONDARY, DEMAND, OR CHAINED FILE

Code: T—Terminal
Specification Type: I
Explanation: The file named in columns 7-14 was not previously defined in file description specifications as an input, combined, or update file with a designation of primary, secondary, demand, or chained.
System Action: The job is terminated.
User Response: Make sure the file named in columns 7-14 is properly defined in file description specifications. Resubmit the job.

RG161—AND OR OR LINE INVALID WITH LOOK AHEAD RECORDS OR RLABL

Code: T—Terminal
Specification Type: I, C
Explanation: An AND or OR line was used with look ahead fields or RLABL.
System Action: The job is terminated.
User Response: Make sure that AND or OR lines are not specified for look ahead fields (** in columns 19-20) or for RLABL. Resubmit the job.

RG162—RECORD IDENTIFYING INDICATOR IN COL 19-20 INVALID FOR AN AND LINE

Code: W—Warning
Specification Type: I
Explanation: A record identifying indicator is in columns 19-20 of an AND line.
System Action: Blanks are assumed.
User Response: To avoid this message next time this job is run, leave columns 19-20 of the AND line blank.

RG163—ENTRIES IN COL 17-18 AND 21-42 INVALID FOR LOOK AHEAD RECORD.ENTRIES IN 59-74 INVALID FOR LOOK AHEAD FIELD

Code: T—Terminal
Specification Type: I
Explanation: Columns 17-18 and 21-42 must be blank for look ahead records, columns 59-74 must be blank for look ahead fields.
System Action: The job is terminated.
User Response: Leave columns 17-18 and 21-42 blank for look ahead records; leave columns 59-74 blank for look ahead fields. Resubmit the job.

RG164—STACKER SELECT ENTRY IN COL 42 INVALID FOR DEVICE SPECIFIED; ASSUME BLANK

Code: W—Warning
Specification Type: I
Explanation: Stacker select entry is invalid for devices on this system.
System Action: No action taken.
User Response: Replace stacker select entry in column 42 with blanks.

RG165—INDICATORS IN COL 65-70 INVALID FOR A TABLE/ARRAY

Code: T—Terminal
Specification Type: I
Explanation: Field indicators cannot be used if columns 53-58 contain a table/array name.
System Action: The job is terminated.
User Response: Use the field indicators to test numeric fields.

RG166—PLUS OR MINUS INDICATOR IN COL 65-68 INVALID FOR ALPHAMERIC FIELD

Code: T—Terminal
Specification Type: I
Explanation: A Plus or Minus indicator in columns 65-68 cannot be used to test an alphameric field.
System Action: Blank is assumed; the job is terminated.
User Response: Use Plus or Minus indicators only to test numeric fields. An alphameric field can only be tested for a blank condition (entry in columns 69-70). Resubmit the job.

RG167—RECORD ID POSITION 21-38 OR TO ENTRY IN COL 48-51 EXCEEDS RECORD LENGTH, ASSUME RECORD LENGTH

Code: T—Terminal
Specification Type: I
Explanation: Field location entries (columns 21-38 and 48-51) exceed record length specified in file description specifications.
System Action: Record length is assumed; the job is terminated.
User Response: Make the field location entries (columns 21-38 and 48-51) equal to or less than the record length specified on file description specifications. Resubmit the job.

RG168—FIELD NAME IN COL 53-58 IS A RESERVED WORD OTHER THAN PAGE

Code: T—Terminal
Specification Type: I
Explanation: The field name entry in columns 53-58 is a reserved word other than PAGE.
System Action: The job is terminated.
User Response: Make the proper field name entry in columns 53-58 (PAGE is the only RPG II reserved word that can be entered in these columns). Resubmit the job.

RG169—CONTROL OR MATCHING FIELDS INVALID FOR ARRAY OR TRAILER RECORD

Code: T—Terminal
Specification Type: I
Explanation: Control or matching fields must not be specified for arrays or trailer records.
System Action: The job is terminated.
User Response: Make sure no control or matching fields are specified for array or trailer records. Resubmit the job.

RG170—MATCHING OR CONTROL FIELDS INVALID WITH DEMAND OR CHAIN FILES

Code: T—Terminal
Specification Type: I
Explanation: Matching or control fields cannot be specified for demand or chain files.
System Action: The job is terminated.
User Response: Make sure that matching or control fields are not specified for demand or chain files. Resubmit the job.

RG171—LOOK AHEAD RECORDS INVALID WITH DEMAND FILES, CHAIN FILES, FILES CONTAINING SPREAD CARDS, OR WITH THIS DEVICE

Code: T—Terminal
Specification Type: I
Explanation: Look ahead records cannot be specified for demand files, chained files, files containing spread cards, or with the ledger card device.
System Action: The job is terminated.
User Response: Make sure that look ahead records are not specified for demand files, chained files, a file containing spread cards, or with the ledger card device. Resubmit the job.

RG172—INCORRECT SEQUENCE OF INPUT SPECIFICATIONS

Code: T—Terminal
Specification Type: I
Explanation: All records from one input, update, or combined files are not specified consecutively.
System Action: The job is terminated.
User Response: Specify all records from one input, update, or combined file consecutively before starting to describe records from a different file.

RG173—NO FIELDS SPECIFIED FOR LOOK AHEAD RECORD

Code: T—Terminal
Specification Type: I
Explanation: A look ahead record is specified (** in columns 19-20), but no look ahead fields are defined (columns 53-58).
System Action: The job is terminated.
User Response: Make the proper look ahead field specifications in columns 53-58 for a look ahead record. Resubmit the job.

RG178—STERLING OR BINARY INVALID WITH CONTROL OR MATCHING FIELDS

Code: T—Terminal
Specification Type: I
Explanation: Binary fields have been used as control or matching fields.
System Action: The job is terminated.
User Response: Do not specify a binary field as a control or matching field.

RG180—ARRAY LENGTH EXCEEDS LENGTH SPECIFIED IN COL 36-42 IN EXTENSION SPECIFICATIONS OR NOT A MULTIPLE OF THE ENTRY LENGTH IN COL 40-42 IN EXTENSION SPECIFICATIONS

Code: T—Terminal
Specification Type: I
Explanation: The array length either exceeds the length specified in columns 36-42 of your extension specifications, or is not a multiple of the length entry in columns 40-42 of the extension specification, or both.
System Action: The job is terminated.
User Response: Make the array length equal to or less than the length specified in columns 36-42 of extension specifications. The length must also be a multiple of the length of an array element (columns 40-42 of extension specifications). Resubmit the job.

RG181—INCONSISTENT FIELD LENGTHS FOR CONTROL OR MATCHING FIELDS OF ONE LEVEL. ASSUME FIRST VALID LENGTH

Code: T—Terminal
Specification Type: I
Explanation: All control or matching fields of one level were not assigned the same field length.
System Action: The job is terminated.
User Response: Assign the same field length to all control or matching fields of the same level. Resubmit the job.

RG182—INVALID SPLIT CONTROL FIELD SPECIFICATION, ASSUME PREVIOUS TOTAL LENGTH FOR THIS LEVEL

Code: T—Terminal
Specification Type: I
Explanation: Specifications for split control fields of the same level are not specified on successive lines.
System Action: The job is terminated.
User Response: Place the specifications for split control fields of the same level on successive lines. Resubmit the job.

RG183—CONTROL OR MATCHING FIELDS OF A LEVEL SPECIFIED AS BOTH ALPHAMERIC AND NUMERIC, ASSUME NUMERIC

Code: W—Warning
Specification Type: I
Explanation: All control and matching fields assigned the same level are not the same type (alphameric or numeric).
System Action: Numeric is assumed for all fields assigned the same control or matching level. If any of the fields specified as alphameric is greater than 15 characters, only a portion of the field will be used.
User Response: If this assumption is wrong, make the proper entry and resubmit the job.

RG184—ALL OF THE VALID MATCH LEVELS WERE NOT REFERENCED IN LAST RECORD GROUP

Code: T—Terminal
Specification Type: I
Explanation: The same number of match levels are not specified to all record types in a file.
System Action: The job is terminated.
User Response: Make sure that all record types in a file either have no match levels or have the same number of match levels specified. Resubmit the job.

RG186—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

Code: T—Terminal
Specification Type: I
Explanation: All match or control fields without field record relation entries (columns 63–64) do not precede those fields that do have field record relation entries.
System Action: The job is terminated.
User Response: Place all match or control fields without field record relation entries before those match or control fields with field record relation entries. Resubmit the job.

RG187—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

Code: T—Terminal
Specification Type: I
Explanation: When field record relation is used, all match and control fields assigned the same indicator (columns 63–64) must be grouped together.
System Action: A new group is assumed, but the job is terminated.
User Response: Group all match and control fields with the same field record relation indicator together. Resubmit the job.

RG188—FIELD RECORD RELATION INDICATOR USED IMPROPERLY WITH MATCH OR CONTROL FIELDS

Code: T—Terminal
Specification Type: I
Explanation: When used with match or control fields, the field record relation indicator in columns 63–64 does not match a record identifying indicator used for this file.
System Action: The job is terminated.
User Response: Make the field record relation indicator in columns 63–64 match a record identifying indicator for this file. Resubmit the job.

RG189—INVALID SEQUENCE FOR CALCULATION SPECIFICATIONS OR SR NOT SPECIFIED IN COLUMNS 7–8 WITH BEGSR OR ENDSR

Code: T—Terminal
Specification Type: C
Explanation: Calculation specifications are not specified in this order: detail, total, and subroutines.
System Action: The job is terminated.
User Response: Place calculation specifications in this order: detail, total, and subroutines. Resubmit the job.

RG190—INVALID SEQUENCE FOR BEGSR AND ENDSR OPERATION CODES

Code: T—Terminal
Specification Type: C
Explanation: BEGSR operation code does not precede ENDSR operation code.
System Action: The job is terminated.
User Response: Place the BEGSR specification before the ENDSR specification in a subroutine. Resubmit the job.

RG191—A SUBROUTINE MUST NOT CALL ITSELF

Code: T—Terminal
Specification Type: C
Explanation: An EXSR specification within a subroutine must not call the subroutine it is in.
System Action: The job is terminated.
User Response: If you wish to branch to another point within the same subroutine use a GOTO and TAG operation. Resubmit the job.

RG192—BRANCHING BETWEEN SUBROUTINE AND OTHER CALCULATIONS INVALID

Code: T—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.
System Action: The job is terminated.
User Response: When using subroutines, make sure branching between a subroutine and other calculations is not specified. Make the necessary changes and resubmit the job.

RG193—BRANCHING BETWEEN DETAIL, TOTAL AND LR CALCULATIONS INVALID

Code: T—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.
System Action: The job is terminated.
User Response: Remove any operations that attempt to branch from detail to total calculations or vice versa. Resubmit the job.

RG194—SETOF OPERATION INVALID FOR LR INDICATOR

Code: T—Terminal
Specification Type: C
Explanation: The LR indicator cannot be turned off by the SETOF operation code.
System Action: The job is terminated.
User Response: Remove the SETOF LR specification and resubmit the job.

RG195—LENGTH OF SEARCH WORD NOT EQUAL TO LENGTH OF ELEMENT IN TABLE OR ARRAY

Code: T—Terminal
Specification Type: C
Explanation: Length of search word (Factor 1) is not equal to length of element in table or array being searched.
System Action: The job is terminated.
User Response: Make the length of the search word (Factor 1) equal to the length of the element in the table or array being searched. Resubmit the job.

RG196—FACTOR 2 OR RESULT FIELD INVALID FOR LOKUP OPERATION

Code: T—Terminal
Specification Type: C
Explanation: Either Factor 2 or Result Field is invalid for this LOKUP operation.
System Action: The job is terminated.
User Response: Specify LOKUP operation with table or array name in Factor 2 or in Result Field. Resubmit the job.

RG197—SEARCH TABLE HAS MORE ENTRIES THAN ITS RELATED TABLE

Code: W—Warning
Specification Type: C
Explanation: The search table (Factor 2) contains more entries than its related table.
System Action: No action taken.
User Response: To avoid this message the next time this job is run, make the number of entries in the table being searched (Factor 2) equal or less than the number of entries in the related table (result field).

**RG198—INDICATOR ENTERED IN COL 54-57
INVALID WITH LOKUP ON AN UNSEQUENCED
TABLE**

Code: T—Terminal
Specification Type: C
Explanation: You must not specify a search for high or low in a LOKUP operation on an unsequenced table.
System Action: The job is terminated.
User Response: Specify the LOKUP operation on an unsequenced table for an equal condition only (indicator in columns 58-59). Resubmit the job.

**RG199—TEST FOR BOTH HIGH AND LOW INVALID
FOR LOKUP OPERATION**

Code: T—Terminal
Specification Type: C
Explanation: You must not specify a search for both high and low in the same LOKUP operation.
System Action: The job is terminated.
User Response: Specify the LOKUP for either high or low, but not both. Resubmit the job.

**RG200—RESULTING INDICATORS IN COL 54-59
REQUIRED OR NOT ALLOWED FOR OPERATION
SPECIFIED**

Code: T—Terminal
Specification Type: C
Explanation: The resulting indicator entry in columns 54-59 is not specified properly.
System Action: The job is terminated.
User Response: 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, KQ), resubmit the job.

**RG201—HALF ADJUST ENTRY IN COL 53 FOR
DIVISION OPERATION FOLLOWED BY A MVR
OPERATION: ASSUME NO HALF ADJUST**

Code: W—Warning
Specification Type: C
Explanation: When an MVR operation follows a DIV operation, the DIV operation must not be half adjusted.
System Action: No half adjusting is done.
User Response: To avoid this message the next time this job is run, leave column 53 (Half Adjust) blank.

**RG202—MVR OPERATION CODE DOES NOT FOLLOW
DIV OPERATION**

Code: T—Terminal
Specification Type: C
Explanation: The MVR operation must immediately follow a DIV operation.
System Action: The job is terminated.
User Response: Place an MVR operation immediately after a DIV operation or remove the MVR operation and resubmit the job.

**RG204—HALF ADJUST ENTRY IN COL 53 INVALID
FOR OPERATION OR NUMBER OF DECIMAL
POSITIONS SPECIFIED; ASSUME BLANK**

Code: W—Warning
Specification Type: C
Explanation: Half adjusting (H in column 53) cannot be done for this operation or half adjusting is invalid for the number of decimal positions specified.
System Action: Column 53 is assumed to be blank; therefore no half adjusting is done.
User Response: To avoid this message the next time the job is run, leave column 53 blank for this operation.

RG205—COMP, TESTZ, OR MVR INVALID FOR AN ARRAY

Code: T—Terminal
Specification Type: C
Explanation: COMP, TESTZ, and MVR must not be specified for an array.
System Action: The job is terminated.
User Response: Delete any COMP, TESTZ and MVR operations specified for an array. Resubmit the job.

RG206—INVALID USE OF COMP OR LOKUP

Code: T—Terminal
Specification Type: C
Explanation: COMP or LOKUP operation specified improperly.
System Action: The job is terminated.
User Response: 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. Resubmit the job.

RG207—FIELD TYPE, ALPHAMERIC OR NUMERIC, INVALID FOR OPERATION SPECIFIED

Code: T—Terminal
Specification Type: C
Explanation: This operation requires a different field type (alphameric or numeric).
System Action: The job is terminated.
User Response: Make the proper field type entry (alphameric or numeric) and resubmit the job.

RG208—FORCE OPERATION INVALID AT TOTAL TIME

Code: T—Terminal
Specification Type: C
Explanation: FORCE operation must be specified at detail time only.
System Action: The job is terminated.
User Response: Specify the FORCE operation at detail time and resubmit the job.

RG209—FILE TYPE INVALID FOR USE WITH THIS OPERATION CODE

Code: T—Terminal
Specification Type: C
Explanation: DEBUG and EXCPT must be used with an output file; FORCE must be used with an input or combined primary or secondary file.
System Action: The job is terminated.
User Response: Specify the proper file type or a different operation code, and resubmit the job.

RG211—DEBUG SPECIFIED FOR MORE THAN ONE OUTPUT FILE

Code: T—Terminal
Specification Type: C
Explanation: The filename entered in Factor 2 is not the same for all DEBUG operations.
System Action: The job is terminated.
User Response: Place the same filename in Factor 2 for all DEBUG operations and resubmit the job.

RG212—EXCPT OPERATION CODE SPECIFIED BUT NO EXCPT OUTPUT RECORDS SPECIFIED

Code: W—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).
System Action: No action taken.
User Response: To avoid this message, either delete the EXCPT operation code or specify the proper exception records in output specifications.

RG213—EXSR DOES NOT REFERENCE A BEGSR NAME

Code: T—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.
System Action: The job is terminated.
User Response: Make the label in Factor 2 of the EXSR operation the same as the label in Factor 1 of a BEGSR operation. Resubmit the job.

RG214—GOTO BRANCHES TO A BEGSR NAME

Code: T—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 operation.
System Action: The job is terminated.
User Response: Make the label in Factor 2 of a GOTO operation the same as the label in Factor 1 of a TAG operation. Resubmit the job.

RG215—FACTOR 1 ENTRY IN COL 18-27 MISSING

Code: T—Terminal
Specification Type: C
Explanation: No entry specified in Factor 1 for this operation.
System Action: The job is terminated.
User Response: Make the proper entry in Factor 1 and resubmit the job.

RG216—FACTOR 1 ENTRY IN COL 18-27 INVALID FOR THIS OPERATION

Code: T—Terminal
Specification Type: C
Explanation: An entry must not be specified in Factor 1 for this operation.
System Action: The job is terminated.
User Response: Make Factor 1 blank for this operation and resubmit the job.

RG217—FACTOR 2 ENTRY IN COL 33-42 MISSING

Code: T—Terminal
Specification Type: C
Explanation: No entry specified in Factor 2 for this operation.
System Action: The job is terminated.
User Response: Make the proper entry in Factor 2 and resubmit the job.

RG218—FACTOR 2 ENTRY IN COL 33-42 INVALID FOR THIS OPERATION

Code: T—Terminal
Specification Type: C
Explanation: An entry must not be specified in Factor 2 for this operation.
System Action: The job is terminated.
User Response: Make Factor 2 blank for this operation, and resubmit the job.

RG219—RESULT FIELD ENTRY IN COL 43-48 MISSING

Code: T—Terminal
Specification Type: C
Explanation: No entry specified in the Result Field for this operation.
System Action: The job is terminated.
User Response: Make the proper entry in the Result Field for this operation and resubmit the job.

RG220—RESULT FIELD ENTRY IN COL 43-48 INVALID FOR THIS OPERATION

Code: T—Terminal
Specification Type: C
Explanation: An entry must not be specified in Result Field for this operation.
System Action: The job is terminated.
User Response: Make the Result Field blank for this operation and resubmit the job.

RG221—RESULT FIELD LENGTH MAY NOT BE LARGE ENOUGH

Code: W—Warning
Specification Type: C
Explanation: The result field specified may not be large enough to hold the largest possible result obtained in the calculation operations specified.
System Action: No action taken.
User Response: Check to make sure the result field specified is large enough. If it is not, make it larger and resubmit the job.

RG223—SUBROUTINE SPECIFICATIONS ARE THE ONLY CALCULATION SPECIFICATIONS

Code: T—Terminal
Specification Type: C
Explanation: Subroutine specifications do not follow detail and total calculations.
System Action: The job is terminated.
User Response: Place detail and total calculations before subroutine operations. Re-submit the job.

RG224—A ZERO CONSTANT INVALID AS DIVISOR IN COL 33–42

Code: T—Terminal
Specification Type: C
Explanation: The constant entered in Factor 2 of a DIV operation must not be zero.
System Action: The job is terminated.
User Response: Place the proper divisor in Factor 2 of the DIV operation and resubmit the job.

RG225—CONDITIONING INDICATORS IN COL 9–17 INVALID WITH TAG, BEGSR, ENDSR, OR RLABL OPERATION

Code: T—Terminal
Specification Type: C
Explanation: Conditioning indicators must not be specified in columns 9–17 for TAG, BEGSR, ENDSR, or RLABL operations.
System Action: The job is terminated.
User Response: Make columns 9–17 blank for TAG, BEGSR, ENDSR, or RLABL operations. Resubmit the job.

RG226—A RESERVED WORD OTHER THAN PAGE INVALID

Code: T—Terminal
Specification Type: C, I, O
Explanation: No reserved word other than PAGE can be specified as a result field. CONTD is a reserved word, for compatibility with other systems.
System Action: The job is terminated.
User Response: Make sure no reserved word other than PAGE is specified in columns 43–48 as the result field. Resubmit the job.

RG227—RESULT FIELD IN COL 43–48 IS A LOOK AHEAD FIELD OR CONSTANT

Code: T—Terminal
Specification Type: C
Explanation: The result field must not be a look ahead field or a constant.
System Action: The job is terminated.
User Response: Make the proper result field entry in columns 43–48 and resubmit the job.

RG228—INVALID INDEX

Code: T—Terminal
Specification Type: C
Explanation: Array index not specified properly. Index field name must contain a valid combination of characters. Index constant of field value must be a positive number which does not exceed the number of elements in the array and have zero decimal positions.
System Action: The job is terminated.
User Response: Make the proper array index entry and resubmit the job.

RG229—INDEXING INVALID FOR TABLES OR FIELDS

Code: T—Terminal
Specification Type: C
Explanation: Indexing must be specified for arrays only.
System Action: The job is terminated.
User Response: Remove specifications for indexing tables or fields. Resubmit the job.

RG231—GOTO DOES NOT BRANCH TO A TAG

Code: T—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.
System Action: The job is terminated.
User Response: Make the label in Factor 2 of the GOTO operation the same as the label in Factor 1 of a TAG or ENDSR operation. Resubmit the job.

RG232—THIS NAME WAS PREVIOUSLY USED ON A TAG, BEGSR, OR ENDSR

Code: T—Terminal
Specification Type: C
Explanation: The label in Factor 1 was previously specified in another TAG, BEGSR, or ENDSR operation.
System Action: The job is terminated.
User Response: Make the label in Factor 1 of each TAG, BEGSR, and ENDSR operation unique. Resubmit the job.

RG233—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

Code: T—Terminal
Specification Type: T
Explanation: The configuration entry in column 15 of your telecommunications specifications is not P, S, M or blank.
System Action: The job is terminated.
User Response: Make the proper entry (P, S, M, or blank) in column 15 and resubmit the job.

RG234—TRANSMITTER/RECEIVER, COLUMN 16, DOES NOT CONTAIN T OR R

Code: T—Terminal
Specification Type: T
Explanation: The type of station entry in column 16 is neither T nor R.
System Action: The job is terminated.
User Response: Enter T (for a transmitter station) or R (for a receiver station) and resubmit the job.

RG235—CONTROL/TRIBUTARY, COLUMN 17, CONTAINS A CHARACTER OTHER THAN T OR BLANK. IF THIS IS A SWITCHED OR POINT-TO-POINT NETWORK, COLUMN 15, ASSUME BLANK; IF MULTIPOINT, ASSUME T

Code: W—Warning
Specification Type: T
Explanation: The type of control entry in column 17 is neither T nor blank.
System Action: Blank is assumed if this is a switched network or a point-to-point leased line; T is assumed if this is a multipoint leased line.
User Response: To avoid this message when this job is run again, enter T in column 17 for tributary on multipoint network. Leave column 17 blank if switched line or point-to-point line is used.

RG236—ASCII/EBCDIC, COLUMN 18, IS NOT U, A, E, OR BLANK. ASSUME EBCDIC

Code: W—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.
System Action: EBCDIC is assumed.
User Response: If the assumption was wrong, make the proper entry and resubmit the job.

RG237—TRANSPARENT FEATURE, COLUMN 19, IS NOT Y, N, OR BLANK. ASSUME NO TRANSPARENCY

Code: W—Warning
Specification Type: T
Explanation: The entry in column 19 is not Y for transparency or N or blank for no transparency.
System Action: No transparency is assumed.
User Response: If the assumption was wrong, make the proper entry and resubmit the job.

RG238—AUTOCALL/AUTOANSWER, COLUMN 20, IS NOT E, S, M, A, B, OR BLANK. COLUMNS 21-31 WILL BE IGNORED

Code: T—Terminal
Specification Type: T
Explanation: The entry in column 20 is not E, S, M, A, B, or blank.
System Action: Entries in columns 21-31 are ignored; the job is terminated.
User Response: Make the proper entry in column 20 and resubmit the job.

RG239—ENTRY FOR DIAL NUMBER, COLUMNS 21-31, IS NOT VALID FOR THE AUTOCALL/AUTOANSWER ENTRY IN COLUMN 20

Code: T—Terminal
Specification Type: T
Explanation: The entry in columns 21-31 is not valid for the entry in column 20.
System Action: The job is terminated.
User Response: Enter dial number in columns 21-31 if the entry in column 20 is E; enter symbolic name in columns 21-31 if the entry in column 20 is S. Resubmit the job.

RG240—IDENTIFICATION TYPE FOR THIS STATION, COLUMN 32, IS NOT S, E, OR BLANK. COLUMNS 33-39 WILL NOT BE CHECKED

Code: W—Warning
Specification Type: T
Explanation: The station identification entry in column 32 is not S, E, or blank.
System Action: Columns 33-39 will not be checked for a valid entry.
User Response: If the assumption is not correct, make the identification entry in column 32 (S, E, or blank) that properly describes this station and resubmit the job.

