REFERENCE MANUAL

REPORT PROGRAM GENERATOR
Edition B

SYSTEM TEN COMPUTER BY SINGER

SINGER
BUSINESS MACHINES

REFERENCE MANUAL

REPORT PROGRAM GENERATOR
Edition B

SYSTEM TEN* COMPUTER BY SINGER

PUBLICATION NO. 40-226-1

CONTROL NO. B241PB

MARCH 1972

SINGER
BUSINESS MACHINES

2350 WASHINGTON AVE.
SAN LEANDRO, CALIF. 94577

*A trademark of the Singer Company.

PRINTED IN U.S.A.

REVISION RECORD

REPORT PROGRAM GENERATOR

EDITION LETTER	UPDATE NUMBER	DESCRIPTION
A	None	Original printing - October 1971
B	None	This revision incorporates new and changed information pertinent to the RPG 10K compiler. They are embodied in Section 1 and in Sections 3 through 8, and summarized in Appendix H.

PREFACE

The Report Program Generator (RPG) is a program that can be used to read input data, perform specified operations on the data, and produce formatted output reports and files.

The RPG compiler and object program require the use of the System Ten Disc Management Facility (DMF).

There are two versions of the System Ten RPG compiler, which require 9K and 10K core locations to compile, respectively. Those features which are available with the 10K compiler only are so marked in the text.

The purpose of this RPG Reference Manual is to completely describe the use of the System Ten RPG compiler and language. While a System Ten user with no previous RPG experience will be able to learn the use of RPG from this manual, it may be helpful for him to consult one of the several basic textbooks on RPG which are available.

Sections 1-6 of this Reference Manual provide the basic information necessary for a programmer to write source programs for the System Ten RPG compiler.

Section 7 discusses the RPG compiler and how it is used in conjunction with other System Ten facilities.

Section 8 describes the RPG object program and the various tables generated by the compiler.

A number of Appendixes present some details of RPG programming for the System Ten in greater depth and give additional examples of the use of RPG, linkage with Assembler subroutines, debugging procedure, and differences between the 9K and 10K compilers.

TABLE OF CONTENTS

Section 7 RPG COMPILER

TABLE OF CONTENTS

Section 8 RPG OBJECT PROGRAM

LIST OF FIGURES

LIST OF TABLES

Section 1

REPORT PROGRAM GENERATOR

REPORT PROGRAM GENERATOR

INTRODUCTION

Report Program Generator (RPG) is a system that takes input data, performs calculations and operations on it, and produces formatted reports and files. RPG has two basic components:

- A language with which the user specifies how his input data is to be processed and how the output reports are to be arranged.
- A compiler program that translates the user's RPG specifications into machine language instructions (object program), which can be used directly by the computer.

The object program then uses the input files specified by the user, performs calculations and operations on them as specified, and produces the output reports or files desired by the user.

Every RPG program has the same logical flow and structure. The flowcharts in Appendix A depict the sequence of actions that take place in every RPG program. Thus the programmer merely needs to provide the parameters of the specific case: data on the files to be used, the placement and structure of the input and output files, records and fields, and the calculations to be performed.

TWO COMPILER VERSIONS AVAILABLE

The System Ten RPG compiler is available in two versions: a compiler that requires a 9K partition to compile, and a compiler that requires a 10K partition to compile. The 10K compiler version provides certain additional features not available with the 9K compiler. These features are designated "10K Compiler Feature Only" where they are described in the manual.

USE OF RPG SPECIFICATIONS FORMS

The use of RPG to produce reports includes several steps. First, the user must carefully analyze the problem to determine what input files will be used, what format they will have, what calculations will be done on the input data, what output files will be needed, how they will be laid out and the information arranged in the completed report.

After the user has thoroughly analyzed his problem and determined his specific needs, he must communicate these specifications to the RPG compiler program. To do this he fills out a series of "specifications" forms.

- The Control Card and File Description Specifications Form is used to provide information to the RPG compiler and to name and describe the input and output files for the RPG object program.

- The Input Specifications Form is used to describe the exact layout of the input records.

- The Calculation Specifications Form is used to describe calculations and operations to be done on the input data and resulting data.

- The Output Format Specifications Form is used to describe the arrangement and placement of information on the output files and reports.

The contents of the specification forms are punched into cards (one line from a specification form to one card) to make up the RPG source deck.

With the 9K compiler, the source deck is placed on disc using the Disc Management Facility (DMF) FILE program.

The 10K compiler provides the option of compiling either from the disc or of using direct input from a Model 30 card reader.

COMPILING THE SOURCE PROGRAM

The source deck is read into System Ten core from a disc file or the card reader and the RPG specifications are translated into machine language instructions by the RPG compiler. The resulting object program is then stored as a file on disc for subsequent execution of the RPG job. At execution time, the input files are read and the RPG object program produces the desired output files and reports.

Refer to Section 7 for further discussion.

OBJECT PROGRAM TERMINATION

When an RPG object program has been successfully executed, control returns to the DMF conversational loader and the message:

A) ENTER PROGRAM NAME

is displayed on the workstation (device 0).

When an RPG object program terminates abnormally, it places an error code in a core location and branches to that location, causing a load condition. On a standard System Ten core dump, the address of the error code is determined by subtracting 11 from the address in locations 41-44 in the partition in which the program is executing. That is,

LOC (41P-44P) - 11 = Address of Halt Code

For additional details, refer to Section 8 and Appendix F.

HARDWARE REQUIREMENTS

9K Compiler

The 9K RPG compiler requires the following minimum hardware configuration:

- One Model 20 CPU (9K partition is required for compilation)
- One Model 70 Workstation
- One Model 40 Disc Drive
- One Model 50 Line Printer

The Model 50 Line Printer can be omitted if the user does his source program listing on the Model 70 Workstation.

In addition, a card reader (or other IOC input device) is required to read the source input if the DMF FILE program is used, or some other means of placing the source input into a DMF file must be provided.

10K Compiler

The 10K RPG compiler requires the following minimum hardware configuration:

- One Model 20 CPU (10K partition is required for compilation)
- One Model 70 Workstation
- One Model 40 Disc Drive
- One Model 50 Line Printer

Optional Hardware

- One to nine additional Model 40 Disc Drives
- One Model 30 Card Reader (or equivalent) for RPG source input or parameter input.
- One Model 35 Card Punch for punching out the object program.

SOFTWARE REQUIREMENTS

9K Compiler

The 9K RPG compiler requires the use of the Disc Management Facility (DMF). The user may become familiar with DMF by consulting the System Ten DMF Reference Manual.

10K Compiler

The 10K RPG compiler has essentially the same software requirements as the 9K compiler. Specifically, DMF is needed. In addition, the 10K compiler requires the LIOCS module, R_OPEN.

Section 2

RPG SPECIFICATIONS FORMS -- COMMON ELEMENTS

Figure 2.1 RPG CONTROL CARD AND FILE DESCRIPTION SPECIFICATIONS FORM

RPG CONTROL CARD AND FILE DESCRIPTION SPECIFICATIONS

Date ______________

Program ______________

Programmer ______________

Punching Instruction								
	Graphic							
	Punch							

Page (1 2) ☐☐

Program Identification (75 76 77 78 79 80) ☐☐☐☐☐☐

Control Card Specifications

Line (3 4 5)	Form Type (6)	NOT USED (7–11)	Core Size to Execute (12 13 14)	NOT USED (15–26)	Printer Line Count for Object Program (if other than 60) (27 28)	Comments (29–74)
010	H		500			

File Description Specifications

Line (3 4 5)	Form Type (6)	Filename (7–12)	NOT USED (13 14)	I/O File Type (15)	Input Files Only: P/S File Designation (16)	Input Files Only: End of File (17)	Input Files Only: A/D Sequence (18)	NOT USED (19–23)	Record Length (24–27)	NOT USED (28–32)	Overflow Indicator (33 34)	NOT USED (35–39)	Device (40–46)	Pool Name (Disc) / Mail to (Common) / Device Number (47–52)	NOT USED (53)	File Name (Disc) / Mail From (Common) / (Leave blank for all other devices) (54–59)
020	F	MASTER		I	P		A		94				DISC	INPØØL		FILE1
030	F	ØUT		Ø					132		ØA		PRINTER	2		
04	F															
05	F															
06	F															
07	F															
021		TRNSAC		I	S	E	A		80				READER	1		

FORM 40-344 (OBSOLETES FORM 40-254)

SINGER BUSINESS MACHINES

RPG SPECIFICATIONS FORMS – COMMON ELEMENTS

There are certain columns that contain information common to the four different types of RPG Specifications Forms, as follows:

Column Number(s)	Contents
1-2	Page
3-5	Line
6	Form Type
7	Comments
75-80	Program Identification

Refer to the sample RPG Control Card and File Description Specifications form in Figure 2.1. The allowed values and explanatory notes concerning the contents of these columns are given below.

PAGE (Columns 1-2)

Recommended Values	Meaning
01-99	Page Number

Each page of the specification forms may be numbered at the upper right hand portion of the form. Page numbers should be assigned in ascending order; for example: 01,02,03,05,07. It is possible to have more than one of each type of specification form, but all specification forms of the same type must be kept together. The proper sequence in which to arrange the forms is as follows:

- Control Card and File Description Specifications
- Input Specifications
- Calculation Specifications
- Output Format Specifications

LINE (Columns 3-5)

Recommended Values	Meaning
010-999	Line Number

The lines of the RPG Specifications Forms are numbered in columns 3 through 5 of the forms. The forms are actually pre-printed with numbers; the control card is always line 01. The remaining lines on the Control Card and File Description Specifications Form are numbered from 02 through 07. The additional lines on the form may be used to add lines (for example, 08 and 09) or to indicate insertion lines.

Figure 2.1 shows a Control Card and File Description Specifications Form with line numbers filled in. The extra line 021 will be inserted between line 020 and line 030.

The other three forms (Input Specifications, Calculation Specifications, and Output Format Specifications) are pre-numbered 01-15. There are blank lines at the bottom of the form which the user may number as lines to follow line 15 or lines to be inserted. It is possible to skip lines on the form. The user doesn't need to use every line number that is pre-printed, but the line numbers he uses should all be arranged in increasing order.

It is also possible to use any other System Ten characters in line numbers; for example: 02A, 02B, 02C, 03, 04. In this case the cards must be arranged in System Ten collating sequence.

If a card is found to be out of sequence during compilation, an S will be printed in the source listing beside the line that's out of sequence. If the page/line number field is left blank, the card is assumed to be in sequence where it is located.

After the cards have been punched from the specification forms, they must be arranged in the proper sequence (increasing page and line numbers) by the user.

FORM TYPE (Column 6)

Allowed Values	Meaning
H	"Header" card (Control Card)
F	File Description Specifications
I	Input Specifications
C	Calculation Specifications
O	Output Format Specifications

Column 6 on all specification forms contains a pre-printed letter which must be one of those listed above. The letter code informs the RPG compiler what type of specifications the line contains. For example, an I in column 6 indicates that the line contains Input specifications. If the user adds lines to the forms, he must enter the appropriate character in column 6.

When the source deck is prepared for input, all cards of the same type must be kept together, and the cards must be organized in the sequence shown above. In other words, the source input must be arranged as follows:

Header Card

All File Description Specifications

All Input Specifications

All Calculation Specifications

All Output Format Specifications

COMMENTS (Column 7)

Allowed Value	Meaning
*	Comment Line

The user may include comments in the listing of his source program by placing an asterisk (*) in column 7 on any line (with the exception of the Control Card Specifications line). The remainder of the line may then be used for a comment. This line is punched as a card and included in the input source deck. The comment will be listed "as is" in the source program listing. The user should include enough commentary on the coding so that another programmer unfamiliar with the program can understand it fully by looking at the source listing. Any characters allowed in the character set may be used in a comment line.

PROGRAM IDENTIFICATION (Columns 75-80)

Allowed Values	Meaning
Any characters	Name of program.

The user may provide a name for his program in columns 75-80 of the control Card Specifications Form. This name will be printed on the RPG computer listing. It is recommended that all cards of a program have the same identification in columns 75-80.

Section 3

CONTROL CARD AND FILE DESCRIPTION SPECIFICATIONS FORM

CONTROL CARD SPECIFICATIONS
FILE DESCRIPTION SPECIFICATIONS

Figure 3.1 RPG CONTROL CARD AND FILE DESCRIPTION SPECIFICATIONS FORM

RPG CONTROL CARD AND FILE DESCRIPTION SPECIFICATIONS

Date ______________

Program ______________

Programmer ______________

Punching Instruction							
Graphic							
Punch							

Page 1 2

Program Identification 75 76 77 78 79 80

Control Card Specifications

Line	Form Type	NOT USED	Core Size to Execute	NOT USED	Printer Line Count for Object Program (if other than 60)	Comments
3 4 5	6	7 8 9 10 11	12 13 14	15 16 17 18 19 20 21 22 23 24 25 26	27 28	29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74
0 1	H					

File Description Specifications

Line	Form Type	Filename	NOT USED	I/O File Type	Input Files Only: P/S File Designation	Input Files Only: E End of File	Input Files Only: A/D Sequence	NOT USED	Record Length	NOT USED	Overflow Indicator	NOT USED	Device	Pool Name (Disc) / Mail to (Common) / Device Number	NOT USED	File Name (Disc) / Mail From (Common) / (Leave blank for all other devices)
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
0 2	F															
0 3	F															
0 4	F															
0 5	F															
0 6	F															
0 7	F															

FORM 40-344 (OBSOLETES FORM 40-254)

SINGER
BUSINESS MACHINES

CONTROL CARD AND FILE DESCRIPTION SPECIFICATIONS FORM

CONTROL CARD SPECIFICATIONS

A control card (or heading) specification is necessary at the beginning of an RPG source program. Only one control card is permitted and it must be the first card of the source deck.

A line for the control card is included on the RPG Control Card and File Description Specifications Form, which is shown in Figure 3.1. If there is more than one such form in the source program, the control card line is filled in only on the first form (having the lowest page number).

FORM TYPE (Column 6)

Must contain an H.

CORE SIZE TO EXECUTE (Columns 12-14)

Allowed Values	Meaning
001 - 999	The amount of core storage available for the object program, specified in units of instructions (of 10-character length).
000 or Blank	Core storage available to the object program equals 10,000 characters or 1000 instructions.

PRINTER LINE COUNT FOR OBJECT PROGRAM (Columns 27-28)

Allowed Values	Meaning
01-99	The line on the page at which the Overflow Indicator is turned on. This does not mean the number of lines to be printed.
Blank	The printer line count is set at 60.

When using a Model 70 Workstation as a line printer, printer line count can be used to indicate the number of printer lines between the first lines of the continuous form being used for printing.

COMMENTS (Columns 29-74)

The programmer may insert whatever comments he wishes in columns 29-74 of the control card.

FILE DESCRIPTION SPECIFICATIONS

Each file to be used by an RPG program must be clearly defined and described. The file description specifications provide the compiler with necessary information concerning the various files used. The specifications are written on the lower part of the Control Card and File Description Specifications Form. One file is described on each line.

FORM TYPE (Column 6)

Must contain an F.

FILE NAME (Columns 7-12)

The file name may be one to six characters long. The first character must be alphabetic. The remaining characters may be alphabetic (A-Z) or numeric (0-9); other characters are not allowed. Also, blank spaces are not allowed in the middle of a file name. The file name must be left-justified.

Examples of valid file names:

INPUT

SALES

OUTREC

INFILE

TRNX10

FILE TYPE (Column 15)

Allowed Values	Meaning
I	Input File
O	Output File

Column 15 is used to specify whether this file is used as input or output for the program. An input file may be a collection of data records on magnetic disc, on cards, or in common. An output file is a series of records which will be produced by the program on magnetic disc, line printer, on a card punch, or in common.

Note: Disc input is fixed-allocation Read only. Disc output files are assumed to be in non-contention mode. Therefore, all output disc files specified should be in separate pools.

FILE DESIGNATION (Column 16)

Allowed Values	Meaning
P	Primary File
S	Secondary File

A primary file is the principal input file. When many files are being processed, the primary file always specifies the order of processing. There must be one, and only one, primary file for a program. The first file specified must be designated as the primary file.

A secondary file is any input file other than the primary file.

Output files must have a blank in column 16.

END OF FILE (Column 17)

Allowed Values	Meaning
E	The program can end only when all records of this file have been read.
Blank	The program can end even if all records of this file have not been processed.

Column 17 applies to programs having more than one input file. The processing of some of the files may be complete while other files still have records to be read. Thus, if column 17 of a line describing an input file is left blank, it is not necessary to process all records of that file before ending the program.

An E in column 17 of every input file description will insure that all records from all files will be read.

SEQUENCE (Column 18)

This applies to input files only.

Allowed Values	Meaning
A	Sequence of record fields matched against record fields in another file is checked. The program will expect fields in ascending order.
D	Sequence of record fields matched against record fields in another file is checked. The program will expect fields in descending order.
Blank	Sequence of fields on records is not checked. No matching is used with fields or records of this file.

Column 18 should always be blank for output files. The program determines the location of the sequenced fields in the input records from the Input Specifications Form, where columns 61-62 are used to indicate matching fields.

When data fields of records in different files are to be matched, sequencing is required. If a record is found to be out of sequence during input file matching, the program will be cancelled.

If matching fields are used but column 18 is left blank, A (ascending order of sequence numbers) will be assumed. For further information, see the description of matching fields, columns 61-62, on the Input Specifications Form in Section 4.

RECORD LENGTH (Columns 24-27)

Allowed Values	Meaning
1-415 (9K Compiler)	The number of characters per record in the file.
1-940 (10K Compiler)	The number of characters per record in the file
Blank	The record length is assumed equal to the maximum record length allowed by the I/O device used by this file, except that the default for disc is 94 characters for both the 9K and 10K compilers.

All records of a file must have the same length. The entry in columns 24-27 must always end in column 27, and leading zeros may be omitted. For example, if the record length is 80, the characters 80 appear in columns 26 and 27. Maximum record lengths for various devices are as follows:

Device	Maximum Length (characters)
Card Reader	80
Card Punch	80
Printer	132
Common	415
Disc (9K Compiler)	94
Disc (10K Compiler)	940

10K Compiler Feature Only

The 10K compiler allows multiple-sector disc records to be defined. The maximum record length for disc records is ten sectors or 940 data characters. Only one record per block is allowed for multiple-sector records and the record must start on a sector boundary. An example of the File Description Specifications for multi-sector records is shown in Fig. 3.2.

Note: The default record length assumed for disc when columns 24-27 are blank is 94 characters; this is true for both the 9K and 10K compilers.

RPG CONTROL CARD AND FILE DESCRIPTION SPECIFICATIONS

Date ____________

Program ____________

Programmer ____________

Punching Instruction	Graphic							
	Punch							

Page (1 2)

Program Identification (75 76 77 78 79 80)

Control Card Specifications

Line (3–5)	Form Type (6)	NOT USED (7–11)	Core Size to Execute (12–14)	NOT USED (15–26)	Printer Line Count for Object Program (if other than 60) (27–28)	Comments (29–74)
01	H					

File Description Specifications

Line (3–5)	Form Type (6)	Filename (7–12)	NOT USED (13–14)	I/O File Type (15)	P/S File Designation (16)	E End of File (17)	A/D Sequence (18)	NOT USED (19–23)	Record Length (24–27)	NOT USED (28–32)	Overflow Indicator (33–34)	NOT USED (35–39)	Device (40–46)	Pool Name (Disc) / Mail to (Common) / Device Number (47–52)	NOT USED (53)	File Name (Disc) / Mail From (Common) / (Leave blank for all other devices) (54–59)
020	F	INPUT1		I	P				580				DISC	RPGTST		PRIMRY
030	F	INPUT2		I	S				940				DISC	RPGTST		SCNDRY
04	F															
050	F	* MULTI-SECTØR RECØRDS														
06	F															
07	F															

FORM 40-344 (OBSOLETES FORM 40-254)

SINGER
BUSINESS MACHINES

Figure 3.2 SPECIFICATION OF MULT-SECTOR DISC RECORDS

OVERFLOW INDICATORS (Columns 33-34)

Allowed Values	Meaning
OV,OA-OG	A symbol for the Overflow Indicator is assigned. Used for output file control in printing page headings.
Blank	No Overflow Indicator is used.

The user assigns a symbol from among the allowed values for the Overflow Indicator.

The Overflow Indicator specified here must agree with that used on the Calculation Specifications Form and the Output Format Specifications Form. For example, if the Overflow Indicator is designated OE in columns 33-34, then OE must be consistently used as the Overflow Indicator later in the program.

DEVICE (Columns 40-46)

Allowed Values	Meaning
READER	Card Reader
PUNCH	Card Punch
PRINTER	Line Printer
COMMON	Common Mailbox
DISC	Disc Drive

The symbol placed in columns 40-46 signifies the type of device used by the file. The entry must start in column 40.

All disc files must be linked sequential files under the System Ten Disc Management Facility (DMF).

The Model 70 Workstation can be used as a substitute for the line printer on output by specifying PRINTER in columns 40-46 and the workstation device number in column 52.

The Common Mailbox can be used to transfer information between programs operating in different partitions. RPG programs set and expect the Common Mailbox End-Of-File Indicator to be six commercial "at" signs (@@@@@@) in the first six positions of the Mailbox I/O area. Therefore, common should not be defined as containing less than six characters. The sending RPG program automatically transmits the EOF record (@@@@@@) when LR is turned on, but not when the program terminates abnormally. Thus the programmer need not and <u>must not</u> attempt to send the six @ signs. If he does, the receiving partition will return to the conversational mode and the sending partition will be left waiting in an idle state. Additional information about the Common Mailbox is given in Appendix D.

SYMBOLIC DEVICE (Columns 47-52)

Allowed Values	Meaning
Pool Name	Used when entry in columns 40-46 is DISC. Name of pool where this file is located. Pool name consists of 1-6 alphanumeric characters. The first character must be alphabetic (A-Z).
Device Number	Used when entry in columns 40-46 is READER, PUNCH, or PRINTER. This is the standard number assigned to the device used by this file. Device number must be in column 52. Columns 47-51 must be blank.
Mail To	For the Common Mailbox, a partition number or symbolic partition name of the partition receiving the data via the mailbox must be entered in columns 51-52. Columns 47-50 must be blank.

FILE NAME (Columns 54-59)

Allowed Values	Meaning
File Name	For files using the disc, a file name must be supplied in columns 54-59. This is the name by which the file is known to the Disc Management Facility (DMF). The file name must be 1-6 alphanumeric characters long and must begin with a letter (A-Z).
Mail From	For the Common Mailbox, a partition number or symbolic partition name of the partition sending the data via the mailbox must be entered in columns 58-59. Columns 54-57 must be blank.
Blank	For any other device than disc or common, columns 54-59 must be left blank.

Example

Figure 3.3 shows a valid set of File Description specifications. The primary file YRTODT (year-to-date) is maintained on the disc in MYPOOL.FILEA. The secondary input file WKLY is read in on the card reader. The output file called REPORT goes to the printer.

Figure 3.3 SAMPLE CONTROL CARD AND FILE DESCRIPTION SPECIFICATIONS

RPG CONTROL CARD AND FILE DESCRIPTION SPECIFICATIONS

Date ____________
Program ____________
Programmer ____________

Punching Instruction	Graphic							
	Punch							

Page (1 2) ☐☐

Program Identification (75 76 77 78 79 80) ☐☐☐☐☐☐

Control Card Specifications

Line (3 4 5)	Form Type (6)	NOT USED (7–11)	Core Size to Execute (12–14)	NOT USED (15–26)	Printer Line Count for Object Program (if other than 60) (27–28)	Comments (29–74)
01	H					

File Description Specifications

Line (3 4 5)	Form Type (6)	Filename (7–12)	NOT USED (13–14)	I/O File Type (15)	P/S File Designation (16)	E End of File (17)	A/D Sequence (18)	NOT USED (19–23)	Record Length (24–27)	NOT USED (28–32)	Overflow Indicator (33–34)	NOT USED (35–39)	Device (40–46)	Pool Name (Disc) / Mail to (Common) / Device Number (47–52)	NOT USED (53)	File Name (Disc) / Mail From (Common) / (Leave blank for all other devices) (54–59)
020	F	YRTØDT		I	P		A		94				DISC	MYPØØL		FILEA
030	F	WKLY		I	S	E	A		80				READER	1		
040	F	REPØRT		Ø					132		OV		PRINTER	2		
05	F															
06	F															
07	F															

FORM 40-344 (OBSOLETES FORM 40-254)

SINGER
BUSINESS MACHINES

Section 4
INPUT SPECIFICATIONS FORM

RPG INPUT SPECIFICATIONS

10K Compiler Feature

Date ____________
Program ____________________
Programmer ____________________

Punching Instruction	Graphic							
	Punch							

Page 1 2 [ ][ ]

Program Identification 75 76 77 78 79 80 [ ][ ][ ][ ][ ][ ]

				AND OR				Record Identification Codes														Field Location							Field Indicators		
								1				2				3															
Line	Form Type	Filename	NOT USED	Sequence	Number (1/N)	Option (O)	Record Indentifying Indicators	Position	Not (N)	Type (C/D/Z)	Character	Position	Not (N)	Type (C/D/Z)	Character	Position	Not (N)	Type (C/D/Z)	Character	NOT USED	Hollerith Indicator	From	To	Decimal Positions	Field Name	Control Level (L1-L9)	Matching Fields	Field-Record Relation	Plus	Minus	Zero or Blank
3 4 5	6	7 8 9 10 11 12	13	14 15 16	17	18	19 20	21 22 23 24	25	26	27	28 29 30 31	32	33	34	35 36 37 38	39	40	41	42	43	44 45 46 47	48 49 50 51	52	53 54 55 56 57 58	59 60	61 62	63 64	65 66	67 68	69 70
0 1	I																														
0 2	I																														
0 3	I																														
0 4	I																														
0 5	I																														
0 6	I																														
0 7	I																														
0 8	I																														
0 9	I																														
1 0	I																														
1 1	I																														
1 2	I																														
1 3	I																														
1 4	I																														
1 5	I																														
	I																														
	I																														
	I																														
	I																														
	I																														

FORM 40-342 (OBSOLETES FORM 40-253)

SINGER
BUSINESS MACHINES

Figure 4.1 RPG INPUT SPECIFICATIONS FORM

INPUT SPECIFICATIONS FORM

The Input Specifications Form (Figure 4.1) provides the RPG compiler with detailed information about the input files, the records that compose the input files, and the arrangement of data fields within the records.

The lines on the form may be considered to be divided in two parts logically. Columns 7-41 provide data about the input file and the relationships between the records of the file. Columns 43-70 describe the structure of the data fields in the records.

The description of the input file (columns 7-41) and the description of fields in the records of that file (columns 43-70) must be on separate specification lines.

For example, if a record contained three fields, there would be one line on the Input Specifications that described the input file and the record identification in columns 7-41. Each of the next three lines would contain the description of <u>one</u> input field.

FORM TYPE (Column 6)

Must contain an I.

FILE NAME (Columns 7-12)

The file name entered here identifies which input file is being described. The file name must begin in column 7 and must agree with the file name given in the File Description Specifications. Every input file named in the File Description Specifications must also appear in the Input Specifications.

If a line of the Input Specifications form does not contain a file name, the data in that line refers to the last file name appearing in a previous line.

SEQUENCE (Columns 15-16)

Allowed Values	Meaning
Any two alphabetic characters.	No sequence checking will be done.
01-99	Sequence checking will be done.

A numeric entry in columns 15-16 specifies a particular sequence for records of different types, for example, names and addresses.

An entry of any two letters here implies that different types of records do not have any special order.

In any one file, all the record types with alphabetic characters in columns 15-16 are described first and then the records with numerical entries are described. An entry in columns 15-16 is required; blank is not valid.

When the user wishes to specify a sequence for his record types, he assigns numbers in ascending order. The first record type must be numbered 01. Numbers may be skipped, but all numbers must be in ascending order.

If a type of record is out of sequence, the program will be cancelled.

AND, OR (Columns 14-16)

Refer to the descriptions of the use of AND and OR in conjunction with Record Identification Codes under the sub-section "Columns 21-41 (Record Identification Codes)" later in this section.

NUMBER (Column 17)

Allowed Values	Meaning
Blank	Record types not checked for sequence number.
1	There is one record of this type in each sequenced group.
N	There are multiple records of this type per sequenced group.

When different record types are arranged in numerical sequence (numeric characters in columns 15-16), the contents of column 17 indicate whether the group contains just one record (digit 1) or multiple records (letter N).

OPTION (Column 18)

Allowed Values	Meaning
Blank	At least one record of this type must be present.(This applies when sequence checking is used.)
O	For "Option." Presence of records of this type is optional. (This applies only if sequence checking is used.)

When different record types are arranged in numerical sequence (numeric characters in columns 15-16), the contents of column 18 indicate whether the group must contain a record of this type or if records of this type are optional.

When the contents of columns 15-16 are alphanumeric (non-sequenced records), then columns 17-18 must be blank.

RECORD IDENTIFYING INDICATORS (Columns 19-20)

Allowed Values	Meaning
01-99	Record Identifying Indicator.
LR	Last Record Indicator.
H1-H3	Halt Indicator. Indicates that the program is checking for a record type that results in an error condition. If these indicators (H1 - H3) are turned on without being turned off later, the program will terminate.

Columns 19-20 are used to assign record identifying indicators to each type of record. Different record types are usually processed in different ways. When a record of a particular type is selected for processing, the Record Identifying Indicator for that type of record is turned on. This Record Identifying Indicator can be used to specify what calculations and output format operations will later be performed with this record, and under what conditions they will be performed.

It is not necessary to specify Record Identifying Indicators in any particular numeric sequence.

The same indicator may be specified for different record types. This causes the different record types to turn on the same indicator when they are selected for processing.

There may be only one Record Identifying Indicator for each record type. That is, an OR clause may not be used to assign a second identifying indicator to a particular record type.

RECORD IDENTIFICATION CODES (Columns 21-41)

The information in columns 21-41 is used to identify the type of record being processed. This identification code will then determine the manner in which the records are handled by the program.

When a file contains several types of records, one type of record will be processed during one cycle. A Record Identifying Indicator is set when the record is selected and remains on until the completion of that cycle.

If there is only one type of record present, columns 21-41 may remain blank.

Level Indicators (L1-L9) may not be used as record identification codes.

Use of AND

Each specification line between columns 21 and 41 may be used to specify up to three identifying characters. If more identifying characters are needed, an additional line (or lines) must be used and the word AND must be entered in columns 14-16 of the additional line(s). The programmer may use as many AND lines as he wishes to specify the record identification code. However, all the specified identifying characters must be present in the record before it will be processed properly. Columns 17-20 must be left blank when AND is used in columns 14-16.

Use of OR

Several different record identification codes may be used to specify a record type. To indicate that one of several valid identifications will be found in the record, the characters OR are placed in columns 14-15 of the specification line containing the alternate code. Columns 16-20 must be left blank when OR is used in columns 14-15.

Record Identification Code Sub-fields

It is possible to specify three characters per line. Each identification code is given in a seven-column field consisting of four parts: Position, Not (N), Type (C/D/Z), and Character. The three fields are clearly labeled 1, 2, and 3 on the Input Specifications Form.

The first record identification code field includes columns 21-27, with Position in columns 21-24, Not (N) in column 25, Type (C/D/Z) in column 26, and Character in column 27.

Field 2 covers columns 28-34, with Position in columns 28-31, Not (N) in column 32, Type (C/D/Z) in column 33, and Character in column 34.

Field 3 extends across columns 35-41, with Position in columns 35-38, Not (N) in column 39, Type (C/D/Z) in column 40, and Character in column 41.

The contents of these fields are as follows:

Position (Columns 21-24, 28-31, or 35-38)

Allowed Values	Meaning
1-415 (9K Compiler)	Specifies the position in the input record of a character in the record identification code. Must be right-justified.
1-940 (10K Compiler)	Specifies the position in the input record of a character in the record identification code. Must be right-justified.
Blank	The record requires no identification code. If Position is left blank, then Not (N), Type (C/D/Z), and Character must also be blank.

Columns 21-24, 28-31, and 35-38 are used to give the positions in the input record of the characters that make up the record identification code. The characters of the record identification code are arranged from left to right (fields 1, 2, and 3) and on subsequent specification lines if AND is used.

Not (N) (Columns 25, 32, or 39)

Allowed Values	Meaning
N	The specified character must not be present in the specified position of the input record.
Blank	The specified character must be present in the specified position of the input record.

The entry in columns 25, 32, or 39 indicates whether the specified character should or should not be present in a particular position of the input record.

RPG INPUT SPECIFICATIONS

10K Compiler Feature

Date ____________
Program ____________
Programmer ____________

Punching Instruction	Graphic							
	Punch							

Page 1 2

Program Identification 75 76 77 78 79 80

Line (3-5)	Form Type (6)	Filename (7-14)	AND/OR Sequence (15-16)	Number (1/N) (17)	Option (O) (18)	Record Identifying Indicators (19-20)	Record Identification Codes 1: Position (21-24)	Not (N) (25)	Type (C/D/Z) (26)	Character (27)	2: Position (28-31)	Not (N) (32)	Type (C/D/Z) (33)	Character (34)	3: Position (35-38)	Not (N) (39)	Type (C/D/Z) (40)	Character (41)	Hollerith Indicator (43)	Field Location From (44-47)	To (48-51)	Decimal Positions (52)	Field Name (53-58)	Control Level (L1-L9) (59-60)	Matching Fields (61-62)	Field-Record Relation (63-64)	Field Indicators Plus (65-66)	Minus (67-68)	Zero or Blank (69-70)
01O	I	INPUT	AA			01	1		C	R																			
02O	I					02	80		D	4																			
03	I					03	25		Z	A																			
04	I																												

Line	Form Type	Comment
05	I	* ANY INPUT RECØRD WITH AN R IN CØL. 1 WILL SET ØN INDICATØR 01.
06	I	* ANY INPUT RECØRD WITH DIGIT BITS SAME AS 4 IN CØL. 80 WILL
07	I	* SET ØN INDICATØR 02.
08	I	* ANY INPUT RECØRD WITH ZØNE BITS SAME AS A IN CØL. 25 WILL
09	I	* SET ØN INDICATØR 03.
10	I	
11	I	
12	I	
13	I	
14	I	
15	I	

FORM 40-342 (OBSOLETES FORM 40-253)

SINGER BUSINESS MACHINES

Figure 4.2 USE OF DIGIT AND ZONE IN RECORD IDENTIFICATION CODES

Type (C/D/Z) (Columns 26, 33, or 40)

With the 9K compiler, an entry of C is mandatory. This indicates that any valid System Ten character may be entered in columns 27, 34, or 41, and that the exact character specified must (or must not) be present in the specified position in order to turn on the record identifying indicator.

The 10K compiler provides the option of using the entire character, only the digit portion, or only the zone portion of the character in the record identification code. Thus, with the 10K compiler, allowed entries in columns 26, 33, or 40 are C, Z, or D. If C is entered, the entire character (six bits) will be used for the record identification. If D is entered, only the digit portion of the character (four rightmost bits) will be used as the record identification code.

Refer to Figure 4.2, line 020. The entry for Position is '80', there is no entry for Not, Type is D and Character is 4. Thus, any card that is read having a 4 in column 80 will cause the record identifying indicator to be turned on. However, cards having any of the characters $, D, or T in column 80 will also cause the record identifying indicator to be turned on, since the digit portions of the character codes for $, 4, D, and T are identical.

In line 030 of Fig. 4.2, the record identifying indicator is 03, Position is 25, Not is blank, Type is Z, and character is A. Thus, any card that is read having a character in column 25 with the same zone configuration as A (that is, A,B,C...0 and @) will turn on the indicator 03.

Character (Columns 27, 34, or 41)

Allowed Values	Meaning
Any alphabetic (A-Z), numeric (0-9), or special character in the character set.	This is the character that will appear in the specified position of the input record to serve as a record identification code, or part of the record identification code. Blank is a valid character.

For example, in Fig. 4.3 the indicator 01 is turned on every time a record is read having an A in column 1. If a record is read that contains anything else but an A in column 1, indicator 02 will be turned on. Thus, records having a missing or improper identification code may be detected and an indicator turned on which can be used to cause further action by the program.

RPG INPUT SPECIFICATIONS

10K Compiler Feature

Date ____________
Program ____________
Programmer ____________

Punching Instruction: Graphic / Punch

Page 1 2

Program Identification 75 76 77 78 79 80

Line (3-5)	Form Type (6)	Filename (7-12)	NOT USED (13)	AND/OR Sequence (14-16)	Number (1/N) (17)	Option (O) (18)	Record Indentifying Indicators (19-20)	Record Identification Codes 1: Position (21-24)	Not (N) (25)	Type (C/D/Z) (26)	Character (27)	Record Identification Codes 2: Position (28-31)	Not (N) (32)	Type (C/D/Z) (33)	Character (34)	Record Identification Codes 3: Position (35-38)	Not (N) (39)	Type (C/D/Z) (40)	Character (41)	NOT USED (42)	Hollerith Indicator (43)	Field Location From (44-47)	Field Location To (48-51)	Decimal Positions (52)	Field Name (53-58)	Control Level (L1-L9) (59-60)	Matching Fields (61-62)	Field-Record Relation (63-64)	Field Indicators Plus (65-66)	Field Indicators Minus (67-68)	Field Indicators Zero or Blank (69-70)
010	I	IN		AA			01	1		C	A																				
020	I																					2	5	0	ACCTNØ						
030	I																					6	30		NAME						
040	I																					31	50		ADDRES						
050	I																					51	65		CITY						
060	I																					66	70		STATE						
070	I																					71	75	0	ZIP						
080	I																				H	76	80	2	AMTØWD						
090	I			BB			02	1	N	C	A																				
10	I																														
11	I																														
12	I																														
13	I																														
14	I																														
15	I																														
	I																														
	I																														
	I																														
	I																														
	I																														

FORM 40-342 (OBSOLETES FORM 40-253)

SINGER BUSINESS MACHINES

Figure 4.3 SAMPLE RPG INPUT SPECIFICATIONS

If the last input record specification has an indicator assigned and no identification codes designated, it will act as a "catch-all", and that indicator will be turned on whenever a record is encountered which does not conform to any designated record identification. If such a catch-all indicator is not assigned, a record with an improper identification code might cause the program to terminate.

HOLLERITH INDICATOR (Column 43)

Applies to card input only.

Allowed Values	Meaning
Blank	Input field uses ANSI code for negative numbers (minus sign in low-order position), P through Y being equivalent to -0 through -9.
H	Input field uses Hollerith code for negative numbers (minus sign in low-order position), an 11-zone punch over the digits 0 through 9 being equivalent to -0 through -9.

The Hollerith indicator pertains to a field and not to an entire record. In addition, H applies to numeric fields only.

> Note: In Hollerith code a 12-zone overpunch in the low-order position of a numeric field signifies an explicitly positive number. For example, the Hollerith punch code for +1 is a 12-1 punch, which is equivalent to the ANSI code for the character A.

9K Compiler

With the 9K compiler, an entry of H in column 43 will not cause the proper recognition of 12-zone overpunches. A numeric field with a 12-zone overpunch in the low order position will be translated to zero in core.

10K Compiler

With the 10K compiler, an entry of H in column 43 will cause a 12-zone overpunch in the low order position of a numeric field to be translated according to the Hollerith code. For example, an A (12-1) will be entered as +1 into the system. The translation is done according to the following table:

INPUT CHARACTER	HOLLERITH CODE	RPG INTERPRETATION
A	12 and 1	+1
B	12 and 2	+2
C	12 and 3	+3
D	12 and 4	+4
E	12 and 5	+5
F	12 and 6	+6
G	12 and 7	+7
H	12 and 8	+8
I	12 and 9	+9

To summarize the differences between the two compilers: with 9K compiler, an H entry will cause Hollerith punched input for negative fields to be properly translated, but not positive fields; with the 10K compiler, an H entry will cause Hollerith input for both negative and positive fields to be properly translated.

FIELD LOCATION (Columns 44-51)

Columns 44-51 give the location in the input record of the field named in columns 53-58 (Field Name). The Field Location is divided into two portions: "From" in columns 44-47, and "To" in columns 48-51.

From (Columns 44-47)

Allowed Values	Meaning
1-415 (9K Compiler)	Location in record where the named field begins.
1-940 (10K Compiler)	Location in record where the named field begins.

To (Columns 48-51)

Allowed Values	Meaning
1-415 (9K Compiler)	Location in record where the named field ends.
1-940 (10K Compiler)	Location in record where the named field ends.

The location specified in "From" must be numerically less than or equal to the location specified in "To". The difference between the two numbers given for "From" and "To" is, of course, the field length minus one. If the numbers in "From" and "To" are the same, this implies the field is one character in length.

Numerical data has a maximum field length of 18 digits. The maximum length of a field of alphanumeric characters is 100 characters, or the maximum record length of the device, whichever is less.

The "From" and "To" numbers must end in columns 47 and 51, respectively. Leading zeros may be left out.

A separate line must be used for each field description.

DECIMAL POSITIONS (Column 52)

Allowed Values	Meaning
Blank	Record field is alphanumeric.
0-9	Number of decimal places to the right of the decimal point.

Whenever the field named in columns 53-58 is numeric, there must be an entry in column 52. The record field must be numeric if calculations or edit operations are to be performed on the contents of the field. If the number of decimal places specified is greater than the field length, an error diagnostic will be issued. In this case, object program output will not be reliable.

FIELD NAME (Columns 53-58)

Allowed Values	Meaning
1-6 alphanumeric characters.	Name of field in input records. First character must be alphabetic. Special characters and spaces within a name are not permitted. Must be left-justified.
PAGE	Input field will contain a number that is one less than the starting page number of the output.
UDATE	Input field will contain a date to be used on reports. Entering UDATE in the Input Specifications causes the object program to ignore the date in common.

Columns 53-58 are used to name an input field. The programmer must refer to the field by this name throughout his program. Only fields that will be used by the program should be named.

The name must start in column 53.

Within one type of record, the different fields must have different names.

However, a field name used in one type of record may be identical to that used in another type of record, so long as the field length and data type are the same. This is true even if the fields have different locations in the different record types.

When PAGE appears as an input field name, it indicates that the input field locations specified will contain a number which determines the output page starting number. The starting page number is calculated by adding one to the number in the input field. For example, if the input field named PAGE contains a 9, the output will start with page 10.

PAGE may also appear as a field name on the Output Format Specifications Form (see description of columns 32-37 in Section 6). If PAGE appears in the Output Format Specifications but not in the Input Specifications, page numbering will start with page 1.

Thus, including a PAGE field in the Input Specifications allows the user to have an initial page number other than 1, or page numbers that are governed by the input records.

The PAGE field may have any length but may not use decimal positions. That is, the page numbers must be integers.

The number in the PAGE input field must be right-justified. For example, if the PAGE field has four locations and the initial page number is to be 16, the PAGE input field must contain 0015.

The page number may be reset by including a new PAGE input field with a new input record.

Calculations may be done on the PAGE field, as with any other numerical field.

CONTROL LEVEL (Columns 59-60)

The calculation and printing of totals is governed by Control Level Indicators. An input field that has a Control Level Indicator is called a control field. When the information in a control field changes, a control break is said to occur. A group of records that have the same information in their control fields is called a control group.

When there is a control break, the appropriate Control Level Indicator is turned on. This Control Level Indicator can then be used to govern the calculation and printing of totals.

Allowed Values	Meaning
L1 - L9	Control Level Indicators

The Control Level Indicators are numbered 1-9. Larger numbers indicate a higher rank. When a Control Level Indicator with a higher value is turned on, all lower value indicators are also turned on. For example, if L5 is on, L1-L4 will also be on.

There is also a Control Level Indicator L0, which is always turned on; thus, it may not be assigned or turned off by the programmer (refer to the section on the Calculation Specifications Form, Columns 7-8).

As an example of the use of control levels, the following fields could be assigned Control Level Indicators as shown:

ACCT	L1
SALSMN	L2
DEPT	L3
BRANCH	L4

Thus, whenever the contents of ACCT changes (that is, a different account number is read in the input data) the calculation and printing of total values can be caused: for example, "total sales," "total amount owed," or whatever input data one wishes to total. If the value of DEPT (department number) changes, then indicators L1, L2, and L3 are turned on and can be used to cause the printing of totals for Account, Salesman, and Department. The Branch totals would not be printed until there was a change in the BRANCH control field.

With respect to control level indicators, the following points should be noted.

- When the same control level indicator has been assigned to fields in records of different types, the fields must be of the same length and type (numeric or alphanumeric).

- Within one record type, a maximum of 100 characters may be assigned to control fields.

- It is not necessary to assign Control Level Indicators in a particular order. For example, L3 may be assigned first, L7 next, and then L2.

- If a field contains numeric information, only the numerical digits are compared to see if a change of control group has occurred. In other words, minus signs and decimal positions are ignored. For example, -2 is considered equivalent to +2, and 3.8 is considered equivalent to 38.

- For all control fields having the same control level indicator, if the first control field specified is numeric, then all others are assumed to be numeric; if alphanumeric, then all others are assumed to be alphanumeric.

- The initial contents of all control fields are set as spaces.

- A control break is likely to occur when the first control field in a program is processed, since the contents of this field are compared to a storage area containing spaces. This is not a "true" control break and calculations of totals are omitted. Thus, calculations and output operations resulting in totals are not done until the second record containing control fields is read.

MATCHING FIELDS (Columns 61-62)

Records or parts of records (data fields) from a file can be compared against records or fields of one or more other files to determine when the field or record contents are identical.

Allowed Values	Meaning
M1-M9	Matching Field Level

The matching field level (M1-M9) identifies the fields to be matched. If matching fields are found during processing, the Matching Record (MR) Indicator is turned on. The MR Indicator can be used to cause certain operations to occur (refer to columns 9-17 of the Calculation Specifications, and columns 23-31 of the Output Format Specifications).

The programmer should keep the following points in mind when assigning matching field levels:

a. Record types with matching fields specified must be in sequential order and are checked for proper sequence (ascending or descending order, as specified in column 18 of the File Description Specifications). If a record is found to be out of sequence, the program is cancelled.

b. Matching of fields is optional. It isn't necessary that matching fields be present in all files of the program. Nor do matching fields have to be present in different record types of a file.

c. If one record type has fields that match those in another record type, the number of matching fields in each record type must be equal.

d. When matching fields are assigned the same matching level (M1-M9), the length and type of the matched fields must be the same.

e. Matching fields in different records may have overlapping locations. However, 100 characters is the maximum allowed for the total of all field lengths.

f. When there are several matching fields in a record type, all the fields are concatenated into a single combined match field. The combination of fields is in order of significance with M9 being most significant. The order is from M9 to M1.

g. A particular matching level (M1-M9) can be used only once within a particular record type.

h. The presence of decimal positions is ignored when matching numeric fields. For example, if 5.42 were the contents of a field to be matched with a field containing 542, the contents would be considered identical.

RPG INPUT SPECIFICATIONS

10K Compiler Feature

Date ____________
Program ____________
Programmer ____________

Punching Instruction: Graphic / Punch

Page 1 2

Program Identification 75 76 77 78 79 80

Line (3-5)	Form Type (6)	Filename (7-14)	AND/OR, Sequence (14-16)	Number (1/N) (17)	Option (O) (18)	Record Identifying Indicators (19-20)	Record Identification Codes 1: Position (21-24)	Not (N) (25)	Type (C/Z/D) (26)	Character (27)	Field Location From (44-47)	Field Location To (48-51)	Decimal Positions (52)	Field Name (53-58)	Control Level (L1-L9) (59-60)	Matching Fields (61-62)	Field-Record Relation (63-64)	Field Indicators Plus (65-66)	Field Indicators Minus (67-68)	Field Indicators Zero or Blank (69-70)
010	I	INFILE	01	1		01	05		C	6										
020	I		ØR			02	05		C	7										
030	I										1	4		NØ						
040	I										41	46	2	AMT						
050	I										47	49	2	DSCNT						
060	I										50	55		PART						
070	I										56	60		TYPE						
080	I										61	69		REF						
090	I										70	74		ØNHAND						
100	I										15	25	2	DEBIT			01		99	99
110	I										75	76		FLAG		M1	01			
120	I										25	36	2	DEBIT			02		88	88
130	I										77	78		FLAG		M1	02			
14	I																			
15	I																			

FORM 40-342 (OBSOLETES FORM 40-253)

SINGER BUSINESS MACHINES

Figure 4.4 USE OF FIELD-RECORD RELATION AND FIELD INDICATORS

i. Minus signs are ignored. For example, -71 will be considered identical to +71.

j. If several fields in one record type are to be matched to fields in another record type, all the fields must match before the matching record (MR) Indicator will turn on.

k. The program disregards field names when matching records or fields. It is permissible to use the same field name for matching fields in different record types.

l. The sequence (A or D in column 18) of the first file encountered when matching fields is used for all files involved in that matching level.