RG241—IDENTIFICATION FOR THIS STATION, COLUMNS 33-39 CONTAINS AN INVALID ENTRY FOR THE ID TYPE INDICATED IN COLUMN 32

Code: T—Terminal
Specification Type: T
Explanation: The identification entry in columns 33-39 is invalid for the identification type specified in column 32.
System Action: The job is terminated.
User Response: Enter identification sequence in columns 33-39 if column 32 contains an E; enter symbolic name in columns 33-39 if column 32 contains an S. Resubmit the job.

RG242—IDENTIFICATION TYPE FOR THE REMOTE STATION. COLUMN 40, IS NOT S, E, OR BLANK. COLUMNS 41-47 WILL NOT BE CHECKED

Code: W—Warning
Specification Type: T
Explanation: The identification entry in column 40 is not S, E, or blank.
System Action: Columns 41-47 will not be checked for an entry.
User Response: Make the identification entry in column 40 (S, E, or blank) that properly describes the remote station. Resubmit the job.

RG243—IDENTIFICATION FOR REMOTE STATION, COLUMNS 41-47, CONTAINS AN INVALID ENTRY FOR THE ID TYPE GIVEN IN COLUMN 40

Code: T—Terminal
Specification Type: T
Explanation: The identification entry specified for a remote station in columns 41-47 is invalid for the identification type (column 40).
System Action: No action taken.
User Response: Be sure the entry in columns 41-47 is valid for the identification type (S, E, or blank) specified in column 40. Resubmit the job.

**RG244—INVALID REMOTE TERMINAL SPECIFIED,
COLUMNS 48-51**

Code: T—Terminal
Specification Type: T
Explanation: The entry in columns 48-51 is not a valid remote terminal.
System Action: The job is terminated.
User Response: Specify a valid remote terminal and resubmit the job.

**RG245—ITB, COLUMN 52, IS NOT I OR BLANK
ASSUME I**

Code: W—Warning
Specification Type: T
Explanation: The entry in column 52 is neither I nor blank.
System Action: I is assumed.
User Response: If the assumption was wrong, leave column 52 blank and resubmit the job.

**RG246—PERMANENT ERROR INDICATOR, COLUMNS
53-54, IS INVALID**

Code: T—Terminal
Specification Type: T
Explanation: The indicator specified in columns 53-54 is not 01-99, L1-L9, LR, or H1-H9.
System Action: The indicator is ignored and the job is terminated.
User Response: Make the proper entry in columns 53-54 and resubmit the job.

**RG247—WAIT TIME, COLUMNS 55-57, IS INVALID.
ASSUME SYSTEM CONVENTION FOR TIMEOUT, 180
SECONDS**

Code: W—Warning
Specification Type: T
Explanation: The wait time entry specified in columns 55-57 is not 1-999 or blank.
System Action: System convention for timeout, 180 seconds, will be assumed.
User Response: If the assumption was not acceptable, make the proper entry (1-999 or blank) and resubmit the job.

**RG248—RECORD AVAILABLE INDICATOR,
COLUMNS 58-59, IS INVALID**

Code: T—Terminal
Specification Type: T
Explanation: The record available indicator specified in columns 58-59 is not 01-99, L1-L9, LR, or H1-H9.
System Action: The indicator is ignored and the job is terminated.
User Response: Make the proper entry in columns 58-59 and resubmit the job.

**RG249—LAST FILE PROCESSED, COLUMN 60, IS NOT
L OR BLANK**

Code: T—Terminal
Specification Type: T
Explanation: The last record processed entry in column 60 is not L or blank.
System Action: The job is terminated.
User Response: Enter L in column 60 if the BSC input file must be processed last; blank if not. Resubmit the job.

**RG250—POLLING CHARACTERS, COLUMNS 61-62,
CONTAIN AN INVALID CHARACTER FOR THE CODE
TYPE ENTRY IN COLUMN 18**

Code: T—Terminal
Specification Type: T
Explanation: The polling characters specified in columns 61-62 are invalid, or are missing on a line configuration that requires them.
System Action: The job is terminated.
User Response: Make the proper entry in columns 61-62. (A list of the valid polling characters is included in the *IBM System/3 RPG II Telecommunications Reference Manual*, SC21-7507.) Resubmit the job.

RG251—ADDRESSING CHARACTERS, COLUMNS 63-64, ARE INVALID FOR THE CODE TYPE ENTRY IN COLUMN 18. THE ENTRY IS IGNORED

Code: T—Terminal
Specification Type: T
Explanation: The addressing characters in columns 63-64 are invalid for the code type specified in column 18, or are missing on a line configuration that requires them.
System Action: The job is terminated.
User Response: Make the proper entry in columns 63-64. (A list of the valid addressing characters is included in the *IBM System/3 RPG II Telecommunications Reference Manual*, SC21-7507.) Resubmit the job.

RG252—IF BSCA IS SPECIFIED, ONLY TWELVE FILES ARE ALLOWED OF WHICH NINE MAY BE BSCA FILES

Code: T—Terminal
Specification Type: T
Explanation: When BSCA is used, your program should not use more than twelve files, nine of which can be BSCA files.
System Action: The job is terminated.
User Response: Do not use more than twelve files.

RG253—INVALID REMOTE DEVICE SPECIFIED, COLUMNS 65-70

Code: T—Terminal
Specification Type: T
Explanation: The entry in columns 65-70 is not a valid remote terminal.
System Action: The job is terminated.
User Response: Specify a valid remote terminal and resubmit the job.

RG254—REMOTE DEVICE SPECIFIED WHEN REMOTE TERMINAL IS BLANK OR INVALID; ASSUME COLUMNS 65-70 BLANK

Code: W—Warning
Specification Type: T
Explanation: A remote device cannot be specified if a remote terminal is not specified.
System Action: Blank is assumed for columns 65-70.
User Response: If this assumption is wrong, specify a valid remote terminal and resubmit the job.

RG256—STACKER SELECT ENTRY IN COL 16 INVALID FOR OUTPUT DEVICE; ASSUME BLANK

Code: W—Warning
Specification Type: O
Explanation: An entry in these columns is invalid for this system.
System Action: Blank is assumed.
User Response: Leave column 16 blank.

RG257—INVALID STACKER SELECT ENTRY IN COL 16; ASSUME DEFAULT STACKER

Code: W—Warning
Specification Type: O
Explanation: A stacker select entry must not be made in column 16.
System Action: No action taken.
User Response: Column 16 must be blank.

RG258—SPACE AND/OR SKIP ENTRIES IN COL 17-22 INVALID FOR DEVICE, ASSUME BLANK

Code: W—Warning
Specification Type: O
Explanation: The space and/or skip entries in columns 17-22 are invalid for the device.
System Action: Blank is assumed for invalid space and/or skip entries.
User Response: To avoid this message when the job is run again, leave columns 17-22 blank for all devices except the console, the CRT, and the printer.

RG259—INVALID SKIP ENTRIES IN COL 19-22 OR GREATER THAN THE FORM LENGTH SPECIFIED, ASSUME BLANK

Code: W—Warning
Specification Type: O
Explanation: The skip entries in columns 19-22 are not specified properly or they exceed the form length in your line counter specifications.
System Action: Blanks are assumed.
User Response: If this assumption was wrong, make the proper skip entries and resubmit the job.

**RG260—INVALID SPACE ENTRIES IN COL 17-18;
ASSUME SPACE 1 AFTER OR BLANK**

Code: W—Warning
Specification Type: O
Explanation: The space entries in columns 17-18 are not a number from 0 to 3 or blank.
System Action: If space and skip before entries 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.
User Response: If the assumption was wrong, make the proper space entries in columns 17-18 and resubmit the job.

**RG261—FETCH OVERFLOW ENTRY IN COL 16
INVALID FOR DEVICE; ASSUME BLANK**

Code: W—Warning
Specification Type: O
Explanation: The fetch overflow entry specified in column 16 is invalid for the device.
System Action: Blank is assumed; therefore, no fetch overflow is done.
User Response: To avoid the message when the job is run again, specify fetch overflow for printer files only.

**RG262—OVERFLOW INDICATOR INVALID FOR AN
EXCPT RECORD**

Code: T—Terminal
Specification Type: O
Explanation: An overflow indicator must not be specified for an exception record (E in column 15).
System Action: The job is terminated.
User Response: Remove overflow indicators from exception output lines. Resubmit the job.

**RG263—FETCH OVERFLOW INVALID WITH OVER-
FLOW INDICATOR ENTERED IN COL 23-31;
ASSUME NO FETCH**

Code: W—Warning
Specification Type: O
Explanation: An overflow indicator and fetch overflow (F in column 16) must not be specified on the same output line.
System Action: Blank in column 16 is assumed; therefore, no fetch overflow is done.
User Response: If this assumption was wrong, make the proper fetch overflow specification and resubmit the job.

**RG264—OVERFLOW INDICATOR USED IS NOT
ASSIGNED TO THIS FILE**

Code: T—Terminal
Specification Type: O
Explanation: The overflow indicator specified was not assigned to this file in your file description specifications.
System Action: The job is terminated.
User Response: Assign the overflow indicator to this file in file description specifications. Resubmit the job.

**RG265—1P INDICATOR INVALID WITH TOTAL OR
EXCPT RECORDS**

Code: W—Warning
Specification Type: O
Explanation: First page (1P) indicator must not be specified for total or exception records.
System Action: No action taken.
User Response: To avoid this message when this job is run again, specify the 1P indicator with heading and detail records only.

RG266—FETCH OVERFLOW INVALID WITH 1P INDICATOR, ASSUME NO FETCH OVERFLOW

Code: W—Warning
Specification Type: O
Explanation: A fetch overflow line (F in column 16) must not be conditioned by the 1P indicator.
System Action: No fetch overflow is assumed.
User Response: To avoid this message when this job is run again, remove the 1P indicator from lines in which fetch overflow is specified; or if the assumption was wrong, remove the 1P indicator.

RG267—1P INDICATOR INVALID FOR A COMBINED FILE

Code: T—Terminal
Specification Type: O
Explanation: The 1P indicator must not be specified for records in a combined file.
System Action: The job is terminated.
User Response: Specify the 1P indicator to condition records in an output file only.

RG268—SPECIFIED OR IMPLIED SPACE BEFORE OF ZERO IS INVALID FOR CONSOLE FILE. ASSUME SPACE BEFORE OF ONE.

Code: W—Warning
Specification Type: O
Explanation: The console forces one space before printing. A zero or blank entry in Space Before will be defaulted to one.
System Action: Space Before of one is assumed.
User Response: To avoid this message specify at least one for Space Before.

RG269—INVALID INDICATORS USED IN AN AND RELATIONSHIP WITH 1P

Code: T—Terminal
Specification Type: O
Explanation: Only external indicators (U1–U8) can be specified in an AND relationship with the 1P indicator.
System Action: The job is terminated.
User Response: Specify the 1P indicator in an AND relationship with external indicators only. Resubmit the job.

RG270—END POSITION ENTRY IN COL 40–43 FOR CONSTANT, EDIT WORD, FIELD, OR ARRAY EXCEEDS RECORD LENGTH

Code: T—Terminal
Specification Type: O
Explanation: The end position entry in columns 40–43 exceeds the records length specified in your file description specifications.
System Action: The job is terminated.
User Response: Make the proper end position entry in columns 40–43; it must be equal to or less than the record length. Resubmit the job.

RG271—LENGTH OF ARRAY, ARRAY ELEMENT, OR FIELD EXCEEDS RECORD LENGTH

Code: T—Terminal
Specification Type: O
Explanation: Length specified for array, array element, or field exceeds the record length specified in your file description specifications.
System Action: The job is terminated.
User Response: Make the proper entry; it must be equal to or less than the record length or increase the record length entry to handle the length. Resubmit the job.

RG272—END POSITION ENTRY IN COL 40–43 FOR CONSTANT, EDIT WORD, FIELD, OR ARRAY TOO LOW

Code: T—Terminal
Specification Type: O
Explanation: The end position entry in columns 40–43 is too small to allow the first field, array, or array element to be written, printed, or punched in its entirety.
System Action: The job is terminated.
User Response: Make the end position entry large enough for the field, array, or array element to be written, printed, or punched. Resubmit the job.

RG273—OUTPUT INDICATORS IN COL 23-31 MISSING OR ALL NEGATIVE

Code: W—Warning
Specification Type: O
Explanation: No output indicators are specified in columns 23-31 or all those indicators specified are negative. Output may not be written when desired.
System Action: No action taken.
User Response: To avoid this message when this job is run again, specify at least one positive indicator to condition output records to ensure that output is written only when desired.

RG274—OUTPUT INDICATORS MISSING FOR AN AND OR OR LINE

Code: W—Warning
Specification Type: O
Explanation: No conditioning indicators were specified in columns 23-31 or an AND or OR line.
System Action: No action taken.
User Response: To avoid this message when this job is run again, place the proper conditioning indicators in columns 23-31 of the AND or OR line. Resubmit the job.

RG276—INVALID EDIT CODE IN COL 38

Code: T—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.
System Action: The job is terminated.
User Response: Make the proper edit code entry in column 38 and resubmit the job.

RG277—INVALID EDIT WORD SIZE

Code: T—Terminal
Specification Type: O
Explanation: The number of replaceable characters in this edit word (columns 45-70) exceeds the length of the field to be edited.

System Action: The job is terminated.
User Response: Make the number of replaceable characters in the edit word equal to the length of the field to be edited. Resubmit the job.

RG278—EDIT CODES INVALID WITH FIELDS OTHER THAN UNPACKED NUMERIC FIELDS OR CONSTANTS OTHER THAN * OR \$

Code: T—Terminal
Specification Type: O
Explanation: Edit codes cannot be specified with edit words or with constants other than \$ or * or with fields other than unpacked numeric fields.
System Action: The job is terminated.
User Response: Make the proper edit code entry and resubmit the job.

RG279—CONSTANTS IN COL 45-70 INVALID FOR X, Y, AND Z EDIT CODES

Code: T—Terminal
Specification Type: O
Explanation: Edit codes X, Y, and Z must not be specified for edit words with '\$' or '*' in columns 45-47.
System Action: The job is terminated.
User Response: Use either edit codes or edit words, but not both. Resubmit the job.

RG280—FIELD LENGTH FOR Y EDIT CODE LESS THAN 3 OR GREATER THAN 6

Code: W—Warning
Specification Type: O
Explanation: Field edited by Y edit code is not from 3 to 6 characters long.
System Action: If less than 3 characters long, field is not edited; if more than 6 characters long, only the 6 low order digits are edited.
User Response: Make the field to be edited by Y edit code 3 to 6 characters long or change the edit code to prevent error message.

RG281—DECIMAL POSITIONS INVALID FOR FIELD EDITED BY Y CODE

Code: T—Terminal
Specification Type: O
Explanation: Decimal positions must not be specified for field edited by Y code.
System Action: The job is terminated.
User Response: Specify no decimal positions for field edited by Y edit code, or use a different edit code. Resubmit the job.

RG282—NAME OF FIELD TO BE EDITED, BY CODE SPECIFIED IN COL 38, MISSING

Code: T—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 32–37.
System Action: The job is terminated.
User Response: Specify the name of the field to be edited in columns 32–37 and resubmit the job.

RG283—INVALID FILE TYPE FOR OUTPUT RECORD

Code: T—Terminal
Specification Type: O
Explanation: The file specified in columns 7–14 of your output specifications is not a combined file, update file, output file, or a file associated with ADD.
System Action: The job is terminated.
User Response: Make sure the file specified in output specifications is a combined file, update file, output file, or a file associated with ADD. Resubmit the job.

RG286—*PRINT INVALID FOR DEVICE

Code: T—Terminal
Specification Type: O
Explanation: *PRINT cannot be used for devices on this system.
System Action: The job is terminated.
User Response: Remove *PRINT and resubmit the job.

RG287—OPERATION IN COLUMN 40 INVALID FOR DEVICE

Code: T—Terminal
Specification Type: O
Explanation: * Entry or N in column 40 is invalid for this device.
System Action: The job is terminated.
User Response: Remove * entry or N and resubmit the job.

RG288—BLANK AFTER ENTRY IN COL 39 INVALID WITH RESERVED WORD OTHER THAN PAGE; ASSUME BLANK

Code: W—Warning
Specification Type: O
Explanation: Column 39 contains a B entry with a reserved word other than PAGE.
System Action: Blank is assumed.
User Response: Leave column 39 blank and resubmit the job.

RG290—*PLACE PRECEDES ALL FIELD NAMES AND CONSTANTS

Code: T—Terminal
Specification Type: O
Explanation: When *PLACE is used, it must be specified after fields which are to be placed in different location.
System Action: The job is terminated.
User Response: Specify the fields to be moved before you specify *PLACE and resubmit the job.

RG291—INVALID ENTRIES IN COL 38, 39, OR 44–74 FOR OUTPUT OPERATION, ASSUME BLANKS

Code: T—Terminal
Specification Type: O
Explanation: Columns 38, 39, and 44–74 must be blank for *PLACE; columns 38 and 44–74 must be blank for the ledger card control number.
System Action: Blanks are assumed; the job is terminated.
User Response: Leave columns 38, 39, and 44–74 blank for *PLACE; leave columns 38 and 44–74 blank for the ledger card control number. Resubmit the job.

RG292—TOO MANY AND/OR LINES

Code: T—Terminal
Specification Type: I or O
Explanation: More than 20 AND/OR lines specified in your input or output specifications.
System Action: The job is terminated.
User Response: Make the number of AND/OR lines specified 20 or less. Resubmit the job.

RG293—BLANK AFTER SPECIFIED FOR A CONSTANT

Code: W—Warning
Specification Type: O
Explanation: Blank after should not be specified for a constant since constants will be blanked out whenever they are used.
System Action: Blank is assumed.
User Response: If you do not want the message to come out, do not specify blank after (B in columns 39) for constants.

RG300—VALUE OF ARRAY INDEX EXCEEDS NUMBER OF ARRAY ELEMENTS

Code: T—Terminal
Specification Type: O
Explanation: The array index specified exceeds the number of elements in the array.
System Action: The job is terminated.
User Response: Specify the proper array index value; the index must not exceed the number of array elements specified for the array in columns 36–39 of your extension specifications. Resubmit the job.

RG302—BLANK AFTER ENTRY IN COL 39 INVALID FOR LOOK AHEAD FIELD; ASSUME BLANK

Code: W—Warning
Specification Type: O
Explanation: Column 39 must be blank for a look ahead field.
System Action: Blank is assumed.
User Response: To avoid this message the next time the job is run, leave column 39 blank for look ahead field.

RG304—INVALID INDICATOR OR IMPROPER USE OF A VALID INDICATOR

Code: T—Terminal
Specification Type: I, C, or O
Explanation: The indicator specified is invalid or used improperly.

System Action: The job is terminated.
User Response: 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, KQ can be assigned). If the indicator has been used improperly, see the restrictions concerning proper use of indicators under *Operation Codes, Setting Indicators*. Resubmit the job.

RG305—INDICATOR ASSIGNED BUT NOT USED TO CONDITION OPERATIONS

Code: W—Warning
Specification Type: I, C, or O
Explanation: The indicator was assigned but was not used to condition an operation.
System Action: No action taken.
User Response: Determine whether the indicator assigned is needed to condition any operation. If not, remove this indicator to avoid this message the next time this job is run.

RG306—INDICATOR USED TO CONDITION OPERATIONS BUT NOT ASSIGNED

Code: T—Terminal
Specification Type: I, C, or O
Explanation: All indicators except LR, MR, 1P and L0 must be assigned before they can be used to condition operations.
System Action: The job is terminated.
User Response: Make sure the indicator is assigned before it is used to condition operations. Resubmit the job.

**RG307—FILE NAME DEFINED BUT NEVER USED.
SPECIFICATION IS DROPPED.**

Code: W—Warning
Specification Type: F
Explanation: A filename was defined in columns 7-14 but no input or output specifications exist for this file.
System Action: Specification is dropped.
User Response: To avoid this message when this job is run again, remove the filename in columns 7-14 in the fields not used in the program.

**RG308—SEQUENCING INVALID FOR FILE WITH NO
MATCH FIELD, ASSUME COLUMN 18 ON FILE
DESCRIPTION SPECIFICATION BLANK**

Code: W—Warning
Specification Type: F
Explanation: Sequence checking specified in column 18 for a file with no match fields.
System Action: Assume column 18 is blank.
User Response: Leave column 18 blank for files with no match fields.

**RG309—SEQUENCE ENTRY IN COL 18 INVALID OR
BLANK FOR FILES WITH MATCH FIELDS SPECIFIED,
ASSUME FIRST VALID SEQUENCE OR A**

Code: W—Warning
Specification Type: F
Explanation: No sequence entry or an invalid sequence entry is specified in column 18 for a file with match fields.
System Action: For a primary file, A is assumed. If no valid sequence entry is specified for a secondary file, the primary sequence value is assumed.
User Response: If this assumption was wrong, make the proper sequence entry (A or D) in column 18 and resubmit the job.

**RG310—EXTENSION CODE SPECIFIED IN COL 39 ON
FILE DESCRIPTION SPECIFICATION FOR THIS FILE,
BUT EXTENSION SPECIFICATION MISSING**

Code: T—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.
System Action: The job is terminated.
User Response: You must 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. Resubmit the job.

**RG311—AN EXTENSION OR LINE COUNTER
SPECIFICATION WAS PROVIDED FOR THIS FILE
BUT AN EXTENSION CODE WAS NOT ENTERED IN
COL 39 ON THE FILE DESCRIPTION SPECIFICATION**

Code: W—Warning
Specification Type: F
Explanation: Extension or line specifications were supplied, but no extension code (E in column 39 of file description) was specified.
System Action: No action is taken.
User Response: To avoid this message the next time this job is run, enter E in column 39 for this file.

**RG314—FIELD, TABLE OR ARRAY NAME DEFINED
BUT NEVER USED**

Code: W—Warning
Specification Type: I, E, or C
Explanation: A name is assigned to a field, table, or array but the field, table, or array is not used in the program.
System Action: No action taken.
User Response: To avoid this message when this job is run again, remove the field table, or array name if it is not used.

**RG315—FIELD NAME USED BUT NEVER DEFINED
OR TABLE NAME OR ARRAY ELEMENT USED AS AN
ARRAY INDEX**

Code: T—Terminal
Specification Type: C or O
Explanation: (1) The field name is used in calculation or output operations but was not defined, or (2) a table name or array element is used as an array index.
System Action: The job is terminated.
User Response: (1) Be sure the field is defined before it is used in calculation or output operations, or (2) be sure that the array index is not a table name or array element. Resubmit the job.

**RG316—INVALID DEFINITION FOR RESERVED
WORD; ASSUME VALID DEFINITION**

Code: T—Terminal
Specification Type: I or C
Explanation: The field named by one of the RPG II reserved words is not specified according to the predefined format.
System Action: The predefined format for this reserved word is assumed, but the job is terminated.
User Response: Make the proper entry for the reserved word and resubmit the job.

**RG317—NUMBER OF DECIMAL POSITIONS
SPECIFIED EXCEEDS FIELD LENGTH**

Code: T
Specification Type: I, C, or O
Explanation: The number of decimal positions specified exceeds the field length.
System Action: The job is terminated.
User Response: Make the proper decimal position entry, it can be equal to or less than the field length. Resubmit the job.

**RG318—MISSING A RECORD CONDITIONED BY 1P
AND FORMS POSITIONING SPECIFIED ON CONTROL
CARD**

Code: W—Warning
Specification Type: H and O
Explanation: Repetitive 1P output for forms positioning is specified in your control card specifications but 1P is not used to condition an output record.
System Action: No action taken.
User Response: Use 1P to condition the proper output record to avoid this message the next time this job is run.

**RG319—NO DATA FOR ALTERNATE COLLATING
SEQUENCE, OR FILE TRANSLATION**

Code: T—Terminal
Specification Type: H
Explanation: Alternate collating sequence or file translation is specified in your header line, but no alternate collating sequence table or file translation table was supplied.
System Action: The job is terminated.
User Response: Provide the proper tables for alternate collating sequence or file translation or delete the specifications. Resubmit the job.

**RG320—INVALID ALTERNATE COLLATING
SEQUENCE DATA RECORD**

Code: T—Terminal
Specification Type: Not applicable.
Explanation: Columns 1-6 in your alternate collating sequence data records do not contain ALTSEQ.
System Action: The job is terminated.
User Response: Check your alternate collating sequence data records to make sure the data is specified properly; each record must contain ALTSEQ in columns 1-6. Resubmit the job.

RG321—INVALID, UNDEFINED, OR TABLE FILENAME ON FILE TRANSLATION DATA RECORD

Code: T—Terminal
Specification Type: Not applicable
Explanation: The entry in columns 1-8 of the file translation data record is invalid, not previously defined, or is a table filename.
System Action: The job is terminated.
User Response: Make the entry in columns 1-8 of each file translation data record a filename previously defined in file description specifications or the characters *FILES $\emptyset\emptyset$ (\emptyset = blank). The entry must not be a table filename. Resubmit the job.

RG322—ALTERNATE COLLATING SEQUENCE OR FILE TRANSLATION DATA INVALID

Code: T—Terminal
Specification Type: Not applicable.
Explanation: The data supplied for alternate collating sequence or file translation is invalid (not 0-9 and A-F).
System Action: The job is terminated.
User Response: Make the data specified for alternate collating sequence or file translation consist of the characters A-F and 0-9. Resubmit the job.

RG324—TOTAL LENGTH OF ALL CONTROL OR ALL MATCHING FIELDS EXCEEDS 144 CHARACTERS

Code: T—Terminal
Specification Type: I
Explanation: The total length of all control or all matching fields is too large.
System Action: The job is terminated.
User Response: Make the total length of all matching fields (M1-M9) or all control fields (L1-L9) equal to or less than 144. Resubmit the job.

RG325—ALL INPUT, UPDATE, AND COMBINED FILES CONDITIONED BY EXTERNAL INDICATORS

Code: W—Warning
Specification Type: I

Explanation: When all input, update, and combined 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.
System Action: No action taken.
User Response: When all input, update, or combined files are conditioned by external indicators, be sure all indicators are not off.

RG326—COMPILE-TIME TABLES SPECIFIED NO DATA FOUND

Code: T—Terminal
Specification Type: Not applicable
Explanation: Compile time table specified (From filename in columns 11-18 of extension specifications blank), but no table input records were supplied after the source program.
System Action: The job is terminated.
User Response: For compile time tables, supply the table input records immediately after the source program. Resubmit the job.

RG327—SPLIT CONTROL FIELDS SPECIFIED MAY NOT HAVE PARTS THAT ARE PACKED

Code: T—Terminal
Specification Type: I
Explanation: All parts of a split control field must be either packed or unpacked.
System Action: The job is terminated.
User Response: Make proper entries so that all parts of the split control field are either packed or unpacked. Resubmit the job.

RG329—PACKED OR BINARY DATA NOT VALID FOR DEVICE

Code: W—Warning
Specification Type: I or O
Explanation: Packed or binary data should be specified only for disk, BSCA, and 1442 files.
System Action: Data errors may occur if program is executed.
User Response: Specify packed or binary data for disk, BSCA, and 1442 files only. Resubmit the job.

**RG330—ALPHAMERIC FIELD SPECIFIED AS
PACKED OR BINARY**

Code: T—Terminal
Specification Type: O
Explanation: Packed data cannot be specified for alphameric fields.
System Action: The job is terminated.
User Response: Specify packed data for numeric fields only. Resubmit the job.

RG331—NO INPUT SPECIFICATIONS FOUND

Code: T—Terminal
Specification Type: Not applicable
Explanation: No valid input specifications are supplied for this job.
System Action: The job is terminated.
User Response: Supply valid input specifications and resubmit this job.

**RG332—SEQUENCE ERROR FOUND IN COMPILE
TIME TABLE/ARRAY**

Code: T—Terminal
Specification Type: Not applicable
Explanation: Compile time table or array is not in the sequence specified in columns 45 to 57.
System Action: The job is terminated.
User Response: Make sure the data is in the sequence specified (A or D) in column 45 or 57. Resubmit the job.

**RG333—TABLE/ARRAY FULL OR NO TABLE/
ARRAYS FOR FOLLOWING DATA**

Code: W—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.
System Action: No more data is accepted for tables or arrays.
User Response: Make sure the data supplied does not exceed the maximum table size or that a table or array is defined for the data you supply. Resubmit the job.

RG334—SHORT TABLE

Code: W—Warning
Specification Type: Not applicable
Explanation: The number of entries supplied is less than the maximum number of entries the table can contain.
System Action: The remaining entries are filled with blanks or zeros.
User Response: None required.

**RG335—EDIT WORD SPECIFIED WITH OTHER THAN
UNPACKED NUMERIC FIELDS**

Code: T—Terminal
Specification Type: O
Explanation: Edit words are allowed only with unpacked numeric fields.
System Action: The job is terminated.
User Response: Specify edit words for unpacked numeric fields only. Resubmit the job.

**RG337—INVALID SEQUENCE FOR EXIT AND RLABL
OP CODES**

Code: T
Specification Type: C
Explanation: The RLABL operation code does not immediately follow an EXIT operation.
System Action: The job is terminated.
User Response: Make sure that RLABL operations follow an EXIT operation in your calculation specifications. Resubmit the job.

RG338—SUBR MUST BE USED WITH EXIT OP CODE

Code: T—Terminal
Specification Type: C
Explanation: The entry specified in Factor 2 of an EXIT operation does not start with SUBR.
System Action: The job is terminated.
User Response: Make sure the subroutine name in Factor 2 starts with SUBR. Resubmit the job.

RG339—AN OUTPUT REFERENCE IS REQUIRED FOR EACH COMBINED OR UPDATE FILE

Code: T—Terminal
Specification Type: F
Explanation: The proper output specifications have not been specified for the combined or update file.
System Action: The job is terminated.
User Response: Specify the proper output specifications for the combined or update file. A table output specification will meet the requirements for a combined file. Resubmit the job.

RG340—CONTROL/TRIBUTARY, COLUMN 17, CONTAINS A BLANK FOR A MULTIPOINT LINE. ASSUME T

Code: W—Warning
Specification Type: T
Explanation: Column 17 was left blank for a multipoint line (M in column 15).
System Action: T is assumed.
User Response: To avoid this message when this job is run again, enter a T in column 17, or change the configuration entry in column 15.

RG341—CONTROL/TRIBUTARY, COLUMN 17, CONTAINS A T FOR A SWITCHED OR A POINT TO POINT NETWORK. ASSUME BLANK

Code: W—Warning
Specification Type: T
Explanation: Column 17 contains a T for a point-to-point network (P in column 15).
System Action: Blank is assumed.
User Response: To avoid this message when this job is run again, leave column 17 blank, or change the configuration entry in column 15.

RG342—TRANSPARENT MODE IS SPECIFIED, COLUMN 19, WHEN ASCII CONTROL CHARACTERS, COLUMN 18, ARE TO BE USED

Code: T—Terminal
Specification Type: T
Explanation: The transparent mode cannot be specified on an adapter using ASCII data link characters.
System Action: The job is terminated.
User Response: Make the proper entry in column 19 and resubmit the job.

RG343—AUTOCALL/AUTOANSWER, COLUMN 20, IS NOT BLANK FOR NON-SWITCHED NETWORK

Code: T—Terminal
Specification Type: T
Explanation: Column 20 contains an entry for a network that is not switched.
System Action: The job is terminated.
User Response: Leave column 20 blank for a network that is not switched. Resubmit the job.

RG344—SYMBOL FOR DIAL NUMBER, COLUMNS 21-31, IS AN ARRAY

Code: T—Terminal
Specification Type: T
Explanation: An array name was used as the dial number.
System Action: The job is terminated.
User Response: Enter the table element or field name to be used as the dial number in columns 21-31. If you want to use an array element as the dial number, you must use calculation specifications to move the contents of the array element into the field you specify in columns 21-31. Resubmit the job.

RG345—FIELD OR TABLE HOLD AREA FOR THE DIAL NUMBER WAS NOT DEFINED AS NUMERIC

Code: T—Terminal
Specification Type: T
Explanation: The field or table hold area for the dial number specified in columns 21–31 was not defined as numeric.
System Action: The job is terminated.
User Response: Define the field or table hold area for the dial number specified in columns 21–31 as numeric. Re-submit the job.

RG346—COLUMN 32 IS NOT BLANK FOR A NON-SWITCHED NETWORK

Code: T—Terminal
Specification Type: T
Explanation: Column 32 was not left blank for a non-switched network.
System Action: The job is terminated.
User Response: Leave column 32 blank for a non-switched network and resubmit the job.

RG347—IDENTIFICATION FOR THIS STATION, COLUMNS 33–39, CONTAINS AN ARRAY

Code: T—Terminal
Specification Type: T
Explanation: An array name was used as the station identification.
System Action: The job is terminated.
User Response: Enter the table element or field name to be used as the station identification in columns 33–39. 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 33–39. Resubmit the job.

RG348—COLUMN 40 IS NOT BLANK FOR A NON-SWITCHED NETWORK

Code: T—Terminal
Specification Type: T
Explanation: Column 40 was not left blank for a non-switched network.
System Action: The job is terminated.
User Response: Leave column 40 blank for a non-switched network and resubmit the job.

RG349—IDENTIFICATION FOR THE REMOTE STATION, COLUMNS 41–47, CONTAINS AN ARRAY

Code: T—Terminal
Specification Type: T
Explanation: An array name was used as the remote station identification.
System Action: The job is terminated.
User Response: Enter the table element or field name to be used as the remote station identification in columns 41–47. 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 41–47. Resubmit the job.

RG350—RECORD AVAILABLE INDICATOR IS PRESENT ON TRANSMIT FILE, OR IN A PROGRAM WITH ONLY 1 BSCA FILE. INDICATOR IS DROPPED

Code: W—Warning
Specification Type: T
Explanation: A record available indicator was specified for a transmit file or in a program which has only one BSCA file.
System Action: The indicator is ignored.
User Response: Remove the record available indicator or define the other BSCA file if a transmit interspersed with a receive program is desired. Resubmit the job.

RG351—LAST FILE PROCESSED, COLUMN 60, IS NOT BLANK ON A TRANSMIT FILE OR A PRIMARY INPUT FILE. THE ENTRY IS IGNORED

Code: W—Warning
Specification Type: T
Explanation: L was entered in column 60 for a transmit file or for a primary input file.
System Action: The entry is ignored.
User Response: 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. Resubmit the job.

RG352—POLLING CHARACTERS WERE GIVEN ON OTHER THAN A TRANSMIT FILE ON A MULTIPOINT NETWORK. THE ENTRY IS IGNORED

Code: W—Warning
Specification Type: T
Explanation: Polling characters are specified in columns 61–62 for a file other than a transmit file on a multipoint network.
System Action: The entry in columns 61–62 is ignored.
User Response: To avoid this message when this job is run again, remove the entry from columns 61–62.

RG353—THERE IS AN ENTRY IN THE ADDRESSING CHARACTERS, COLUMNS 63–64, ON A FILE THAT IS NOT A MULTIPOINT RECEIVER FILE. THE ENTRY IS IGNORED

Code: W—Warning
Specification Type: T
Explanation: Addressing characters are specified in columns 63–64 for a file that is not a multipoint receiver file.
System Action: The entry in columns 63–64 is ignored.
User Response: To avoid this message when this job is run again, remove the entry from columns 63–64.

RG354—CORRESPONDING FILE DESCRIPTION SPEC FILE IS NOT A BSC FILE

Code: T—Terminal
Specification Type: T
Explanation: A BSC device entry was not made for this file on the File Description sheet.
System Action: The job is terminated.
User Response: Make a BSC device entry for this file on the File Description sheet. Resubmit the job.

RG355—A CONVERSATIONAL FILE WAS DEFINED WHEN NO CONVERSATIONAL FILE IS ALLOWED

Code: T—Terminal
Specification Type: T
Explanation: A conversational file is not allowed with 2770/2780.
System Action: The job is terminated.
User Response: Correct the telecommunications specification and resubmit the job.

RG356—PACKED FIELD OR BINARY FIELD SPECIFIED IN A FILE WITHOUT THE TRANSPARENT FEATURE

Code: T—Terminal
Specification Type: T
Explanation: A packed or binary field was specified for a file that does not have the transparent feature.
System Action: The job is terminated.
User Response: Be sure packed or binary fields are only specified for files with the transparent feature. Resubmit the job.

RG357—THE FILE CORRESPONDING TO THIS TRANSMITTER SPECIFICATION IS NOT A COMBINED OR AN OUTPUT FILE ON THE FILE DESCRIPTION SPECIFICATION

Code: T—Terminal
Specification Type: T
Explanation: The transmitter file was not defined as a combined or output file on the File Description sheet.
System Action: The job is terminated.
User Response: Define the transmitter file as a combined file or an output file on the File Description sheet. Resubmit the job.

RG358—CORRESPONDING FILE DESCRIPTION SPEC FILE IS NOT DEFINED AS A COMBINED OR AN INPUT FILE FOR THIS RECEIVE FILE

Code: T—Terminal
Specification Type: T
Explanation: The receive file was not defined on the File Description sheet as a combined file or as an input file.
System Action: The job is terminated.
User Response: Define the receive file as a combined file or as an input file on the File Description sheet. Resubmit the job.

RG359—BLOCKED RECORD DEFINED FOR A FILE WITH CONVERSATIONAL RESPONSES. ASSUME NO BLOCKING

Code: W—Warning
Specification Type: T
Explanation: Blocked records must not be defined for a file with conversational responses.
System Action: No blocking is assumed.
User Response: To avoid this message when this job is run again, remove the blocked records specification.

RG360—THERE IS NO TELECOMMUNICATIONS SPEC FOR A FILE DEFINED AS A BSCA FILE ON THE FILE DESCRIPTION SPECS

Code: T—Terminal
Specification Type: T
Explanation: No telecommunications specifications were supplied for a file that was described as a BSCA file on the File Description sheet.
System Action: The job is terminated.
User Response: Supply the proper telecommunications specifications and resubmit the job.

RG361—LOOK AHEAD FIELDS SPECIFIED FOR BSC FILE

Code: T—Terminal
Specification Type: T
Explanation: Look ahead fields are not allowed for a BSC file.
System Action: The job is terminated.
User Response: Remove the look ahead specification for BSC file and resubmit the job.

RG362—MATCHING FIELDS DEFINED ON A TRANSMIT FILE WITH CONVERSATIONAL RESPONSE

Code: T—Terminal
Specification Type: T
Explanation: Matching fields are not allowed for a transmit file with conversational responses.
System Action: The job is terminated.
User Response: Remove the matching fields definition for transmit file with conversational responses.

RG363—MATCHING FIELDS DEFINED FOR A FILE DESIGNATED TO BE THE LAST FILE PROCESSED IN COLUMN 60 OF THE TELECOMMUNICATIONS SPEC

Code: T—Terminal
Specification Type: T
Explanation: Matching fields were defined for a file designated as the last file to be processed (L in column 60).
System Action: The job is terminated.
User Response: Remove the matching fields definition if the file was the last one to be processed, or remove the L entry in column 60. Resubmit the job.

RG364—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

Code: W—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 sheet.
System Action: EBCDIC is assumed if end of file (E in column 17 of the File Description sheet) is specified for any input file the program uses.
User Response: If the assumption was wrong, remove the L from column 60 or make the proper end of file entry on the Input sheet. Resubmit the job.

RG365—ITB IS SPECIFIED ON A FILE WITHOUT BLOCKED RECORDS. ITB IS DROPPED

Code: W—Warning
Specification Type: T
Explanation: Intermediate block check (ITB) was specified for a file which does not have blocked records.
System Action: The intermediate block check specification (I in column 52) is ignored.
User Response: To avoid this message when this job is run again, remove the I from column 52 or define blocked records. Resubmit the job.

RG366—AUTOCALL/AUTOANSWER, COLUMN 20, IS BLANK FOR A SWITCHED NETWORK

Code: T—Terminal
Specification Type: T
Explanation: Column 20 was left blank for a switched network.
System Action: The job is terminated.
User Response: Make the proper entry (M, E, S, A, or B) in column 20 for a switched network.

RG367—A TRANSMIT WITH CONVERSATIONAL RESPONSE FILE IS USED WITH FORCE OR READ OPERATION CODE OR AS A PRIMARY FILE

Code: T—Terminal
Specification Type: T
Explanation: (1) Neither the FORCE nor the READ operation code can be used with a transmit file which has conversational responses.
(2) A transmit file with conversational responses cannot be a primary file.
System Action: The job is terminated.
User Response: Remove the FORCE or READ operation code or change the file designation from primary. Resubmit the job.

RG368—THE FIELD OR TABLE HOLD AREA USED FOR A STATION IDENTIFICATION, COLUMNS 33-39 OR COLUMNS 41-47, IS MORE THAN FIFTEEN CHARACTERS IN LENGTH, OR DIAL NUMBER IS MORE THAN TWELVE DIGITS

Code: T—Terminal
Specification Type: T
Explanation: Either the field or table hold area used for a station identification (columns 33-39 or 41-47) contains more than 15 characters, or the dial number (columns 21-31) contains more than 12 digits.
System Action: The job is terminated.
User Response: Be sure that the field or table hold area used for a station identification is numeric and from 2 to 15 characters long. If you specify a dial number, be sure it is not more than 12 characters long. Resubmit the job.

RG369—WARNING: ONLY ONE I/O AREA WAS SPECIFIED ON A NON-CONVERSATIONAL FILE. THROUGHPUT MAY BE SLOW

Code: W—Warning
Specification Type: T
Explanation: Because only one I/O area is specified for a non-conversational file, processing time is likely to be slow.
System Action: No action taken.
User Response: To avoid this message when the job is run again, specify dual I/O areas if the program size permits.

RG370—THE LINE CONFIGURATION AND LINE CONTROL ENTRIES, COLUMN 15 OR 17-47, ARE NOT THE SAME ON EACH TELECOMMUNICATIONS SPEC

Code: T—Terminal
Specification Type: T
Explanation: The line configuration and line control entries (column 15 or 17-47) are not the same for each BSC file.
System Action: The job is terminated.
User Response: Make the same entries in columns 15 and 17-47 for each BSC file in the program. Resubmit the job.

RG371—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

Code: W—Warning
Specification Type: T
Explanation: The station identification entry (columns 33-39 or 41-47) was specified as a 1-character field.
System Action: The character is duplicated to provide a two-character identification field.
User Response: If the assumption was wrong, specify a station identification which is at least 2 characters, but no more than 12 character long. Resubmit the job.

RG372—A B IN COLUMN 37 OF THE CONTROL CARD IS AN INVALID ENTRY IN A BSCA PROGRAM

Code: T—Terminal
Specification Type: H
Explanation: A B entry must not be specified in column 37 of the control card specifications for a BSCA program.
System Action: The job is terminated.
User Response: Remove the B entry from column 37 of the control card specifications and resubmit the job.

RG373—THE SAME FILENAME WAS GIVEN ON TWO TELECOMMUNICATIONS SPECS

Code: T—Terminal
Specification Type: T
Explanation: A BSCA file must not have multiple definitions.
System Action: The job is terminated.
User Response: Specify a unique filename on each Telecommunications sheet used in this program. Resubmit the job.

RG374—ENTRY IN COL 16 INVALID

Code: W—Warning
Specification Type: F
Explanation: The entry in column 16 of the file description specifications is not P, S, C, R, T, D, or blank.
System Action: Blank is assumed if the file is an output file; otherwise, S is assumed.
User Response: If the assumption was wrong, make the proper entry in column 16 and resubmit the job.

RG375—ID IN COL 75-80 OF CONTROL CARD MUST NOT BE BLANK WHEN C IS SPECIFIED IN COL 10, ASSUME BLANK IN COL 10

Code: W—Warning
Specification Type: H
Explanation: A C is specified in column 10 of your control card specifications, but no program identification is specified in columns 75-80.
System Action: Column 10 is assumed to be blank.
User Response: When C is specified in column 10 of your control card specification, place the proper program name in columns 75-80. Resubmit the job.

RG376—INVALID NAME IN COLS 75-80 OF CONTROL CARD, ASSUME BLANK

Code: W—Warning
Specification Type: H
Explanation: The entry in columns 75-80 of your header line is neither a valid RPG program name nor blanks.
System Action: Blanks are assumed.
User Response: If this assumption was wrong, make the proper program name entry and resubmit the job.

RG377—RAF, COLUMN 31, IS NOT ALLOWED ON A BSCA FILE

Code: T—Terminal
Specification Type: F
Explanation: A BSCA file cannot be specified as a record address file.
System Action: The job is terminated.
User Response: Remove the record address file specification for a BSCA file and resubmit the job.

RG378—NO LINE COUNTER SPECIFICATION FOR THIS BSCA FILE. ASSUME PAGE SIZE—66, OVERFLOW LINE—60

Code: W—Warning
Specification Type: T
Explanation: Entries must be specified if the page size and overflow line differ from assumed values.
System Action: Page size of 66 is assumed; overflow line of 60 is assumed.
User Response: Verify that page size of 66 is correct for the job.

RG379—MULTI-POINT INVALID WITH 2770 OR 2780

Code: T—Terminal
Specification Type: T
Explanation: Column 15 must be P, S, or blank.
System Action: The job is terminated.
User Response: Correct column 15 and resubmit the job.

RG380—2770 AND 2780 CANNOT BE SPECIFIED IN THE SAME PROGRAM

Code: T—Terminal
Specification Type: T
Explanation: Both 2770 and 2780 have been specified for the same job.
System Action: The job is terminated.
User Response: Specify either 2770 or 2780 but not both.

RG381—INVALID DEVICE SPECIFIED FOR THE REMOTE TERMINAL USED

Code: T—Terminal
Specification Type: T
Explanation: Device specified in columns 65-70 is not a valid remote device.
System Action: The job is terminated.
User Response: Specify a valid device for the remote terminal used.

RG382—INVALID REMOTE DEVICE FOR FILE TYPE SPECIFIED

Code: T—Terminal
Specification Type: T
Explanation: An output device was specified for an input file or an input device was specified for an output file.
System Action: The job is terminated.
User Response: Specify a valid remote device for the type of operation being performed and resubmit the job.

RG383—ITB AND TRANSPARENCY SPECIFIED FOR 2770. ITB IS DROPPED

Code: W—Warning
Specification Type: T
Explanation: When 2770 is specified, specify either ITB (column 52) or Transparency (column 19) but not both.
System Action: Blank is assumed for column 52 (ITB).
User Response: Verify that the assumption is correct for this job.

RG388—FACTOR MUST BE EITHER A FIELD NAME OR A LITERAL WHEN USED WITH DEBUG OPERATION

Code: T—Terminal
Specification Type: C
Explanation: Factor 1 can only be a field name or a literal when the DEBUG operation is specified.
System Action: The job is terminated.
User Response: Make Factor 1 either a field name or a literal and resubmit the job.

RG389—L0-L9 INDICATOR SPECIFIED IN AN OR RELATIONSHIP WITH LR

Code: W—Warning
Specification Type: O
Explanation: An L0-L9 indicator should not be specified in an OR relationship with an LR indicator.
System Action: Operations specified in this relationship will be done twice at LR time.
User Response: Do not specify an L0-L9 indicator in an OR relationship with an LR indicator unless you want the specified operations to be done twice. Resubmit the job if necessary.

RG390—SEQUENCE CHECKING IS NOT PERFORMED ON EXECUTION TIME ARRAYS

Code: W—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.
System Action: A sequenced array is assumed.
User Response: Be sure the array is in ascending or descending sequence.

RG391—A FIELD WITH A LENGTH GREATER THAN 8 CHARACTERS CANNOT BE USED IN FACTOR 1 WITH DEBUG OPERATION

Code: T—Terminal
Specification Type: C
Explanation: The length of a Factor 1 field cannot be greater than eight characters when a DEBUG operation is specified.
System Action: The job is terminated.
User Response: Limit the length of the Factor 1 field to eight characters. Resubmit the job.

RG392—LAST ENTRY IN ONE OR MORE COMPILE TIME TABLE/ARRAYS WAS BLANK.

Code: W—Warning
Specification Type: E
Explanation: The compile time table/array contains fewer entries than the number of entries specified in columns 36-39 of the Extension specifications.
System Action: A warning message is given.
User Response: If the assumption is wrong, review your compile time tables/arrays and fill the table.

RG394—'ADD' IN COL 16-18 NOT ALLOWED ON AND/OR LINES, ASSUME BLANK

Code: T—Terminal
Specification Type: O
Explanation: ADD was specified in columns 16-18 of an AND/OR line in output specifications.
System Action: Blank is assumed, but the job is terminated.
User Response: Remove the ADD entry from columns 16-18 of the AND/OR line and resubmit the job.

RG397—FILE DESCRIBED AS 'ADD' TYPE FILE, EACH OUTPUT LINE MUST HAVE 'ADD' IN COL 16-18. ASSUME 'ADD'

Code: W—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 16-18 of the Output sheet for each record type output line to be written.
System Action: ADD in columns 16-18 is assumed.
User Response: To avoid this message the next time this job is run, remove the A from column 66 of the file description specifications or specify ADD in columns 16-18 of the output specifications for each record type output line to be written.

RG398—COLS 54-59, INVALID FOR DEVICE, OR WRONG ENTRY, ASSUME BLANK

Code: T—Terminal
Specification Type: F
Explanation: Columns 54-59 contain an entry for a file which was not assigned to a SPECIAL device (SPECIAL in columns 40-46).
System Action: Blank is assumed, but the job is terminated.
User Response: Leave columns 54-59 blank for file not assigned to a SPECIAL device. Resubmit the job.

RG399—INVALID ENTRY IN COLS 54-59

Code: T—Terminal
Specification Type: F
Explanation: The entry in columns 54-59 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).
System Action: The job is terminated.
User Response: Enter the name of the user-written subroutine (SUBxx) or IBM-written subroutine (SRyzzz) which will perform the input/output operations for the SPECIAL file. Resubmit the job.

RG400—INVALID MODE OF PROCESSING ENTRY IN COLUMN 28

Code: T—Terminal
Specification Type: F
Explanation: The entry in column 28 is not R, L, or blank.
System Action: R is assumed for valid file type or mode of processing; the job is terminated.
User Response: Make proper mode of processing entry in column 28 and resubmit the job.

RG401—ONLY ONE TABLE/ARRAY PER FILENAME ALLOWED FOR THIS DEVICE

Code: T—Terminal
Specification Type: E
Explanation: Only one table or array can be specified per file (except for a card file).
System Action: The job is terminated.
User Response: Specify only one table or array per file (except for card files) and resubmit the job.