FIELD-RECORD RELATION (Columns 63-64)

10K Compiler Feature Only

The user may wish to combine records which contain the same information but have different format. For example, he may wish to combine, in the same input file, cards prepared under an old adminstrative system with cards prepared under a newer system. The Field-Record Relation indicator allows this.

Refer to Figure 4.4. Here two types of records are being combined in the input file INFILE. The first record format is distinguished by the character '6' in column 5 and causes indicator 01 to be turned on. The second record format is distinguished by a character '7' in column 5 and causes indicator 02 to be turned on. Most of the fields within the two record formats are the same and are simply listed. However in the case of the DEBIT field, it is located in columns 15-25 in the first record format and in columns 25-36 in the second record format. Also, the field named FLAG is located in columns 75-76 in the first record format and in columns 77-78 in the second record format. The record identifying indicators 01 and 02 are entered in the Field-Record Relation columns opposite the appropriate field location description.

The procedure is simply to place the appropriate record identifying indicator in the Field-Record Relation (columns 63-64) opposite the name and location of the field from the record with that indicator.

Allowed Values	Meaning
01-99	Record identifying indicator for record to which this field belongs.

Note: All fields which have no Field-Record Relation must be described before fields with Field-Record Relations. (See Figure 4.4.)

FIELD INDICATORS (Columns 65-70)

10K Compiler Feature Only

Columns 65-70 may be used to test the contents of a field named in columns 53-58. Columns 65-66 and 67-68 are used with numeric fields only, while columns 69-70 may be used in conjunction with numeric or alphanumeric fields. If the field named contains a positive number, the indicator specified in columns 65-66 is turned on. If the field named contains a negative number, the indicator specified in columns 67-68 will be turned on. If the field named contains zeros or blanks, the indicator specified in columns 69-70 will be turned on.

Allowed Values	Meaning
01-99, H1-H3	This is the indicator that will be turned on if the condition specified at the top of the columns where the indicator appears is satisfied.

The Halt indicators may be used to terminate the execution of an object program if data is encountered which is clearly incorrect: for example, a negative number entered for an employee's hourly pay rate. Figure 4.4 gives an example of the use of Field Indicators.

Section 5

CALCULATION SPECIFICATIONS FORM

RPG CALCULATION SPECIFICATIONS

10K Compiler Feature

Date ____________

Program ____________

Programmer ____________

Punching Instruction	Graphic							
	Punch							

Page (1 2) ______

Program Identification (75 76 77 78 79 80) ______

Line (3 4 5)	Form Type (6)	AN/OR; Control Level (L0-L9, LR) (7 8)	Indicators: Not, And Not, And Not (9–17)	Factor 1 (18–27)	Operation (28–32)	Factor 2 (33–42)	Result Field (43–48)	Field Length (49 50 51)	Decimal Positions (52)	Half Adjust (H) (53)	Resulting Indicators: Arithmetic Plus / Compare High F1>F2 (54 55)	Arithmetic Minus / Compare Low F1<F2 (56 57)	Arithmetic Zero / Compare Equal F2=F1 (58 59)	Comments (60–74)
01	C													
02	C													
03	C													
04	C													
05	C													
06	C													
07	C													
08	C													
09	C													
10	C													
11	C													
12	C													
13	C													
14	C													
15	C													
	C													
	C													
	C													
	C													
	C													

FORM 40-343 (OBSOLETES FORM 40-252)

SINGER
BUSINESS MACHINES

Figure 5.1 RPG CALCULATION SPECIFICATIONS FORM

CALCULATION SPECIFICATIONS FORM

The Calculation Specifications Form (Fig.5.1) describes the operations to be performed on the input data. The form is divided into three major areas:

- Columns 7-17 contain indicators that specify the conditions causing calculations to be performed.
- Columns 18-53 state what calculations are to be done, and on what data.
- Columns 54-59 specify tests to be made on the results of calculations. The outcome of these can then be used to govern certain output operations or further calculations.

FORM TYPE (Column 6)

Must contain a C.

CONTROL LEVEL (Columns 7-8)

Allowed Values	Meaning
L0, L1-L9	The calculation described on this line is done only when a control break occurs for the control level indicator specified in columns 7-8.
LR	The calculation described on this line will be done after the last input record has been read.
Blank	The calculation described on this line is done after each record is read, (if columns 9-17 are also blank). These are called detail calculations.

Detail calculations must be specified before control level (total) calculations.

Although the L0 Indicator may be used by the programmer, it cannot be assigned, since it is always turned on. Whenever there is a control break, the program performs all calculations and operations which have been assigned Control Level Indicators. The L0 Indicator may be used to cause totals to be printed.

If L1-L9 are used to determine when certain calculations will be done, the calculations will be done only when these indicators are on. (Refer to the Input Specifications Form, columns 59-60, where Control Level Indicators are assigned.)

When a Control Level Indicator is turned on by a control break, all Control Level Indicators with lower numbers are turned on also. Thus, if L6 is turned on, not only will the operations governed by L6 be done, but the operations corresponding to L1-L5 as well.

When the last record has been read, the LR Indicator turns on. When LR is turned on, all other Control Level Indicators (L1-L9) will be on.

Thus, using L1-L9 and LR, up to ten different levels of totals can be calculated and printed out. For example, suppose we want to calculate total sales for a world wide company using ten different categories of totals. We could assign Control Level Indicators to control fields as follows:

ACCT	L1
SALSMN	L2
DEPT	L3
BRANCH	L4
CITY	L5
STATE	L6
REGION	L7
NATION	L8
CONTNT	L9

L9 gives the continental total. When the last record is read, LR is turned on and we can obtain the world total for sales.

When a Control Level Indicator is set on or set off using the SETON or SETOF operation code, Control Level Indicators with lower numbers are not affected.

When LR is set on using the SETON operation, Control Level Indicators L1-L9 are not affected. LR cannot be set off using the SETOF operation code. LR must be on for the program to terminate normally.

INDICATORS (Columns 9-17)

Columns 9-17 are used to specify the indicators that must or must not be on if the operation specified on the line is to be performed.

For example, a particular type of input record may have been assigned a Record Identifying Indicator of 66 in columns 19-20 of the Input Specifications Form. If a line on the Calculation Specifications Form then specifies an indicator of 66 in columns 10-11, the specified operation will be done only when an input record of that type is read.

Columns 9-17 provide space for three different indicators: in columns 9-11, 12-14, and 15-17. Columns 9, 12, 15 may contain an N, which means that the operation is to be done only if the specified indicator is <u>not</u> on.

If two or three indicators are entered in columns 9-17, they are assumed to have an AND relationship. That is, all specified conditions must be satisfied before the operation will be done.

The results of a calculation can be used to turn on a resulting indicator. These resulting indicators are described under columns 54-59 of the Calculation Specifications Form. The indicators in columns 9-17 may refer to indicators resulting from a previous operation.

Allowed Values	Meaning
Blank	Operation is performed every time a card is read. (Assumes columns 7-8 are blank also.)
01-99	Record Identifying Indicators or resulting indicators specified earlier in the program.
L1-L9	Control Level Indicators specified previously.
LR	Last Record Indicator.
MR	Matching Record Indicator.
H1-H3	Halt Indicators specified previously.
OA-OG, OV	Overflow Indicators specified earlier in the program.
RS (10K Compiler Only)	Service Request Indicator. This indicator is turned on whenever a service request is received.

The Halt Indicators H1-H3 are used to terminate the object program. They can be utilized to prevent a record from being processed when an input error is detected or the result of a previous calculation indicated an error. A Halt Indicator can also be used to cause a certain operation to be done only when an error is detected. Halt Indicators are set on only when specified by the user. RPG does not set on Halt Indicators automatically.

The MR Indicator will cause the specified operation to occur only when matching records have been found.

The LR Indicator is used to specify all operations to be done when the last record of a job has been read.

An Overflow Indicator used in columns 9-17 must agree with the Overflow Indicator specified in columns 33-34 of the File Description Specifications Form. The Overflow Indicator will be turned on when the last line to be printed on a page of output has been reached.

If a Control Level Indicator (L1-L9) appears in columns 7-8 and an MR is specified in columns 9-17, the Matching Record Indicator refers to the record previously read (just prior to the control break). When all operations caused by Control Level Indicators have been done, the MR Indicator indicates a matching condition of the current record (the one causing the control break).

If a Control Level Indicator appears in columns 9-17, the operation specified is done only on the record causing the control break (or on a record causing a control break of higher level).

Within a program cycle, the operations caused by a Control Level Indicator in columns 7-8 will all be done before operations conditioned by Control Level Indicators in columns 9-17.

With the 10K compiler only, a Service Request indicator RS, is available. It is turned on whenever the executing program receives a service request and remains on for one program cycle, after which it is automatically turned off. This indicator may be used like any other indicator to determine when operations are to be done.

AN, OR (Columns 7-8)

10K Compiler Feature Only

With the 10K compiler, more than one line may be used to specify the indicators that determine when a calculation or operation is to be done. AN stands for logical AND; when it appears in columns 7-8 of a calculation specifications line, it implies that all indicator values specified on this line as well as indicator values specified on the previous line must be satisfied before the specified calculation or operation will be done. OR is the logical OR: when it appears in columns 7-8 of a calculation specifications line, it implies that if either the indicator values on this line or the indicator values specified on the previous line are satisfied, the specified calculation or operation will be performed.

Many lines of indicators may be used in combination with AN and OR specified in columns 7-8. The operation to be done is specified only on the last line of a group of lines connected by AN or OR. Refer to Figure 5.2. Here we see that the calculation of adding EXTRA to SALES will be done only if indicators 01, 02, 03, 04, and 05 are all on, or if indicator 06 is on and indicators 03 and 05 are not on.

FACTOR 1 AND FACTOR 2 (Columns 18-27 and Columns 33-42)

The data on which the specified operation is to be performed is specified by Factor 1 and Factor 2. For example, if the operation is subtraction, Factor 2 is subtracted from Factor 1. Some operations require two operands; some require one operand; and some require none. These will be discussed further under Operation (columns 28-32).

The data on which the operation is to be performed may be specified by naming the field or by entering the actual data (literal).

The allowed entries are

- The name of any defined field.
- Literal data (either numeric or alphanumeric).
- The date field name, UDATE.
- The special name, PAGE.
- A label for a TAG operation (Factor 1 only).
- A label for a GOTO or EXIT operation (Factor 2 only).

Entries in Factor 1 or Factor 2 must be left-justified.

Literal data is the actual data used in an operation rather than the name of a field which contains the data. Examples of literals would be 23174.02 or 'MR. JONES'. A literal may be alphanumeric or numeric.

RPG CALCULATION SPECIFICATIONS

10K Compiler Feature

Date ____________

Program ____________

Programmer ____________

Punching Instruction	Graphic							
	Punch							

Page 1 2

Program Identification 75 76 77 78 79 80

Line (3-5)	Form Type (6)	Control Level (L0-L9, LR) / AN OR (7-8)	Not (9)	Indicator (10-11)	Not (12)	Indicator (13-14)	Not (15)	Indicator (16-17)	Factor 1 (18-27)	Operation (28-32)	Factor 2 (33-42)	Result Field (43-48)	Field Length (49-51)	Decimal Positions (52)	Half Adjust (H) (53)	Resulting Indicators: Plus / Compare High F1>F2 (54-55)	Minus / Low F1<F2 (56-57)	Zero / Equal F2=F1 (58-59)	Comments (60-74)
010	C			01		02		03											
020	C	AN		04		05													
030	C	OR		06	N	03	N	05	SALES	ADD	EXTRA	ADJSLS	6	2			10	20	ADJUST SALES
04	C																		
05	C																		
06	C																		
07	C																		
08	C																		
09	C																		
10	C																		
11	C																		
12	C																		
13	C																		
14	C																		
15	C																		
	C																		
	C																		
	C																		
	C																		
	C																		

FORM 40-343 (OBSOLETES FORM 40-252)

SINGER BUSINESS MACHINES

Figure 5.2 USE OF AN, OR WITH INDICATORS

The following rules pertain to <u>alphanumeric</u> literals:

a. They must be enclosed between apostrophes (single quote marks).

b. Any characters of the System Ten character set are permitted.

c. The maximum length is 8 characters.

d. Imbedded blanks are permitted.

e. Alphanumeric literals may not be used in arithmetic operations.

f. An apostrophe within an alphanumeric literal must be represented by two apostrophes. For example, the literal IT'S OK would be represented as 'IT''S OK'.

The following rules pertain to <u>numeric</u> literals:

1. They may not be enclosed within apostrophes.

2. Permitted characters include the numerals 0-9, the decimal point and the minus sign.

3. The maximum length is 10 characters, which includes the decimal point and minus sign, if present.

4. Imbedded blanks are not allowed.

5. Numeric literals are used in arithmetic operations exactly as numeric fields would be.

6. If a minus sign is present, it must be the leftmost character.

   <u>Note</u>: If literals are used in a COMP (Compare) or MOVE operation, the literal field will be expanded automatically by the object program to fill the required size.

RPG CALCULATION SPECIFICATIONS

10K Compiler Feature

Date ______________
Program ______________
Programmer ______________

Punching Instruction	Graphic							
	Punch							

Page (1 2) ____

Program Identification (75 76 77 78 79 80) ____

Line (3-5)	Form Type (6)	Control Level (L0-L9, LR) / AN OR (7-8)	Indicators (9-17)	Factor 1 (18-27)	Operation (28-32)	Factor 2 (33-42)	Result Field (43-48)	Field Length (49-51)	Decimal Positions (52)	Half Adjust (H) (53)	Resulting Indicators Plus / High F1>F2 (54-55)	Minus / Low F1<F2 (56-57)	Zero / Equal F2=F1 (58-59)	Comments (60-74)
010	C		05		Z-ADD	BLNCE	BLNCE				06	07	08	TEST FØR SIGN
020	C		06		GØTØ	INTRST								
030	C		07		GØTØ	CREDIT								
040	C		08		GØTØ	PDØFF								
05	C													
06	C													
07	C													
08	C													
09	C													
10	C													
11	C													
12	C													
13	C													
14	C													
15	C													
	C													
	C													
	C													
	C													
	C													

FORM 40-343 (OBSOLETES FORM 40-252)

SINGER BUSINESS MACHINES

Figure 5.3 USE OF Z-ADD AS A TEST FOR SIGN

OPERATION (Columns 28-32)

The operation code in columns 28-32 specifies what operation is to be performed. For example, ADD is the operation code used to add Factor 2 and Factor 1. The operation code must start in column 28.

OPERATION CODES

The operation codes are as follows:

ADD

Causes the contents of field named in Factor 2 to be added to contents of field named in Factor 1. Result is stored in Result Field (see columns 43-48). Numeric literals rather than field names may be used in Factor 1 and Factor 2.

Z-ADD

"Zero and Add" sets the contents of the Result Field to zeros and adds the contents of the field named in Factor 2 to the Result Field. Factor 2 may also contain a numeric literal. Factor 1 is not used in the Z-ADD operation. Z-ADD has the effect of transferring the contents of the numeric field in Factor 2 to the Result Field.

With the 9K compiler, an attempt to Z-ADD a field to itself is ignored by the compiler. No diagnostic is issued.

10K Compiler Feature Only

With the 10K compiler, the Z-ADD operation can be used to determine the sign of the contents of a field. This is possible because in the Z-ADD operation with the 10K compiler, the contents of the field named in Factor 2 are moved to a temporary storage location, then the field named in Result Field is set to zeros and the contents are moved from the temporary location to the field named in Result Field. Thus, a Z-ADD with the same field named in Factor 2 and Result Field will not affect the contents of the named field and the Resulting Indicators can be used to determine the sign of the field. Figure 5.3 shows how Z-ADD can be used to perform such a test.

SUB

Causes numeric field named in Factor 2 (or literally presented there) to be subtracted from numeric field named in Factor 1 (or literally presented there). The difference is stored in the field named in Result Field.

Z-SUB

"Zero and Subtract" causes the contents of the Result Field to be set to zeros and the numeric field literally presented or referenced in Factor 2 then to be subtracted from the Result Field. This has the effect of storing the negative of the contents of Factor 2 into the Result field. Factor 1 is not used in Z-SUB.

Z-SUB can be used to reverse the sign of a number if the same numeric field is specified in both Factor 2 and Result Field.

MULT

Causes the contents of the numeric field named in Factor 1 (or the numeric literal present there) to be multiplied by the contents of the field named in Factor 2 (or the numeric literal present there). The product is stored in the field named in Result Field. The contents of Factor 1 and Factor 2 are limited to a length of 10 decimal digits. The product stored in the Result Field may have a maximum length of 18 decimal digits.

DIV

Causes the numeric field named or literally present in Factor 1 to be divided by the numeric field named or literally present in Factor 2. The quotient is stored in the field named in Result Field. The field named in Factor 1 may have a maximum length of 18 decimal digits. The field named in Factor 2 may have a maximum length of 10 decimal digits. The contents of Result Field have a maximum length of 18 digits, but the maximum number of significant digits is ten. If the Half Adjust (column 53) is specified, nine is the maximum number of significant digits. The numeric contents of the field specified in Factor 2 must not be zero. If Factor 2 is zero, the program will be cancelled.

An additional requirement is that the lengths and decimal positions of the fields involved in a DIV operation must be such that

$$L(F1)-D(F1)+D(F2)+D(R) < 20$$

where:

L(F1) = Length of Factor 1
D(F1) = Number of decimal places in Factor 1
D(F2) = Number of decimal places in Factor 2
D(R) = Number of decimal places in the Result Field.

MVR

The Move Remainder operation may be used after a Divide (DIV) operation. It stores the remainder from the division operation in the field named in the Result Field. MVR uses only Result Field; entries in Factor 1 and Factor 2 are not valid with this operation. The RPG program will keep track of proper placement of the decimal point in the remainder.

The result of an MVR operation will not be valid if any operation other than SETON, SETOF, GOTO, TAG, EXIT, or RLABL is performed between the DIV and the MVR operations. It is recommended that the MVR operation immediately follow the associated DIV operation.

MOVE

In the Move operation, the contents of the field named or the literal present in Factor 2 is moved to the field specified in the Result Field. Factor 1 is not used. The field named or the literal in Factor 2 can be either alphanumeric or numeric, as can the field named in Result Field. Characters are transferred from the field in Factor 2 to the Result Field starting from the right. Thus, if the field in Factor 2 is longer than the Result Field, only the right-hand portion of the field in Factor 2 is transferred. If the field in Factor 2 is shorter than the Result Field, the characters from Factor 2 will be moved to the rightmost portion of the Result Field. The excess characters on the left in the Result Field will not be changed. For example, if Factor 2 contains a three-character numeric literal 444, and the field specified in Result Field is numeric, six characters long, and contains 888888 prior to the MOVE operation, after the move the Result Field will contain 888444.

Examples of MOVE Operations

Factor 2	ABCD		
Result Field	BAY ST		
After operation		BAABCD	BA0000
When result field is:		Alphanumeric	Numeric

Factor 2	0175		
Result Field	BAY ST		
After operation		BA0175	BA0175
When result field is:		Alphanumeric	Numeric

Factor 2	J15Y		
Result Field	BAY ST		
After Operation		BAJ15Y	BA015Y
When result field is:		Alphanumeric	Numeric

Factor 2	567A		
Result Field	BAY ST		
After Operation		BA567A	BA0567
When result field is:		Alphanumeric	Numeric

Factor 2	B23QR		
Result Field	****		
After Operation		23QR	023R
When result field is:		Alphanumeric	Numeric

Factor 2	ABCDE		
Result Field	****		
After Operation		BCDE	0000
When result field is:		Alphanumeric	Numeric

Factor 2	12345		
Result Field	****		
After Operation		2345	2345
When result field is:		Alphanumeric	Numeric

MOVEL

The Move Left operation causes the contents of the field named or the literal present in Factor 2 to be moved to the field named in the Result Field. Factor 1 is not used. This operation differs from MOVE in that MOVEL starts moving the characters in Factor 2 from the <u>left</u> and places them left-justified in the Result Field. Thus, if the field named in Factor 2 was four characters long and contained ABCD and the field named in Result Field was six characters long and contained ZZZZZZ before the MOVEL operation, the contents of the Result Field after the MOVEL would be ABCDZZ.

If the numeric field specified in Factor 2 is shorter than the Result Field, the sign of Factor 2 (if any) is not moved. If the numeric field in Factor 2 is equal to or longer than the field in Result Field, the sign of Factor 2 is placed in the rightmost position of the Result Field.

<u>Examples of MOVEL Operations</u>

Factor 2	J234		
Result Field	BAY ST		
After Operation		J234ST	0234ST
When result field is:		Alphanumeric	Numeric

Factor 2	12345		
Result Field	****		
After Operation		1234	1234
When result field is:		Alphanumeric	Numeric

Factor 2	1#3YZ		
Result Field	****		
After Operation		1#3Y	013Y
When result field is:		Alphanumeric	Numeric

COMP

The Compare operation compares the contents of the fields specified or literals present in Factor 1 and Factor 2 with each other. As a result of the Compare operation, one of the resulting indicators specified in columns 54-59 is turned on. The Result field is not used with COMP. The resulting indicators can be used to control subsequent calculations or output operations. If the field in Factor 1 is greater than the field in Factor 2 (F1>F2), the resulting indicator in columns 54-55 is turned on. If F1<F2, then the indicator in columns 56-57 is turned on. Similarly, if the contents of the fields in Factors 1 and 2 are equal (F1=F2), the indicator specified in columns 58-59 is turned on.

Before a Compare operation, all indicators referenced in this Compare operation (columns 54-59) are set off. After the Compare, the appropriate indicator is set on.

The Compare operation can be used on both numeric and alphanumeric fields. When numeric fields are being compared, the numbers are aligned at the implied decimal point and any excess spaces are assumed filled with zeros. For example, if Factor 1 contains a numeric field of 8 digits with two decimal places (123456.12), and Factor 2 references a field specified as ten digits long with four decimal places (123456.1201), then Factor 1 would be set equal to 123456.1200 and Factor 2 is 123456.1201. Thus, F1<F2 and the resulting indicator specified in columns 56-57 would be turned on. Comparison of numeric fields is always algebraic. If one desires to compare the absolute values, he must first write the necessary instructions that will obtain the absolute values.

When alphanumeric fields are to be compared, the comparison is in accordance with the System Ten collating sequence. Alphanumeric fields are aligned at their leftmost characters (they are left-justified). When alphanumeric fields being compared have unequal lengths, the excess locations are assumed to be filled with blanks.

RPG CALCULATION SPECIFICATIONS

10K Compiler Feature

Date ____________
Program ____________
Programmer ____________

Punching Instruction	
Graphic	
Punch	

Page 1 2

Program Identification 75 76 77 78 79 80

Line (3–5)	Form Type (6)	Control Level (L0-L9, LR) (7–8)	Indicators (9–17)	Factor 1 (18–27)	Operation (28–32)	Factor 2 (33–42)	Result Field (43–48)	Field Length (49–51)	Decimal Positions (52)	Half Adjust (H) (53)	Plus / High F1>F2 (54–55)	Minus / Low F1<F2 (56–57)	Zero / Equal F2=F1 (58–59)	Comments (60–74)
01	C				Z-ADD	AMT	SAVAMT	7	2				30	
02	C			ST14	CØMP	'XX'					43	34	42	
03	C		42		GØTØ	TØTAL								
04	C		N30	TØT1	DIV	SAVAMT	QUØT	7	2					
05	C													
06	C													
07	C													
08	C			DETAIL	TAG									DETAIL CALCS
09	C													
10	C													
11	C													
12	C	L0		TØTAL	TAG									TØTAL CALCS
13	C	L1			EXCPT									
14	C	L1		AMT2	ADD	AMT3	AMT3	6	2					
15	C													
16	C													
17	C	L1			GØTØ	DETAIL								
	C													
	C													
	C													

FORM 40-343 (OBSOLETES FORM 40-252)

SINGER
BUSINESS MACHINES

Figure 5.4 EXAMPLE OF BRANCH BETWEEN DETAIL AND TOTAL CALCULATIONS

GOTO

The GOTO operation causes a branch from one point in the program to another. The point arrived at after the branch is specified by a name (tag) in Factor 2. This name must also appear in Factor 1 of a TAG operation. That is, the origin point of a branch is designated by a GOTO Operation. The destination point of the GOTO operation is marked by a TAG. For example, if DEDUCT is the TAG for a sequence of operations to be done when certain indicators are on, one would specify a GOTO operation with the Tag DEDUCT written in Factor 2. Columns 7-17 would contain the control level and other indicators that would cause the branch to be taken. A TAG operation would be placed just before the sequence of operations to be done with the name DEDUCT in Factor 1.

A GOTO operation thus has the branch designated in Factor 2 and the conditions under which the branch is taken specified in columns 7-17. Factor 1, Result Field and the rest of the line must be blank.

The "name tag" placed in Factor 2 must be alphanumeric, may have a maximum of six characters, must begin with a letter and may have no imbedded blanks. The name placed in Factor 2 must be identical to the name placed in Factor 1 of the associated TAG operation.

With 9K Compiler

The GOTO may branch forward or backward in the program within detail calculations, or within total calculations, but the branch may not go from a point within the detail calculations to a point within the total calculations (or vice versa).

With 10K Compiler

With the 10K compiler, a GOTO operation may branch from a point within detail calculations to a point within total calculations or vice versa. This feature requires that all names used with TAG operations in the program be unique. (With the 9K compiler, it is permissible to use the same TAG within detail and total calculations since branching between these parts of the program is not allowed.)

After a branch between detail and total calculations has been completed, the type of output associated with the initial type of calculations is performed. For example, if a GOTO occurs from total to detail calculations, total output will be done after the detail calculations have been completed. Figure 5.4 shows a GOTO operation that branches between the detail and total calculations.

TAG

The TAG operation is used to identify the destination point of a branch (GOTO) operation. The name in Factor 1 must be the same as that specified in Factor 2 of the corresponding GOTO operations. (See the preceding description of GOTO for further details.)

For a TAG operation, Factor 2, Result Field and Indicators, (columns 9-17), and Resulting Indicators must be blank. If the operations following the TAG operation are to be done at total time, the Control Level Indicator LO must be entered in columns 7-8.

With the 9K compiler, the same name may appear as a tag in the detail calculations and in the total calculations, and will be considered to be distinct, since branching from detail to total calculations (or vice versa) is not allowed. However, it is recommended that all tags be unique names.

With the 10K compiler, tag names must be unique, since branching between detail and total calculations is allowed.

SETON

This operation is used to turn on or "set on" the indicator or indicators specified in columns 54-59. Control Level Indicators (columns 7-8) and indicators (columns 9-17) can be used to determine when the SETON operation will be executed.

SETOF

The Set Off (SETOF) operation is used to turn off the indicators specified in columns 54-59. Control Level Indicators (columns 7-8) and indicators (columns 9-17) can be used to determine when the SETOF operation is to be done. SETOF may not be used to turn off indicators LR or LO.

EXIT

This operation is used to exit from the RPG program to a specified external subroutine. The name of the subroutine which will be executed is placed in Factor 2. Control Level Indicators (columns 7-8) and indicators (columns 9-17) may be used to determine when the exit is to be taken. Factor 1 and Result Field must be blank. Refer to Appendix B for additional information on linkage conventions.

RLABL

The Reference Label (RLABL) operation is used to specify a field or indicator which can be referenced by a subroutine external to the RPG program. Indicators are specified by the letters IN followed by the indicator name, for example, IN05 or INL2.

<u>CAUTION</u>: Do not SETOF the following indicators within the assembly routine:

LR
LO
1P
MR

It is the responsibility of the assembly-language programmer to adhere to the above rule.

If an IN name has been used as a field name (such as IN16), it will be considered as a field name and not an indicator when used in the RLABL operation. The name of this field is entered in Result Field (columns 43-48). If the field has not been previously defined in the RPG Input or Calculation Specifications, the field length and number of decimal positions must be entered in columns 49-51 and column 52, respectively.

All RLABL operations must immediately follow their associated EXIT operation. Control Level indicator LO must be used (columns 7-8) if any Control Level Indicator was used on the EXIT operation preceding the RLABL. Columns 7-8 must be blank on RLABL if they are blank on the preceding EXIT operation. (See the description of linkage conventions in Appendix B.)

RPG CALCULATION SPECIFICATIONS

10K Compiler Feature

Date ____________
Program ____________
Programmer ____________

Punching Instruction	Graphic							
	Punch							

Page (1 2) ____

Program Identification (75 76 77 78 79 80) ____

Line (3-5)	Form Type (6)	Control Level (L0-L9, LR) (7-8)	Indicators (9-17)	Factor 1 (18-27)	Operation (28-32)	Factor 2 (33-42)	Result Field (43-48)	Field Length (49-51)	Decimal Positions (52)	Half Adjust (H) (53)	Resulting Indicators: Arithmetic Plus/Minus/Zero; Compare High F1>F2 / Low F1<F2 / Equal F2=F1 (54-59)	Comments (60-74)
01	C		02N05	SALES	DSPLY	2						DISPLAY A FIELD
02	C	*									ØN THE	LINE PRINTER.
03	C		04		DSPLY	0	NET					DISPLAY A FIELD
04	C	*									ØN WØRK	STATIØN, AND
05	C	*									ENTER	NEW VALUE.
06	C	L2	15N04	SALES	DSPLY	0	NET					DISPLAY 2 FLDS
07	C	*									ØN WRK	STN , CHNGE 2ND
08	C	*									FIELD,	(NET).
09	C											
10	C											
11	C											
12	C											
13	C											
14	C											
15	C											
	C											
	C											
	C											
	C											
	C											

FORM 40-343 (OBSOLETES FORM 40-252)

SINGER
BUSINESS MACHINES

Figure 5.5 USE OF DISPLAY OPERATION

ADDITIONAL OPERATIONS WITH 10K COMPILER

DSPLY (10K Compiler Feature Only)

The 10K compiler provides a Display operation (DSPLY) that allows the Model 70 workstation to be used to print out the contents of a field; or print out the contents of a field, blank the field and enter new contents into that field; or print out two fields, blank the contents of the second field, and then enter new contents into the second field.

Enter the operation code DSPLY in columns 28-32. If a field is to be printed out only, the name of the field is entered in Factor 1, left justified. The number of the device to be used for the output display is entered in Factor 2, left-justified. In this case, Result Field must be blank. Figure 5.5, line 010, specifies that when indicator 02 is on and indicator 05 is not on, the field SALES will be displayed on the line printer (device 0).

If the contents of a field are to be printed out, replaced in core by blanks or zeros, and then replaced with new contents entered by the user, the field to be so handled is entered in Result Field. Line 030 in Figure 5.5 specifies that the field NET will be printed out on the workstation, the contents of the field will be set to zeros (if the field is numeric) or blanks (if the field is alphanumeric), and then the user will be expected to enter new contents for NET from the workstation. Since this option involves an entry of data by the user as well as output of data by the program, a conversational device such as the workstation must be used.

Line 050 in Figure 5.5 specifies that after a control break of level L2 and when indicator 15 is on and 04 is not on, the contents of the fields SALES and NET will be printed out on the workstation (device 0), the field NET (assumed to be numeric) will be set to zeros in core and the user will be expected to enter a new value for NET from the workstation.

Data entered from the workstation into an alphanumeric field will be assumed to be left-justified. For example, if the field NAME is alphanumeric and 20 characters long and the user enters 'JOHN SMITH' from the terminal, the left-hand ten characters of NAME will be filled and the right-hand ten characters will remain blank.

If the result field of a DSPLY operation specifies a numeric field, the following points should be noted:

1. Numeric data need not be entered with leading zeros; the data will be automatically right-justified and padded on the left with the necessary number of zeros.

2. If the data to be entered is negative, the first character entered must be a hyphen (-).

3. The number of characters permitted to be entered is the (field length + 1) to allow for the sign. It is the user's responsibility not to enter more data than is permitted by the field length specification.

4. If too much data is entered, truncation will occur on the left.

5. Characters other than 0-9 and the hyphen (minus sign) will be ignored.

Examples:

Assuming that a six-digit Result Field has been specified, we have:

Entry	Internal Result
1	000001
-1	00000Q
000123	000123
bbbbbb	000000
+56	000056
(nothing entered)	000000
-1ABQ2	00001R
1234567 (seven digits)	234567 (truncated on left)

EXCPT (10K Compiler Feature Only)

The Exception (EXCPT) operation is used to print out lines while calculations are being done. The programmer simply enters EXCPT in columns 28-32 (Operation) of the Calculation Specifications Form. At that point in the calculations, each output record that contains an 'E' code in column 15 of the Output Format Specifications Form will be output to the selected output medium.

Figure 5.6 shows Calculation and Output Format Specifications that will cause a card to be punched out during calculation time whenever an account is found to be overdue (overdue amount greater than zero). The output file specified, PUNCH, is assumed to have been previously specified to use a card punch as its output device. The card punched out will contain a name, account number, the overdue amount, and the new balance.

Figure 5.6 USE OF EXCPT OPERATION, Part 1

RPG CALCULATION SPECIFICATIONS

10K Compiler Feature

Date ______
Program ______
Programmer ______

Punching Instruction	
Graphic	
Punch	

Page (1 2) ______

Program Identification (75 76 77 78 79 80) ______

Line (3-5)	Form Type (6)	Control Level (L0-L9, LR) (7-8)	Indicators (9-17)	Factor 1 (18-27)	Operation (28-32)	Factor 2 (33-42)	Result Field (43-48)	Field Length (49-51)	Decimal Positions (52)	Half Adjust (H) (53)	Plus / High F1>F2 (54-55)	Minus / Low F1<F2 (56-57)	Zero / Equal F2=F1 (58-59)	Comments (60-74)
01	C				Z-ADD	ØVRDUE	ØVRDUE				06	07		PYMT ØVERDUE ?
02	C		06	ØVRDUE	ADD	CURRNT	NEWBAL							
03	C		06	ØVRDUE	MULT	LTRATE	SURCHG							
04	C		06	NEWBAL	ADD	SURCHG	NEWBAL							
05	C		06		EXCPT									
06	C		07		Z-SUB	ØVRDUE	CREDIT							
07	C		07	CURRNT	SUB	CREDIT	NEWBAL							
08	C													
09	C													
10	C													
11	C													
12	C													
13	C													
14	C													
15	C													
	C													
	C													
	C													
	C													
	C													

FORM 40-343 (OBSOLETES FORM 40-252)

SINGER BUSINESS MACHINES

Figure 5.6 USE OF EXCPT OPERATION, Part 2

RPG OUTPUT - FORMAT SPECIFICATION

10K Compiler Feature

Date ____ ____
Program ______________
Programmer ______________

Punching Instruction	
Graphic	
Punch	

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

Line (3-5)	Form Type (6)	Filename (7-12)	Type (H/D/T/E) (15)	Space Before (17)	Space After (18)	Skip Before (19-20)	Skip After (21-22)	Output Indicators (23-25)	And (26-28)	And (29-31)	Field Name (32-37)	Edit Codes (38)	Blank After (B) (39)	End Position in Output Record (40-43)	Hollerith Indicator (44)	Constant or Edit Word (45-70)
01	O	PUNCH	E		2			05	06							
02	O										NAME			20		
03	O										ACCTNØ			40		'00-000-00'
04	O										ØVRDUE			60		' , 0.00-'
05	O										NEWBAL			80		' , 0.00-'
06	O															
07	O															
08	O															
09	O															
10	O															
11	O															
12	O															
13	O															
14	O															
15	O															
	O															
	O															
	O															
	O															
	O															

Ȥ = Zero Suppress; ƀ = Use Edit Word or No Editing

FORM 40-341 (OBSOLETES FORM 40-251)

SINGER BUSINESS MACHINES

CALCULATION SPECIFICATIONS FORM

Table 5-1 USE OF FIELDS ON CALCULATION SPECIFICATIONS FORM

OPERATION Cols.28-32	CONTROL LEVEL Cols.7-8	INDICATORS Cols.9-17	FACTOR 1 Cols.18-27	FACTOR 2 Cols.33-42	RESULT FIELD Cols.43-48	FIELD LENGTH Cols.49-51	DECIMAL POSITIONS COL.52	HALF ADJUST Col.53	RESULTING INDICATORS Cols.54-59
ADD	0	0	R	R	R	0	0	0	0
Z-ADD	0	0		R	R	0	0	0	0
SUB	0	0	R	R	R	0	0	0	0
Z-SUB	0	0		R	R	0	0	0	0
MULT	0	0	R	R	R	0	0	0	0
DIV	0	0	R	R	R	0	0	0	0
MVR	0	0			R	0	0	0	0
MOVE	0	0		R	R	0	0		
MOVEL	0	0		R	R	0	0		
COMP	0	0	R	R					R
SETON	0	0							R
SETOF	0	0							R
GOTO	0	0		R					
TAG	0		R						
RLABL	0				R	0	0		
EXIT	0	0		R					
DSPLY*	0	0	0	R	0				
EXCPT*	0	0							

O-OPTIONAL R-REQUIRED -BLANK

* 10K Compiler Feature Only

SUMMARY OF OPERATIONS

A summary of the operations which can be coded on the Calculation Specifications Form is presented in Table 5-1.

DECIMAL ALIGNMENT IN ARITHMETIC OPERATIONS

In ADD, Z-ADD, SUB, Z-SUB, and COMP (with numeric fields), the lengths of Factor 1 and Factor 2 must not exceed 18 digits after any adjustment to align the implied decimal points. Zeros are assumed to be added to the right of the field with the fewer number of decimal places, and these assumed positions must be included when considering the 18-digit maximum length. For example, if Factor 1 is a 5-digit field with three decimal places and Factor 2 is a 9-digit field with one decimal place, Factor 2 would be considered to have 11 positions (9 + 2 for decimal alignment).

In MULT, the maximum length for Factor 1 and Factor 2 is 10 digits, and decimals do not have to be aligned. The maximum length for the Result Field is 18 digits.

In DIV, the maximum length for Factor 2 is 10 digits regardless of the number of decimal places. The length of Factor 1 and the Result Field must not exceed 18 digits each. There is an additional requirement that the following formula be satisfied.

$$L(F1)-D(F1)+D(F2)+D(R) < 20$$

where:

L(F1) = Length of Factor 1
D(F1) = Number of decimal places in Factor 1
D(F2) = Number of decimal places in Factor 2
D(R) = Number of decimal places in the Result Field.

In other words, the length of Factor 1 minus the number of decimal places in Factor 1 plus the number of decimal places in Factor 2 plus the number of decimal places in the Result Field must not be greater than 20.

RESULT FIELD (Columns 43-48)

The field named in Result Field will be used in different ways depending on the operation specified in columns 28-32. The column labeled "Result Field" in Table 5-1 indicates when an entry is required in Result Field and when it is to be left blank.

The Result Field name may be from 1-6 alphanumeric characters. The first character must be alphabetic, and special characters and embedded blanks are not allowed. The field named here may have been defined previously in the Input Specifications or the Calculation Specifications. Or, it may be defined at this point in the RPG program. If the field is defined at this point, there must be entries in columns 49-51 (field length) and, in the case of numeric fields, in column 52 (decimal positions) also.

A field must be numeric if it is to be used in arithmetic operations or numeric compare operations or is to be edited or zero-suppressed by the Output Format Specifications.

In the case of division, the user must follow System Ten rules. The user is responsible for verifying the validity of his results.

If a field name entered in Result Field has been specified earlier, then columns 49-52 must be left blank.

FIELD LENGTH (Columns 49-51)

Allowed Values	Meaning
1-100	Allowed length for an alphanumeric field named in Result Field (columns 43-48).
1-18	Allowed length for a numeric field named in Result Field (columns 43-48).
Blank	The result field is not used in this operation, or, if used, its length has been specified already.

The programmer should make certain that he allows enough space in the Result Field to store the result of the specified operation. For example, if two fields, AMT1 and AMT2, each have a field length of 4 numeric characters, their product would require a field length of 8 numeric characters.

DECIMAL POSITIONS (Column 52)

Allowed Values	Meaning
Blank	The field whose length is specified in columns 49-51 is alphanumeric. If field length is also blank, then the length of the field named in Result Field has been previously specified.
0-9	The number of decimal places in the field, which is assumed to consist of numeric characters.

The number of decimal positions specified must not be greater than the field length. If an arithmetic operation results in a number with fewer decimal places than the number specified, the remaining decimal places are filled with zeros. If an arithmetic operation results in more decimal places than have been specified for the Result Field, the extra decimal places on the right are dropped.

Duplicate Specification of Fields

9K Compiler

With the 9K compiler, if a field has been specified earlier in the Calculation Specifications or in the Input Specifications, the field specifications may not be written again. That is, if a field named in Result Field has had its field length and decimal positions specified earlier in the source program, then columns 49-52 must be blank.

10K Compiler

The 10K compiler permits the user to enter the field length and decimal positions of a field named in Result Field even if it has been specified previously in the Calculation Specifications or the Input Specifications. (With the 9K compiler, the length and number of decimal positions may be specified only the first time the field is named, whether in the Calculation Specifications or the Input Specifications.)

This applies to both numeric and alphanumeric fields. Number of decimal positions, of course, is only specified for numeric fields. If a field is specified again, the field length and number of decimal positions must agree with the previous definition for that field. Thus it is not possible to change the length of a field by this feature. The feature is intended for the convenience of the user who names a field many times and who may or may not wish to enter the length and decimal positions of the field each time he writes it in the specifications.

HALF ADJUST (Column 53)

"Half Adjust" is the term used to describe the process of rounding off numbers resulting from arithmetic operations when the Result Field has fewer decimal positions than Factor 1 or Factor 2. When Half Adjust is specified, a 5 is added to the digit following the last digit specified for the result field. (If the number is negative, -5 is added.) Then, all digits following the last specified digit are dropped.

Allowed Values	Meaning
H	Half Adjust. (Rounding is done.)
Blank	No Half Adjust. (Rounding is not done.)

Example

Suppose, the number resulting from an arithmetic operation is 386.174 and the result field is specified as having five numeric characters with two decimal places. With Half Adjust specified, a 5 would be added to the third decimal place giving 386.179 and then the last decimal place is dropped giving 386.17 as the contents of the Result Field.

If the result of an arithmetic operation were 1209.786 to be half-adjusted to a Result Field of two decimal places, a 5 would be added in the third decimal place, giving 1209.791 and then the third decimal place would be dropped, giving 1209.79.

RESULTING INDICATORS (Columns 54-59)

Columns 54-59 are used to specify what indicators will be turned on as a result of arithmetic or compare operations. An entry is also required for the SETON or SETOF operations to specify what indicator(s) will be turned on or off. The use of Resulting Indicators is optional in the case of arithmetic operations and required in the case of COMP, SETON or SETOF.

Allowed Values	Meaning
01-99	Any two numeric characters in this range are assigned to indicate under what conditions subsequent calculations or output operations are to be done.
H1-H3	Halt Indicators. Usually turned on when an error condition is detected. These indicators will cause program termination if not turned off during the calculation cycle. They must be SETON to cause program termination, since RPG does not turn on any Halt Indicators automatically.
OV,OA-OG	Overflow Indicators. Usually turned on when the maximum number of output lines per page has been reached. Can be used to control output operations, such as skip to top of new page.
L1-L9	Control Level Indicators.
LR	Last Record Indicator. (LR may not be used with the SETOF operation).

When an arithmetic operation is being done (i.e, ADD, SUB, MULT, DIV, Z-ADD, Z-SUB, or MVR) and the contents of the result field is positive, then the indicator specified in columns 54-55 is turned on. If an arithmetic operation results in a negative number, then the indicator specified in columns 56-57 is turned on. If the contents of the Result Field is zero after an arithmetic operation is performed, the indicator specified in columns 58-59 is turned on.

Similarly in the case of a Compare (COMP) operation, if the number specified by Factor 1 is greater than the number specified by Factor 2 (F1>F2), the indicator in columns 54-55 is turned on. If F1<F2, the indicator given in columns 56-57 is turned on. If F1=F2, the indicator specified in columns 58-59 is turned on.

RPG CALCULATION SPECIFICATIONS

10K Compiler Feature

Date ______
Program ______
Programmer ______

Punching Instruction								
	Graphic							
	Punch							

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

Line (3-5)	Form Type (6)	Control Level (L0-L9, LR) (7-8)	Indicators (9-17)	Factor 1 (18-27)	Operation (28-32)	Factor 2 (33-42)	Result Field (43-48)	Field Length (49-51)	Decimal Positions (52)	Half Adjust (H) (53)	Resulting Indicators: Plus / High F1>F2 (54-55)	Minus / Low F1<F2 (56-57)	Zero / Equal F2=F1 (58-59)	Comments (60-74)
010	C				SETØF						47	48	20	
020	C				SETØF						21	30	31	
030	C				MØVE	1	ZERØ	1	0					
040	C		01		Z-ADD	AFLD1	ATØT1	3	1					
050	C		01	ATØT1	ADD	AFLD2	ATØT2	3	1					
060	C		01		Z-SUB	AFLD3	ATØT3	15	9					
070	C		01	ATØT3	SUB	AFLD4	ATØT4	15	9					
080	C		01		MØVE	AFLD5	ATØT5	15	6					
090	C		01	ATØT5	SUB	ATØT2	ATØT6	15	6					
100	C		01		MØVEL	ATØT6	ATØT7	17	6					
110	C		01	AFLD6	SUB	ATØT7	ATØT8	18	7					
120	C		01	ATØTAL	ADD	1	ATØTAL	4	0					
130	C		01		GØTØ	END								
140	C		02	BFLD1	MULT	BFLD2	BTØT1	15	3					
150	C			END	TAG									
	C													
	C													
	C													
	C													
	C													

FORM 40-343 (OBSOLETES FORM 40-252)

SINGER BUSINESS MACHINES

Figure 5.7 SAMPLE RPG CALCULATION SPECIFICATIONS

For SETON or SETOF operations, there is no distinction among columns 54-55, 56-57, and 58-59. The indicator or indicators to be turned on a SETON (or off by a SETOF) are simply written starting in columns 54-55. If more than one indicator is to be set on or set off, the entries are made from left to right: the first in columns 54-55, the second in columns 56-57, the third (if present) in columns 58-59. The SETOF operation may not be used to turn off indicators LR or LO.

COMMENTS (Columns 60-74)

The programmer may and should include comments in columns 60-74. Any information he desires may be entered here to help him or other persons understand the program. Comments in columns 60-74 have no effect on the RPG compiler, but are simply printed in source listing.

The programmer can make additional comments by placing an asterisk (*) in column 7 of a specification line. In that case the entire line is treated as a comment, and is printed in the source listing but is not compiled.

Example

A Calculation Specifications Form is shown in Figure 5.7.

Section 6

OUTPUT FORMAT SPECIFICATIONS FORM

RPG OUTPUT - FORMAT SPECIFICATION

10K Compiler Feature

Date ____________

Program ______________________

Programmer ___________________

Punching Instruction	Graphic							
	Punch							

Page 1 2 ☐☐

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

Line (3 4 5)	Form Type (6)	Filename (7 8 9 10 11 12)	NOT USED (13)	AND / OR (14)	Type (H/D/T/ E) (15)	(16)	Space Before (17)	Space After (18)	Skip Before (19 20)	Skip After (21 22)	Output Indicators Not (23)	(24 25)	And Not (26)	(27 28)	And Not (29)	(30 31)	Field Name (32 33 34 35 36 37)	Edit Codes (38) Z = Zero Suppress, b = Use Edit Word or No Editing	Blank After (B) (39)	End Position in Output Record (40 41 42 43)	Hollerith Indicator (44)	Constant or Edit Word (45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70)
0 1	O																					
0 2	O																					
0 3	O																					
0 4	O																					
0 5	O																					
0 6	O																					
0 7	O																					
0 8	O																					
0 9	O																					
1 0	O																					
1 1	O																					
1 2	O																					
1 3	O																					
1 4	O																					
1 5	O																					
	O																					
	O																					
	O																					
	O																					
	O																					

FORM 40-341 (OBSOLETES FORM 40-251)

SINGER BUSINESS MACHINES

Figure 6.1 RPG OUTPUT FORMAT SPECIFICATIONS FORM

OUTPUT FORMAT SPECIFICATIONS FORM

The layout of the generated output report or file is determined by the Output Format Specifications Form (Fig. 6.1).

The form has two main portions:

- Columns 7-31 name the output file and identify the type of record that is to be processed to produce lines of output. They also specify (by means of indicators) the conditions that must exist for certain output lines to be produced.

- Columns 32-70 are used to specify the exact fields to be printed on the report and their location on the printed output line. Editing of numeric fields for output (insertion of decimal points, dollar signs, etc.) can also be specified here.

FORM TYPE (Column 6)

Must contain an O.