RG403—INVALID LENGTH OF KEY FIELD IN COLUMN 29-30, ASSUME 03

Code: T—Terminal
Specification Type: F
Explanation: The length of key field entry in columns 29-30 is not specified properly. The entry must be 29 or less for unpacked keys, 8 for packed keys.
System Action: 03 is assumed, but the job is terminated.
User Response: Make the length of key field entry in columns 29-30 a valid key length. Resubmit the job.

**RG404—INVALID RECORD ADDRESS TYPE ENTRY
IN COLUMN 31, ASSUME A**

Code: T—Terminal
Specification Type: F
Explanation: The entry in column 31 is not A, P, I, or blank.
System Action: A is assumed; the job is terminated.
User Response: Make the proper record address type entry in column 31 and re-submit the job.

**RG405—INVALID KEY START LOCATION ENTRY IN
COLUMNS 35-38, ASSUME 1**

Code: T—Terminal
Specification Type: F
Explanation: Columns 35-38 do not contain a number from 1-4096 for an indexed file.
System Action: 1 is assumed; the job is terminated.
User Response: Make the proper key start location entry in columns 35-38 and re-submit the job.

**RG406—INVALID CORE INDEX ENTRY IN COLS
60-65, ASSUME BLANK**

Code: W—Warning
Specification Type: F
Explanation: Columns 60-65 do not contain a number from 6-9999 for an indexed file processed randomly.
System Action: Blank is assumed.
User Response: If this assumption was wrong, make the proper core index entry in columns 60-65 and resubmit the job.

**RG407—INVALID FILE ADDITION OR UNORDERED
ENTRY IN COLUMN 66, ASSUME A**

Code: T—Terminal
Specification Type: F
Explanation: The file addition or unordered load entry in column 66 is not A, U, or blank.
System Action: A is assumed; the job is terminated.
User Response: Make the proper file addition or unordered load entry in column 66 and resubmit the job.

**RG408—NUMBER OF EXTENTS ENTRY IN COLS
68-69 IS INVALID OR NOT ALLOWED WITH DEVICE,
ASSUME BLANK**

Code: T—Terminal
Specification Type: F
Explanation: Columns 68-69 contain an entry for some device other than disk or contain an entry which is not 01-50 or blank.
System Action: Blank is assumed; the job is terminated.
User Response: Make the proper entry in columns 68-69 and resubmit the job.

**RG409—ENTRY OF K MADE IN COLUMN 31 FOR
RECORD ADDRESS TYPE, ASSUME A**

Code: W—Warning
Specification Type: F
Explanation: An entry of K is not allowed in column 31 for record address type.
System Action: A is assumed.
User Response: If this assumption was wrong, make the proper entry in column 31 and resubmit the job.

**RG451—CONTINUATION (K IN COL 53) INVALID
FOR DEVICE**

Code: T—Terminal
Specification Type: F
Explanation: Continuation is only allowed on SPECIAL files.
System Action: Continuation is ignored; job is terminated.
User Response: Remove the continuation (K in column 53) and resubmit the job.

**RG452—ENTRY IN COL 54-59 OF A CONTINUATION
CARD IS INVALID OR MISSING**

Code: T—Terminal
Specification Type: F
Explanation: For a SPECIAL file you must enter a table or array name in columns 54-59. The table or array must be defined on the Extension specifications. Blanks are invalid.
System Action: The continuation card is ignored; job is terminated.
User Response: Correct or remove the entry.

RG453—CONTINUATION ENTRY IN COL 54—59 IS REPEATED FOR A FILE, SECOND ENTRY IGNORED

Code: W—Warning
Specification Type: F
Explanation: For a SPECIAL file only one continuation is allowed.
System Action: The second usage of the entry is ignored.
User Response: To avoid this message on the next run, remove the repeated continuation entry.

RG455—COLUMNS 7—52 AND 66—72 ARE NOT BLANK FOR A CONTINUATION LINE, ASSUME BLANK

Code: W—Warning
Specification Type: F
Explanation: If continuation is specified, these columns must be blank.
System Action: Entries in columns 7—52 and 66—72 are ignored.
User Response: If this assumption is incorrect, remove the continuation entries and resubmit the job.

RG457—ENTRY IN COL 70 INVALID FOR DEVICE, ASSUME BLANK

Code: W—Warning
Specification Type: F
Explanation: Column 70 is not used.
System Action: Entry in column 70 is ignored.
User Response: To avoid this message on the next run, leave column 70 blank.

RG460—INVALID ENTRY IN COL 53, ASSUME BLANK

Code: W—Warning
Specification Type: F
Explanation: Valid entries are K or blank.
System Action: Blank is assumed.
User Response: To avoid this message on the next run, correct entry in column 53.

RG461—INVALID ENTRY IN COL 70, ASSUME BLANK

Code: W—Warning
Specification Type: F
Explanation: Column 70 must be blank.
System Action: Blank is assumed.
User Response: To avoid this message on the next run, remove the entry in column 70.

RG462—CONTINUATION, K IN COL 53, INVALID FOR MAIN FILE DESCRIPTION LINE, ASSUME BLANK

Code: W—Warning
Specification Type: F
Explanation: K is valid only on a continuation File Description specification.
System Action: Blank is assumed.
User Response: To avoid this message on the next run, leave column 53 blank.

RG500—FROM NAME INVALID OR MISSING FROM RA FILE

Code: T—Terminal
Specification Type: E
Explanation: The From Filename entry in columns 11—18 is missing or not specified properly for an RA file.
System Action: The job is terminated.
User Response: Enter the proper record address filename in columns 11—18 and resubmit the job.

RG502—FROM FILENAME IS AN RA FILE THAT IS USED MORE THAN ONCE

Code: T—Terminal
Specification Type: E
Explanation: The RA file named in columns 11—18 is used more than once in the Extension specifications.
System Action: The job is terminated.
User Response: Since only one RA file is allowed in a program, either remove the Extension specification or correct the entry in columns 11—18.

RG503—TO FILENAME FOR A RA FILE TYPE IS EITHER: 1—NOT A PRIMARY, SECONDARY, OR DEMAND FILE OR 2—IS MISSING, INVALID OR NON DISK FILE

Code: T—Terminal
Specification Type: E
Explanation: The To Filename entry in columns 19–26 must be a primary or secondary disk file, to be processed by an RA file.
System Action: The job is terminated.
User Response: Make the proper To Filename entry in columns 19–26 and re-submit the job.

RG504—TO FILENAME IS INCORRECT FILE TYPE

Code: T—Terminal
Specification Type: E
Explanation: The filename specified in columns 19–26 is not an input, output, or update file.
System Action: The job is terminated.
User Response: Make sure the file named in columns 19–26 is an input, output, or update file. Resubmit the job.

RG510—LENGTH GIVEN FOR BINARY FIELD IS NOT 2 OR 4, ASSUME 2

Code: T—Terminal
Specification Type: I, O
Explanation: Binary field length specified is neither 2 nor 4 bytes.
System Action: The job is terminated.
User Response: Make the length of the binary field either 2 or 4 bytes. Resubmit the job.

RG511—PACKED LENGTH GREATER THAN 8 FOR A FIELD, TABLE, OR ARRAY

Code: T—Terminal
Specification Type: I, O
Explanation: The length specified for a packed field, table, or array is greater than 8.
System Action: The job is terminated.
User Response: Specify a length of 8 or less for a packed field, table, or array. Re-submit the job.

RG516—MORE THAN 7 AN/OR LINES SPECIFIED

Code: T—Terminal
Specification Type: C
Explanation: More than 7 consecutive AN/OR line specified in the calculation specifications.
System Action: The job is terminated.
User Response: Specify up to 7 consecutive AN, OR, or AN/OR lines to condition an operation. Resubmit the job.

RG517—AN/OR LINES OUT OF ORDER

Code: T—Terminal
Specification Type: C
Explanation: The line immediately following a line with an operation code is an AN/OR line.
System Action: The job is terminated.
User Response: Remove the AN/OR entry in columns 7–8 from the first line in an AN/OR group and resubmit the job.

RG518—NO INDICATORS GIVEN WITH AND/OR LINES

Code: W—Warning
Specification Type: C
Explanation: At least one indicator should be given in an AN or OR line.
System Action: Warning is issued.
User Response: Correct or remove the specification to eliminate the warning error.

RG519—COLUMNS 18-59 ARE INVALID WITH AN/OR LINES OR OP CODE IS MISSING WITH INDICATORS PRESENT, ASSUME BLANK

Code: T—Terminal
Specification Type: C
Explanation: Only the last line of a group of AN/OR lines can have entries in columns 18-59 or indicators are specified in columns 7-17, but no operation is specified in columns 28-32.
System Action: The job is terminated.
User Response: Make sure that entries are made only in columns 18-59 of the last line of a group of AN/OR lines or make the proper operation code entry in columns 28-32 Resubmit the job.

RG520—THIS LINE IS NOT AN AN/OR LINE AND PREVIOUS LINE HAS NO OP CODE; OR THIS LINE HAS NO INDICATORS AND NO OP CODE

Code: T—Terminal
Specification Type: C
Explanation: This line is not an AN/OR line and previous line has no operation code specified.
System Action: The job is terminated.
User Response: If this line should be an AN/OR line, enter an AN/OR entry in columns 7-8; 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 28-32. Resubmit the job.

RG521—MINUS INDICATOR IS NOT ALLOWED FOR TEST BIT OPERATION OF ONLY 1 BIT

Code: W—Warning
Specification Type: C
Explanation: Columns 56-57 (Minus) must be blank when only one bit is specified for a TESTB operation.
System Action: Blank is assumed.
User Response: To avoid the message the next time this job is run, leave columns 56-57 blank.

RG522—ALL THREE RESULTING INDICATORS ARE THE SAME

Code: W—Warning
Specification Type: C
Explanation: Usually the same indicator is used for only one or two of the conditions.
System Action: The indicator specified will be set on each time the calculation is executed.
User Response: Make sure the proper resulting indicator entries have been made in columns 54-59. If the entries were incorrect, resubmit the job.

RG523—A NEGATIVE FACTOR FOR THE SQUARE ROOT OPERATION IS NOT ALLOWED

Code: T—Terminal
Specification Type: C
Explanation: The entry specified in Factor 2 of a SQRT operation is negative.
System Action: The job is terminated.
User Response: Make the entry in Factor 2 of a SQRT operation a positive value. Resubmit the job.

RG524—WHOLE ARRAYS ARE NOT ALLOWED AS FACTOR 1 WITH DISPLAY OR CHAIN OP CODE

Code: T—Terminal
Specification Type: C
Explanation: The entry in Factor 1 of a DSPLY or CHAIN operation cannot be a whole array.
System Action: The job is terminated.
User Response: Enter the array name and index in Factor 1 of a DSPLY or CHAIN operation. Resubmit the job.

RG525—OPERATION CODE IS INVALID FOR DEVICE TYPE OR MODE OF PROCESSING

Code: T—Terminal
Specification Type: C
Explanation: The CHAIN operation can only be specified for disk files processed randomly; READ may not be used with the KEYBORD file.
System Action: The job is terminated.
User Response: Make sure that CHAIN is only specified for disk files processed randomly. Do not use READ with a KEYBORD file. Resubmit the job.

RG540—FILE TYPE ENTRY IS U OR C FOR A CONSOLE, ASSUME I

Code: W—Warning
Specification Type: F
Explanation: Neither an update nor a combined file can be assigned to the console.
System Action: The file is assumed to be an input file.
User Response: If this assumption was wrong, make the proper entry in column 15 and resubmit the job.

RG541—FILE DESIGNATION IS INVALID FOR ADDRROUT FILE, ASSUME R

Code: T—Terminal
Specification Type: F
Explanation: The file designation entry in column 16 is not R for an ADDRROUT file.
System Action: The job is terminated.
User Response: Enter an R in column 16 for the ADDRROUT file and resubmit the job.

RG543—LENGTH OF KEY COL 29-30, OR LENGTH OF KEY AND KEY START LOCATION GREATER THAN RECORD LENGTH

Code: T—Terminal
Specification Type: F
Explanation: The key field entry in columns 29-30 must be less than 29 characters and must be less than the record length. The sum of the key field starting location plus the key length must not exceed the record length.
System Action: Key field length of 03 is assumed, key field starting location of 01 is assumed. The job is terminated.
User Response: Make the proper key field length (columns 29-30) and key field starting location (columns 35-38) entries. Resubmit the job.

RG544—LENGTH OF RA OR KEY FIELD, COLS 29-30 BLANK OR INVALID, ASSUME 03

Code: T—Terminal
Specification Type: F
Explanation: Columns 29-30 are blank or the entry specified is invalid for files that contain limits or for ADDRROUT files.
System Action: 03 is assumed; the job is terminated.
User Response: Make the entry in columns 29-30 a number from 1 to 29 for files that contain limits and for ADDRROUT files. Resubmit the job.

RG548—FILE ADDITION IS INVALID FOR FILE OR DEVICE, ASSUME BLANK

Code: T—Terminal
Specification Type: F, O
Explanation: File addition (A in column 66) can be specified for sequential and indexed output files on disk only.
System Action: The job is terminated.
User Response: Make the proper file addition entry in column 66 and resubmit the job.

RG549—KEY FIELD START LOCATION IS BLANK OR EXCEEDS RECORD LENGTH

Code: T—Terminal
Specification Type: F
Explanation: Columns 35–38 are blank or the entry specified exceeds the record length in your file description specifications.
System Action: The job is terminated.
User Response: Make the key field starting location entry (1-4096) in columns 35-38 equal to or less than the record length. Resubmit the job.

RG550—NO MORE THAN 20 FILE DESCRIPTION SPECS ALLOWED

Code: T—Terminal
Specification Type: F
Explanation: More than 20 file description lines were specified.
System Action: The job is terminated.
User Response: Specify a maximum of 20 file description lines per program. Resubmit the job.

RG551—RECORD LENGTH MISSING OR INVALID FOR DISK FILE, ASSUME 256

Code: T—Terminal
Specification Type: F
Explanation: The record length entry in columns 24–27 is missing.
System Action: The job is terminated.
User Response: Make the proper record length entry in columns 24–27; it can be a number from 1 to 4096. Resubmit the job.

RG552—FACTOR 1 AND RESULT FIELD MUST NOT BOTH BE BLANK WITH DSPLY OP CODE

Code: T—Terminal
Specification Type: C
Explanation: Both the Result Field and Factor 1 were left blank on a DSPLY operation.
System Action: The job is terminated.
User Response: Make the proper entry under Factor 1 or the Result Field for the DSPLY operation and resubmit the job.

RG553—CORE INDEX IS INVALID FOR DEVICE TYPE OR MODE OF PROCESSING

Code: W—Warning
Specification Type: F
Explanation: Core index can be specified in columns 60–65 for indexed disk files processed randomly.
System Action: Blank is assumed.
User Response: To avoid this message on the next run, make the proper core index entry in columns 60–65 and resubmit the job.

RG554—ADD SPECIFIED ON THE FILE DESCRIPTION SPEC BUT ADD NOT REFERENCED ON OUTPUT

Code: T—Terminal
Specification Type: Not applicable
Explanation: Column 66 contains an A, but record addition ADD in columns 16–18 is not specified in your output specifications.
System Action: The job is terminated.
User Response: Place ADD in columns 16–18 of your output specifications when A is specified in column 66 of file description. Resubmit the job.

RG555—NO ADD SPECIFIED ON FILE DESCRIPTION

Code: T—Terminal
Specification Type: Not applicable
Explanation: ADD is specified in columns 16–18 of your output specifications, but the add function was not specified in file description specifications (col. 66) for this file.
System Action: The job is terminated.
User Response: Place A in column 66 of your file description specifications when ADD is specified in columns 16–18 of the output specifications. Resubmit the job.

RG557—MASK FOR BIT OPERATION IS NOT 0-7

Code: T—Terminal
Specification Type: C
Explanation: The mask specified for the bit operation is not 0-7.
System Action: The job is terminated.
User Response: Specify bits 0-7 as the mask for the bit operation and resubmit the job.

RG558—INVALID USE OF (OR MISSING) RESULTING INDICATORS WITH THIS OP CODE. ASSUME INVALID RESULTING INDICATORS BLANK.

Code: W—Warning
Specification Type: C
Explanation: For CHAIN: Columns 56-59 must be blank. It is suggested that columns 54-55 contain an indicator that can be tested for record not found.
For READ: Columns 54-57 must be blank. It is suggested that columns 58-59 contain an indicator that can be tested for end of file.
System Action: Blank is assumed for columns that must be blank.
User Response: To avoid this message the next time the job is run, make the necessary corrections, as mentioned above.

RG560—MODE OF PROCESSING COL 28 GIVEN BUT NOT ALLOWED, ASSUME BLANK

Code: T—Terminal
Specification Type: F
Explanation: The mode of processing entry specified in column 28 is invalid.
System Action: The job is terminated.
User Response: An entry is allowed only for limits or random processing of disk files.

RG561—KEY FIELD START LOCATION (COLS 35-38) GIVEN BUT NOT ALLOWED, ASSUME BLANK

Code: T—Terminal
Specification Type: F
Explanation: The key field start location entry specified in columns 35-38 is invalid.
System Action: The job is terminated.
User Response: Place the proper entry in columns 35-38 of file description specifications for indexed files only. Resubmit the job.

RG562—FILE TYPE FOR FROM FILENAME AND/OR TO FILENAME INVALID WITH TABLE/ARRAY

Code: T—Terminal
Specification Type: Not applicable
Explanation: The From Filename and/or the To Filename specified is invalid.
System Action: The job is terminated.
User Response: Make sure the From Filename specified in columns 11-18 of extension specifications is an input file and that the To Filename in columns 19-26 is an output file. Resubmit the job.

RG564—RECORD LENGTH IS NOT AT LEAST TWICE THE KEY LENGTH

Code: T—Terminal
Specification Type: F
Explanation: The record length must be at least twice the key length.
System Action: The job is terminated.
User Response: Specify the record length to be at least twice the key length, and resubmit the job.

RG565—COLUMN 31 INVALID FOR DEVICE TYPE

Code: T—Terminal
Specification Type: F
Explanation: The entry in column 31 is valid for update, chained output (or ADDROUT) disk files only.
System Action: The job is terminated.
User Response: Leave column 31 blank or change the file type entry. Resubmit the job.

RG566—INVALID USE OF DEVICE AS FROM FILENAME

Code: T—Terminal
Specification Type: E
Explanation: The file named in columns 11-18 of extension specifications is not assigned to the disk, data recorder, or console.
System Action: The job is terminated.
User Response: Place the proper From Filename entry in columns 11-18 and resubmit the job.

**RG567—TABLE RECORD SIZE GREATER THAN FROM
FILENAME DEVICE RECORD SIZE**

Code: T—Terminal
Specification Type: E
Explanation: Table or array record length specified exceeds the maximum record allowed for the device.
System Action: The job is terminated.
User Response: Make the table or array record length equal to or less than the maximum record length for the device. Resubmit the job.

**RG568—LENGTH OF KEY FIELD OR RA LENGTH
COLS 29-30 GIVEN BUT NOW ALLOWED, ASSUME
BLANK**

Code: T—Terminal
Specification Type: F
Explanation: Length of key field or RA length specified in columns 29-30 is invalid for this file type.
System Action: The job is terminated.
User Response: Leave columns 29-30 blank, and resubmit the job.

**RG569—ENTRY OF I COL 32 NOT GIVEN FOR AN IN-
DEXED FILE, ASSUME I**

Code: T—Terminal
Specification Type: F
Explanation: The entry specified in column 32 for an indexed file is not I.
System Action: I is assumed, the job is terminated.
User Response: Enter I in column 32 for an indexed file and resubmit the job.

**RG570—LOOK AHEAD WITH NUMERIC SEQUENCE
OR LOOK AHEAD FOLLOWS A NUMERIC RECORD**

Code: T—Terminal
Specification Type: I
Explanation: A look ahead record type (** in columns 19-20) cannot be specified on the same line as a numeric sequence entry in columns 15-16.
System Action: The job is terminated.
User Response: Specify look ahead record types (** in columns 19-20) on the same line with an alphabetic entry in columns 15-16. Resubmit the job.

**RG571—MORE THAN ONE LOOK AHEAD RECORD
IN A FILE**

Code: T—Terminal
Specification Type: I
Explanation: Look ahead is specified more than once for this file.
System Action: The job is terminated.
User Response: Make only one look ahead specification for a file. Resubmit the job.

**RG572—LOOK AHEAD CANNOT BE THE ONLY
RECORD IN A FILE**

Code: T—Terminal
Specification Type: I
Explanation: Look ahead records specified do not follow other file or record type specifications.
System Action: The job is terminated.
User Response: Specify look ahead records following other file or record type specifications. Resubmit the job.

RG573—MULTI RA FILES DEFINED

Code: T—Terminal
Specification Type: F
Explanation: More than one record address file is defined in this program.
System Action: The job is terminated.
User Response: Specify only one record address file per program. Resubmit the job.

**RG574—EXTERNAL INDICATOR COLS 71-72 NOT
THE SAME AS RA FILES**

Code: T—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.
System Action: The job is terminated.
User Response: When external indicators are used, specify the same external indicator for both the record address file and the file it is used to process. Resubmit the job.

RG575--NO INPUT SPECIFICATIONS FOUND FOR THIS FILE

Code: T--Terminal
Specification Type: Not applicable
Explanation: Input specifications required for this file, but none were supplied.
System Action: The job is terminated.
User Response: Supply input specifications for all input files (except record address and tables) and for update files. Resubmit the job.

RG576--COMPILE TIME TABLE DATA FOUND. COMPILE TIME TABLE OR ARRAY NOT SPECIFIED IN EXTENSION

Code: W--Warning
Specification Type: E
Explanation: No extension specifications were supplied for compile time table.
System Action: Table data is not processed.
User Response: Supply the proper extension specifications and resubmit the job.

RG577--ONLY ONE FILE ASSOCIATED WITH RA FILE IS ALLOWED IN A PROGRAM

Code: T--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.
System Action: The job is terminated.
User Response: Specify only one record address file per program or associate only one file with a record address file. Resubmit the job.

RG578--RA FILE OR A FILE ASSOCIATED WITH RA FILE IS REQUIRED BUT NOT DEFINED

Code: T--Terminal
Specification Type: F
Explanation: A record address file or a file associated with the RA file is required for this job, but was not defined.
System Action: The job is terminated.
User Response: Supply the proper record address file or file associated with an RA file. Resubmit the job.

RG579--FIRST 1P LINE NOT FOR PRINTER, ASSUME COL 41 IN CONTROL CARD BLANK

Code: W--Warning
Specification Type: O
Explanation: Forms alignment is requested but the first 1P line is not specified for a printer file.
System Action: Column 41 of the control card specifications is assumed to be blank; therefore, no forms alignment is done.
User Response: For forms alignment, specify the first 1P line for a printer file.

RG580--REFERENCED A MATCH LEVEL WHICH IS NOT VALID, OR DEFINED A LEVEL MORE THAN ONCE

Code: T--Terminal
Specification Type: I
Explanation: Either an invalid match level is used or a match level is defined more than once.
System Action: The job is terminated.
User Response: Be sure that each record group contains the same match levels, and that each match level is defined only once. Resubmit the job.
job.

RG581--MISSING OR INVALID AN/OR ENTRY IN COL 7-8

Code: T--Terminal
Specification Type: C
Explanation: An AN/OR entry in columns 7-8 is missing or the entry specified is not AN or OR.
System Action: The job is terminated.
User Response: Make the proper AN/OR entry in column 7-8 and resubmit the job.

RG582—THE RELATIVE RECORD NUMBER FOR THE CHAIN OPERATION MUST BE NUMERIC WITH 0 DECIMAL

Code: W—Warning
Specification Type: C
Explanation: The relative record number specified for a CHAIN operation is not a numeric field with zero decimal positions.
System Action: The decimal positions are ignored.
User Response: To avoid this message the next time this job is run, make the relative record number for a CHAIN operation a numeric field with zero decimal positions.

RG583—BINARY LENGTH SPECIFIED GREATER THAN 9, ASSUME 9

Code: T—Terminal
Specification Type: I
Explanation: The binary length specified is greater than 9.
System Action: The job is terminated.
User Response: Make the binary length entry 9 or less and resubmit the job.

RG584—THIS MATCH LEVEL WAS REFERENCED PREVIOUSLY IN THIS RECORD GROUP

Code: T—Terminal
Specification Type: I
Explanation: A match level was referenced more than once within one record group.
System Action: The job is terminated.
User Response: Be sure that each match level is referenced only once within a record group. Resubmit the job.

RG585—DISPLAY, CHAIN, OR DEMAND FILE SPECIFIED, BUT APPROPRIATE OPERATION CODE NOT FOUND IN CALCULATION SPECIFICATIONS

Code: T—Terminal
Specification Type: C
Explanation: Display, chain, or demand files are specified but the appropriate operation codes are not specified in calculation specifications.
System Action: The job is terminated.
User Response: Specify the appropriate operation code and resubmit the job.

RG586—MORE THAN ALLOWABLE TABLE/ARRAY NAMES USED IN THE PROGRAM

Code: T—Terminal
Specification Type: E
Explanation: More than 60 compile-time tables and/or arrays were defined or a total of more than 63 tables or arrays were defined in this program.
System Action: The job is terminated.
User Response: Reduce the number of compile-time tables and/or arrays to 60 or less and the total number of tables and/or arrays to 63 or less.