FILE NAME (Columns 7-12)

The name of the output file is entered here. The output file name entered here must be identical to that given on the File Description Specifications Form. The name may be one to six alphanumeric characters long, must start with a letter, and may not include special characters or imbedded blanks. It must start in column 7.

The file name appears only on the first line of the Output Format Specifications for a given file. That is, if many lines are used to specify the format of an output file, the file name need not be repeated.

TYPE (Column 15)

Allowed Values	Meaning
H	Heading Record.
D	Detail Record.
T	Total Record.
E	Exception Record (With 10K Compiler Only).

The entry in column 15 identifies the type of record that is being processed for the output report.

A Heading Record (H) is a record usually containing literal data that is used as a heading or title for a page or a table. The printing of column headings for a table of numerical data would be accomplished with a Heading Record containing the column headings as data literals.

A Detail Record (D) usually contains the detailed information that is generated by the input data and the calculation specifications (exclusive of total calculations). A Detail Record could be simply the printing of data from an input card on the output report.

A Total Record (T) would be a record of data resulting from calculations performed by the RPG program, particularly totals accumulated. The printing of Total Records could be governed by various indicators or Control Level Indicators. It is quite common to cause total records to be printed when a control break occurs and after the last record has been read (LR Indicator is turned on).

10K Compiler Feature Only

An Exception Record (E) is a record that is transmitted to the output device during calculation time. Whenever an EXCPT operation is performed, all records that are designated with an E in column 15 of the Output Format Specifications are placed onto the output device specified for the file, in the order specified.

AND, OR (Columns 14-16)

AND (in columns 14-16) and OR (in columns 14-15) are used in conjunction with the Output Indicators (in columns 23-31 of the Output Format Specifications Form). The Output Indicators state what conditions must exist (or must not exist) before the output record specified by this line is produced. A maximum of three indicators can be specified in columns 23-31. If the programmer wishes to specify more than three indicators having an AND relationship, he must enter AND in columns 14-16 on the second line. When AND is specified in columns 14-16, columns 17-22 must be blank.

If an OR relationship is to be specified between two Output Indicators, the first indicator is entered on one line and the second indicator is entered on the next line with OR in columns 14-15. When OR is specified, columns 16-22 must be blank.

OR allows more than one AND clause to qualify the record specifications as an output record. Thus, complex logical tests on alternate groups of conditions are possible. The only restriction in the use of AND clauses connected by ORs is that any AND clause that references the Overflow Indicator must occur prior to any AND clause that does not reference the Overflow Indicator. The Overflow Indicator may appear in any position within the AND clause.

SPACE AND SKIP (Columns 17-18 and 19-22)

The entries in columns 17-22 control spacing between printer lines and skipping to various line numbers on the page being printed or the following page. If all these columns are left blank, the printer will single-space after each line is printed. It is possible to specify a wide variety of printer spacing and skipping between lines of output. The spacing and skipping may be done before or after printing the output line. If skipping and spacing are both specified with a printed line, the actions are done according to the following sequence:

- Skip before printing
- Space before printing
- Skip after printing
- Space after printing

The Overflow Indicator (which was designated in the File Description Specifications) is turned on whenever the printer prints on the last line of the page or spaces beyond it.

However, when the printer skips past the last line on a page to a line on the following page, the Overflow Indicator is not turned on. If the programmer wishes to turn on the Overflow Indicator in this case, he may use a SETON operation.

Spacing and skipping <u>after</u> printing may save time since the output file does not have to wait for the paper to advance before it prints a line.

> <u>Note</u>: Only one space/skip pattern may be specified per output line (whether Heading, Detail or Total).

SPACE (Columns 17-18)

Allowed Values	Meaning
1-9	Number of spaces paper is to be advanced before printing the current line (if entry is in column 17).
	Number of spaces paper is to be advanced after printing the current line (if entry is in column 18).
Blank or Zero (0)	There is no way to suppress line spacing; a blank or zero (0) entry is interpreted as a one (1) due to the hardware control of spacing, i.e., the line printer automatically spaces one line after printing each line.

If the entry is placed in column 17, the spacing will be done before the line is printed. If the entry is placed in column 18, the spacing will be done after the line is printed.

SKIP (Columns 19-22)

Skipping permits paper movement from one line to another without stopping at the intermediate lines. The programmer enters the line number of the next line.

Allowed Values	Meaning
01-99	Number of the line on the printer form. Line 01 is the first line on the form where printing is done (the top of the page).
00, Blank	No skipping will be done.

When skipping is specified to a line number less than the current line number, the paper is advanced to the next page. For example, if the line being printed is on line 10 of the printer form and the user specifies a skip to line 05, the next printing will be done on line 05 of the following page.

If the skip is to be done before the line is printed, the entry is placed in columns 19-20. If the skip is to be done after the line is printed, the entry is placed in columns 21-22.

The skip entry can be greater than the Printer Line Count (columns 27-28 on the Control Card Specifications Form), but if a skip goes beyond an end-of-page punch in the printer carriage control tape, results may not be as expected. It is recommended that the skip entry not exceed the Printer Line Count. If the printer carriage control tape has an end-of-page punch at the overflow line, there will be an automatic skip from the overflow line to the top-of-page punch in the carriage control tape.

Table 6-1 SYSTEM TEN RPG VALID INDICATOR USAGE

Indicator	File Description Specifications Overflow Indicator (Cols. 33-34)	Input Specifications Record Indentifying Indicators (Cols.19-20)	Input Specifications Control Level Indicators (Cols.59-60)	Calculation Specifications Control Level Indicators (Cols.7-8)	Calculation Specifications Indicators (Cols.9-17)	Calculation Specifications Resulting Indicators (Cols.54-59)	Output Format Specifications Output Indicators (Cols.23-31)
01-99		X			X	X	X
L0				X			X
L1-L9		X	X	X	X	X	X
MR					X		X
1P							X*
OV,OA-OG	X				X	X	X
H1-H3		X			X	X	X
LR		X		X	X	X**	X
RS***					X	X	X

*1P is effective only with Detail and Heading Lines.

**LR may not be the result of a SETOF operation.

***10K Compiler Feature Only.

OUTPUT INDICATORS (Columns 23-31)

The indicators which must be turned on before the specified output operation is done are entered in columns 23-31. If several indicators must be on before the output operation is done, the required indicators are listed in columns 24-25, 27-28, and 30-31. If more than three indicators are required, one or more additional lines must be used with AND in columns 14-16. If any one of several possible indicators will allow the output operation to be done, the alternative indicators are written on different lines in columns 24-25 with an OR in columns 14-15 of every alternate line.

The programmer may specify that an output operation be done when a particular indicator is not turned on by specifying an N before the indicator (in columns 23, 26, or 29).

The indicator may be used to govern the output of an entire record or may be used to control the output of a single field.

Allowed Values	Meaning
01-99	Any indicator specified previously, as a Record Identifying Indicator, or as a resulting indicator from the Calculation Specifications Form.
L0,L1-L9	Control Level Indicator.
OA-OG,OV	Overflow Indicators (must be previously assigned).
H1-H3	Halt Indicators previously assigned.
LR	Last Record indicator.
1P	First Page Indicator.
MR	Matching Record Indicator.
RS (10K Compiler Only)	Service Request Indicator. This indicator is turned when a service request is received.

An Overflow Indicator is turned on when the last print line of the page has been reached (assuming the Overlow Indicator was previously assigned). If the Overflow Indicator was not assigned on the File Description Specifications Form, it may not be used in the Output Format Specifications. The same Overflow Indicator must be used on both specifications forms.

RPG OUTPUT - FORMAT SPECIFICATION

10K Compiler Feature

Date ____________
Program ____________
Programmer ____________

Punching Instruction	Graphic							
	Punch							

Page (1 2) ____

Program Identification (75 76 77 78 79 80) ____

Line (3-5)	Form Type (6)	Filename (7-12)	NOT USED (13)	AND / OR (14-16)	Type (H/D/T/E) (15)	Space Before (17)	Space After (18)	Skip Before (19-20)	Skip After (21-22)	Output Indicators Not (23)	(24-25)	And Not (26)	(27-28)	And Not (29)	(30-31)	Field Name (32-37)	Edit Codes (38)	Blank After (B) (39)	End Position in Output Record (40-43)	Hollerith Indicator (44)	Constant or Edit Word (45-70)
010	O	ØUT			H		3	01			ØF										
020	O			Ø	R						1P										
030	O																		50		'HEADING'
040	O				D		2				Ø5										
050	O															SALES			20		' 0.00-'
060	O															RATE			35		'0.00-'
070	O															GRØSS			50		' , 0.00-'
080	O															NET			65		' , 0.00-'
09	O																				
10	O																				
11	O																				
12	O																				
13	O																				
14	O																				
15	O																				
	O																				
	O																				
	O																				
	O																				
	O																				

Ƶ = Zero Suppress
ƀ = Use Edit Word or No Editing

FORM 40-341 (OBSOLETES FORM 40-251)

SINGER BUSINESS MACHINES

Figure 6.2 SAMPLE RPG OUTPUT FORMAT SPECIFICATIONS

An Overflow Indicator may not appear on either AND or OR lines. When it is used in an AND relationship with a Record Identifying Indicator, the results might not be as expected. The reason is that the record type might be the one read when Overflow occurs, and lines conditioned by both overflow and Record Type Indicators may not all be printed.

The First Page Indicator (1P) is usually used to control the printing of literal information, especially headings and titles. It is used in connection with Header or Detail output lines (see column 15). It may not be used with the printing of Total output lines. It is not permissible to use the 1P Indicator in AND or OR relationships with Control Level Indicators.

If the 1P indicator is used in an OR relationship with an Overflow Indicator, the Overflow Indicator must appear <u>first</u> in the Output Format Specifications. (See lines 010 and 020 in the sample form in Fig. 6.2)

10K Compiler Feature Only

The RS indicator is usually turned on by a service request. It remains on for one program cycle, after which it is automatically turned off.

FIELD NAME (Columns 32-37)

The names of fields which will be printed in the output are entered in columns 32-37. The field name must have been previously defined in the Input Specifications or the Calculation Specifications.

Allowed Values	Meaning
1-6 alphanumeric characters	Field name. First character must be alphabetic. Special characters and imbedded blanks are not permitted.
PAGE	Causes automatic numbering of output pages.
UDATE	Causes the date to be printed in the format MM/DD/YY, where MM is the number of the month, and YY is the last two digits of the year. (See Appendix D on Common Core Conventions and description of UDATE in Section 8.)
*PLACE (With 10K Compiler Only)	This 10K compiler option allows the repeated printing of a field or fields in the same output record.

Each field name must appear on a separate specification line. Also, a field name may not appear on the same line as the output file name. Thus, whenever a Field Name appears in columns 32-37 of a line, columns 7-22 must be blank.

Field names may appear on the form in any order. The actual placement of the fields on the output form is controlled by the End Position specified in columns 40-43. If two specified fields overlap, the field specified last on the Output Format Specifications takes precedence.

If a literal constant is to be printed, Field Name must be left blank. The literal to be printed is entered in columns 45-70 (see description of those columns further on in this section).

For a numeric field, a minus sign is stored in the rightmost position and prints as an alphabetic character unless editing is specified

PAGE, which causes automatic page numbering, is assumed to be a four-character numeric field, unless it has been previously defined with a different field length. The page numbering will begin with 1 unless another number has been specified in a PAGE input field (refer to Input Specifications Form, columns 53-58).

To reset the page number field to zero at some point in the job, the user may specify a B in Blank After (see column 39 description below). The programmer can also arrange to reset the PAGE field to zero when a particular indicator is turned on.

Example

In Fig. 6.2, we see the specifications for printing a detail output line containing the fields SALES, RATE, GROSS, and NET. The numbers in columns 40-43 specify the position on the output form where the last character of the field will be printed.

10K Compiler Feature Only

*PLACE allows a field or a group of fields to be repeatedly placed across an output record in record locations specified by the End Position in Output Record (refer to columns 40-43). See the paragraph "Use of *PLACE Option" later in this section for more details.

EDIT CODES (Column 38)

Allowed Values	Meaning
Z	Leading zeros will be suppressed, and the sign on negative numbers is dropped.
Blank	The field is not edited or editing is governed by the Edit Word (see columns 45-70).

Editing is used upon the contents of the output field to make them more readable or comprehensible. For example, if the contents of a four-digit numeric field are 0002, the three leading zeros can be suppressed so that only the 2 is printed.

As another example, an output field named AMOUNT contains six characters with two decimal places implied. The field can store a dollar amount as high as $9,999.99. However, the contents of the field would actually be 999999. The dollar sign, comma, and decimal point, if desired in the printed output, must be placed using a literal constant and an edit word (see columns 45-70).

BLANK AFTER (Column 39)

Column 39 can be used to reset the contents of a field to zeros after the field is printed in the output.

Allowed Values	Meaning
B	The contents of the field named in columns 32-37 will be reset to zeros after the output operation is performed.
Blank	The field will not be reset to zeros after the output operation.

Resetting an output field to zeros by means of Blank After is useful when accumulated totals are to be printed out for several control groups. If a particular field is used to accumulate a total, when the total line is printed, a B in column 39 will cause the named field to be reset to zeros. Thus, a new total can be accumulated for a new control group.

END POSITION IN OUTPUT RECORD (Columns 40-43)

Allowed Values	Meaning
1-415	The position in the output record which will contain the last character of the output field.

The number entered must end in column 43. Leading zeros may be omitted.

Example:

If a four-digit field to be printed has a 98 specified in columns 42-43, the output field will be printed in positions 95-98.

Figure 6.3 USE OF *PLACE OPTION

RPG OUTPUT - FORMAT SPECIFICATION

10K Compiler Feature

Date ____ ____
Program ______________
Programmer ______________

Punching Instruction: Graphic / Punch

Page (1 2)

Program Identification (75 76 77 78 79 80)

Line (3-5)	Form Type (6)	Filename (7-12)	NOT USED (13)	AND/OR (14-16) Type (H/D/T/E)	Space Before (17)	Space After (18)	Skip Before (19-20)	Skip After (21-22)	Output Indicators (23-31)	Field Name (32-37)	Edit Codes (38)	Blank After (B) (39)	End Position in Output Record (40-43)	Hollerith Indicator (44)	Constant or Edit Word (45-70)
01	O	ØUTPUT		D		2									
02	O									FLD1			10		
03	O									FLD2			25		
04	O												35		'STØP'
05	O									*PLACE			75		
06	O									*PLACE			115		
07	O														
08	O														
09	O														
10	O														
11	O														
12	O														
13	O														
14	O														
15	O														

Z = Zero Suppress; b = Use Edit Word or No Editing

FORM 40-341 (OBSOLETES FORM 40-251)

SINGER BUSINESS MACHINES

OUTPUT FORMAT SPECIFICATIONS FORM

USE OF *PLACE OPTION (10K Compiler Only)

10K Compiler Feature Only

When a field is to be printed a number of times across the output record, or a group of fields is to be repeated across the output record, the user may enter *PLACE under Field Name and the final position of the field or fields to be repeated in End Position in Output Record. The placement of fields will then be done automatically and the user does not have to write the field names over and over again in the Output Format Specifications. A typical use of *PLACE is shown in Figure 6.3. The fields FLD1, FLD2 and the literal STOP will be repeated ending in position 75 and then again ending in position 115.

HOLLERITH INDICATOR (Column 44)

10K Compiler Feature Only

With the 10K compiler, it is possible to generate punched card output for negative numeric fields in Hollerith code (using an 11-zone punch for negative numbers).

Allowed Values	Meaning
H	The output numeric field will use the Hollerith punched card code.
Blank	ANSI sign convention is used: P through Y in the low-order position corresponding to -0 through -9.

Figure 6.4 shows the Input and Output Specifications for several numeric fields. Some of the fields are in Hollerith code on input (those marked with an H in column 43 of the Input Specifications); some of the fields are marked with H in column 44 of the Output Specifications and thus will be punched out in Hollerith code.

The table below gives some examples of converted fields. H stands for Hollerith code and A stands for ANSI code.

Input Code	Input Field	Numeric Value in System Ten	Output Code	Output Field
A	4786S	-47863	H	4786L
H	37851	+37851	H	37851
H	2814A	+28141	H	28141
H	0833N	-08335	A	0833U
A	0099T	-00994	A	0099T

RPG INPUT SPECIFICATIONS

10K Compiler Feature

Date ____________
Program ____________
Programmer ____________

Punching Instruction	
Graphic	
Punch	

Page (1 2) ____

Program Identification (75 76 77 78 79 80) ____

Line (3-5)	Form Type (6)	Filename (7-12)	NOT USED (13)	AND/OR Sequence (15-16)	Number (1/N) (17)	Option (O) (18)	Record Identifying Indicators (19-20)	Rec. Id. Code 1 Position (21-24)	Not (N) (25)	Type (C/Z/D) (26)	Character (27)	Rec. Id. Code 2 Position (28-31)	Not (N) (32)	Type (C/Z/D) (33)	Character (34)	Rec. Id. Code 3 Position (35-38)	Not (N) (39)	Type (C/Z/D) (40)	Character (41)	NOT USED (42)	Hollerith Indicator (43)	Field Location From (44-47)	Field Location To (48-51)	Decimal Positions (52)	Field Name (53-58)	Control Level (L1-L9) (59-60)	Matching Fields (61-62)	Field-Record Relation (63-64)	Field Indicators Plus (65-66)	Field Indicators Minus (67-68)	Field Indicators Zero or Blank (69-70)
01	I	INFILE		AA			01	51		C	Z																				
02	I																					1	5	0	FLDA						
03	I																				H	11	15	0	FLDB						
04	I																				H	21	25	0	FLDC						
05	I																				H	31	35	0	FLDD						
06	I																					41	45	0	FLDE						
07	I																														
08	I																														
09	I																														
10	I																														
11	I																														
12	I																														
13	I																														
14	I																														
15	I																														
	I																														
	I																														
	I																														
	I																														
	I																														

FORM 40-342 (OBSOLETES FORM 40-253)

SINGER
BUSINESS MACHINES

Figure 6.4 USE OF HOLLERITH INDICATOR, Part 1

Figure 6.4 USE OF HOLLERITH INDICATOR, Part 2

RPG OUTPUT - FORMAT SPECIFICATION

10K Compiler Feature

Date ____________
Program ____________
Programmer ____________

Punching Instruction	Graphic						
	Punch						

Page 1 2

Program Identification 75 76 77 78 79 80

Line	Form Type	Filename	NOT USED	AND / OR	Type (H/D/T/E)	Space Before	Space After	Skip Before	Skip After	Output Indicators Not		And Not		And Not		Field Name	Edit Codes (Z = Zero Suppress, b = Use Edit Word or No Editing)	Blank After (B)	End Position in Output Record	Hollerith Indicator	Constant or Edit Word
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	45–70
01	O	PRINT			D		2				01										
02	O															FLDA			10	H	
03	O															FLDB			20	H	
04	O															FLDC			30	H	
05	O															FLDD			40		
06	O															FLDE			50		
07	O																				
08	O																				
09	O																				
10	O																				
11	O																				
12	O																				
13	O																				
14	O																				
15	O																				
	O																				
	O																				
	O																				
	O																				
	O																				

FORM 40-341 (OBSOLETES FORM 40-251)

SINGER BUSINESS MACHINES

OUTPUT FORMAT SPECIFICATIONS FORM

CONSTANT OR EDIT WORD (Columns 45-70)

Columns 45-70 may contain either a literal constant to be printed or a word used to edit the named output field.

Constants

Literal constants in the Output Format Specifications are most often used for titles, page headings, and column headings. The literal constant is written with apostrophes (single quotes). The first apostrophe must be in column 45.

When a constant is specified in columns 45-70, Field Name must be left blank. The constant may be all numeric, all alphabetic, or mixed. Blanks and special characters are permitted. All characters will be printed exactly as they appear in the constant within quotes, with the exception of an apostrophe. If a constant contains an apostrophe, it must be represented by <u>two</u> adjacent apostrophes. Thus if APPLICANT'S STATUS is to be printed as a column heading, the proper entry in columns 45-70 would be 'APPLICANT''S STATUS'.

The maximum length of a constant entered in columns 45-70 is 24 characters, since two columns are required for the beginning and ending apostrophes. Constants longer than 24 characters may be specified on two separate specification lines.

```
Examples:             'SALESMAN NO.'
                      'TOTAL DEBT'
                      'ACCT #'
                      '25 AND OLDER'
```

Edit Word

One often desires to edit a numeric field, so that the printed output will include decimal points and commas.

For example, the output field TOTSAL may contain 451236 with two decimal places specified. By means of an edit word, this field can be printed as 4,512.36.

The following rules must be observed when using an edit word:

- o A Field Name must be specified in columns 32-37.
- o The edit word is enclosed within apostrophes.
- o The first apostrophe must appear in column 45.
  The edit word must start in column 46.

Editing uses the standard System Ten editing conventions (see System Ten Assembler 1 Reference Manual, description of the Edit instruction).

The edit word consists of filler characters and punctuation marks or @ signs, which are interspersed among the filler characters. A filler character is defined as any valid System Ten character other than the @ sign or a punctuation mark. A punctuation mark is any of the following: comma, period (decimal point), hyphen, or slash.

The number of filler characters must equal the number of characters in the numeric field to be edited plus one character for printing the sign of the number.

Editing works as follows. The leftmost character of the numeric field to be edited is examined to determine whether it is significant. (In a numeric field, the leftmost nonzero character and all characters to the right of that are called "significant" digits.) If the leftmost character is significant (nonzero), it replaces the leftmost filler character. If it is a non-significant digit (leading zero), the filler character remains. (Thus, blank filler characters have the effect of suppressing leading zeros.) The editing proceeds similarly from left to right across the numeric field being edited and across the filler characters.

Punctuation marks are left unchanged and the @ sign is replaced by a blank.

If the contents of the numeric field being edited are positive or zero, the rightmost character of the edit word is changed to a blank. If the number being edited is negative, the rightmost filler character of the edit word is left unchanged. A minus sign is usually placed in the rightmost position of the edit word. Thus, the minus sign will be printed if the number being edited is negative; a blank will replace it if the number is positive or zero.

The editing operation includes scanning the character string to be printed to see if any punctuation marks remain to the left of the first significant digit. Any such punctuation mark is then replaced with the filler character immediately to its left. The leftmost position of the edit word must not contain a punctuation mark. To summarize the use of characters in the edit word:

Allowed Characters	Meaning
@	Will be replaced by a blank.
Punctuation Marks Period (.) Comma (,) Slash (/) Hyphen (-)	These characters will be printed exactly as they appear, interspersed among the filler characters. If a punctuation mark is to the left of the first significant digit, it will be replaced by the character to its left.
Filler Characters Any other valid System Ten character besides @ or punctuation marks given above.	Will be replaced by a character from the numeric field if the character is significant. If the character from the numeric field is not significant, the filler character will be printed.

Examples

Note: In these examples, b means a blank character.

(A) Printing an Amount:

Numeric Field	00187604
Edit Word	'bbb,bbb.00-'
Printed Result	1,876.04

(B) Printing a Social Security Number:

Numeric Field	067235418
Edit Word	'000-00-0000-'
Printed Result	067-23-5418

(C) Check Protection:

Numeric Field	000150000
Edit Word	'*,***,***.00-'
Printed Result	****1,500.00

(D) Suppressing Leading Zeros

Numeric Field	000235
Edit Word	'bbbbbb-'
Printed Result	235

(E) Negative Amount

Numeric Field	0053240V (negative number)
Edit Word	'bbb,bbb.00-'
Printed Results	5,324.06-

The printing of a dollar sign preceding an amount is most easily done by using a literal constant to specify the dollar sign, and then using an edit word to punctuate the numeric field for printing.

Section 7
RPG COMPILER

FUNCTIONAL DESCRIPTION
USAGE CONSIDERATIONS
INSTALLATION CONSIDERATIONS
INSTALLATION PROCEDURE
RPG PARAMETERS
PARAMETER OPTIONS WITH 10K COMPILER
RPG COMPILER OPERATION
INTERPRETING THE COMPILATION OUTPUT
COMPILER MESSAGE SUMMARY AND
ERROR RECOVERY PROCEDURE

RPG COMPILER

FUNCTIONAL DESCRIPTION

The RPG compiler processes standard RPG specification card images and produces an executable load module (program) conforming to the specifications entered as input to the compiler.

9K Compiler

The 9K compiler executes under DMF and requires that the source file, the object file and a work pool be disc resident. The compiler allows the user to specify these files at compilation time.

10K Compiler

The 10K compiler executes under DMF and requires the work pool to be disc resident. Source input is optionally from disc or a card reader and object output goes optionally to a disc file or a card punch.

USAGE CONSIDERATIONS

Input Requirements

Parameter Input

The parameter input to the RPG compiler is optional and may be used to specify the source, object, and work pool at compilation time. A parameter input device is recommended; however, if the equipment configuration does not include an input device (or the parameter input specifies a non-existent IOC device), the default RPG pool/files will be assumed.

Source Input

With the 9K compiler, the source input must be a linked sequential disc file.

With the 10K compiler, the input is optionally from a linked sequential disc file or from the card reader.

Output Requirements

Object File

With the 9K compiler, the object output (text card images) must be a linked sequential disc file. The object program may be placed in either a null file (containing no data) or a non-null file. If a non-null file is specified, the object program will be placed behind the existing data in the file, extending the file.

This facility allows "spooling" successive RPG compilations and permits inclusion of user EXITs (subroutines) called by the RPG object program.

With the 10K compiler, the object file may be specified as a disc file or may be punched out on a card punch.

Compiler Listings

The RPG compiler requires a printer or similar output device on which to print the source/diagnostic listings, object program storage map and any errors which might prevent a successful compilation.

Work Areas

The RPG compiler requires a work pool to contain the internal tables and temporary work files. The work pool need not be initialized as the compiler will destroy the "links" upon first usage of the pool. Also, the work pool must not contain any DMF files. SYSPOL and RPGPOL must not be specified as the work pool.

INSTALLATION CONSIDERATIONS

Residency

The RPG compiler and the files it accesses must reside under DMF.

Default Pool/Files

If the RPG parameter input does not specify either the source or object files or the work pool, the default file or pool name is assumed. If a parameter device is not available, or the system has no parameter input device, all the default names are assumed, as follows:

Input Source File	-	RPGPOL.TEMP
Output Object File	-	RPGPOL.RPGOBJ
Work Pool (9K)	-	_WKAnn
Work Pool (10K)	-	_WKAnn

Where nn is the partition in which the RPG compiler resides.

Recommended Pool Limits

RPGPOL should contain a minimum of 2,000 sectors. (One disc sector contains 100 characters.)

_WKAnn should contain approximately 1,000 sectors for each 1K of object program size. A work pool of 10,000 sectors should be sufficient to compile the largest program capable of executing on the System Ten. (The compiler will attempt execution with a minimum of approximately 110 sectors. However, it is doubtful if a successful compilation could be completed with so small a work pool.)

9K Compiler

SYSPOL must have approximately 1,800 free sectors to contain the 9K compiler.

10K Compiler

SYSPOL must have approximately 2,100 free sectors to contain the 10K compiler.

Additional System Ten Software Requirements

The installation procedure requires the following DMF support utilities:

- CREATE
- FILE
- UDATE

In addition, if the RPG object program is to access disc devices, the following LIOCS support programs must be accessible in SYSPOL.

- OPEN
- CLOSE

10K Compiler Requirement

The 10K compiler requires the use of the module R_OPEN.

Optimizing Compilation Speed

Compilation speed can be optimized by reducing disc access time to a minimum. This can be accomplished by having each file that is accessed and the work pool on different disc drives. In addition, the RPG compiler should be in contiguous sectors to minimize the load time as many phases are loaded repeatedly during compilation. Such complete optimization requires four disc drives, which may not be feasible at an installation.

Examples below illustrate a more or less optimum disc allocation for one, two and three available drives, with the following assumptions:

1. The source and object files reside in RPGPOL,
2. The RPG compiler resides in SYSPOL, and
3. _WKAnn is the work pool.

RPG Disc Allocation Examples:

1. One Disc Drive

SYSPOL, RPGPOL and _WKAnn pools lie on even 20,000 sector boundaries.

- SYSPOL Limits: 000100*- 019999
- RPGPOL Limits: 020000 - 021999
- _WKAnn Limits: 040000 - 049999

*Allows for 100 sectors of DMF system overhead preceding SYSPOL.

2. Two Disc Drives

- SYSPOL resides on disc drive 0.
- RPGPOL resides on disc drive 1.
- _WKAnn resides on disc drive 0.

Thus, the source and object files will reside on drive 1 while the work pool resides on drive 0.

3. Three Disc Drives

- SYSPOL resides on disc drive 0.
- RPGPOL resides on disc drive 1.
- _WKAnn resides on disc drive 2.

The combination of the use of two or three disc drives (Examples 2 and 3) with the placement of RPGPOL and _WKAnn on even 20,000 sector boundaries (Example 1) will result in a further decrease of compilation time.

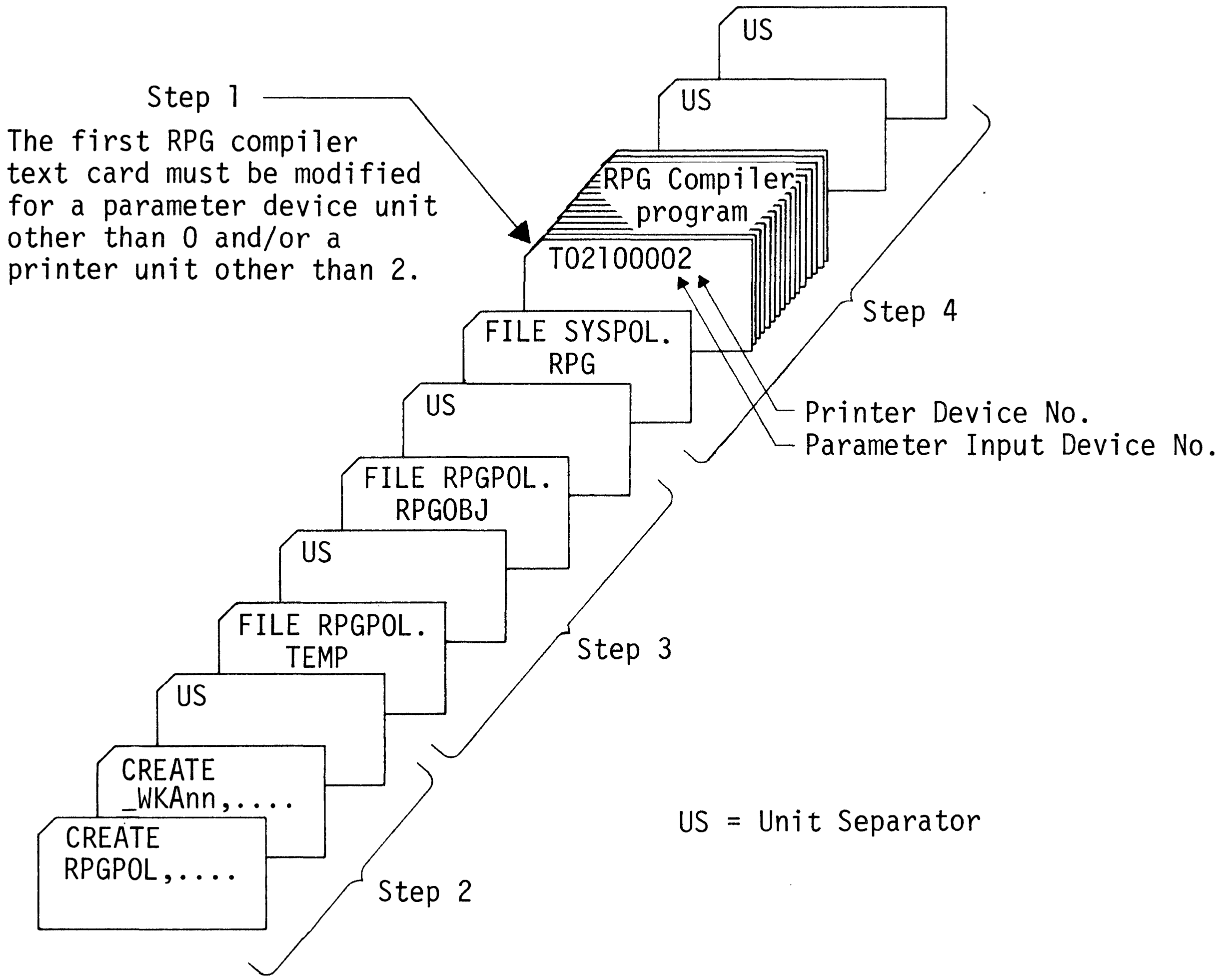


Figure 7.1 DECK SETUP FOR RPG COMPILER INSTALLATION

INSTALLATION PROCEDURE

IOC Device Number Assignments

9K Compiler

The following IOC device numbers are preset by the RPG compiler.

- Parameter Input Device - Device 0.
- Printer Output Device - Device 2.

These assignments may be altered by modifying the first compiler text card (#0001) to reflect the correct device numbers. The text card format is as follows:

T021000rp

where

r(col. 8) is the parameter input device number and

p(col. 9) is the printer device number.

As stated above, this text card is preset to:

T02100002

That is,

Parameter Device, r=0

Printer Device, p=2

Refer to Figure 7.1.

10K Compiler

With the 10K compiler, the first compiler text card (#0001) not only specifies the parameter input device and the printer output device, but provides options for source input, object output and compile-and-go. The text card format is as follows:

T051000rpstg

where

r(col. 8) is the parameter input device number,

p(col. 9) is the printer device number,

s(col. 10) is the source input device type (see Note, below),

t(col. 11) is the object output device type (see Note, below),

g(col. 12) is the compile-and-go option

0 = compile only

1 = compile and go

Note: For parameters 's' and 't', a value of Q through Y (-1 through -9) signifies a disc device; an entry of 0 through 9 signifies the device number of an IOC device. The standard default values for 's' and 't' are:

s, source -- disc (RPGPOL.TEMP)

t, object -- disc (RPGPOL.RPGOBJ)

Thus, these default values are normally set at installation time:

r = 0 (workstation)

p = 2 (line printer)

s = Q (disc, RPGPOL.TEMP)

t = Q (disc,RPGPOL.RPGOBJ)

g = 0 (compile only)

The first compiler text card then reads as follows:

T05100002QQ0

It is possible to override these values by parameter input from the workstation at compilation time.

Installation Steps

Step 1. Alter the first text card to assign the parameter device and printer device if the desired unit numbers differ from the preset default values.

Note: If the user desires to always use the default pool/files for compilation, he may change the parameter device to a nonexistent IOC device number.

Step 2. Create the following RPG default pools utilizing the DMF support utility CREATE:

- RPGPOL
- _WKAnn

Step 3. Create the following RPG default files in RPGPOL utilizing the DMF support utility FILE:

- TEMP
- RPGOBJ

Step 4. File the RPG compiler in SYSPOL utilizing the DMF support utility FILE.

When these steps have been completed successfully, the RPG compiler is ready for use.

Figure 7.1 illustrates this installation procedure.

RPG PARAMETERS

General

The RPG parameters are optional and indicate to the RPG compiler the following:

- The name of the source file.
- The name of the object file.
- The name of the work pool.

The parameters are entered via the parameter device, normally the workstation.

Format

[INPUT=poolname.filename] [,] [OUTPUT=poolname.filename] [,]

[WORK=poolname] [;] [comment]

where:

poolname is the pool name, one to six characters starting with an alphabetic character, and

filename is the file name, one to six characters starting with an alphabetic character.

Preparation Rules

1. All parameters are optional. The default values are

   INPUT=RPGPOL.TEMP

   OUTPUT=RPGPOL.RPGOBJ

   WORK=_WKAnn

2. If the pool name is not specified for INPUT and OUTPUT, RPGPOL is assumed. Example:

   INPUT=TEMP is interpreted to mean INPUT=RPGPOL.TEMP

3. If both pool name and file names are specified, they must be separated by a period (.) and contain no imbedded blanks.

4. The parameters may appear in any order and must be separated by either commas or blanks.

5. The command terminator (;) is needed only if the user wishes to include comments following the last parameter. That is, the command terminator terminates the scan for additional parameters.

6. The parameters may begin in any position of the parameter record.

Parameter Examples

```
INPUT=FILE,OUTPUT=SYSPOL.OBJECT,WORK=WORKPO

INPUT=RPGSRC.TEMPƀOUTPUT=RPGOBJƀWORK=DUMMY

INPUT=SRCFIL,ƀƀWORK=WORKFL;THIS IS A COMMENT

OUTPUT=OBJPOL.RPGOBJƀƀƀ;COMMENT FOLLOWS
```

where

ƀ=blank space.

Default Options

To obtain all default parameters, perform one of the following actions:

- Press Enter Key on the workstation.
- Enter a single semicolon (;) on the workstation or card reader, whichever is used.
- Enter a unit separator card, if the card reader is used.
- Enter a blank card in the card reader or an 80-column line of blanks on the workstation.
- Specify the parameter device to a nonexistent IOC number (at RPG installation time).

PARAMETER OPTIONS WITH 10K COMPILER

The 10K compiler provides the following options in addition to those supported by the 9K compiler:

1. The source input may come directly from a card reader (or equivalent device). The user specifies

   ```
   INPUT=n
   ```

   on the workstation in response to the request

   ```
   A) ENTER RPG PARAMETERS
   ```

   where n is the device number of the source input device. For the card reader, n=1.

2. The object output may be directed to a card punch (or equivalent device). The user specifies

   ```
   OUTPUT=n
   ```

   where n is the device number of the output device. For the card punch, n=4.

3. Also "compile-and-go" is allowed; that is, it is possible to compile and execute an RPG program in one operation. If "compile and go" is chosen as an option, the object output must be placed on disc; compile and go can not be done if the output of compilation is punched out as an object deck.

   The "compile and go" option is specified by typing GO followed by a semicolon on the workstation in response to the request for RPG parameters. That is,

   ```
   A)ENTER RPG PARAMETERS
   GO;
   ```

4. If a default value of "compile-and-go" has been installed at a particular installation (by altering the first text card (#0001) of the 10K RPG compiler), then it is possible to override this at compilation time. The user enters

   ```
   NOGO
   ```

   as his compilation option.

The standard default options for the 10K compiler are

Input Source File -- RPGPOL.TEMP

Output Object File -- PRGPOL.RPGOBJ

Work Pool -- _WKAnn where nn is the partition in which the RPG compiler resides.

Compilation -- Compile only

The following examples show various possibilities for the RPG parameters:

```
A) ENTER RPG PARAMETERS.
OUTPUT=4,WORK=WORKPL;
```

The default value for the source file is used; the object deck is punched out.

```
A) ENTER RPG PARAMETERS.
GO;
```

All default names will be used. This is "compile and go".

```
A) ENTER RPG PARAMETERS.
INPUT=1,GO;
```

Source input is from a card reader; "compile and go" with object file and work pool default values.

```
A) ENTER RPG PARAMETERS.
INPUT=1,OUTPUT=4;
```

Source input is from a card reader; object deck is punched; "compile and go" is not permitted.

```
A) ENTER RPG PARAMETERS.
;
```

A semicolon alone means all 10K default values.

```
A) ENTER RPG PARAMETERS.

INPUT=SRC.RPGSCR,OUTPUT=OBJ.RPGOBJ,
WORK=WRKPOL,GO;
```

This is a "compile and go" with user-assigned source file, object file, and work pool names.

Additional Notes on GO and NOGO Options

1. Options specified at compilation time will override the default options specified when the compiler was installed.

2. The device used for object output must not be the same as that used for listing the source program.

3. If GO and NOGO appear in the same parameter string, the last parameter appearing will be the option used.

Re-entry of Parameters from Workstation

With the 10K compiler, the user may re-enter his RPG parameters if he makes an error. He simply depresses the ERROR key on the workstation and then enters the correct RPG parameters.

RPG COMPILER OPERATION

Pre-Compilation Procedure

Source/Object File Initialization

The RPG compiler only accepts the source input from a linked sequential disc file and likewise places the object program text output into a linked sequential disc file. It is the user's responsibility to initialize these files according to his or the installation's requirements. The following examples illustrate typical pre-compilation procedures. The default pool/files are used in the illustrations as a matter of convenience; however, the illustrations also apply if other object or source file names or work pool names are substituted for the default values.

Example 1

Compilation into a null object file (one containing no data).

1. FILE object file with no data to insure the file is null.

2. FILE the source deck into the source file.

3. Compile RPG source deck.

This is shown in Figure 7.2

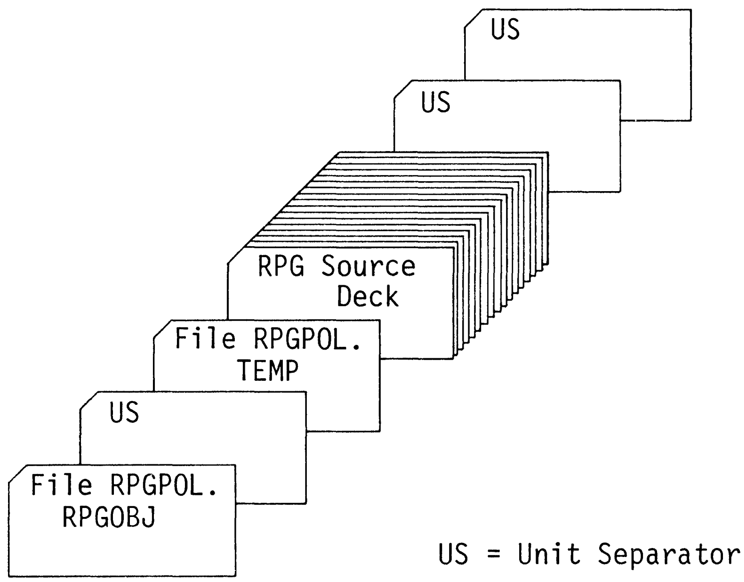


Figure 7.2 COMPILATION INTO A NULL OBJECT FILE

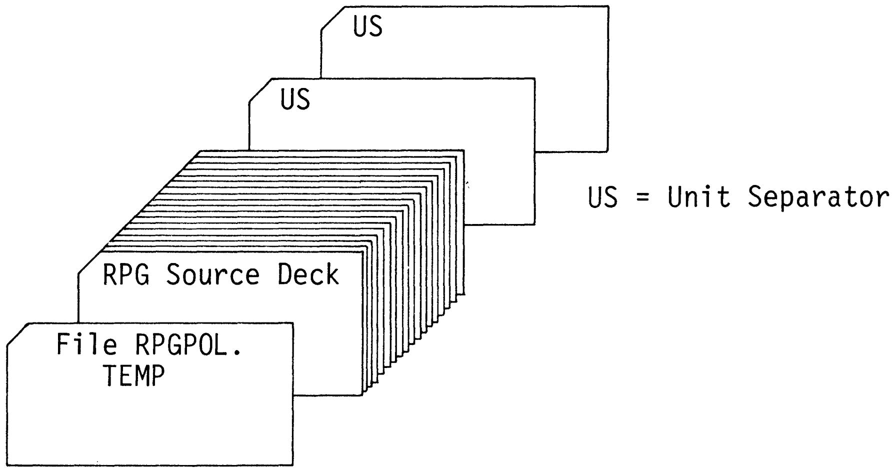


Figure 7.3 COMPILATION INTO A NON-NULL OBJECT FILE

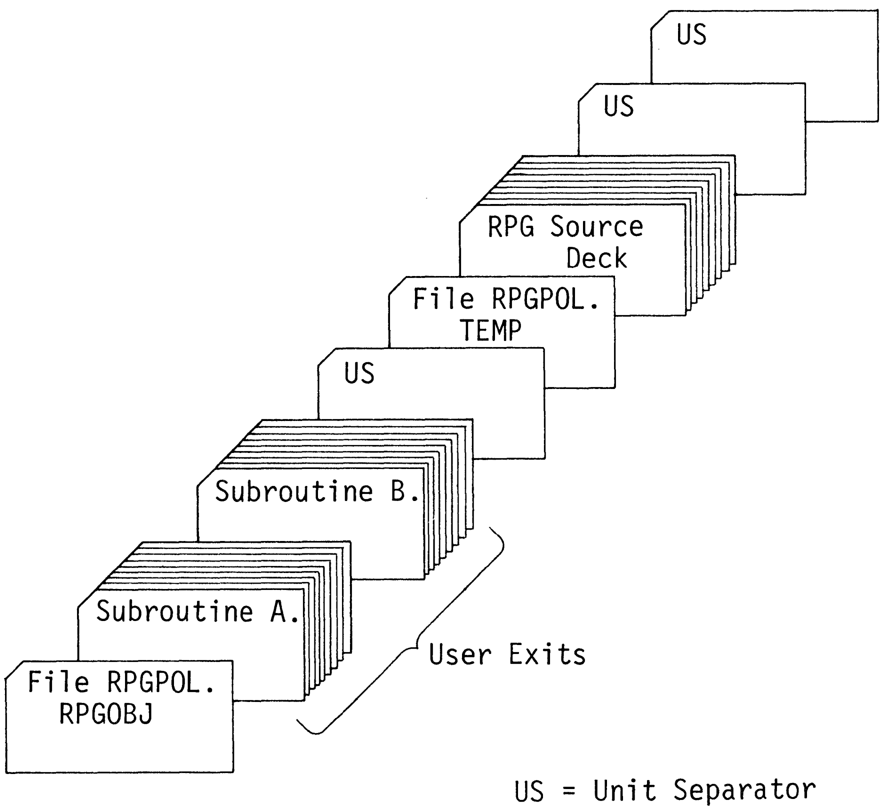


Figure 7.4 COMPILATION WITH USER SUBROUTINES

Example 2

Compilation into a non-null object file (one which already contains data).

1. FILE the source deck into the source file.

2. Ensure that the proper data is in the object file.

3. Compile the RPG source deck.

This is shown in Figure 7.3

Example 3

Compilation with User "EXITs."

If an RPG program is to call an assembly subroutine, the subroutine text cards must be placed in front of the RPG object text cards in order for the object program to execute correctly. This can be accomplished in two ways:

A: 1. Compile the RPG source program.

2. Punch out the RPG program object deck.

3. Place the subroutine object deck in front of the RPG object program.

4. FILE this deck back into the object file.

or

B: 1. FILE the subroutine object text cards into the RPG object file.

2. Compile the RPG source program. (The compiler will place the RPG text behind the subroutine text already in the object file.)

Note: The subroutine text cards should not contain any START (S in column 1) cards. However, if the user desires to delineate his subroutine decks, a START card with an execution address of the DMF loader (S0060) will permit correct loading of the RPG object program.

Figure 7.4 shows method B described above.

Current Date Initialization

The RPG compiler will print the date on the compilation listings if it is found in locations 0306-0313 of common in accord with standard System Ten conventions. The format expected is:

mm/dd/yy

where:

mm is the two-digit month,

dd is the two-digit day, and

yy is the two-digit year.

It is the user's responsibility to insure that the current date is located correctly in common if it is desired on the compiler-produced listings. (See the description of UDATE in Section 8.)

Compilation

The RPG compiler executes under DMF; therefore the workstation is assumed to be the parameter input device for both DMF and RPG. (However, the RPG compiler will accept input from a card reader.)

Operation Procedure

1. Press Enter Key to load DMF.

2. In response to the conversational loader's

   A)ENTER PROGRAM NAME.

   enter:

   RPG

3. The RPG compiler will load and ask for parameters:

   A)ENTER RPG PARAMETERS.

   Enter any RPG Parameters (refer to RPG Parameters within this section), or simply press Enter Key to obtain default parameters.

Notes:

1. No ERROR entry is implemented for parameter errors. If an error is encountered, the compilation will be aborted (indicated by a load condition) and a appropriate message will be displayed on the RPG printer device.

2. If the operator recognizes an error before the parameters are entered (e.g., an erroneous pool name, or an invalid character), entering a slash or asterisk surrounded by spaces will insure that the compilation will not be attempted.

3. If the compilation is aborted due to parameter errors, the Operation Procedure outlined above must be repeated.

4. If the parameter device is assigned to the card reader, the "ENTER PARAMETERS" message will not be displayed. A blank card or unit separator will cause the default parameters to be used.

5. If the RPG parameter device is specified as a nonexistent IOC unit, the default parameters will automatically be assumed.

6. The initialization phase of the 10K RPG compiler verifies that disc devices required for use by the compiler are on line and ready before compilation begins. If a required disc device is not accessible, the compiler issues a request to the operator to ready the device and then waits for a reply to either continue or cancel the compilation. If a disc is placed off line during compilation, the compilation is aborted and the standard disc I/O abort message is issued.

Compilation Termination

The termination of a compilation is indicated by the conversational loader's request for the next program:

A)ENTER PROGRAM NAME.

Abnormal termination is indicated by a load condition. (The RPG compiler prints all error and abort messages on the RPG printer device.)