RG587—IF FACTOR 1 OR FACTOR 2 IS A WHOLE ARRAY, RESULT FIELD MUST BE WHOLE ARRAY

Code: T—Terminal
Specification Type: C
Explanation: The entry in Factor 1 or Factor 2 is a whole array, but the Result Field does not refer to a whole array.
System Action: The job is terminated.
User Response: When the entry in Factor 1 or Factor 2 is a whole array, place an array name in the Result Field. Resubmit the job.

RG588—TESTB, BITON, AND BITOF MAY NOT REFERENCE AN ENTIRE ARRAY

Code: T—Terminal
Specification Type: C
Explanation: An entire array must not be referenced in a TESTB, BITON, or BITOF operation.
System Action: The job is terminated.
User Response: When using arrays with TESTB, BITON, or BITOF operations, specify array elements not the whole array. Resubmit the job.

RG589—RESULT FIELD MUST BE A ONE-POSITION ALPHAMERIC FIELD. IF FACTOR 2 IS A FIELD NAME, IT MUST BE A ONE-POSITION ALPHAMERIC FIELD

Code: T—Terminal
Specification Type: C
Explanation: The Result Field is not a one-byte alphameric field for TESTB, BITON, and BITOF, or Factor 2 is a field name but is not a one-byte alphameric entry.
System Action: The job is terminated.
User Response: Make the Result Field a one-byte alphameric field for TESTB, BITON, or BITOF. If Factor 2 contains a field name, make it a one-byte alphameric field. Resubmit the job.

RG590—WHENEVER HIGH IS USED IN A MOVE ZONE OPERATION, IT MUST REFERENCE AN ALPHAMERIC FIELD

Code: T—Terminal
Specification Type: C
Explanation: The high portion of a move zone instruction does not reference an alphameric field.
System Action: The job is terminated.
User Response: Make the high portion of a move zone instruction reference an alphameric field and resubmit the job.

RG591—LENGTH OF FIELD IN FACTOR 1 NOT EQUAL TO KEY LENGTH OF FILE SPECIFIED IN FACTOR 2

Code: T—Terminal
Specification Type: C
Explanation: The length of the field in Factor 1 of a CHAIN operation is not equal to the key field length specified in Factor 2.
System Action: The job is terminated.
User Response: For a CHAIN operation, make the length of the chaining field (Factor 1) equal to the length of the key field (Factor 2). Resubmit the job.

RG592—FOR SEQUENTIALLY PROCESSED UPDATE FILE—T ENTRY IN COL 15 IS INVALID OR L0-L9 INDICATOR USED WITH E IN COL 15

Code: T—Terminal
Specification Type: O
Explanation: Total output cannot be specified for update files processed sequentially.
System Action: The job is terminated.
User Response: Remove the T or E entry from column 15 and resubmit the job.

RG593—TABLE/ARRAY NAME MISSING FOR 'TO' AND/OR 'FROM' FILENAME

Code: W—Warning
Specification Type: E
Explanation: No table name was specified in columns 27-32 for a table load operation (From Filename in columns 11-18) or for a table output operation (To Filename in columns 19-26).
System Action: No action taken.
User Response: To avoid the message when this job is run again, specify the proper table name in columns 27-32.

RG594—'TO' FILENAME MAY NOT BE USED WITH EXECUTION TIME TABLE/ARRAY

Code: T—Terminal
Specification Type: E
Explanation: An array output operation (To Filename in columns 19-26) must not be specified for execution time arrays.
System Action: The job is terminated.
User Response: Remove the To Filename entry in columns 19-26 for execution time arrays. Resubmit the job.

RG595—COLS 27-32 AND 46-51 MUST BE BOTH TABLE OR BOTH ARRAY NAMES

Code: T—Terminal
Specification Type: E
Explanation: For alternating tables, columns 27-32 and 46-51 do not both contain table names; or columns 27-32 and 46-51 do not both contain array names for alternating arrays.
System Action: The job is terminated.
User Response: For alternating tables or arrays, specify either table names or array names in both columns 27-32 and 46-51. Resubmit the job.

RG596—INDICATORS MAY NOT BE USED WITH *PLACE

Code: T—Terminal
Specification Type: O
Explanation: *PLACE is conditioned automatically by the same indicators which conditioned the line or lines to be repeated.
System Action: The job is terminated.
User Response: Remove the conditioning indicators from the *PLACE statement and resubmit the job.

RG597—END POSITION SPECIFIED FOR *PLACE LESS THAN TWICE THAT OF HIGHEST PREVIOUSLY SPECIFIED FIELD END POSITION OR END POSITION GREATER THAN 256

Code: T—Terminal
Specification Type: O
Explanation: The end position specified for *PLACE is lower than end position specified for the preceding field or is greater than 256.
System Action: The job is terminated.
User Response: Make the proper end position entry for *PLACE, and resubmit the job.

RG598—ALPHA TABLE/ARRAY SPECIFIED AS PACKED, ASSUME NUMERIC

Code: T—Terminal
Specification Type: E
Explanation: An alphameric table or array was specified as packed.
System Action: The job is terminated.
User Response: Specify the table or array as numeric, and resubmit the job.

RG599—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

Code: T—Terminal
Specification Type: E
Explanation: The binary length was not specified as 4 or 9.
System Action: The job is terminated.
User Response: Make the proper binary length entry and resubmit the job.

RG602—CONSOLE AND KEYBOARD BOTH SPECIFIED

Code: T—Terminal
Specification Type: F
Explanation: Both device names CONSOLE and KEYBOARD specified in this program.
System Action: The job is terminated.
User Response: Specify either CONSOLE or KEYBOARD but not both in this program. Resubmit the job.

RG603—INPUT SPECIFICATION FROM KEYBOARD INVALID

Code: T—Terminal
Specification Type: I
Explanation: Records to be supplied from the keyboard are specified in your input specifications.
System Action: The job is terminated.
User Response: Specify keyboard operations with the KEY operation code in calculation specifications, not in input specifications. Resubmit the job.

RG604—INVALID OR MISSING FIELD LIGHTS SPECIFIED ON SET OR KEY OPERATION

Code: T—Terminal
Specification Type: C
Explanation: Field lights entries in columns 31–32 are missing or the entries specified are not 0, blank, or 1–8.
System Action: The job is terminated.
User Response: Make the proper field light entry in columns 31–32 and resubmit the job.

RG605—SAME VALID FIELD LIGHT USED IN COLUMNS 31 AND 32

Code: W—Warning
Specification Type: C
Explanation: The same field light entry is specified in both column 31 and column 32.
System Action: The field light specified is turned on.
User Response: To avoid this message the next time this job is run, specify different field lights in column 31 and column 32.

RG607—TAB SET MUST BE FOLLOWED BY KEY IN MANUAL MODE

Code: T—Terminal
Specification Type: C
Explanation: The tab set positions specified are not immediately followed by a KEY operation.
System Action: The job is terminated.
User Response: Specify the proper tab set position immediately before a KEY. Resubmit the job.

RG608—TAB SET POSITION INVALID

Code: T—Terminal
Specification Type: C
Explanation: Tab set position in columns 49–51 is not specified properly or is not a number from 1–132.
System Action: The job is terminated.
User Response: Make the proper tab set entry in columns 49–51 and resubmit the job.

RG609—KEYBOARD SPECIFIED AS PRIMARY FILE BUT FUNCTION NEVER USED

Code: T—Terminal
Specification Type: C
Explanation: The keyboard is specified as the primary file but it is not used.
System Action: The job is terminated.
User Response: Use normal input or input from KEYBORD using the KEY operation code. Resubmit the job.

RG610—ONLY COMMAND KEY INDICATORS VALID AS RESULTING INDICATORS ON SET OPERATION

Code: T—Terminal
Specification Type: C
Explanation: The resulting indicator entries specified in column 54–59 of a SET operation are not command key indicators.
System Action: The job is terminated.
User Response: Specify the proper command key indicators in columns 54–59 of the SET operation and resubmit the job.

RG611—SKIPXX, SPACEX, OR POSITION OF PRINT ELEMENT GIVEN WITHOUT FILENAME

Code: T—Terminal
Specification Type: C
Explanation: No filename is specified in columns 33–42 of a SET operation to perform skipping, spacing, or positioning the print element.
System Action: The job is terminated.
User Response: Specify, in columns 33–42 of a SET operation, the name of the file that skipping, spacing, or positioning of print element is to be performed on. Resubmit the job.

RG612—POSITION OF PRINT ELEMENT OUT OF LIMITS FOR FILE

Code: T—Terminal
Specification Type: C
Explanation: The entry for positioning the print element specified in columns 49–51 exceeds the record length defined for this file in file description specifications.
System Action: The job is terminated.
User Response: Make the entry for positioning the print element equal to or less than the record length, and resubmit the job.

RG613—X FOR SPACEX OR XX FOR SKIPXX INVALID

Code: T—Terminal
Specification Type: C
Explanation: 1. The SKIPxx entry in columns 43–48 is not 01–99, A0–A9, or B0–B2 or it is greater than the form length.
2. The SPACEx entry in columns 43–48 is not 1–3.
System Action: The job is terminated.
User Response: Make the proper SKIPxx or SPACEx entry in columns 43–48 and resubmit the job.

RG614—OPERATION REQUIRES KEYBOARD SPEC ON FILE DESCRIPTION

Code: T—Terminal
Specification Type: C
Explanation: The KEYBOARD file is not defined in your file description specifications, but KEY and SET operations are used.
System Action: The job is terminated.
User Response: Make the proper file description specifications for the KEYBOARD file and resubmit the job.

RG615—CANNOT KEY FIELD GREATER THAN 128 CHARACTERS

Code: T—Terminal
Specification Type: C
Explanation: The field entered from the keyboard is more than 128 characters long.
System Action: The job is terminated.
User Response: Make fields to be keyed into the system 128 characters or less. Resubmit the job.

RG616—KEYBOARD MUST HAVE PRIMARY OR DEMAND DESIGNATION, ASSUME DEMAND

Code: T—Terminal
Specification Type: F
Explanation: The file assigned to the keyboard is not specified as a primary or a demand file.
System Action: Demand is assumed; the job is terminated.
User Response: Designate the file assigned to the keyboard as either primary (P in column 16) or demand (D in column 16). Resubmit the job.

RG617—RESULT FIELD MAY NOT BE AN ARRAY

Code: T—Terminal
Specification Type: C
Explanation: The entry in Result Field (columns 43–48) must not be an array name.
System Action: The job is terminated.
User Response: Remove the array name as an entry in Result Field; make the proper entry, and resubmit the job.

**RG618—CANNOT HAVE ANY SECONDARY FILES
WHEN KEYBOARD IS SPECIFIED AS PRIMARY**

Code: T—Terminal
Specification Type: F
Explanation: When the keyboard is specified as a primary file, no other file can be specified as primary or secondary.
System Action: The job is terminated.
User Response: Remove the secondary file designation entry (column 16) and resubmit the job.

**RG621—TRAILER RECORD OVERLAPS HEADER
RECORD**

Code: T—Terminal
Specification Type: I
Explanation: The trailer field overlaps the header field in a spread card.
System Action: The job is terminated.
User Response: Make the first trailer field start after the last position in the header field. Resubmit the job.

RG622—NO TRAILER FIELDS FOR SPREAD CARD

Code: T—Terminal
Specification Type: I
Explanation: No trailer fields are specified for the spread card.
System Action: The job is terminated.
User Response: Make the proper trailer field entries for the spread card (TR in columns 19-20). Resubmit the job.

**RG623—ENTRIES IN COLUMNS 7-18 AND 21-74
INVALID FOR TR SPECIFICATION, ASSUME NO TR**

Code: T—Terminal
Specification Type: I
Explanation: Entries specified in columns 7-18 and 21-74 of a TR line.
System Action: Columns 19-20 are assumed blank; no spread cards are accepted. The job is terminated.
User Response: If spread cards are to be used, leave columns 7-18 and 21-74 blank for the TR line (TR in columns 19-20). Resubmit the job.

RG624—TR SPECIFICATION OUT OF ORDER

Code: T—Terminal
Specification Type: I
Explanation: The TR specification line is not preceded by a definition of a header record.
System Action: The job is terminated.
User Response: Place the TR specification line immediately after a definition of a header record. Resubmit the job.

**RG625—FACTOR 1 MUST BE NUMERIC FOR CHAIN
OPERATION WHEN FACTOR 2 FILENAME HAS
PACKED KEYS**

Code: T—Terminal
Specification Type: C
Explanation: The entry specified in Factor 1 of a CHAIN operation is not numeric even though the file named in Factor 2 has packed keys.
System Action: The job is terminated.
User Response: Make the entry in Factor 1 of a CHAIN operation numeric when the file named in Factor 2 has packed keys. Resubmit the job.

**RG626—MORE THAN 255 TR SPECIFICATIONS
GIVEN**

Code: T—Terminal
Specification Type: I
Explanation: More than 255 valid TR lines are specified in this program.
System Action: The job is terminated.
User Response: Make the number of valid TR lines in this program 255 or less. Resubmit the job.

RG627—FACTOR 2 GIVEN ON SET OPERATION WITHOUT EITHER RESULT FIELD OR POSITION OF PRINT ELEMENT

Code: T—Terminal
Specification Type: C
Explanation: A filename is specified in Factor 2 of a SET operation without an entry in either Result Field (columns 43–48) or Field Length (columns 49–51).
System Action: The job is terminated.
User Response: If the entry in Factor 2 is valid, make the proper entry in columns 43–48 or 49–51 of the SET operation. Resubmit the job.

RG628—INVALID FILE TYPE FOR SPREAD CARD CARD

Code: T—Terminal
Specification Type: I
Explanation: The file containing spread cards is not a card input file designated as primary or secondary.
System Action: The job is terminated.
User Response: Make sure the file containing spread cards is a card input file designated as primary or secondary. Resubmit the job.

RG629—BLOCK LENGTH, START PRINT POSITION, INVALID OR NOT GIVEN

Code: T—Terminal
Specification Type: F
Explanation: The block length or start print position is not specified or is not specified properly.
System Action: The job is terminated.
User Response: Make the proper block length or start print position entry in columns 20–23 of file description specifications and resubmit the job.

RG630—TWO OUTPUT FILES, EITHER TRACTR1, TRACTR2, OR LEDGER CARD, HAVE OVERLAPPING POSITIONS.

Code: W—Warning
Specification Type: F
Explanation: If overlapping print positions are not desired, make the proper entries in the file description specifications and resubmit the job. If overlapping is desired, continue with the job.
System Action: A warning message is given
User Response: Verify if you intended to have overlays.

RG631—FACTOR 1 MUST HAVE SAME LENGTH WHEN PACKED AS LENGTH OF PACKED KEYS FOR FACTOR 2 FILENAME

Code: T—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.
System Action: The job is terminated.
User Response: Make sure the entry in Factor 1 of a CHAIN operation is the same length when packed as the record key in file named in Factor 2. Resubmit the job.

RG632—SET OR KEY OPERATION SPECIFIED WITH NO FUNCTION

Code: T—Terminal
Specification Type: C
Explanation: The SET or KEY operation code was entered in columns 28–32 without specifying the SET or KEY operation to be performed.
System Action: The job is terminated.
User Response: Specify the function of the SET or KEY operation you want to perform (columns 7–27, 33–39) and resubmit the job.

RG633—FACTOR 2 MUST BE A PRINTER FILE

Code: T—Terminal
Specification Type: C
Explanation: Factor 2 (columns 33–42) must contain a printer filename in order to perform the operation specified.
System Action: The job is terminated.
User Response: Enter the proper printer filename in columns 33–42 and resubmit the job.

RG635—NUMERIC SEQUENCE CHECKING SPECIFIED FOR A SPREAD CARD, BUT N NOT SPECIFIED FOR NUMBER, ASSUME N

Code: W—Warning
Specification Type: I
Explanation: An N entry was not made in column 17 even though sequence checking was specified (numeric entry in columns 15–16).
System Action: N is assumed.
User Response: To avoid this message when this job is run again, enter N in column 17.

RG636—CONDITIONING INDICATORS INVALID

Code: T—Terminal
Specification Type: C
Explanation: Conditioning indicators in columns 9–17 cannot be used with this operation.
System Action: The job is terminated.
User Response: Remove the conditioning indicators from columns 9–17 and resubmit the job.

RG637—LEDGER EJECT SPECIFIED BUT NO LEDGER IN PROGRAM

Code: T—Terminal
Specification Type: C
Explanation: No ledger file defined, but a SET calculation specification specified EJECT.
System Action: The job is terminated.
User Response: Either define the ledger file or remove the calculation specifying EJECT.

RG638—CARBON COPY SPECIFIED BUT NO LEDGER IN PROGRAM

Code: T—Terminal
Specification Type: F
Explanation: Block length and record length have been specified as 1 for TRACTR2, but no ledger file has been defined.
System Action: The job is terminated.
User Response: Either define the ledger file or specify a valid block length and record length for TRACTR2.

RG640—END OF FILE INDICATOR IGNORED FOR READ FROM LEDGER

Code: W—Warning
Specification Type: C
Explanation: An end-of-file indicator has been specified in columns 58–59 on the calculation specification, but end of file is not allowed for a ledger file.
System Action: The indicator is ignored.
User Response: Remove the end-of-file indicator specification.

RG641—*BOTH SPECIFIED WITHOUT VALID TRACTR1 AND TRACTR2 SPECIFICATIONS

Code: T—Terminal
Specification Type: C
Explanation: TRACTR1 and TRACTR2 must be defined if *BOTH is specified.
System Action: The job is terminated.
User Response: Define TRACTR1 and TRACTR2 or remove the calculation specification for *BOTH.

RG642—LEDGER CARD CONTROL NUMBER, N IN COL 40, MUST APPEAR BEFORE ANY NORMAL OUTPUT FIELDS IN RECORD

Code: T—Terminal
Specification Type: O
Explanation: Ledger card control number was not specified before the output fields.
System Action: The job is terminated.
User Response: Specify the control number first.

**RG643—CONTROL NUMBER MAY NOT BE USED WITH
A LEDGER CARD FILE THAT USES MATCHING
RECORDS**

Code: T—Terminal
Specification Type: O
Explanation: Control number was specified for
a ledger card file that uses matching
records.
System Action: The job is terminated.
User Response: Do not specify a control number
for the file.

**RG645—IMPROPER USE OF THE PACK/UNPACK
FEATURE FOR LIMITS FILE PROCESSING**

Code: T—Terminal
Specification Type: E
Explanation: The unpacked key length must be
one or two less than twice the packed
key length.
System Action: The job is terminated.
User Response: Correct the unpacked key length
and resubmit the job.

**RG646—RESULT FIELD FOR XFOOT MAY NOT BE
AN ENTIRE ARRAY**

Code: T—Terminal
Specification Type: C
Explanation: An entire array can not be specified
as the result field for XFOOT.
System Action: The job is terminated.
User Response: Make the result field a numeric
field, an array element, or a table
element. Resubmit the job.

**RG647—UNEQUAL KEY LENGTHS SPECIFIED FOR
KEYS OF IDENTICAL FORMAT**

Code: W—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.
System Action: The key length of the file processed
by limits is assumed as the key
length of the limits file.
User Response: Specify the key lengths to be equal,
to eliminate the message.

RG799—ERROR FILE FULL

Code: W—Warning
Specification Type: Not applicable
Explanation: Too many errors were made in this
program.
System Action: No action taken.
User Response: To avoid this message when this
job is run again, correct the errors
already diagnosed in this program.

**RG999—PROGRAM EXCEEDS CORE IN COL 12-14 OF
HEADER CARD**

Code: W—Warning
Specification Type: H
Explanation: The program requires more core
storage for execution than specified
in columns 12-14 of the control
card specifications.
System Action: No action taken.
User Response: To avoid this message when this
job is run again, make the proper
entry in columns 12-14.

Appendix C: Summary of RPG II Specifications

This appendix contains a brief column-by-column description of each of the RPG II specification sheets. It is intended as a quick reference by programmers who are acquainted with RPG II for the IBM System/3 Model 6. For a complete description of each entry, refer to the applicable section of this manual. For Telecommunications specification descriptions, see *IBM System/3 RPG II Telecommunications Program Reference Manual, SC21-7507*.

INFORMATION COMMON TO ALL FORMS

RPG II source records should be in ascending numeric sequence by columns 1 through 5. Records that are out of sequence are flagged. Adjacent records with duplicate sequence numbers are not flagged.

Columns 1-2 (Page)

Arrange the specifications sheets in the following order and number them in ascending sequence:

1. Control Card and File Description.
2. Extension and Line Counter.
3. Telecommunications.
4. Input.
5. Calculation.
6. Output-Format

Columns 3-5 (Line)

The first two digits of the line number are preprinted. Use the unnumbered lines on the sheet for additional specifications or, along with column 5, to insert a line between two other completed lines. For example, line 025 would be inserted between lines 02 and 03.

Column 6 (Form Type)

This column contains a preprinted code (H, F, E, L, T, I, C, or O) which must be placed in all RPG II specification records.

Column 7 (Comments)

Enter an asterisk in each line used as a comment line. The control card specification line (line 01) cannot be used as a comment line.

Columns 75-80 (Program Identification)

Insert any valid characters in column 75-80 of the control card specifications to identify the program. This name is used in a program directory which contains the location of your program on disk. If these columns are left blank, RPGOBJ is assumed. Columns 75-80 on all other specification sheets except the Control Card Specifications sheet can contain any entries.

CONTROL CARD SPECIFICATIONS

Columns 7-9 (Core Size to Compile)

Blank

Column 10 (Object Output)

- Blank — Object program written temporarily in object library
D — Object program written temporarily in object library
C — Object program written permanently in object library
P — Object program punched into cards

Column 11 (Listing Options)

- Blank — Program listing produced
B — No program listing produced
P — Partial program listing produced

Columns 12-14 (Core Size to Execute)

Column 12

- Blank, 0 — No additional 256-byte increments are needed.
Q — One additional 256-byte increment is needed.
H — Two additional 256-byte increments are needed.
T — Three additional 256-byte increments are needed.

Columns 13-14

- Blank – Core size available for execution is same as core size used for compilation.
- 01-13 – Core size available for execution if different from core size used for compilation. Entry is the number of K (1K=1,024 bytes) available.

Column 15 (Debug)

- Blank – DEBUG operation not used
- 1 – DEBUG operation used

Columns 16-20

Blank

Column 21 (Inverted Print)

- Blank – Domestic format
- I – World Trade format
- J – World Trade format (leading zero remains for zero balances)
- D – United Kingdom format

Columns 22-25

Blank

Column 26 (Alternate Collating Sequence)

- Blank – Normal collating sequence used
- S – Alternate collating sequence used

Columns 27-36

Blank

Column 37 (Inquiry)

- Blank – Program not interruptable

- B – Program recognizes inquiry requests
- I – Inquiry program

Columns 38-40

Blank

Column 41 (1P Forms Position)

- Blank – First 1P line printed only once
- 1 – First 1P line can be printed repeatedly to allow forms positioning

Column 42

Blank

Column 43 (File Translation)

- Blank – No file translation needed
- F – Input, output, update, or combined files are to be translated

Column 44 (Punch MFCU Zeros)

- Blank – Leading zeros removed
- 1 – Leading zeros used

Columns 45-47

Blank

Column 48 (Shared I/O)

- Blank – Each disk file has an input/output area
- 1 – Shared input/output area for all disk files

Column 49 (Field Print)

- Blank – System will determine the mode of printing
- 1 – Printing will be done in field mode, from left to right across the page

Columns 50-74

Blank

FILE DESCRIPTION SPECIFICATIONS

Columns 7-14 (Filename)

Enter a name for each file. The filename can be from one to eight characters long, must begin in column 7, and must be a valid RPG II name.

Column 15 (File Type)

I – Input
O – Output
U – Update
C – Combined
D – Display

Column 16 (File Designation)

P – Primary
S – Secondary
C – Chained
R – Record Address
T – Table or Array
D – Demand

Leave blank for display files and all output files except chained output files.

Column 17 (End of File)

E – All records from the file must be processed before the program can end
Blank – The program can end whether or not all records from this file have been processed

If column 17 is blank or contains E for all files, all records from every file must be processed before the program can end. An E can only be specified here if column 15 contains I, U, or C and column 16 contains a P, S, or R.

Column 18 (Sequence)

Blank – No sequence checking is to be done
A – Sequence checking is done. Records are in ascending sequence
D – Sequence checking is done. Records are in descending sequence

Sequence checking is required when matching fields are used. Column 18 applies to update and combined files and all input files except table, array, chained, demand, and record address files.

Column 19 (File Format)

F must be entered.

Columns 20-23 (Block Length)

1-4096 – Disk (record length or multiple of record length)
1-96 – Data Recorder
1-125 – Console
1-128 – Keyboard
83-214 – Ledger (start print position)
1-220 – Printer (220 position, start print position)
1-132 – Printer (132 position, start print position)
1 – TRACTR2 (carbon copy)
1-64 – CRT
1-4096 – SPECIAL

These columns can be left blank for any file except printer (TRACTR1, TRACTR2) and ledger card files.

Columns 24-27 (Record Length)

1-4096 – Disk
1-96 – Data Recorder
1-125 – Console
1-128 – Keyboard
83-214 – Ledger (end print position)
1-220 – Printer (220 position, end print position)
1-132 – Printer (132 position, end print position)
1 – TRACTR2 (carbon copy)
1-64 – CRT
1-4096 – SPECIAL

Column 28 (Mode of Processing)

Blank – ● Sequential by key
 ● Consecutive
L – Sequential within limits
R – ● Random by relative record number
 ● Random by key
 ● By ADDROUT file
 ● Direct file load (random load)

This column must be blank for non-disk files.