Refer to the subsection "Compiler Message Summary and Error Recovery Procedure" later in Section 7 and to Appendix C, "RPG Source Code Diagnostics", for a detailed explanation of all error messages.

INTERPRETING THE COMPILATION OUTPUT

The RPG compiler produces two listings as documentation for each compilation:

- The source/diagnostic listing
- The object program map

Source/Diagnostic Listing

The source line consists of:

LINE NO

The compiler generated line number for programmer and documentation reference purposes.

S

An S will be printed if columns 1-5 (page, line) of the RPG source record are out of sequence. Note that if columns 1-5 are blank, the compiler assumes that the record is in sequence.

RPG SOURCE RECORD

The printed image of the RPG source statement.

The diagnostic line follows the source line in which the error occurs and consists of the following:

ERROR NUMBER

A reference to the diagnostic table (see Diagnostic Messages, Appendix C), to further clarify the error.

COLUMN NUMBER

The column number at which the compiler detected the error. The column number is intended to assist the programmer in locating the language element in question and may not point directly to that element but only to an adjacent position in the source record.

Normally, if the error is syntactic in nature, the column number will point to the left-hand end of the RPG field in error. However, if the error is syntatically correct but is conceptionally or contextually in error, the column number will point to one (1) position past the RPG field in error.

ERROR MESSAGE

Describes the error found.

Refer to Figure 7.5 for an illustration of the Source/Diagnostic Listing.

```
V003    *REPORT*PROGRAM*GENFRATOR*      SOURCE/DIAGNOSTIC LISTING      SA0800  09/20/71    PAGE  1

   1     01010H      700                                                                SA0800
   2     01020FCARD     IPE        0080            READER      1                          SA0800
   3     01030FPRINT    O          0132     0X     PRINTER     2                          SA0800

  **** ERROR 68 ****   COLUMN 34   INVALID INDICATOR SPECIFIED

   4     02010ICARD     AA   10                                                           SA0800
   5     02020I                                           1    3 FLD1                   SA0800
   6     03005C                         SETOF                    AC2050                 SA0800

  **** ERROR 68 ****   COLUMN 54   INVALID INDICATOR SPECIFIED
  **** ERROR 68 ****   COLUMN 56   INVALID INDICATOR SPECIFIED

   7     03006C                         SETOF                    516061                 SA0800
   8     03010C     10       FLD1       COMP '    '              202001                 SA0800
   9     03020C*                                                                        SA0800
  10     03030C     10                  EXIT SUBB                                       SA0800
  11     03040C     10                  EXIT SUBA                                       SA0800
  12     03050C                         RLABL          IN01     10                      SA0800
  13     03060C     10 01               SETON                    50                     SA0800
  14     03070C     10N01               SETON                    51                     SA0800
  15     03080C     20 51               SETON                    60                     SA0800
  16     03090C    N20 50               SETON                    61                     SA0800
  17     03100CLR                       EXIT SUBB                                       SA0800
  18     040100PRINT    H   201    0D                                                   SA0800

  **** ERROR 68 ****   COLUMN 24   INVALID INDICATOR SPECIFIED

  19     040200        OR          1P                                                   SA0800
  20     040300                                    1 ' '                                SA0800
  21     040400*                                                                        SA0800
  22     040500         D  1       60                                                   SA0800
  23     040600        OR          61                                                   SA0800
  24     040700                                   24 'THE SUBROUTINE DID NOT W'         SA0800
  25     040800                                   28 'ORK.'                             SA0800
  26     040900                    60             54 'R.I. 01 WASN''T SETON'            SA0800
  27     041000                    61             54 'R.I. 01 WASN''T SETOF'            SA0800
```

Figure 7.5 SOURCE/DIAGNOSTIC LISTING

Object Program Map

The object program map consists of the following:

EXITS

The EXIT name and the address where the RPG object program expects the entry to each named subroutine to be located.

INDICATORS

The address of each indicator specified in the program and the addresses of all RPG resident indicators.

FIELDS

The field name and its address. An asterisk (*) preceding the field name indicates that the field was not referenced by the RPG program.

The addresses of areas where totals are stored for control breaks are indicated by the appearance of the appropriate Control Level Indicators surrounded by parentheses.

ALLOCATION MAP

The addresses of the object program tables, program entry point and field base address for programmer reference and as a debugging aid. (Refer to Appendix F, RPG Debugging Examples.)

In addition, COMPILATION STATISTICS are included with the object map indicating the object program size, the specified size (from the Header "H" card) as well as the names of the source and object files.

Refer to Figure 7.6 for an illustration of an example of the Object Program Map.

```
V003    *REPORT*PROGRAM*GENERATOR*        OBJECT PROGRAM MAP        SA0800  09/20/71   PAGE  2


                                   ****** EXITS ******

    ADDR  EXIT         ADDR  EXIT         ADDR  EXIT         ADDR  EXIT         ADDR  EXIT

    7000  SUBB         7010  SUBA
                                   **** INDICATORS ****

    ADDR  IN           ADDR  IN           ADDR  IN           ADDR  IN           ADDR  IN

    1900  L0           1901  LR           1902  L9           1903  L8           1904  L7
    1905  L6           1906  L5           1907  L4           1908  L3           1909  L2
    1910  L1           1911  1P           1912  H1           1913  H2           1914  H3
    1915  MR           1920  10           1921  20           1922  50           1923  51
    1924  60           1925  61           1926  01
                                   ***** FIELDS *****

    ADDR  FIELD        ADDR  FIELD        ADDR  FIELD        ADDR  FIELD        ADDR  FIELD

    2095  FLD1


   ****** ALLOCATION MAP ******

   PROGRAM ENTRY              0340
   FIELD BASE ADDRESS         1900
   COMMUNICATION AREA         2080


   *** COMPILATION STATISTICS ***

   PROGRAM SIZE         2,680
   SPECIFIED SIZE       7,000
   SOURCE FILE - RPGPOL.TEMP
   OBJECT FILE - RPGPOL.RPGOBJ

    4 DIAGNOSTICS LISTED


RPG COMPILATION COMPLETED
```

Figure 7.6 OBJECT PROGRAM MAP

COMPILER MESSAGE SUMMARY AND ERROR RECOVERY PROCEDURE

General

All compiler produced messages are printed on the RPG printer device. The messages are grouped into four categories:

- Parameter and Initialization Error Messages
- Diagnostic Messages (See Appendix C.)
- Termination and Information Messages
- Abort Messages

Parameter and Initialization Error Messages

Format

```
*****ERROR***** message
```

The parameter record is listed above the error message for reference.

Table 7-1 PARAMETER AND INITIALIZATION ERROR MESSAGES

MESSAGE	DESCRIPTION OF ERROR	ACTION
SYNTAX ERROR	At or near the flagged position there is a syntactic error.	Check for misspellings, invalid pool or file name, or no semi-colon before the comment field.
DUPLICATE KEYWORD ENCOUNTERED	INPUT, OUTPUT or WORK was entered more than once.	Eliminate the redundant keyword.
SYSPOL CANNOT BE A WORK POOL	SYSPOL was named as the work pool.	Change the work pool to another pool.
RPGPOL CANNOT BE A WORK POOL	RPGPOL was named as the work pool.	Change the work pool to another pool.
WORK POOL CONTAINS ACTIVE FILES	The work pool was found to have files listed in its directory.	Delete the files or change the work pool to another pool.
INPUT FILE IS EMPTY	The source file named did not contain any data.	FILE data into the input file or change the INPUT parameter to the correct input file.
POOL OR FILE 'name' NOT FOUND	'Name' identifies the pool or file which could not be found.	Check for a spelling error or missing pool name specification.
PRINTER AND PUNCH BOTH ASSIGNED TO UNIT n (10K Compiler Only)	('n' is the unit number.) The RPG listing device and the object output device cannot be the same unit.	Change device assignments.
A)READY DEVICE Dn	('n' is the disc unit assignment.) This message is displayed on the conversational output device (CONO) when a requested disc unit is found to be off-line during compiler initialization.	Ready disc device 'n'. Respond with any character on the conversational input device (CONI), or, to cancel the job, depress the Control key if the CONI device is a workstation or submit a Unit Separator card if the CONI device is a card reader.
NOTE: The above errors will cause the compilation to be aborted.		

Termination and Information Messages

Format

These messages appear on the "object program map" page of the compilation listing. If compilation was completed sucessfully, the RPG compiler returns control to the conversational loader.

Table 7-2 TERMINATION AND INFORMATION MESSAGES

MESSAGE	DESCRIPTION	ACTION
NO DIAGNOSTICS LISTED	No errors were found in the RPG specification cards. Successful execution of the object program should occur.	None
'nnnn' DIAGNOSTICS LISTED	'nnnn' is a count of the number of errors encountered in the RPG specification cards. Successful execution of the object program depends on the severity of the diagnostic.	Correct the errors listed.
RPG OBJECT PROGRAM EXCEEDS 10K	The program as compiled is too large to fit in the largest permissible System Ten partition. The object program will not execute.	User must reduce the program size by dividing the program or eliminating data or those operations which require large amounts of core (decimal alignment extended arithmetic,etc.)
INSUFFICIENT AVAILABLE SECTORS TO CONTAIN THE RPG OBJECT FILE(Refer to Note 1 below.)	The free sector list did not contain enough sectors to hold the object file. Compilation is immediately terminated. The output file status is as it was prior to compilation; that is, the object program is not available.	Assign the output to a file in a different pool or delete files in the present pool to free additional sectors.
COMPILATION TERMINATED ABNORMALLY (Refer to Note 2 below.)	The compilation process had to be terminated prematurely; however, the compilation was partially successful and the source/diagnostic listing was produced. The object program is not available.	Correct the condition listed above this message.
RPG COMPILATION COMPLETED	The RPG compilation went to end of job normally.	None

Notes:

1. This message can occur at any time after the printing of the source/diagnostic listing.
2. This message is a direct result of the "insufficient sectors" message referenced implicitly by Note 1 above.

Abort Messages

The presence of these messages is indicated by a "load condition."

Format A:

```
RPG COMPILATION ABORTED pp Vnnn message
```

where

pp is the number of the compiler phase in which the abort occurred, and

Vnnn is the compiler version.

Table 7-3 ABORT MESSAGES

MESSAGE	DESCRIPTION of ABORT	ACTION
PARAMETER ERRORS	The RPG parameters are incorrect.	Correct the error listed above the abort message.
INSUFFICIENT AREA IN WORK POOL	Less than 110 sectors were available in the work pool.	Change the work pool to one with more area.
DISC I/O ERROR SECTOR 'ssssss'	The disc drive was not on-line or was placed off-line during the compilation process. A read/write error occurred on a linked sequential file or the compiler was not able to read a record previously written on the work file. 'ssssss' is the sector in question.	Follow the installation's procedure involving disc read/write errors if the error cannot be traced to an off-line disc drive.
DISC I/O ERROR SECTOR 'ssssss'/'ssssss'	Same as above except that the compiler's error routine could not isolate which of the two sectors listed was in error.	Same as above.
INSUFFICIENT CORE, 10K REQUIRED (10K Compiler Only)	The 10K RPG compiler was loaded into a partition with less than 10K of core.	Load the compiler into a 10K partition.
INSUFFICIENT AREA IN WORK POOL (10K Compiler Only)	Less than 109 sectors were available in the work pool or, if the source input was from an IOC device, less than (109 + number of source cards) were available.	Use a larger work pool.
DISC DRIVE n NOT AVAILABLE (10K Compiler Only)	The job was cancelled by the operator because device 'n' was not available, or the conversational output device (CONO) was ignored and operator communication was prevented.	Attempt the compilation again when the required device is available, or assign different pools or files which do not require disc device 'n'.

Format B:

```
RPG COMPILATION ABORTED pp Vnnn aaaa ERROR CODE ee

    where

  pp  is the number of the compiler phase in which
      the error occurred,

Vnnn  is the compiler version,

aaaa  is the address where the abort routine was called,
      and

  ee  is the error code.
```

Table 7-4 COMPILER ERROR CODES WHEN COMPILER ABORTS

Code No. (ee)	Description of Error	Action
05	Work Pool Overflow.	If 'aaaa' ÷ 29 is equal to a number from 4 to 8, 11, 13, or 16, increase the size of the work pool; otherwise, follow ERROR PROCEDURE following this table.
95	Internal Compiler Error, invalid internal Meta-Code encountered.	Follow ERROR PROCEDURE following this table.
96	Internal Compiler Error, unresolved forward branch reference.	Follow ERROR PROCEDURE following this table.
97	Compiler Installation Error, RPG library modules out of sequence or incomplete.	Verify that the RPG compiler deck is in sequence and re-install. If error occurs again, request a new RPG compiler.
98	Compiler Installation Error, RPG object file incomplete.	Reinstall the RPG compiler. If error occurs again, request a new RPG compiler.
Any Other Code	Unspecified Error Code.	Follow ERROR PROCEDURE following this table.

Note: These error codes are different from the source diagnostic error messages listed in Appendix C. The error codes above refer to errors in the compiler, while the errors detailed in Appendix C result from improper source statements.

Error Procedure

1. Attempt the compilation again to insure that the error condition repeats.

2. Dump core, list the input file, output file and the work pool.

   Note: The work pool must be dumped by sector range as the sector links are destroyed by the RPG compiler.

3. Submit all computer listings, dumps, the work station console sheet and a copy of the source deck which caused the abort to the Systems Engineer assigned to the installation.

Unidentified Abort

If a load condition occurs and no abort message is produced, verify that the abort was not caused by the DMF Loader, or that the printer device was not off-line; then follow the Error Procedure outlined above.

Section 8

RPG OBJECT PROGRAM

RPG OBJECT PROGRAM

FUNCTIONAL DESCRIPTION

The RPG object program produced by the compiler is available for immediate execution, provided that any required subroutines have been FILEd into the assigned object file prior to the RPG compilation. (Refer to the subsection titled "Compilation with User EXITs" in Section 7.)

The results obtained from execution of the object program correspond directly to the RPG source input to the compiler, assuming that compilation and execution have been free from errors.

RPG OBJECT PROGRAM OPERATION

Pre-Execution Procedure

UDATE

The current calendar date (common positions 0306-0313) may be printed in the RPG output by specifying UDATE as an output field name (columns 32-37 of the Output Format Specifications Form).

It is assumed that the correct date has been previously entered in these locations by the operator. To enter the date, the operator can use the UDATE utility program. This is done by the operator typing in UDATE in response to the workstation message:

```
A)ENTER PROGRAM NAME.
```

The workstation then types:

```
A)SET DATE.
```

and the operator enters the date in the format

```
mm/dd/yy
```

where mm is the two-digit month
dd is the two-digit day of the month, and
yy are the last two digits of the year.

UDATE then places this date in locations 0306-0313 of common, and the system returns to the DMF conversational loader.

Input/Output Channel (IOC) Device Availability

The RPG object program does not insure that the requisite input/output devices are "on-line" or "present" on the partition's IOC. It is the user's responsibility to insure that his requisite IOC devices are "on-line" and properly assigned to the correct IOC unit number.

Initialization of Disc Files

The input files used by the RPG object program must be Fixed Allocation Read Only files. Data must be present in the input files.

The output files used by the RPG object program must be "Output-Type" files. They must be null (empty of data) initially.

Refer to the Disc Management Facility Reference Manual for further information on these types of files.

Restrictions for Disc File Access

- The output disc file must be a null file.
- The RPG object program assumes a non-contention mode for each output disc file specified. Therefore only one output file should be specified for a particular pool.
- If a subroutine is to handle disc files not used by the RPG object program, the programmer is advised to include a routine which will be executed prior to RPG to open these files. The routine should then be used to load the RPG object program and its EXIT subroutines.

Object Program Execution

The RPG object program executes under the DMF facility. The workstation is assumed to be the DMF parameter device.

Operation Procedure

1. Press Enter Key to load DMF.
2. In response to the conversational loader's message

   A)ENTER PROGRAM NAME.

   Enter the name of the RPG object program's file.

   Example: RPGPOL.RPGOBJ

Object Program Termination

Normal object program termination will be indicated by the conversational loader's request for the next program:

A)ENTER PROGRAM NAME.

Abnormal termination is indicated by a load condition. The reason for an abnormal termination is indicated by the contents of the partition's error register, locations 41-44. (See Table 8-1) It is mandatory that the installation's standard operating procedure for abnormal terminations of RPG object programs include an immediate core dump of the partition; otherwise, the assessment of the error will normally be impossible.

The exceptions to the above are errors occurring during disc file OPEN and CLOSE. In this case, error messages are produced via the DMF OPEN and CLOSE transients and are output to the DMF parameter device. (Refer to the Disc Management Facility Reference Manual for a summary of the OPEN and CLOSE error messages.)

ASSESSMENT OF ABNORMAL TERMINATIONS

The partition's error register contains a relative pointer to the location causing the abort. Normally, the direct address can be obtained by the following:

LOC(41-44)-11

where LOC(41-44) represents the numeric portion of the four characters in the error register (locations 41-44).

This address normally points to a standard RPG halt error message or to location 10 which indicates an unrecoverable Read error by the DMF loader.

Standard RPG Halt Message

The standard RPG halt messages occupy 10 characters of core and are instruction-boundary aligned.

Format:

HALT ec

where ec is a one or two digit error code. (See Table 8-1.)

Examples:

HALT 4

HALT 71

Note: If 'HALT 5' is the halt message, check the locations specified on the object program map for the Halt Indicators (H1, H2, H3) to determine which indicator has been set on.

Table 8-1 HALT ERROR CODE SUMMARY

ERROR CODE	DESCRIPTION
1	a. SYSPOL pool label not readable. b. DMF logical I/O OPEN or CLOSE transient not locatable in SYSPOL. (Refer to Note 1.)
2	Record found to be out of the sequence specified by a numeric entry in columns 15-16 of the Input Specification cards.
3	Multiple records were encountered within a sequenced group when column 17 (number) of the Input Specification cards specified that only one record per group was permitted. (Col. 17 contained a '1'.
4	An unidentifiable record was encountered.
5	A Halt Indicator was turned on by the object program.
6	The match fields were found to be out of the specified sequence.
7n	Indicates that an unrecoverable I/O error was encountered by the RPG logical disc I/O module. 'n' is the standard LIOCS error code. 'n' - DESCRIPTION OF ERROR
[71]	1 - Unrecoverable read error.
[72]	2 - Unrecoverable write error.
[73]	3 - Free sector list exhausted. (Refer to Note 2.)

Notes:
1. Locations 25-30 contain the name of the transient not located.
2. Index Register 3 (locations 31-34) contains the address of the logical disc I/O module FCB.
   a. A'FCB' + 25 points to 'n'.
   b. A'FCB' + 44 contains the number of the sector in error.

OBJECT PROGRAM DEBUGGING INFORMATION

Allocation Map

The Object Program Map contains pointers which help the programmer evaluate the object code to determine the cause of abnormal terminations. This normally is restricted to evaluating RPG object execution "HALT" indications. The "allocation map" contains the pointers required to evaluate these halts. The "allocation map" entries are described below. (Also refer to the examples in Appendix F.)

Program Entry

This pointer is the logical program entry address. The left hand end of the input buffers can be located using this pointer, as the input buffers always begin 20 characters lower in core than the program entry point.

Therefore, the following calculation produces the left hand end of the input buffers:

```
A(program entry) - 20 = A(left hand end of input buffers)
```

Example:

```
PROGRAM ENTRY IS 1660

        1660-20 = 1640   (Address of left hand end of
                          input buffers)
```

The RPG program initialization code lies in the input buffer areas to conserve core.

The input buffers are allocated in the order of occurrence of the file description entries, and each buffer's length corresponds to the corresponding implicit (or explicit) record length. Figure 8.1 illustrates the input buffer allocation for three input files:

File 1	Record Length	80 characters
File 2	Record Length	60 characters
File 3	Record Length	315 characters

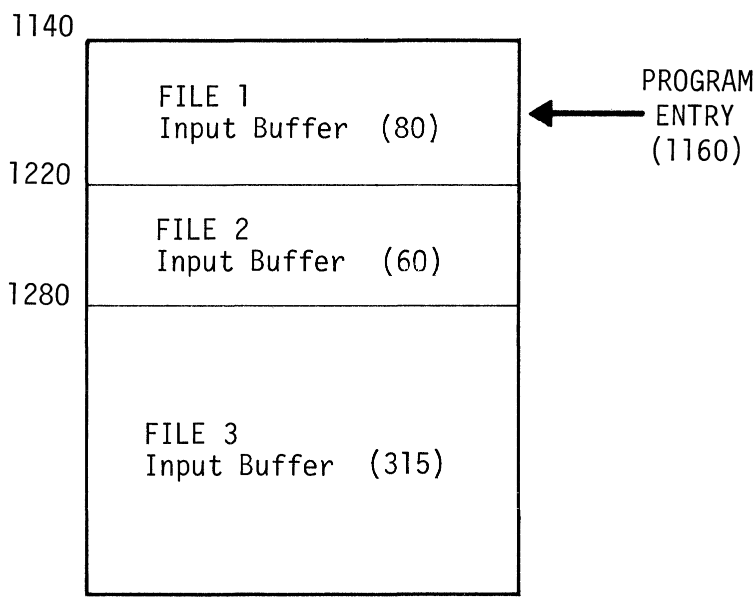


Figure 8.1 INPUT BUFFER ALLOCATION

Field Base Address

The field base address is the address to which all indicators, fields, and internal RPG object program constants are relative. It is listed on the allocation map as an aid to the programmer in field and indicator identification in object code tracing. The field base address is loaded into Index Register 1 upon program initialization and remains resident throughout RPG object program execution.

The object program map contains a list of the indicators and fields and their corresponding absolute addresses. Therefore, to relate the field or indicator symbols to the address representation in the machine instruction, it is necessary to add the field base value to corresponding values found in the A and/or B operands which have an index modifier bit for Index Register 1. If these absolute addresses cannot be matched with some address listed on the object program map referencing a field name or indicator, then they refer to some literal defined in the Calculation or Output Specifications.

Example:

Refer to Figure 8.2. Suppose that the following machine instruction is being decoded:

1P2694P4R5

Operand A is modified by the field base address in Index Register 1.

269 + 2080 = 2349	Which corresponds to the address listed on the object map for field AMTOWD.

Operand B is modified by the field base address in Index Register 1.

425 + 2080 = 2505	No corresponding address found on object map. The address is assumed to contain a constant.

Note: Field base address listed on the object map is 2080.

```
V003    *REPORT*PROGRAM*GENERATOR*          OBJECT PROGRAM MAP        MHR001  09/20/71    PAGE  2


                                   **** INDICATORS ****

        ADDR  IN            ADDR  IN            ADDR  IN            ADDR  IN            ADDR  IN

        2080  L0            2081  LR            2082  L9            2083  L8            2084  L7
        2085  L6            2086  L5            2087  L4            2088  L3            2089  L2
        2090  L1            2091  1P            2092  H1            2093  H2            2094  H3
        2095  MR            2097  OV            2100  01            2101  02

                                   ***** FIELDS *****

        ADDR  FIELD         ADDR  FIELD         ADDR  FIELD         ADDR  FIELD         ADDR  FIELD

        2275  ACCTNO        2279  NAME          2304  ADDRES        2324  CITY          2339  STATE
        2344  ZIP           2349  AMTOWD        2354  TOTAL

       ****** ALLOCATION MAP ******

       PROGRAM ENTRY             0340
       FIELD BASE ADDRESS        2080
       COMMUNICATION AREA        2260

       *** COMPILATION STATISTICS ***

       PROGRAM SIZE        3,090
       SPECIFIED SIZE     10,000
       SOURCE FILE - RPGPOL.TEMP
       OBJECT FILE - RPGPOL.RPGOBJ

       NO DIAGNOSTICS LISTED


RPG COMPILATION COMPLETED
```

Figure 8.2 OBJECT PROGRAM MAP (USE OF FIELD BASE ADDRESS)

FCB Address Table

The FCB Address Table is generated only for RPG programs with more than one input file and contains a four-character address of the File Control Block for each internal RPG file. The FCB Address Table is generated in the same order that the files appear in the input File Description Specifications. This table is used to locate the file control block for each input file. Figure 8.3 illustrates the File Control Address Block for n input files.