Columns 29-30 (Length of Key Field or Record Address Field)

- 1-8 — Length of record keys in packed format (for indexed files)
- 1-29 — Length of record keys in unpacked format (for indexed files and record address files)
- 3 — Length of disk address in ADDRROUT files

Column 31 (Record Address Type)

- P — Indexed file with packed keys
- A — Indexed file
- I — ADDRROUT file or processed by ADDRROUT file
- Blank — Sequential or direct file

Column 31 applies to disk files specified as input, update, or chained output files.

Column 32 (File Organization of Additional I/O Area)

- I — Indexed organization
- T — ADDRROUT file
- 1-9 — Sequential or direct file, use two I/O areas for the file
- Blank — Sequential or direct file, use one I/O area for the file

Columns 33-34 (Overflow Indicator)

- OA-OG, OV — Overflow indicator used to condition records in the file
- Blank — No overflow indicator used

Columns 35-38 (Key Field Starting Location)

For indexed files, enter the beginning position of the key field in the record.

Column 39 (Extension Code)

- E — The file described on this line is a table file, array file, or record address file further described in extension specifications
- L — The file described on this line is a printer file further described in line counter specifications

Columns 40-46 (Device)

Enter the device code for the input/output device used by the file specified in columns 7-14 as follows:

- DISK — Disk
- KEYBOARD — Keyboard
- LEDGER — Ledger Card Device
- TRACTR1 — 132-position printer or primary print unit of 220-position printer
- TRACTR2 — Secondary print unit of 220-position printer
- CONSOLE — Keyboard
- DATA96 — Data Recorder
- CRT65 — Cathode Ray Tube
- SPECIAL — Device not supported by RPG II
- BSCA — Binary Synchronous Communications Adapter

Columns 47-52

Blank

Column 53

Continuation Lines

- K — Continuation record specified for tape

Columns 54-59

Name of Label Exit

- Blank — No SPECIAL device used
- SUBRxx — Name of the user-written subroutine which will perform the I/O operations for a SPECIAL device
- SRyzzz — Name of the IBM-written subroutine which will perform the I/O operations for a SPECIAL device

Continuation Line Option

- Array name — Name of array to be used by user-written IOS subroutine.

Columns 60-65 (Core Index)

- 6-9999 — Number of bytes reserved for core index
- Blank — No core index will be kept in storage

Column 66 (File Addition/Unordered)

- A — New records will be added to the file
- U — Records are to be loaded into an indexed file in unordered sequence

This column applies to sequential and indexed disk files.

Column 67

Blank

Columns 68-69 (Number of Extents)

- Blank — Single-volume file
- 1-50 — Number of volumes that contain the file

For consecutive processing, if any volumes are off-line during processing, then all volumes must be on removable disks. For sequential or random processing, all volumes must be on-line.

Column 70

Blank

Columns 71-72 (File Condition U1-U8)

- U1-U8 — File is conditioned by the specified external indicator
- Blank — File is not conditioned by an external indicator

These columns apply to output files and primary and secondary input (except table or array input files), update, and combined files. A record address file may be conditioned by an external indicator if its associated primary or secondary file is conditioned either by the same indicator or by no indicator.

Columns 73-74

Blank

EXTENSION SPECIFICATIONS**Columns 7-10**

Blank

Columns 11-18 (From Filename)

Enter, left justified, the name of the table or array input file loaded at pre-execution time or the name of the record address file defined on the File Description Sheet.

Columns 19-26 (To Filename)

If the file named in From Filename (columns 11-18) is a record address file, enter the name of the primary or secondary input or update file containing the data records to be processed. If the file named in columns 11-18 is a table or array file, enter the name of the output file to which the table or array is written at end of job. Leave these columns blank if the table or array is not written out.

Columns 27-32 (Table or Array Name)

Enter the 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 are left-justified and must be valid RPG II names. Table names must begin with TAB; array names must not begin with TAB.

Columns 33-35 (Number of Entries Per Record)

Enter, right-justified, the number of entries on each table or array input record. These columns must contain an entry for compile and pre-execution time tables and arrays. These columns must be blank for execution time arrays.

Columns 36-39 (Number of Entries Per Table or Array)

Enter, right-justified, the maximum number of entries in the table or array named in columns 27-32. For alternating tables or arrays, corresponding items are considered one entry.

Columns 40-42 (Length of Entry)

Enter, right justified, the length of each table or array entry. The maximum length is 256 for alphameric entries and 15 for numeric entries. For packed or binary tables and arrays, enter the number of bytes of storage required to represent the data in unpacked format.

Column 43 (Packed or Binary Field)

- Blank — Alphameric or unpacked numeric data
- P — Packed numeric data
- B — Binary numeric data

Column 44 (Decimal Positions)

- Blank – Alphameric table or array
- 0-9 – Number of positions to the right of the decimal

Column 45 (Sequence)

- Blank – No particular sequence
- A – Ascending sequence
- D – Descending sequence

This column describes the sequence of data in a table or array. Column 45 must contain an entry if high or low look-up is to be used.

Columns 46-57

Use these columns when describing a second table or array entered in alternating format with the table or array named in columns 27-32. These entries have the same significance as the corresponding entries in columns 27-45.

Columns 58-74 (Comments)

Enter any information you wish to help you understand or remember what you are doing in each specification line.

LINE COUNTER SPECIFICATIONS

Columns 7-14 (Filename)

Enter the name of a printer file for which you wish to specify a form size and overflow line.

Columns 15-17 (Line Number—Number of Lines Per Page)

- 1-112 – Number of lines available for printing on the printer form

Columns 18-19 (Form Length)

Enter FL to indicate the previous entry is the form length.

Columns 20-22 (Line Number—Overflow Line)

- 1-112 – Number of the overflow line

Columns 23-24 (Overflow Line)

Enter OL to indicate the previous entry is the overflow line.

Columns 25-74

Blank

TELECOMMUNICATIONS SPECIFICATIONS

Columns 7-14 (Filename)

Enter a valid filename for every BSC file your program uses.

Column 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/3 cannot be the control station.
- S – Switched network.

Column 16 (Type of Station)

- T – This station will transmit messages from this file (transmit only or transmit with reception of conversational reply). The file must be designated as an output or combined file on the File Description sheet and must appear on the Output-Format sheet.
- R – This station will receive messages into this file (receive only or receive with transmittal of conversational reply). The file must be designated as an input or combined file on the File Description sheet and must appear on the Input sheet.

Column 17 (Type of Control)

- T — This is a tributary station on a multipoint network. System/3 cannot be the control station and transmit the polling supervisory sequence.
- Blank — Polling is not used.

Column 17 must contain a T if column 15 contains an M (multipoint network).

Column 18 (Type of Code)

- A, U — ASCII data link control characters will be used. When ASCII is used, each station must provide file translation when it is required.
- E or blank — EBCDIC data link control characters will be used.

Column 19 (Transparency)

- Y — This entry is valid only for EBCDIC. The transparency feature must be installed. The data being transferred may contain data link control characters.
- N or blank — The transparency feature is not used. Unpacked numeric or alphanumeric data will be transmitted and received. The data being transferred may not contain data link control characters.

Column 20 (Switched)

- M — The computer operator makes the connection between stations by dialing the number (manual dial).
- E — Autocall is to be used. The dial number is listed in columns 21-31.
- S — Autocall is to be used. The entry in columns 21-31 is the symbolic location of the dial number.
- A — Autoanswer is used by the called station.
- B — Manual answer is used by the called station.
- Blank — This is not a switched network.

Columns 21-31 (Dial Number)

- Numeric — This is the number to be dialed when column 20 contains an E.
- Alphanumeric — Columns 21-31 must contain a symbolic name, other than an array name, referencing the location of the dial number when column 20 contains an S. If the BSC file is an input file other than a demand or conversational receive file, this name must refer to the first (or only) element of a table.

Column 32 (Location of Identification—This Station)

- S — Switched network. This station's identification is located at the position referenced by the symbolic name specified in columns 33-39.
- E — Switched network. The entry in columns 33-39 is this station's identification.
- Blank — This is a nonswitched network or a switched network where no ID is desired for this station.

Columns 33-39 (Identification—This Station)

- Alphanumeric — When column 32 contains an E, this entry is the actual identification sequence of this station (from two to fifteen 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 BSC file is primary or secondary, this symbolic name must refer to the first element of a table.

Column 40 (Location of Identification—Remote Station)

- S — Switched network. The remote station's identification is located at the position referenced by the symbolic name specified in columns 41-47.
- E — Switched network. The entry in columns 41-47 is the remote station's identification.
- Blank — This is a nonswitched network or a switched network where no ID is desired for the remote station.

Columns 41-47 (Identification—Remote Station)

Alphanumeric — When column 40 contains an E, this entry is the actual identification sequence of the remote station (from two to fifteen 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 of the remote station's identification. The symbolic name must not be an array name. If the BSC file is primary or secondary, this symbolic name must refer to the first element of a table.

Columns 48-51 (Remote Terminal)

Blank

Column 52 (ITB)

I — Intermediate block check (ITB) is used.

Columns 53-54 (Permanent Error Indicator)

01-99, L1-L9, LR, H1-H9 — A permanent error indicator should be used with every BSC file. If you are using more than one BSC file, each should have a permanent error indicator. BSC input/output operations must be conditioned on all permanent error indicators being off.

Columns 55-57 (Wait Time)

Numeric — The length of time in seconds, 1-999, that BSC will wait with no messages being sent or received before a permanent error condition occurs.

Blank — The system convention for timeout, 180 seconds, is used.

Columns 58-59 (Record Available Indicator)

01-99, L1-L9, LR, H1-H9 — A record available indicator is used only when System/3 transmits interspersed with receive (no conversational reply) to System/360. The record available indicator is set on when System/360 wishes to transmit to System/3.

Column 60 (Last File)

L — This BSC input file is processed only after all other primary and secondary input files have been processed.

Blank — This BSC input file does not have to be the last input file processed.

Columns 61-62 (Polling Characters)

Alphanumeric — The polling identification of this station is needed if this station is part of a multipoint network and the BSC is a receive (input) file.

Blank — This station is not receiving on a multipoint network.

Columns 65-70 (Remote Device)

Blank — System/3 is not used to communicate with the IBM 2770 Data Communication System or the IBM 2780 Data Transmission Terminal.

1442-1 — The IBM 1442 Card Read/Punch (card read) is a remote device used with the IBM 2780 remote terminal.

1442-2 — The IBM 1442 Card Read/Punch (card punch) is a remote device used with the IBM 2780 remote terminal.

1443 — The IBM 1443 Printer is a remote device used with the IBM 2780 remote terminal.

0545-3 — The IBM 0545 Card Punch, Model 3, is a remote device used with the IBM 2770 remote terminal.

0545-4 — The IBM 0545 Card Punch, Model 4, is a remote device used with the IBM 2770 remote terminal.

2213-1 — The IBM 2213 Printer, Model 1, is a remote device used with the IBM 2770 remote terminal.

2213-2 — The IBM 2213 Printer, Model 2, is a remote device used with the IBM 2770 remote terminal.

2502-1 — The IBM 2502 Card Reader, Model 1, is a remote device used with the IBM 2770 remote terminal.

2502-2 — The IBM 2502 Card Reader, Model 2, is a remote device used with the IBM 2770 remote terminal.

5496-1 — The 5496 Data Recorder (card read) is a remote device used with the IBM 2770 remote terminal.

5496-2 — The 5496 Data Recorder (card punch) is a remote device used with the IBM 2770 remote terminal.

INPUT SPECIFICATIONS

Columns 7-14 (Filename)

Enter a valid RPG II filename for every input, update, and combined file your program uses.

Columns 15-16 (Sequence)

Enter a two-digit number to assign a special sequence to record types in a file and to request that the record type sequence be checked by the program. Enter two alphabetic characters to indicate that record type sequence is not being checked. Alphabetic characters must be used for a chained file. Within a file, record types with an alphabetic sequence entry must be described before record types with a numeric sequence entry.

Column 17 (Number)

- Blank — Columns 15-16 contain alphabetic characters (record type sequence is not being checked)
- 1 — Columns 15-16 contain numeric characters; only one record of this type is present in each sequenced group
- N — Columns 15-16 contain numeric characters; one or more records of this type can be present in the sequenced group

Column 18 (Option)

- Blank — Record type must be present
- O — Optional; record type may or may not be present

Columns 18 is used when record types are being sequence checked (columns 15-16 contain a numeric entry).

Columns 19-20 (Record Identifying Indicator, **, TR)

- 01-99 — Record identifying indicator
- 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 checking for a record type that causes an error condition
- ** — Look-ahead fields
- TR — Spread cards

Columns 21-41 (Record Identification Codes)

This field is divided into three identical subfields:

- Columns 21-27
- Columns 28-34
- Columns 35-41

An AND relationship exists between these three fields.

Position

- Blank — No record identification code is needed
- 1-4096 — Record position of the record identification code

Not

- 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 the identification code is not present in the specified record position

C/Z/D

- C — Entire character
- Z — Zone portion of character
- D — Digit portion of character

Remember that many characters have either the same zone or the same digit portion.

AND and OR Relationships

Enter AND in columns 14-16 on the next line of the Input Sheet if more than three record identification code subfields are needed to identify the record. Enter OR in columns 14-15 if either one of the codes may be present to identify the record. A maximum of 20 AND or OR lines in any combination may be used to describe the record identifying code.

Column 42

Blank

Column 43 (Packed or Binary Field)

- Blank – Input field in unpacked decimal format
- P – Input field in packed decimal format
- B – Input field in binary format

Columns 44–51 (Field Location)

Enter two 1–4 digit numbers to identify the beginning of a field (From) and the end of a field (To) in the input record. These entries are identical for a one-position field.

Column 52 (Decimal Position)

- Blank – Alphameric field
- 0–9 – The number of decimal positions in the numeric field named in columns 53–58

This column must contain an entry for numeric fields.

Columns 53–58 (Field Name)

These columns can contain:

- A valid RPG II field name for each field defined in Field Location.
- An array name or array element.
- PAGE, PAGE1, or PAGE2 special words.

Columns 59–60 (Control Level)

- L1–L9 – Field described on this line is a control field
- Blank – Field described is not a control field

These columns must be blank for chained or demand files.

Columns 61–62 (Matching Fields)

Enter a match value (M1–M9) to indicate matching fields and sequence checking when you have two or more input, update, or combined files with match fields. When you have just one input, update, or combined file with match fields this entry causes only sequence checking.

Columns 63–64 (Field Record Relation)

- 01–99 – Record identifying indicator assigned to a record type
- 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 this entry:

1. All fields without field record relation should be specified before fields with field record relation.
2. All fields with the same field record relation entry should be entered on consecutive lines.
3. All parts of a split control field must have the same field record relation entry and must be described on consecutive specification lines.

Columns 65–70 (Field Indicators)

- 01–99 – Field indicator
- H1–H9 – Halt indicator (when checking for an error condition in the data)

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 which reflects the result of the test is turned on, the others are turned off. If a field is alphameric, an indicator can only be specified in Zero or Blank (columns 69–70).

Columns 71–74

Blank

CALCULATION SPECIFICATIONS

Columns 7–8 (Control Level)

- Blank – Operation done at detail time
- LO – Calculation is performed 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 has been processed or after LR has been set on
- SR – Calculation operation is part of a subroutine

AN or OR can be entered in these columns to indicate that indicators on the line are in an AND or OR relationship with indicators on the preceding line. A maximum of seven AN, OR, or mixed AN and OR lines are allowed to condition an operation. Entries must be in the order listed.

Columns 9-17 (Indicators)

Enter 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 containing indicators in columns 9-17 which are in an AND or OR relationship with those on the first line by entering AN or OR in columns 7-8.

Columns 18-27 (Factor 1) and Columns 33-42 (Factor 2)

Factor 1 and Factor 2 may contain the following entries:

- Name of any field that has been defined.
- Alphameric or numeric literal.
- Subroutine, table or array name, or array element.
- Date field name (UPDATE, UMONTH, UDAY, UYEAR).
- Special name, PAGE, PAGE1, or PAGE2.
- Label for a TAG, BEGSR, or ENDSR operation (Factor 1) or a label for a GOTO or EXSR operation (Factor 2).
- Filename for a CHAIN, DEBUG, DSPLY, READ, FORCE, SET, or KEY operation (Factor 2).

Columns 28-32 (Operation)

Enter an operation code, left justified.

Columns 31-32

Blank, 0, 1-8 — A combination of these entries is used to turn on field lights for SET or KEY operations.

Columns 43-48 (Result Field)

SPACE_n (n=1-3),
SKIP_{nn} (nn=01-B2),
EJECT

Special instructions to control printer output and ledger card ejection when used with the SET operation.

Field name, table names, array name, or array element

These entries hold the results of, or are the object of, the operation specified in columns 28-32.

IN_{xx} (xx=any
RPG II indicator)

The indicator to be transferred to an external subroutine in an RLABL operation.

Columns 49-51 (Field Length)

Blank — Field defined elsewhere
1-256 — Result field length
1-220 — Position to which the print element is to be moved or the position where a tab stop is to be set using the SET operation

Maximum length of a numeric result field is 15 digits; maximum length of an alphameric result field is 256 characters. The entry must be right justified.

Column 52 (Decimal Position)

Blank — Alphameric field or numeric field described elsewhere
0-9 — Number of decimal places in a numeric result field

Column 53 (Half Adjust)

Blank — Do not half adjust the Result Field
H — Half adjust the Result Field

Half adjust is allowed only with arithmetic operations.

Columns 54-59 (Resulting Indicators)

Enter any of the following indicators: 01-99, H1-H9, L1-L9, LR, OA-OG, and OV, KA-KN, KP and KQ. Columns 54-59 are used:

- To test the value of the result field after an arithmetic operation.
- To check the outcome of a CHAIN, LOKUP, COMP, TESTB, or TESTZ operation.
- To specify which indicators to SETON or SETOF.
- To indicate end of file for the READ operation code.
- To allow command keys to be pressed using the SET operation code.

Arithmetic Operations: Enter up to three indicators to be turned on whenever the result is positive (indicator in columns 54–55), negative (indicator in columns 56–57), or zero (indicator in columns 58–59).

Compare Operations: Enter up to three indicators to be turned on whenever Factor 1 is greater than Factor 2 (indicator in columns 54–55), Factor 1 is less than Factor 2 (indicator in columns 56–57), or Factor 1 is equal to Factor 2 (indicator in columns 58–59).

LOKUP Operation: Enter one or two indicators in High, Low, Equal, High and Equal, or Low and Equal. If there is an entry in the High or Low columns, the table name in Factor 2 must be specified as ascending or descending on the Extension sheet.

TESTB Operation: Resulting indicators have the following meaning for this operation:

- Columns 54–55: An indicator in these columns is turned on if each bit specified in Factor 2 is off in the Result Field.
- Columns 56–57: An indicator in these columns is turned on if two or more bits were tested and of mixed status (some bits on and some bits off).
- Columns 58–59: An indicator in these columns is turned on if each bit specified in Factor 2 is on in the Result Field.

TESTZ Operation: Enter one to three indicators to reflect the zone of the leftmost character in the Result Field, as follows:

- Columns 54–55: Turned on by the zone portion of the characters & and A-I.
- Columns 56–57: Turned on by the zone portion of the characters [(bracket), – (minus), and J–R.
- Columns 58–59: Turned on by the zone portion of any character not listed above.

CHAIN Operation: Enter an indicator in columns 54–55 to be turned on in the case of a record-not-found condition.

SETON and SETOF Operations: Enter up to three indicators in columns 54–59 to be turned on (SETON) or turned off (SETOF).

READ Operation: Enter an indicator in columns 58–59 to be turned on after each read operation if an end-of-file condition is reached. Once end-of-file is reached, a halt occurs after each read operation if no indicator is entered.

KEY Operations: Enter up to three indicators to be turned on whenever the numeric field keyed in store only or display mode is positive (indicator in columns 54–55), negative (indicator in columns 56–57), or zero (indicator in columns 58–59).

Columns 54–59 must be left blank for all fields keyed in manual mode.

SET Operations: Enter up to three command keys (KA–KN, KP, KQ) that the operator is allowed to press when the program has paused at this specification line. Field lights must be specified in columns 31–32 of a SET operation when command key indicators are entered in columns 54–59.

Columns 60–74 (Comments)

Enter any meaningful information you wish, to help you understand or remember what you are doing in each specification line.

OUTPUT-FORMAT SPECIFICATIONS

Columns 7–14 (Filename)

Enter a valid RPG II filename for each output, combined, and update file used by your program. Each filename need be specified only once, on the first line describing that file.

Columns 14–16 (AND/OR Relationship)

Enter AND in columns 14–16 or OR in columns 14–15 if output records are in an AND or OR relationship. A maximum of twenty AND, OR, or mixed AND and OR lines are allowed to condition an output record.

Column 15 (Type)

- H – Heading records
- D – Detail records
- T – Total records
- E – Exception records (lines to be written during calculation time)

Columns 16-18 (Add a Record)

Enter ADD 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 sheet for the file to a record is added.

Column 16 (Stacker Select/Fetch Overflow)

- E – Ledger card eject
- F – Fetch overflow

If F is entered, the overflow routine is fetched when overflow occurs, before the usual time in the cycle.

Columns 17-22 (Space/Skip)

If these columns are blank, single spacing occurs after each line is printed.

Columns 17-18 (Space)

Enter a number (0-3) under the appropriate column to indicate the number of lines spaced before or after a line is printed.

Columns 19-22 (Skip)

- Blank – No skipping
- 01-99 – Lines 1-99
- A0-A9 – Lines 100-109
- B0-B2 – Lines 110-112

Enter one of the two-digit numbers listed above to indicate the next line printed. All line numbers between are bypassed. Enter the number in the Before or After columns, depending on whether you want skipping to occur before or after the line is printed.

Columns 23-31 (Output Indicators)

Enter one to three 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 will be done only if the indicator is not on. An AND relationship exists between indicators on a line. Additional lines of indicators in an AND or OR relationship may be used by entering AND in columns 14-16 or OR in columns 14-15 of each additional line (up to 20).

Columns 32-37 (Field Name)

Enter one of the following to name every field written out:

- Any field name previously defined in this program.
- The special words, PAGE, PAGE1, PAGE2, *PLACE, UPDATE, UDAY, UMONTH, and UYEAR.
- A previously defined table name, array name, or array element.

These columns must be blank if a constant is entered on columns 45-70 of the line. If any entry is made under Field Name, columns 7-22 must be blank.

Column 38 (Edit Codes)

Enter an edit code in column 38 when you want to:

- Suppress leading zeros for a numeric field.
- Omit a sign from the low order position of a numeric field.
- Punctuate a numeric field without setting up your own edit word.

A table summarizing the edit codes that can be used is printed above columns 45-70 on the Output-Format Sheet.

Column 39 (Blank After)

- B — Field is reset to blank or zero after writing
Blank — Field is not reset after writing

Numeric fields are set to zero and alphameric fields are set to blanks. This column must be blank for look-ahead and UPDATE fields. If the field name specified with Blank After is a table name, the element of the table looked up last will be blanked or zeroed.

Columns 40–43 (End Position in Output Record)

Columns 40–43 indicate the location on the output record of the field or constant written. Enter the number of the position occupied by the right-most character of the output field. The End Position entry must not be greater than the record length.

Column 44 (Packed or Binary Field)

- Blank — Field is unpacked numeric or alphameric data
P — Field is packed decimal numeric data
B — Field is in binary format

Packed and binary fields can only be written on disk; they cannot be printed or punched. Column 44 must be blank for *PLACE fields.

Columns 45–70 (Constant or Edit Word)

Constant: The following rules apply to constants:

- Field Name (columns 32–37) 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.
- Up to 24 characters of constant information can be placed in one line. Additional lines may be used, but each line must be treated as a separate line of constants. The end position of each line must appear in columns 40–43.

Edit Word: Enter any edit word to specify editing of numeric fields. Edit words must be enclosed by apostrophes. Constants are allowed within edit words.

Edit words are not used with edit codes. However, when edit codes 1–4, A–D, and J–M are used, columns 45–47 may contain '*' (to denote asterisk fill) or '\$' (to denote a floating dollar sign).

Columns 71–74

Blank

Appendix D. Programming Tips

CORE SAVING TECHNIQUES

When your program is too large to fit into the execution core size, you may want to use some core saving techniques 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 core available for execution and (2) how the compiler determines when a program is too large to fit into the core available for execution. This section will discuss the overlay process and then give you some suggestions for saving core.

Overlay Process

When your program exceeds the available storage for program execution, the RPG II compiler places some RPG II object program routines on disk. These routines are then called into main storage as they are needed by your 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 program execution. For this reason, the Root segment always remains in main storage. The Root segment may be used by routines in the Overlay area. The Root segment can call a routine in the Overlay area by using a branch instruction.

The main Overlay area contains the major routines of the RPG II object program. Routines in this area may be called by the Root segment or by other routines in the same main overlay.

Some large programs require that storage be divided into two additional parts: the Secondary Root segment and the Suboverlay area. The Secondary Root segment is used to supplement 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 main Overlay area. Figure 187 shows the location of the main storage areas.

Creating the Overlays

In order to create overlays, the compiler must first determine which routines will go into the main Overlay areas and which routines will go into the Suboverlay areas. Then it calculates the size of the largest main overlay and the size of the largest suboverlay. These sizes are rounded off upwards in increments of 256 bytes (1 sector). The compiler then adds the lengths of the Root segment, the largest main overlay, and the largest suboverlay. If the sum is larger than the available storage, your program is too large, and core saving techniques must be used if the program is to be run.

Special Open/Close

Special Open/Close is used 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 core usage map (Figure 188).

The first load will bring in the Root segment, the Overlay Fetch routine, the Overlay Fetch Area and a special transfer vector to call the Open Overlay. Open is completely self-contained and does not need any of the non-overlay code. When Open is complete, Overlay One is loaded. Overlay One consists of all code that is identified as non-overlay and was not loaded during the first load. The program then executes as a normal overlay program until Close is needed. At this time, Close is brought into core starting at the Overlay Fetch Area and using as much core as is needed.

The Overlay Fetch Area size for the rest of the program can be found by subtracting the start of the Overlay Fetch Area from the lowest start address of the non-overlay code that was not included in the first load; that is, Input Mainline starts at '1762', so '1762' minus '1462' equals '0300' — the Overlay Fetch Area size.

Saving Core

When the compiler finds that your program is too large, an error message is written. You can reduce the storage needed for your program either by using some general core saving techniques or by reducing the size of the overlays.

General Core Saving Techniques

Some of the techniques you can use are:

1. Divide your program into separate tasks, creating a separate program for each task. For example, suppose you want to update a file and print a listing of the updated file. You can save main storage by updating the file with one program and printing the listing with another program.

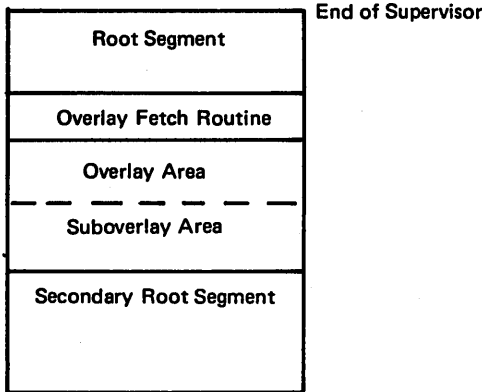


Figure 187. RPG II Storage Map

2. Eliminate unreferenced fields. These unreferenced fields are identified on the RPG II listing. By eliminating these fields, you can eliminate the storage area that is required to hold the data and the instructions that store the data in them.

ADDR	OVERLAY	LENGTH		
0C00		0781	RROOT	ROOT
1381		00E1	RGSUBS	OVERLAY FETCH ROUTINE
1462		0R00	RGSUBS	OVERLAY FETCH AREA
1762		00A3	RGMAIN	INPUT MAINLINE
1865		000B	RGSUBS	TRANSFER VECTOR
1805		003A	RGSUBS	RECORD ID
183F		0026	RGSUBS	CONTROL FIELDS
1462	###002	0008	RGSUBS	SUBSEG
146A	###002	003F	###CSR	SYSTEM SUBR
14A9	###002	0037	###CDI	SYSTEM SUBR
14E0	###002	0046	###CIC	SYSTEM SUBR
1526	###002	002E	###CIF	SYSTEM SUBR
1554	###002	0032	###CRI	SYSTEM SUBR
1586	###002	001C	###CLC	SYSTEM SUBR
15A2	###002	009B	###CMO	SYSTEM SUBR
163D	###002	0015	###CTC	SYSTEM SUBR
1870		005E	RGMAIN	INPUT FIELDS
1928		03FE	RGMAIN	DETAIL CALCS
1DC4		000B	RGSUBS	TRANSFER VECTOR
191F		0009	RGSUBS	CONSTANTS
18CE		0051	RGSUBS	INPUT CTRL RTN
1D81		0043	###PGR	RESET RESULTING INDR
1462	###003	0008	RGSUBS	SUBSEG
146A	###003	0031	###CRR	SYSTEM SUBR
1504	###003	001C	###CLC	SYSTEM SUBR

Figure 188. Core Usage Map

3. Eliminate unreferenced indicators. Eliminating unreferenced indicators can eliminate the instructions required to set the indicators on and off.
4. Eliminate unnecessary conditioning indicators. Two possible forms of unnecessary indicator tests are:
 - a. If only one type of input record is to be processed, the indicator associated with that record will always be on except during the first detail output time. It is, therefore, not necessary for any calculation to be conditioned with this indicator.
 - b. 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
    
```

Note: This technique may 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.

Note: Be sure you do not mix alphameric and numeric fields.

- Reuse input field name areas. In some instances, two or more input files may have fields that always contain identical information. These fields can be given the same field name in order to use the same core storage area.

Another way to reuse input field areas is to use the same names for fields in two files. This can be done only if both fields have the same attributes (length, alphameric/numeric, packed binary) and each field is only used in the cycle in which the record is processed. Both files cannot be used in the same cycle.

- Reduce calculation work area sizes. Be sure that no work area has been defined as larger than it needs to be. This may cause a warning that the result field may not be large enough, but if you know that the largest possible number will fit into the area specified, you may continue.
- Include the necessary intervening blanks when describing alphameric fields and constants for output. This will make the fields adjacent. The output optimization phase will move all adjacent fields and constants with one instruction instead of using one instruction to move each line.

Not Optimized

Optimized

5 'DAILY'

17 'TRANSACTION'

26 'REGISTER'

18 'DAILY TRANS-
ACTION '

26 'REGISTER'

- Design files to contain record lengths that are an even multiple of 256 bytes or that will 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 to be used in a numeric operation in the program. This can save on the amount of storage required to store the field and can allow the input and output fields transfer routine to be optimized.

- Use only one type of file organization in a program (indexed, direct, or sequential). Also, use the same method of processing where possible. This can reduce the disk data management core requirements. Some unit record data management can also be eliminated by transferring unit record files to disk.

- Use the shared input/output access method (SIAM) to process disk files. This will reduce the storage required even on programs with only one disk file.

Note: Using SIAM may decrease program throughput.

- Group calculation statements together 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.

- When using TESTB, BITON, or BITOF, use the actual bit pattern in factor 2.

- 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. (See Appendix E for an example.)

- 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 if the contents of these areas can be reduced to fit into the core available for execution.

The contents of the Root segment, main Overlay area, and Suboverlay area can be found by using the program listing. The Root segment contains the data and routines which are not given an overlay name in the Core Usage of RPG II Code section of the program listing (Figure 189).

Two sections of the program listing are used to determine the contents of the main Overlay and Suboverlay areas. The section shown in Figure 190 tells the:

1. Overlay name
2. Number of sectors in the overlay
3. Start address of the overlay

The start address separates main overlays and suboverlays. Two start addresses appear in the Start Address column. The lower address (1452) identifies a main overlay; the higher address (1C52) identifies a suboverlay.

The Text Sectors column indicates the largest overlays. In Figure 190, overlays 002, 003, and 004 are the largest suboverlays; overlay 001 is the largest main overlay.

Relate the name given in the Overlay Name column shown in Figure 190 to the Core Usage of RPG II Code section shown in Figure 191. 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 has taken place. When this occurs, overlay 001 is not treated as an overlay, but remains in main storage.

PERFORMANCE IMPROVEMENT TECHNIQUES

Some relatively simple program changes may make significant improvements in your program's performance. However, these performance techniques will not improve performance in all programs. Therefore, study these techniques and determine if they will improve your program's performance before you use them. The five performance improvement techniques are:

1. Unblock all randomly processed indexed files. Blocking is not necessary since each record has its own index entry with the direct address of the record.
2. Block all sequentially processed indexed files.
3. Use the core 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.
4. Double buffer unit record files.
5. Reduce or eliminate blocking of consecutive files and double the buffer instead. For example, instead of using a block of 1600 bytes with 80 byte records, use a block of 800 bytes and double buffer.

	START ADDR	NAME IF OVERLAY	CODE LENGTH	NAME	CORE USAGE OF RPGII CODE TITLE
Root {	0C00		0734	RGROOT	ROOT
	1334		011E	RGSUBS	OVERLAY FETCH ROUTINE
	1452		0C00	RGSUBS	OVERLAY FETCH AREA
	2052		0037	RGMAIN	INPUT MAINLINE
	1466	###001	043A	RGMAIN	DETAIL CALCS
	1496	###001	009A	RGSUBS	TRANSFER VECTOR
Root {	2183		0139	RGSUBS	CONSTANTS
	20E6		009D	RGSUBS	OUTPUT CTRL RTN
	2089		005D	RGSUBS	INPUT CTRL RTN
	22BC		0043	##PGRI	RESET RESULTING INDR
	145A	###001	000C	RGSUBS	SUBSEG
	1A81	###001	0015	##BDMC	SYSTEM SUBR
	1452	###001	0008	RGSUBS	SUBSEG
	1A21	###001	0060	##FGMC	MULTIPLY
	1CC0	###002	003E	RGSUBS	SUBSEG
	1C52	###002	0008	RGSUBS	SUBSEG
1C5A	###002	0066	RGSUBS	SUBSEG	
1D81	###002	0031	##SCRR	SYSTEM SUBR	
1CFE	###002	0083	##PGCI	UNPACK	
1E62	###002	001C	##SCLC	SYSTEM SUBR	
1DB2	###002	009B	##SCMO	SYSTEM SUBR	
1E7E	###002	0032	##Scri	SYSTEM SUBR	
1EB0	###002	00B0	##SCSC	SYSTEM SUBR	
1E4D	###002	0015	##SCTC	SYSTEM SUBR	
1F60	###002	0037	##SCDI	SYSTEM SUBR	
1F97	###002	005C	##SCSI	SYSTEM SUBR	
18A0	###001	0086	RGSUBS	EXCEPTION	
1CC0	###003	003E	RGSUBS	SUBSEG	
1C52	###003	0008	RGSUBS	SUBSEG	
1C5A	###003	0066	RGSUBS	SUBSEG	
1D81	###003	0031	##SCRR	SYSTEM SUBR	
1CFE	###003	0083	##PGCI	UNPACK	
1E62	###003	001C	##SCLC	SYSTEM SUBR	
1DB2	###003	009B	##SCMO	SYSTEM SUBR	
1E7E	###003	0032	##Scri	SYSTEM SUBR	
1EB0	###003	00B0	##SCSC	SYSTEM SUBR	
1E4D	###003	0015	##SCTC	SYSTEM SUBR	
1F60	###003	0037	##SCDI	SYSTEM SUBR	
1F97	###003	005C	##SCSI	SYSTEM SUBR	
1CB3	###004	003E	RGSUBS	SUBSEG	
1C5A	###004	0059	RGSUBS	SUBSEG	
1C52	###004	0008	RGSUBS	SUBSEG	
1CF1	###004	0031	##SCRR	SYSTEM SUBR	
1DD2	###004	001C	##SCLC	SYSTEM SUBR	
1D22	###004	009B	##SCMO	SYSTEM SUBR	
1DEE	###004	0032	##Scri	SYSTEM SUBR	
1E20	###004	00B0	##SCSC	SYSTEM SUBR	
1DBD	###004	0015	##SCTC	SYSTEM SUBR	
1ED0	###004	0037	##SCDI	SYSTEM SUBR	
1F07	###004	005C	##SCSI	SYSTEM SUBR	
1926	###001	00C2	RGSUBS	SUBSEG	
1C66	###005	00FA	RGSUBS	SUBSEG	
1C5A	###005	000C	RGSUBS	SUBSEG	
1C52	###005	0008	RGSUBS	SUBSEG	
1D60	###005	0015	##BDMC	SYSTEM SUBR	
19E8	###001	0039	##PGCB	SYSTEM SUBR	
Root {	145E	###004	0254	RGMAIN	DETAIL OUTPUT
	22FF		0036	RGSUBS	CONSTANTS
	1452	###006	000C	RGSUBS	SUBSEG

Figure 189. Overlay Usage Map (Part 1 of 2)

	172E	###006	0016	##SCCO	SYSTEM SUBR
	17FB	###006	0015	##SCTC	SYSTEM SUBR
	1744	###006	001C	##SCDF	SYSTEM SUBR
	1760	###006	009B	##SCMO	SYSTEM SUBR
	1810	###006	0041	##SCUA	SYSTEM SUBR
	1851	###006	0037	##SCDI	SYSTEM SUBR
	16B2	###006	007C	##PGCO	PACK
Root {	1490	###007	0024	RGMAIN	LR & OVERFLOW PROCESSING
	2335		008C	RGSUBS	CONSTANTS
	145E	###007	0032	RGSUBS	OVERFLOW SUBSEGMENT
	14C1	###007	0026	RGSUBS	SUBSEG
	1452	###007	000C	RGSUBS	SUBSEG
	14B4	###007	000D	RGSUBS	SUBSEG
	14E7	###007	0015	##BDMC	SYSTEM SUBR
	145E	###008	00AA	RGMAIN	OPEN
	1452	###008	000C	RGSUBS	SUBSEG
	153B	###008	0015	##BDMC	SYSTEM SUBR
	1515	###008	0026	RGSUBS	SUBSEG
	1508	###008	000D	RGSUBS	SUBSEG
	1452	###009	001D	RGMAIN	CLOSE
	146F	###009	000B	RGSUBS	TRANSFER VECTOR
			04081	AR001	TOTAL CORE USAGE REQUIRED TO EXECUTE

Figure 189. Overlay Usage Map (Part 2 of 2)

	OVERLAY NAME	RELATIVE START	C/T/S	# TEXT SECTORS	START ADDRESS	
Largest Overlay {	###001	00 01 01		07	1452	Suboverlays { 1C52 } Overlays
	###002	00 01 0A		04	1C52	
Largest Suboverlay {	###003	00 01 0F		04	1C52	
	###004	00 01 14		04	1C52	
	###005	01 00 01		02	1452	
	###006	01 00 04		05	1452	
	###007	01 00 0A		01	1452	
	###008	01 00 0C		01	1452	
	###009	01 00 0E		01	1452	
EL	D 2345DF 0 3					
TOTAL # OF LIBRARY SECTORS REQUIRED - 064						

Figure 190. Overlay Identification Area

	START ADDR	NAME IF OVERLAY	CODE LENGTH	NAME	CORE USAGE OF RPGII CODE TITLE
	0C00		0734	RGROOT	ROOT
	1334		011E	RGSUBS	OVERLAY FETCH ROUTINE
	1452		0C00	RGSUBS	OVERLAY FETCH AREA
	2052		0037	RGMAIN	INPUT MAINLINE
Overlay 001	1466	###001	043A	RGMAIN	DETAIL CALCS
	1A96	###001	009A	RGSUBS	TRANSFER VECTOR
	2183		0139	RGSUBS	CONSTANTS
	20E6		009D	RGSUBS	OUTPUT CTRL RTN
	2089		005D	RGSUBS	INPUT CTRL RTN
	22BC		0043	\$\$PGRI	RESET RESULTING INDR
	145A	###001	000C	RGSUBS	SUBSEG
Overlay 001	1A81	###001	0015	\$\$BDMC	SYSTEM SUBR
	1452	###001	0008	RGSUBS	SUBSEG
	1A21	###001	0060	\$\$PGMC	MULTIPLY
Suboverlay 002	1CC0	###002	003E	RGSUBS	SUBSEG
	1C52	###002	0008	RGSUBS	SUBSEG
	1C5A	###002	0066	RGSUBS	SUBSEG
	1DB1	###002	0031	\$\$SCRR	SYSTEM SUBR
	1CFE	###002	0083	\$\$PGCI	UNPACK
	1E62	###002	001C	\$\$SCLC	SYSTEM SUBR
	1DB2	###002	009B	\$\$SCMO	SYSTEM SUBR
	1E7E	###002	0032	\$\$SCRI	SYSTEM SUBR
	1EB0	###002	00B0	\$\$SCSC	SYSTEM SUBR
	1E4D	###002	0015	\$\$SCTC	SYSTEM SUBR
	1F60	###002	0037	\$\$SCDI	SYSTEM SUBR
	1F97	###002	005C	\$\$SCSI	SYSTEM SUBR
	Overlay 001	18A0	###001	0086	RGSUBS
1CC0		###003	003E	RGSUBS	SUBSEG
Suboverlay 003	1C52	###003	0008	RGSUBS	SUBSEG
	1C5A	###003	0066	RGSUBS	SUBSEG
	1DB1	###003	0031	\$\$SCRR	SYSTEM SUBR
	1CFE	###003	0083	\$\$PGCI	UNPACK
	1E62	###003	001C	\$\$SCLC	SYSTEM SUBR
	1DB2	###003	009B	\$\$SCMO	SYSTEM SUBR
	1E7E	###003	0032	\$\$SCRI	SYSTEM SUBR
	1EB0	###003	00B0	\$\$SCSC	SYSTEM SUBR
	1E4D	###003	0015	\$\$SCTC	SYSTEM SUBR
	1F60	###003	0037	\$\$SCDI	SYSTEM SUBR
1F97	###003	005C	\$\$SCSI	SYSTEM SUBR	
Suboverlay 004	1CB3	###004	003E	RGSUBS	SUBSEG
	1C5A	###004	0059	RGSUBS	SUBSEG
	1C52	###004	0008	RGSUBS	SUBSEG
	1CF1	###004	0031	\$\$SCRR	SYSTEM SUBR
	1DD2	###004	001C	\$\$SCLC	SYSTEM SUBR
	1D22	###004	009B	\$\$SCMO	SYSTEM SUBR
	1DEE	###004	0032	\$\$SCRI	SYSTEM SUBR
	1E20	###004	00B0	\$\$SCSC	SYSTEM SUBR
	1DBD	###004	0015	\$\$SCTC	SYSTEM SUBR
	1ED0	###004	0037	\$\$SCDI	SYSTEM SUBR
Overlay 001	1F07	###004	005C	\$\$SCSI	SYSTEM SUBR
	1926	###001	00C2	RGSUBS	SUBSEG
Overlay 001	1C66	###005	00FA	RGSUBS	SUBSEG
	1C5A	###005	000C	RGSUBS	SUBSEG
	1C52	###005	0008	RGSUBS	SUBSEG
	1D60	###005	0015	\$\$BDMC	SYSTEM SUBR
	19E8	###001	0039	\$\$PGCB	SYSTEM SUBR
145E	###006	0254	RGMAIN	DETAIL OUTPUT	
22FF		0036	RGSUBS	CONSTANTS	
1452	###006	000C	RGSUBS	SUBSEG	

Figure 191. Overlay Usage Map (Part 1 of 2)

172E	###006	0016	\$\$SCCO	SYSTEM SUBR
17FB	###006	0015	\$\$SCTC	SYSTEM SUBR
1744	###006	001C	\$\$SCDF	SYSTEM SUBR
1760	###006	009B	\$\$SCMO	SYSTEM SUBR
1810	###006	0041	\$\$SCUA	SYSTEM SUBR
1851	###006	0037	\$\$SCDI	SYSTEM SUBR
16B2	###006	007C	\$\$PGCO	PACK
1490	###007	0024	RGMAIN	LR & OVERFLOW PROCESSING
2335		008C	RGSUBS	CONSTANTS
145E	###007	0032	RGSUBS	OVERFLOW SUBSEGMENT
14C1	###007	0026	RGSUBS	SUBSEG
1452	###007	000C	RGSUBS	SUBSEG
14B4	###007	000D	RGSUBS	SUBSEG
14E7	###007	0015	\$\$BDMC	SYSTEM SUBR
145E	###008	00AA	RGMAIN	OPEN
1452	###008	000C	RGSUBS	SUBSEG
153B	###008	0015	\$\$BDMC	SYSTEM SUBR
1515	###008	0026	RGSUBS	SUBSEG
1508	###008	000D	RGSUBS	SUBSEG
1452	###009	001D	RGMAIN	CLOSE
146F	###009	000B	RGSUBS	TRANSFER VECTOR
		06081	AR001	TOTAL CORE USAGE REQUIRED TO EXECUTE

Figure 191. Overlay Usage Map (Part 2 of 2)

After identifying the Root segment and the largest main overlays and suboverlays, you can determine whether they contain routines that can be manipulated to reduce the overlay size. The following routines can be controlled:

1. Input Records
2. Detail Calculations
3. Total Calculations
4. Detail Output
5. Total Output

Following are some core saving techniques that can be used for these routines. These techniques may not necessarily work for all programs.

Input Records: One or more of the input or update files can be processed 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 Calculations routine.

Note: Total calculations will not be done on the first cycle.

Detail or Total Calculations: Use the following techniques:

1. Use subroutine calculations. In some instances this may increase, rather than decrease, the storage required due to the nature of the existing calculation routines. However, it may reduce the overall core storage requirements.

Note: If one subroutine calls another subroutine, both subroutines must be in core at the same time. This may 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.

2. Eliminate exception output if possible. This will move the logic for those output operations to either Total or Detail Output routines.
3. Eliminate read and/or chain operations by using matching records and processing consecutively. This will move the logic to Input Records routine.
4. Move part of the detail calculation logic to total calculations (or total calculation logic to detail calculations).

Note: Total calculations will not be done on the first cycle.

5. *Detail or Total Output:* Use the following techniques:

1. Use exception output. This will move part of the output logic to Detail or Total Calculation routines.
2. Do some of the output at total (or detail) output time. This moves logic to the Total (or Detail) Output routine.
3. Do not specify blank after for fields. Instead, clear them at the beginning of detail or total calculations.

Appendix E. Bytes of Generated Code for Calculations

This appendix contains the number of bytes of object code generated for RPG II operation codes. When used in conjunction with Appendix D, this information will help you determine the amount of core that may be saved by using certain coding practices.

For example, consider this core saving technique:

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.

If the decimal positions of Factor 1, Factor 2, and the Result Field are all different, an ADD operation will generate 27 bytes.

However, if all the fields were defined as having the same number of decimal positions, the same ADD operation would generate only 15 bytes.

Uniformity of fields will not only save core for ADD and SUB, but for most of the other arithmetic operations as well.

IBM		International Business Machines Corporation																				Form X21-9093 Printed in U.S.A.														
		RPG CALCULATION SPECIFICATIONS																																		
Date _____		Punching Instruction										Graphic										Page 1 2		Program Identification 75 76 77 78 79 80												
Program _____																																				
Programmer _____																																				
Line	Form Type	Control Level (LD, LR, SR)	Indicators						Factor 1	Operation	Factor 2	Result Field	Field Length	Decimal Positions Half Adjust (H)	Resulting Indicators			Comments																		
			Not	Not	Not	Not	Not	Not							Arithmetic	Plus	Minus		Zero																	
														High	Low	Equal																				
														1 > 2	1 < 2	1 = 2																				
														Lookup																						
														Table (Factor 2) is																						
														High	Low	Equal																				
														54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74		
0 1	C	X	ASSUME	FLD1	IS	LENGTH	5,	DECIMAL	POSITIONS	2																										
0 2	C	X	ASSUME	INPUT	IS	LENGTH	4,	DECIMAL	POSITION	1																										
0 3	C			FLD1				ADD	INPUT	RFLD	93																									27 BYTES
0 4	C																																			

IBM		International Business Machines Corporation																				Form X21-9093 Printed in U.S.A.														
		RPG CALCULATION SPECIFICATIONS																																		
Date _____		Punching Instruction										Graphic										Page 1 2		Program Identification 75 76 77 78 79 80												
Program _____																																				
Programmer _____																																				
Line	Form Type	Control Level (LD, LR, SR)	Indicators						Factor 1	Operation	Factor 2	Result Field	Field Length	Decimal Positions Half Adjust (H)	Resulting Indicators			Comments																		
			Not	Not	Not	Not	Not	Not							Arithmetic	Plus	Minus		Zero																	
														High	Low	Equal																				
														1 > 2	1 < 2	1 = 2																				
														Lookup																						
														Table (Factor 2) is																						
														High	Low	Equal																				
														54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74		
0 1	C	X	ASSUME	FLD1	AND	INPUT	ARE	LENGTH	9,	DECIMAL	POSITIONS	3																								
0 2	C			FLD1				ADD	INPUT	RFLD	93																									15 BYTES
0 3	C																																			

Abbreviations and symbols used in the following text:

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
=	— equal
≠	— not equal
-	— minus
>	— greater than
<	— less than
+	— plus

<i>Operation</i>	<i>Bytes</i>
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	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
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 H/A	18
F2 = RF and D1 < DR H/A	18
D1 = D2 < DR	23
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 F1 is a field	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

<i>Operation</i>	<i>Bytes</i>
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
MVR	
D2 = DR	5
D2 ≠ DR	9
XFOOT	
D2 = DR	9
D2 ≠ DR	13
FORCE	13
with UPSI indicator	13 + 7 = 20
DSPLY	
(for factor 1)	40
with variable index	51
with integer index	46
with alphameric field	73
with numeric field	74
(for result field)	10
with variable index	27
with integer index	16
with alphameric field	58
with numeric field	93
SETOF	
1 command key light	7
2 command key lights	13
3 command key lights	18
Conditioning indicators	
(does not apply to CHAIN, FORCE, LOKUP, and READ)	
each indicator	3
each AND type	3
Resulting indicators	5
(does not apply to CHAIN, FORCE, LOKUP, and READ)	
with each resulting indicator	3

<i>Operation</i>	<i>Bytes added to base</i>
CHAIN	Base = 16
with UPSI indicator	6
when Factor 1 has a variable index	11
when key is not packed	14
when key is packed	23
when key is packed and Factor 1 is a table element	6
when key is a record number	8
when key is a record number and Factor 1 is a table element	6
when record-not-found indicator is given	12
when record-not-found indicator not given	16
READ	Base = 29
with UPSI 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 PAF limits	6
LOKUP	Base = 15
when Factor 1 is a table	6
when Factor 1 is a variable	11
with each resulting indicator	12

Operation

Bytes

Operation

Bytes

MOVE, MOVEL, MHHZO, See the following table.
 MHLZO, MLHZO, MLLZO The number of bytes specified includes all array control code lengths.

	MOVE alphanumeric	MOVEL LR < L2 numeric	MOVEL LR > L2 numeric	MOVEL LR = L2 alphanumeric	MOVEL LR < L2 alphanumeric	MOVEL LR > L2 alphanumeric	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, Tag to Array	40	66	43	40	52	40	40	52	52	40
Array to Array, Tag	28	57	38	28	35	35	35	35	47	42
Field to Array, Tag	17	34	27	17	17	24	24	31	24	31
Table to Array, Tag	20	52	33	20	24	30	30	24	36	20
Array, Tag 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, Tag 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

Array control code (initialization and processing) is generated for all calculations except DSPLY, LOKUP, CHAIN, READ, and FORCE.

<i>Operation</i>	<i>Bytes</i>
Array initialization	
F1 or F2 an array	6
F1 or F2 a table	4
F1 or F2 an array and tag	11
Array processing	
F1, F2, RF are arrays	30
F1-RF, F2-RF arrays	22
F2 and RF are arrays	16

Suppose, for example, that a SUB operation code was specified and has the following conditions:

1. F1 = RF
2. D1 = D2 = DR
3. F1, RF = full array
4. F2 = table

The length of object code generated would be as follows:

Array initialization	
F1 array	6 bytes
F2 table	4 bytes
RF array	6 bytes
SUB	6 bytes
Array processing	
F1-RF array	22 bytes

Thus, the total bytes of code generated for a SUB operation code is 44 bytes.

IN-LINE INQUIRY SUBROUTINE (SUBR95)

You can use SUBR95 to perform inquiry-type functions without doing a roll-out/roll-in (see *Control Card Specifications, Inquiry* for a discussion of roll-out/roll-in). You can check at any point in your calculations to determine if an inquiry request has been made.

To use SUBR95, you must call it by specifying the linkage shown in Figure 192 wherever you want to acknowledge an inquiry request. The indicator specified in columns 45–46 can be any RPG II indicator. For a detailed discussion of this linkage, see *Operation Codes, Branching to External Subroutines* under *Supplementary Information*.

When SUBR95 is called, it will check to see if the operator has made an inquiry request. If he has, the indicator specified will be set on; if he has not, the indicator specified will be set off.

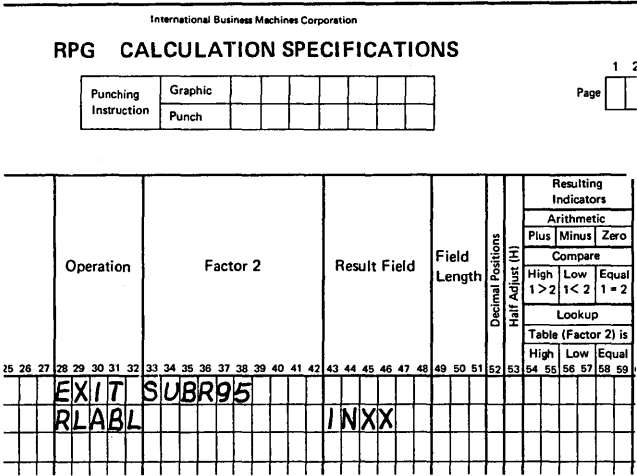


Figure 192. Linkage for SUBR95

RPG II HALT PROCEDURES

Error conditions found in your RPG II program result in a halt during execution or compilation of the program. 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 your program are done, tables are dumped, and files are closed.

3—Immediate Cancel: The job is cancelled without returning control to the RPG II program.

In order to select an option, the operator keys in the option (0, 1, 2, 3) and presses the PROG START key or the ENTER+ key.

A complete discussion of the halts and the necessary operator procedures appears in the *IBM System/3 Model 6 Halt Guide*, GC21-7541. Appendix B contains a detailed list of compilation errors.

OPERATION CONTROL LANGUAGE FOR RPG II

In order to compile an RPG II source program, the RPG II compiler program must be loaded into main storage. This can be done by including an IBM-supplied procedure named RPG (located in the Source Library) in the job stream. The OCL statements that include the library procedure are:

```
/&
// CALL RPG,R1
// RUN
```

The OCL statements included in the Source Library procedure named RPG are shown in Figure 193.

Library procedures can be modified. OCL statements necessary to modify a library procedure are described in the *IBM System/3 Model 6 Operation Control Language and Disk Utility Programs Reference Manual*, GC21-7516.

```
// LOAD $RPG,R1
// FILE NAME-$SOURCE,UNIT-R1,RETAIN-S,TRACKS-10,PACK-SYSTEM
// FILE NAME-$WORK,UNIT-R1,RETAIN-S,TRACKS-10,PACK-SYSTEM
// RUN
```

Figure 193. IBM-Supplied Library Procedure for Compiling an RPG II Source Program

Index

*BOTH	254	
*PLACE	154, 162	
ADD (add)	212	
add a record	141	
adding records to files	46	
additional input/output area		
file description specifications	37	
general information	41	
ADDROUT files		
definition	37	
disk address length	37	
extension code	42	
file description sheet summary chart	57	
file organization	41	
processing by	35	
record address type	37	
allowing command keys to be pressed		
calculation specifications	137	
general information	256	
alternate collating sequence		
coding sheet	19	
control card specifications	16	
general information	17	
specifications	17	
alternate table or array	71	
AND relationship		
calculation specifications	117, 122	
input specifications	76, 87	
output-format specifications	140, 146	
arithmetic operations	212	
arrangement of the RPG II source program	3	
arrays		
adding entries to a short array	292	
alternate	71	
binary field	70	
building (<i>see</i> loading)		
compile time	282, 285	
creating input records	284	
decimal positions	70	
defining	284	
definitions of terms	282	
differences between tables and arrays	281	
dynamic arrays (<i>see</i> execution time arrays)		
editing	293	
entry	282	
execution time	282, 286	
extension specifications	63, 284	
extension sheet summary chart	72	
file description specifications	32	
formatting output	242	
full array	282	
indexing	291	
input records, creating	284	
length of entry	69	
loading		
arrays (continued)		
arrangement of source program	285	
compilation time	64, 285	
considerations	67	
execution time	64, 286	
more than one record	290	
one record	286	
pre-execution time	286	
via input or calculations	282, 286	
LOKUP (<i>see</i> LOKUP)		
modifying the contents	292	
name		
extension specifications	65	
file description specifications	32	
rules for	291	
number of entries per array	68	
number of entries per record	67	
output		
extension specifications	64, 293	
formatting	242	
output-format specifications	293	
packed field	70	
pre-execution time	64, 282	
referencing arrays in calculations (<i>see</i> LOKUP)		
related arrays		
alternating format	71	
definition	282	
length of array entries	69	
searching arrays (<i>see</i> LOKUP)		
sequence	70	
short arrays	282	
square root	213	
summary chart	72	
using arrays	290	
automatic page formatting	264	
BEGSR (begin subroutine)	241	
bit operations	221	
binary fields		
extension specifications	70	
input specifications	89	
output-format specifications	162	
binary format	91	
BITOF (set bit off)	224	
BITON (set bit on)	222	
blank after	161	
block length		
for disk records	33	
general information	33	
branching		
to external subroutines	231	
within RPG II	228	
BSCA (<i>see</i> telecommunications specifications)		
device name	42	
C-character	86	
calculation code size	419	
calculation indicators	116, 122, 136	

calculation specifications	
detailed description	115
summary	404
cathode ray tube (<i>see</i> CRT)	
CHAIN (chain)	248
chained files	31
character grouped by zone or digit	86
combined files	30
combined ledger files	186
command key indicators	
calculation specifications	122, 136
conditioning indicator, used as	122
output-format specifications	144
output indicator, used as	146
resulting indicator, used as	137
summary of use	179
(<i>see also</i> SET, SETON, and SETOF)	
command keys, pressing	
calculation specifications	137
general information	256
comment lines	11
comments	11
common entries	
detailed description	9
summary	395
COMP (compare)	219
compare and testing operations	218
compile time tables and arrays	282, 285
compiling	1
conditioning indicators	
calculation specifications	122
summary of use	178
consecutive processing	36
console (keyboard)	
considerations	44
device name	42
file description sheet summary chart	60
spacing on	144
constant or edit word	
considerations	163
output-format specifications	163
continuation	
line (column 53)	45
line option (column 54-55)	45
record example (figure)	46
control break	96
(<i>see also</i> control fields)	
control card specifications	
detailed description	13
summary	395
control fields	
input specifications	96
split	97
control level indicators	
calculation specifications	116, 123, 136
conditioning indicator, used as	123
field record relation indicator, used as	108
input specifications	81, 96, 108
output-format specifications	144
output indicator, used as	144
record identifying indicator, used as	81
resulting indicator, used as	136
summary of use	182
control number field (ledger files)	184
core index	46

core saving techniques	
general	409
reducing the overlay size	412
core size to compile	14
core size to execute	14.1
CRT	
considerations	43
device name	42
file description sheet summary chart	61
skipping and spacing on	144
C/Z/D	86
D-digit	86
data recorder	
device name	42
file description sheet summary chart	59
date fields	155
DEBUG (debug)	
control card specifications	16
general information	263
specifications	263
decimal positions	
calculation specifications	133
extension specifications	70
input specifications	93
defining an alternate collating sequence	17
definitions of terms	4
demand files	32
detail records	141
detail time	5
detailed RPG II program logic	272
device	42
console	44
CRT	43
printer	44
SPECIAL device support	44
diagnostics, RPG II	325
digit, characters grouped by	86
direct files	
addition to	56
general information	40
loading	56, 249
methods of processing	56
disk	
device name	42
file description sheet summary chart	54-57
file organization (<i>see</i> file organization)	
file processing (<i>see</i> processing methods)	
display files	31
DIV (divide)	212
domestic format	16
DSPLY (display)	247
edit codes	
effect on inverted print	16
output-format specifications	158
summary chart	159
edit words	
considerations	164
formatting	166
output-format specifications	163

EJECT	132		
(see also SET)			
ejecting ledger cards			
calculation specifications	132		
general information	186, 256		
output-format specifications	141		
end of file	32		
end position in output record	162		
ENDSR (end subroutine)	241		
entries per record, number of	67		
entries per table or array, number of	68		
error message format	325		
error messages, RPG II	325		
exception records	141		
EXCPT (exception)	242		
overflow printing	270		
execution time arrays	282, 286		
EXIT (exit to an external subroutine)	231		
EXSR (execute subroutine)	241		
extension code	42		
extension sheet summary chart	72		
extension specifications			
detailed description	63		
summary	397		
extents, number of	52		
external indicators			
calculation specifications	122		
conditioning indicator, used as	123		
file description specifications	53		
field record relation indicator, used as	108		
input specifications	107		
output-format specifications	144		
output indicator, used as	147		
summary of use	183		
factor 1	127		
factor 2	127		
fetching the overflow routine			
general information	268		
output-format specifications	141		
field length	133		
field location	93		
field indicators			
calculation specifications	122		
conditioning indicator, used as	122		
input specifications	112		
output-format specifications	144		
output indicator, used as	144		
summary of use	178		
field lights	132		
(see also KEY and SET)			
field name			
input specifications	93		
in an OR relationship	94		
output-format specifications	153		
rules for forming	93		
field print	27		
field record relation indicator	107		
file addition	47		
file condition	53		
file description sheet summary charts	53		
file description specifications			
detailed description	29		
summary	397		
file designation	31		
file format	33		
file organization			
direct files	40		
file description sheet summary charts	53		
file description specifications	37		
indexed files	38		
sequential files	41		
file processing (see processing methods)			
file translation tables			
control card specifications	22		
general information	23		
specifications	23		
file type	30		
filename			
extension specifications	64		
file description specifications	30		
input specifications	76		
line counter specifications	74		
output-format specifications	140		
first page indicator			
output-format specifications	144		
output indicator, used as	147		
summary of use	180		
fixed dollar sign	164, 168		
fixed length format	33		
floating dollar sign	164		
flowchart			
detailed RPG II program cycle	272		
general RPG II program cycle	4		
FORCE (force)	244		
form length	74		
form type	11		
formatting edit words	166		
forms positioning, 1P	22		
from filename	64		
general core saving techniques	409		
general RPG II object program logic	4		
generated codes for calculations	419		
GOTO (go to)	228		
half adjust	135		
halt indicators			
calculation specifications	122, 136		
conditioning indicator, used as	123		
field indicator, used as	114		
field record relation indicator, used as	108		
input specifications	81, 108, 112		
output-format specifications	144		
output indicator, used as	147		
record identifying indicator, used as	81		
resulting indicator, used as	136		
summary of use	179		
halt procedures	427		
heading records	140		
IBM-written subroutines	45, 398, 425		
indexed files			
addition to	47		
general information	38		
loading	47, 54		
methods of processing	38, 54		

indicators
 calculation specifications 116, 122, 136
 command key (*see* command key indicators)
 conditioning (*see* conditioning indicators)
 control level (*see* control level indicators)
 external (*see* external indicators)
 field (*see* field indicators)
 field record relation 107
 file conditioning 53
indicators (continued)
 file description specifications 42, 53
 first page (*see* first page indicator)
 general information 175
 halt (*see* halt indicators)
 input specifications 81, 102, 107, 112
 last record (*see* last record indicator)
 L0 (*see* L0 indicator)
 matching record (*see* matching record indicator)
 output-format specifications 144
 overflow (*see* overflow indicators)
 record identifying (*see* record identifying indicators)
 resulting (*see* resulting indicators)
 inline inquiry subroutine 425
 input files 30
 input indicators 81, 102, 107, 112
 input/output areas
 additional 37
 shared 27
 input specifications
 detailed description 75
 summary 400
 inquiry
 IBM-written subroutine 425
 RPG 21
 inverted print 16

KEY (key)
 bypassing 252
 calculation specifications 252
 combinations with SET 262
 display mode 250
 manual mode 250
 store only mode 250
 summary chart 253
 used in subroutine 254
 (*see also* SET)
key field
 definition 38
 length 37
 packed format 37
 starting location 42
keyboard
 device name 42
 file description sheet summary chart 58

L0 indicator
 calculation specifications 116
 control level indicator, used as 116
 output-format specifications 144
 output indicator, used as 144
 summary of use 183
last record indicator
 calculation specifications 122, 136
 conditioning indicator, used as 123

 input specifications 81
 output-format specifications 144
 output indicator, used as 144
 record identifying indicator, used as 81
 resulting indicator, used as 136
 summary of use 183
 leading zero suppression 26
ledger card
 control number field 184
 eject
 general information 186
 output-format specifications 141
 SET operation 256
 line finder marks 184
 overflow 194
 printing on 186
 spacing on 186
ledger card device
 device name 42
 file description sheet summary chart 59
ledger file processing
 combined files 186
 general information 184
 output files 187
 overflow 194
length of
 key field 37
 record address field 37
 table or array entry 69
level zero indicator (*see* L0 indicator)
 limits processing 36
line counter specifications
 detailed description 73
 summary 398
line finder marks 184
line number
 coding lines 9
 number of lines per page 74
 overflow lines 74
listing options 14
literals 128
load
 direct file 56, 249
 file description sheet summary charts 53
 indexed file 47, 54
 sequential file 55
 table file 67
logic
 detailed RPG II program cycle 272
 general RPG II program cycle 4
LOKUP (look-up)
 with arrays 238
 general information 232
 with one table 232
 referencing the table item found 234
 starting the search at a particular array item 240
 with two tables 233
look ahead fields
 general information 196
 input specifications 81
 specifications 202
look-up operations (*see* LOKUP)

 machine requirements 4
 matching fields 102

matching record indicator			
assigning matching field values (M1-M9)	103		
calculation specifications	123		
conditioning indicator, used as	123		
matching record indicator (continued)			
field record relation indicator, used as	108		
input specifications	102, 107		
output-format specifications	144		
output indicator, used as	144		
processing matching records	104		
sequence checking	105		
summary of use	182		
maximum number of volumes (extents)	52		
message format, error	325		
MHHZO (move high to high zone)	218		
MHLZO (move high to low zone)	218		
MLHZO (move low to high zone)	218		
MLLZO (move low to low zone)	218		
mode of processing (<i>see</i> processing methods)			
MOVE (move)	214		
move operations	213		
move zone operations	218		
MOVE (move left)	216		
MULT (multiply)	212		
multifile processing			
match fields	204		
no match fields	204		
multivolume files			
consecutive processing	52		
file description specifications	52		
processing by keys	52		
random processing	52		
MVR (move remainder)	212		
name of label exit	45		
negative numbers	87		
not			
calculation specifications	122		
input specifications	85		
output-format specifications	144		
number, record types	80		
number of entries per record	67		
number of entries per table or array	68		
number of extents	52		
object program			
identification	11		
logic			
detailed description	272		
general description	4		
output	14		
OCL	419		
open/close (<i>see</i> special open/close)			
operation codes			
calculation specifications	130		
general information	210		
summary chart	131, 211		
(<i>see also</i> individual operation codes such as ADD, DIV)			
operation control language	427		
option, record type	80		
OR relationship			
calculation specifications	117, 122		
input specifications	76, 87		
output-format specifications	140, 146		
order of record selection, match fields	204		
ordered load	47		
output files	30		
output-format specifications	139		
output indicators			
output-format specifications	144		
summary of use	406		
output ledger files	187		
output record end position	162		
overflow indicators			
calculation specifications	122, 136		
conditioning indicator, used as	123		
considerations	270		
fetching the overflow routine	268		
file description specifications	42		
general information	264		
ledger files	194		
output-format specifications	144		
output indicator	146		
relation to program cycle	271		
resulting indicator, used as	136		
summary of use	182		
overflow line	74		
overflow printing			
automatic page formatting	264		
with EXCPT	270		
overflow indicators (<i>see</i> overflow indicators)			
overlay			
compiler process	409		
core saving techniques	409		
how to find an overlay	412		
reducing the overlay size	412		
packed decimal format	90		
packed key fields	37		
packed or binary data fields			
extension specifications	70		
input specifications	89		
output-format specifications	162		
PAGE			
input specifications	94		
output-format specifications	153		
page formatting, automatic	264		
page numbers	9		
performance improvement techniques	412		
positioning the print element	133, 256		
pre-execution time tables and arrays	64, 282		
primary file	31		
print element positioning	133, 256		
printer			
considerations	44		
device name	42		
file description sheet summary chart	61		
spacing and skipping	143		
processing methods	35		
consecutive	36		
direct file load	56, 250		
random by ADDROUT file	37		
random by key	36		
random by relative record number	36		
sequential by key	36		
sequential within limits	36		
program cycle			
detailed description	272		
general description	4		
program identification	11		
program listing	14		

programmed control of input and output	241
programming tips	409
programs, sample	307
punch MFCU zeros	26
random processing	
by ADDR0UT file	37
by key	36
by relative record number	36
using chain operation	249
READ (read)	244
record addition	47
record address field length	37
record address files	
extension code	42
extension specifications	64
file description specifications	31
format of records, rules for	36
key field length	37
located on disk	57
processing sequentially within limits	36
record address type	37
record identification codes	85
record identifying indicators	
field record relation indicator, used as	107
input specifications	81, 107
output-format specifications	144
output indicator, used as	144
summary of use	178
record length	
for disk files	35
file description specifications	35
for keyboard files	35
reducing the overlay size	412
repeating output fields (*PLACE)	154, 162
result field	132
result field length	133
resulting indicators	
calculation specifications	122, 136
conditioning indicator, used as	122
output-format specifications	144
output indicator, used as	144
summary of use	178
RLABL (RPG II label)	231
rounding	135
RPG inquiry	21
RPG specification sheets	1
RPG II error messages	325
RPG II halt procedures	427
RPG II operation control language	427
RPG II program cycle	
detailed description	272
general description	4
RPG II sample programs	307
RPG II source program record arrangement	3
RPG II specifications summary	393
running an RPG II program	427
sample programs	307
secondary files	31
sequence	
arrays	70
extension specifications	70
file description specifications	33
input specifications	76
tables	70

sequence checking	
input records	76
using M1-M9	105
sequential files	
addition to	55
general information	41
loading	55
methods of processing	55
sequential processing by key	36
sequential processing within limits	36
SET (set)	254
allowing command keys to be pressed	256
calculation specifications	256
combinations with KEY	262
general information	254
ledger card eject	256
print element positioning	256
setting tab stop positions	256
skipping and spacing	256
summary chart	257
used in subroutines	263
(see also KEY)	
SET0F (set off)	228
SET0N (set on)	228
setting indicators	137, 228
setting tab stop positions	133, 256
shared I/O	27
SKIPnn	132
(see also SET)	
skipping	
calculation specifications	132
CRT files	144
output-format specifications	142
printer files	143
source program	
identification	11
listing option	14
record arrangement	3
spacing	
calculation specifications	132
console files	144
CRT files	144
ledger files	144
output-format specifications	142
printer files	143
SPACEn	132
(see also SET)	
SPECIAL device support	
considerations	44
device name	42
file description sheet summary chart	62
special open/close	409
special words	
input specifications	94
output-format specifications	153
specifications summary	395
split control fields	97
spread cards	
general information	82
input specifications	81
processing considerations	84
specifications	84
SQRT (square root)	213
SRyzzz	45, 398
stacker select	141
SUB (subtract)	212
subroutine operations	241

subroutines	
calculation specifications	117
general information	276
IBM-written	425
operation codes	241
specifications	276
user-written	45, 398
SUBRxx	45, 398
SUBR95 (inline inquiry subroutine)	425
summary of specifications	395
supplementary information	175
tab stop positions, setting	133, 256
table files	32
tables	
adding entries to a short table	292
alternate	71
binary field	70
building (see loading)	
compile time	282, 285
creating input records	284
decimal positions	70
defining	284
definitions of terms	282
differences between tables and arrays	281
entry	282
extension specifications	63, 284
extension sheet summary chart	72
file description specifications	32
full table	282
input records, creating	284
length of entry	69
loading	
arrangement of source program	285
compilation time	64, 285
considerations	67
pre-execution time	286
LOKUP (see LOKUP)	
modifying the contents	292
name	
extension specifications	65
file description specifications	32
rules for	291
NOHALT	14
number of entries per record	67
number of entries per table	68
output	
extension specifications	64, 293
formatting	242
output-format specifications	293
packed field	70
pre-execution time	64, 282
referencing tables in calculations (see LOKUP)	
related tables	
alternating format	71
definition	282
length of table entries	69
searching tables (see LOKUP)	
sequence	70
short tables	282
summary chart	72

TAG (tag)	228
telecommunications specifications	
summary	400
terms, definitions of	4
TESTB (test bit)	226
testing and compare operations	218
TESTZ (test zone)	221
to filename	64
total records	141
total time	5
trailer records (see spread cards)	
translation table coding sheet	26
type H/D/T/E	140
UPDATE	155
UDAY	155
UMONTH	155
United Kingdom format	16
unordered load	47
unpacked decimal format	89
update files	30
using RPG II	1
UYEAR	155
volumes, number allowed	52
world trade format	16
XFOOT (crossfoot)	213
Z-ADD (zero and add)	212
Z-SUB (zero and subtract)	212
Z-zone	86
zero suppression	26
zone, characters grouped by	86
IP forms position	22
IP indicator (see first page indicator)	

READER'S COMMENT FORM

IBM System/3
Model 6
RPG II Reference Manual

SC21-7517-3

YOUR COMMENTS, PLEASE. . .

Your comments concerning this publication will help us produce better publications for your use. Each reply will be carefully reviewed by the persons responsible for writing and publishing this material. All comments and suggestions become the property of IBM.

Note: Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.

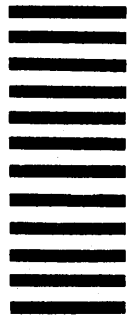
© 1984 IBM Corp.

Fold

Fold

FIRST CLASS
PERMIT NO. 387
ROCHESTER, MINN.

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES



POSTAGE WILL BE PAID BY . . .

IBM Corporation
General Systems Division
Development Laboratory
Rochester, Minnesota 55901

Attention: Publications, Dept. 245

Fold

Fold



International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, New York 10604
(U.S.A. only)

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
(International)

READER'S COMMENT FORM

IBM System/3
Model 6
RPG II Reference Manual

SC21-7517-3

YOUR COMMENTS, PLEASE. . .

Your comments concerning this publication will help us produce better publications for your use. Each reply will be carefully reviewed by the persons responsible for writing and publishing this material. All comments and suggestions become the property of IBM.

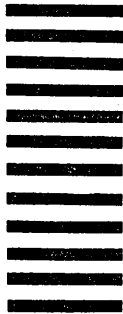
Note: Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.

Fold

Fold

FIRST CLASS
PERMIT NO. 387
ROCHESTER, MINN.

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES



POSTAGE WILL BE PAID BY . . .

IBM Corporation
General Systems Division
Development Laboratory
Rochester, Minnesota 55901

Attention: Publications, Dept. 245

Fold

Fold



International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, New York 10604
(U.S.A. only)

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
(International)