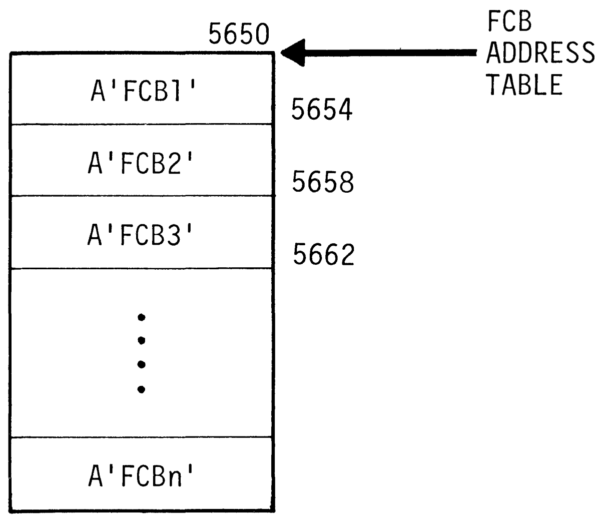


Figure 8.3 FCB ADDRESS TABLE

Communication Area

The communication area contains pointers used to process the current input file.

Table 8-2 illustrates the format of the communication area.

Table 8-2 COMMUNICATION AREA FORMAT

RELATIVE CHARACTER NUMBER*	DESCRIPTION	REMARKS
0-3	Address of current FCB.	Points to FCB of file being processed.
4-7	Address of record type entry for record identified.	If zero, record type was not identified.
8-11	Address of literal pool.	
12-13	Length of record type table entry.	Preset to '22'.
14	Global end-of-file indicator.	Set on if current file reached EOF.

*Communication area address plus relative character number yields the absolute address of the entry.

Multi-File I/O Pointers

The multi-file I/O pointers are generated only when more than one input files are specified. Table 8-3 illustrates the Multi-File I/O Pointers Table.

Table 8-3 MULTI-FILE I/O POINTERS TABLE

RELATIVE CHARACTER NUMBER*	DESCRIPTION	REMARKS
0-3	Address of FCB address table.	
4-7		Does not contain information useful for debugging.
8-9	Input file count.	Used by initialization routines. If not zero, multi-file input initialization is incomplete.
10-11	Total 'E' type file count.	Decremented each time an 'E' type file reaches 'EOF'. The program terminates when this count reaches zero.
12	Flag to indicate if primary file was ever matched to any secondary file.	Set 'on' when 'MR' was set on initially. If 'off' no primary to secondary match ever occurred.

*Multi-File I/O pointers address plus character number yields the absolute address of the entry.

FCB and Record Type Table

The data contained in the file control blocks and record type tables are necessary to evaluate RPG object program terminations. The formats of these tables are described below.

File Control Block

There is one FCB created for each input file. This FCB pertains to RPG only. The FCB for DMF Logical Input/Output Control System (LIOCS) is different. Table 8-4 illustrates the format.

Table 8-4 FILE CONTROL BLOCK

CHARACTER NUMBER	DESCRIPTION	REMARKS
0-3	Address of the record type table for this file.	Points to the L.H.E. of the first record type entry.
4-7	Address of the file read routine.	
8-11	Address of the entry in the record type table for the previous record, to permit checking for numeric sequence.	
12-15	Address of the first entry in the record type table, to be checked for numeric sequence.	If this address is equal to the address of this file's FCB, no records were specified to be checked for sequence.
16-19	Address of the entry in the record type table for the current record.	
20-21	Number of record type entries in the record type table for this file.	
22	End-of-file flag.	On when this file reaches 'EOF'. (Utilized only in multi-file input.)**
24	Match fields flag.	On when match fields are specified for this file.**
25	'E' flag.	On if 'E' was specified for this file.**
26-ff*	Match hold area.	The save area for the current match fields as specified for this file. (Length is the sum of the lengths for all match fields.)

* ff is the final character position, which depends on the length of the Match Hold Area.

** 0 = Off and 1 = On.

Record Type Table

A 22-character entry in the record type table is generated for each record type specified on the Input Specifications Form. The entries are generated in the order of definition of the records specified for each input file. These entries make up the record type table. Table 8-5 illustrates the format of a record type table entry.

Table 8-5 RECORD TYPE TABLE ENTRY

CHARACTER NUMBER	DESCRIPTION	REMARKS
0	Multiple Records Allowed Flag, indicating whether multiple records of a type are allowed.	Contents 'N' - any number may occur. '0' - only one record type is permitted per record group. (This field is not utilized for records not checked for numeric sequence.)
1	Flag indicating whether the record type is optional.	Contents Ø - record type is optional. 0 - record type is required for each record group.* (This field is not utilized for records not checked for numeric sequence.)
2-5	Address of record indicator set on if this record type is identified.	Address is relative to the field base address.
6-9	Address of record identification routine for this record type.	
10-13	Address of routine which moves the fields specified in the input specifications from the input buffer to the field areas.	
14-17	Address of the control break determination routine for this record type.	If the address is zero, no control break fields were specified for this record type.
18-21	Address of the match field concatenation routine.	If the address is zero, no match fields were specified for this record type.

* 'Ø' denotes alphabetic character;
'0' denotes numeric digit.

Halt Code Display

10K Compile Feature Only

In the event of an abnormal termination of the object program, the Halt Code will be displayed on the workstation (device 0). The format of the Halt Code is the same as described in Table 8-1, with the following exceptions;

1. Instead of

    HALT 1

the error code format will be

    HALT 1 OPEN

or

    HALT 1 CLOSE

This indicates which function the object program was attempting to perform.

2. Instead of

    HALT 5

the format is

    HALT 5 xyz

where x is the value of Halt Indicator H1 (0=off, 1-on),

y is the value of Halt Indicator H2 (0=off, 1-on),

and z is the value of Halt Indicator H3 (0=off, 1-on).

Thus it is possible to determine from the workstation display which Halt Indicator was set on by the object program.

3. The additional Halt Code

    HALT 70

will signify that a required disc was off-line at execution time and the operator cancelled the job by depressing a control key.

Examples

HALT 6	(Match fields out of specified sequence.)
HALT 5 010	(Halt Indicator H2 was set on.)

Appendix A

FLOWCHART OF SYSTEM TEN RPG OBJECT PROGRAM

APPENDIX A: FLOWCHART OF SYSTEM TEN RPG OBJECT PROGRAM

The following pages show the flowchart for the object program produced by the System Ten RPG compiler.

Figure A.1 shows the complete cycle of the object program.

Figure A.2 is the flowchart for the subroutine IMFIO, which is used to initialize the input files when there are multiple input files.

Figure A.3 is the flowchart for the subroutine SELNRC, which is used to select the next record when fields are being matched.

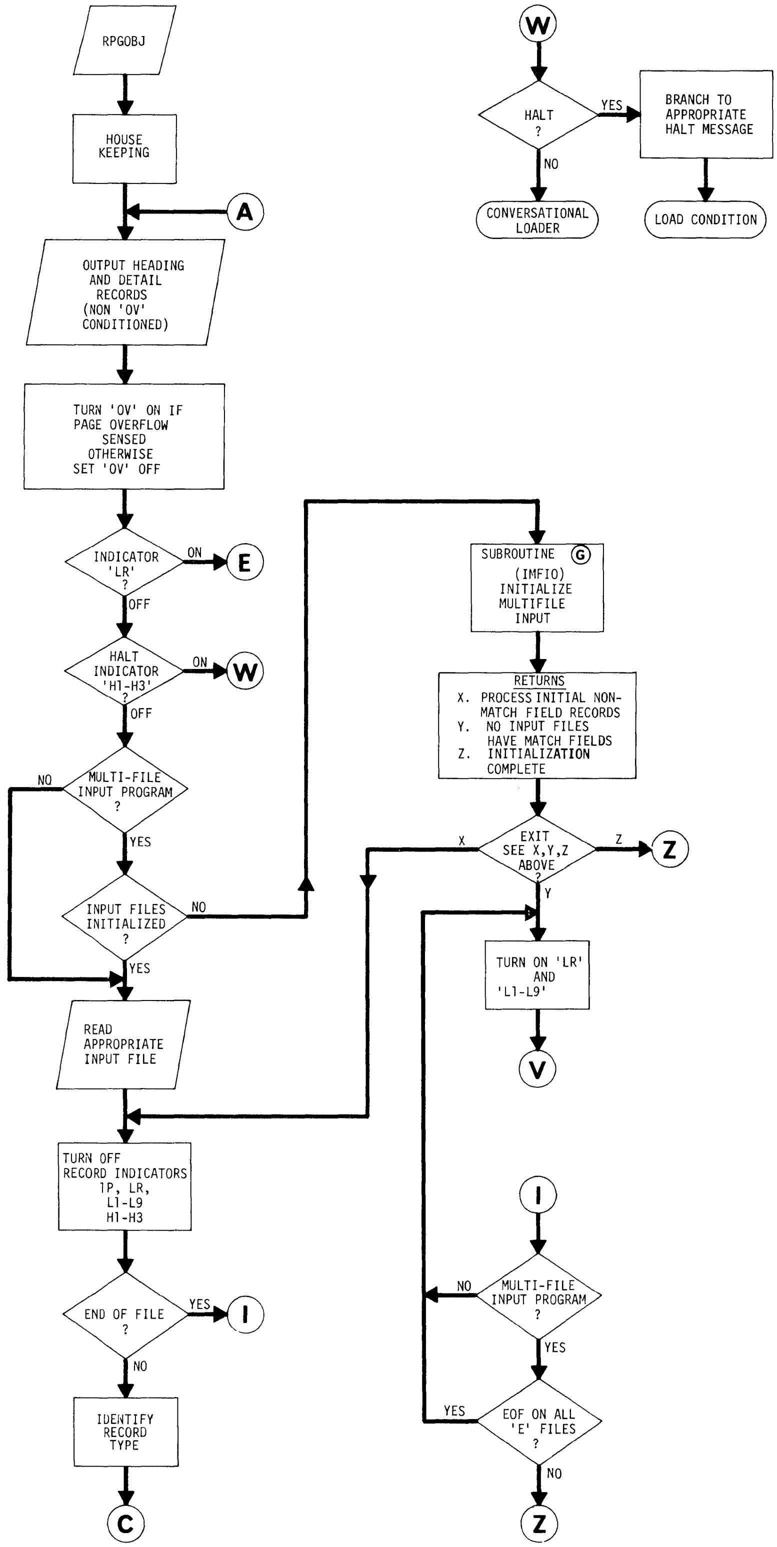
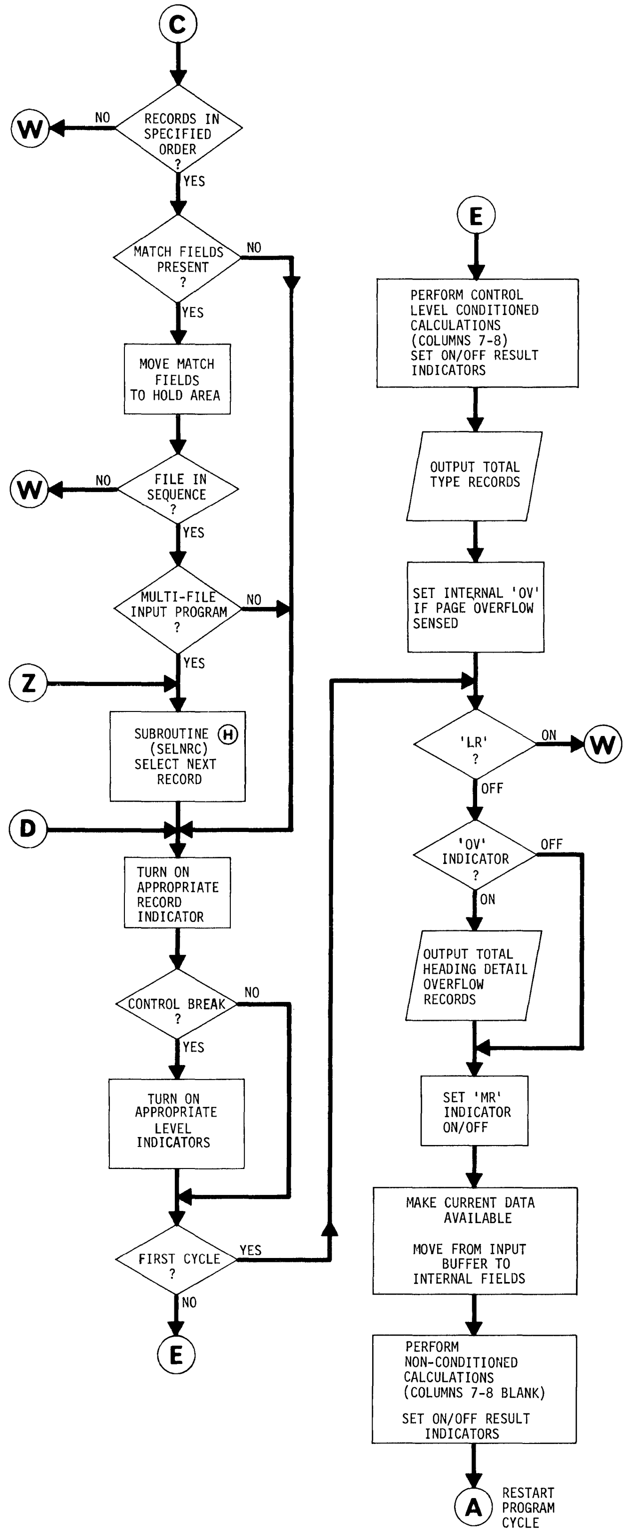


Figure A.1 FLOWCHART FOR SINGER SYSTEM TEN RPG OBJECT PROGRAM (COMPLETE PROGRAM CYCLE)


C
RECORDS IN SPECIFIED ORDER ?
NO
W
YES
MATCH FIELDS PRESENT ?
NO
YES
MOVE MATCH FIELDS TO HOLD AREA
FILE IN SEQUENCE ?
NO
W
YES
MULTI-FILE INPUT PROGRAM ?
NO
YES
Z
SUBROUTINE H (SELNRC) SELECT NEXT RECORD
D
TURN ON APPROPRIATE RECORD INDICATOR
CONTROL BREAK ?
NO
YES
TURN ON APPROPRIATE LEVEL INDICATORS
FIRST CYCLE ?
YES
NO
E
E
PERFORM CONTROL LEVEL CONDITIONED CALCULATIONS (COLUMNS 7-8) SET ON/OFF RESULT INDICATORS
OUTPUT TOTAL TYPE RECORDS
SET INTERNAL 'OV' IF PAGE OVERFLOW SENSED
'LR' ?
ON
W
OFF
'OV' INDICATOR ?
OFF
ON
OUTPUT TOTAL HEADING DETAIL OVERFLOW RECORDS
SET 'MR' INDICATOR ON/OFF
MAKE CURRENT DATA AVAILABLE
MOVE FROM INPUT BUFFER TO INTERNAL FIELDS
PERFORM NON-CONDITIONED CALCULATIONS (COLUMNS 7-8 BLANK)
SET ON/OFF RESULT INDICATORS
A
RESTART PROGRAM CYCLE

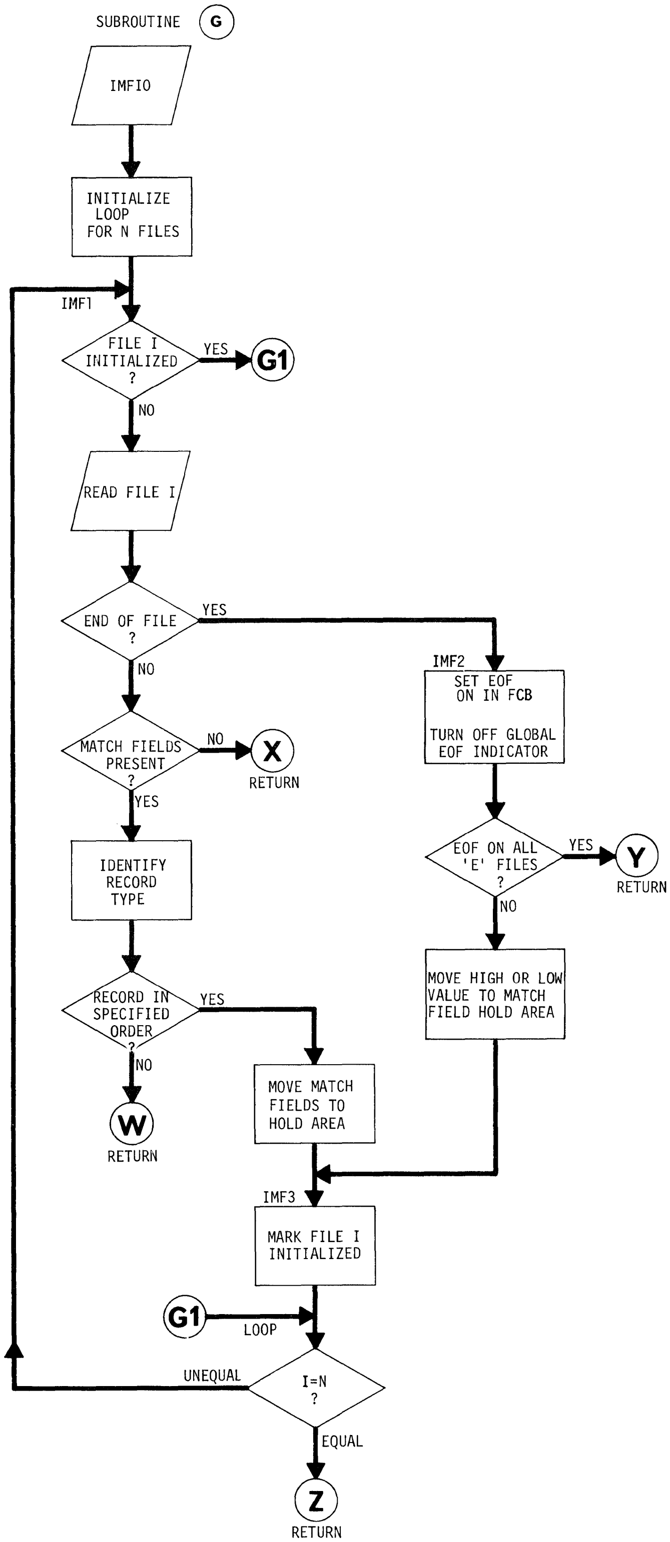


Figure A.2 FLOWCHART FOR RPG MULTI-FILE INPUT INITIALIZATION SUBROUTINE IMFIO

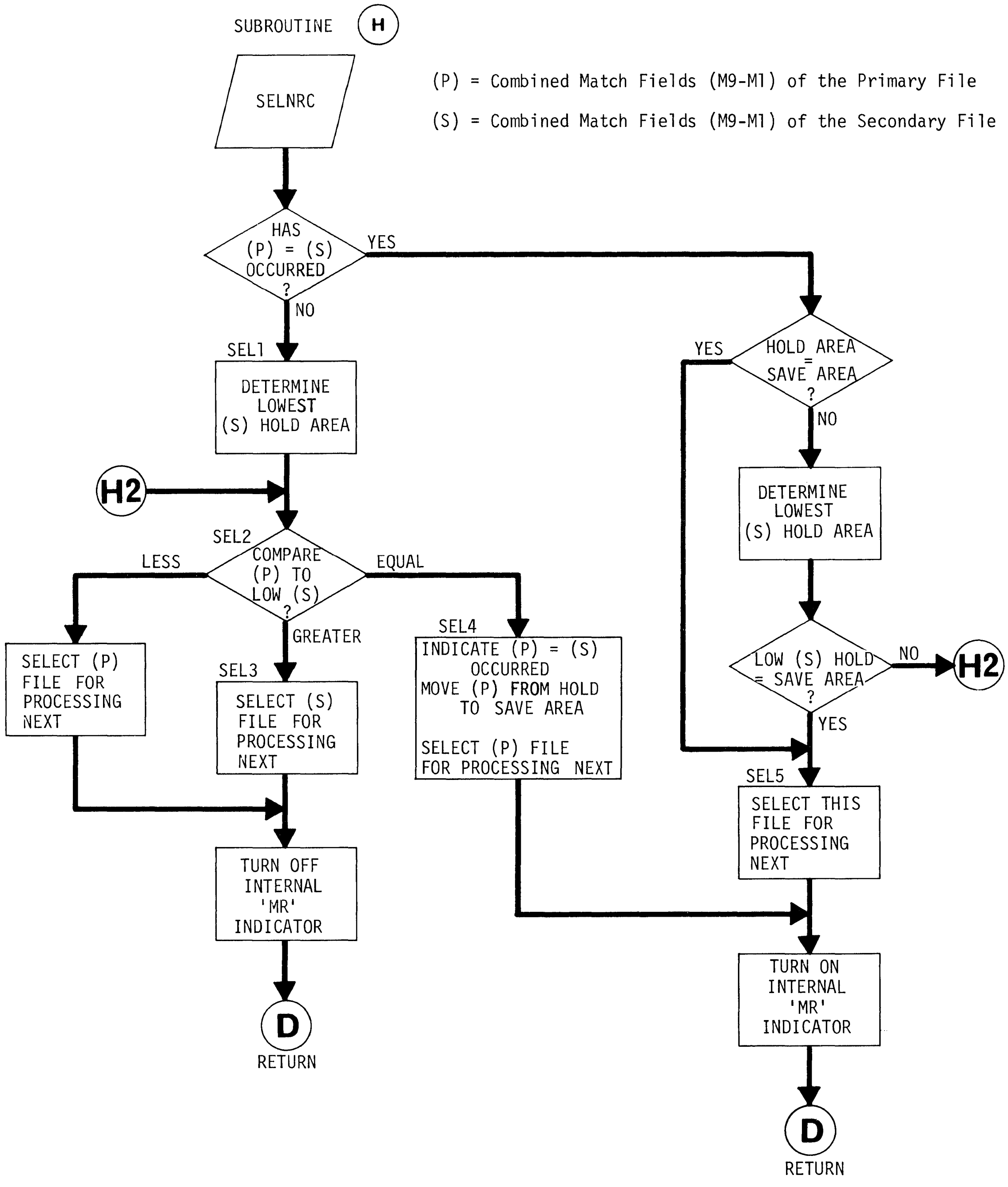


Figure A.3 FLOWCHART FOR RPG SELECT NEXT RECORD SUBROUTINE SELNRC

Appendix B

RPG LINKAGE CONVENTIONS

APPENDIX B: RPG LINKAGE CONVENTIONS

LINKAGE CONVENTIONS

The linkage between the RPG program and Assembler subroutines will be according to standard System Ten conventions with the following exception.

The entry to the user subroutines will be via a control vector contained immediately at the high core address calculated from the control card entry indicating object partition size. The vector is assumed to contain branch instructions to the entry points of the user's subroutines located in core, following the control vector, dependent upon the order in which the EXIT operations (to differently "named" subroutines) are coded on the calculation specifications. The control vector is the sole responsibility of the user and is assumed to be proper by the RPG object program.

RLABL PARAMETERS

Each field or indicator is addressed relative to a base address contained in Index Register 1.

Indicators are one character in length and contain a '0' if off, or a '1' if on.

Refer to Section 7, Example 3, and Figure 7.4 for a detailed discussion of incorporating an object program resulting from an Assembler source program into an object program resulting from an RPG source program.

SUBROUTINE DISC I/O

When handling Disc I/O within an Assembler subroutine, the user should consider several points:

1. Non-disc RPG programs ORG at 340 and have input areas beginning at 320 after initialization is complete. Therefore, calling _OPEN or _CLOSE from a subroutine will overlay vital information. To bypass this obstacle, specify a dummy disc output file (this requires a file label for a null file) to reserve core positions 300-999.

2. To ensure orderly opening of the subroutine disc files, use an EXEC statement in the first subroutine where these files can be opened prior to RPG gaining control. This segment can then be overlaid by subsequent Assembler routines and by the RPG object program.

3. Prior to executing the RPG object program, the user should open all input files by means of an initial overlay. The initial overlay must invoke the loader to bring in the RPG object program and the main Assembler subroutine segment.

4. Closing the files should present no problem if the above points are taken into consideration. _CLOSE should be called only if indicator 'LR' is on.

RPG EXIT EXAMPLE

The following example illustrates how RPG interfaces with an EXIT to an Assembler subroutine. The example is uncomplicated but does show how to extract argument addresses.

The RPG program (Figures B.1, B.2, and B.3) reads a card which contains an action code and two fields. The action code, Field A, Field B, indicator 55, a result field, and the card image are passed as arguments to the Assembler subroutine. Upon regaining control, the RPG program checks for indicator 55 being on. If it is on, the program branches around any further calculations and goes to the print routine to print an error message.

If indicator 55 is not on, the action code is compared to 'A' and 'S'. If it is neither 'A' nor 'S', indicator 57 is turned on. This error condition will happen only if the Assembler subroutine does not detect the invalid action code. An error message will then be printed indicating that SUBR1 did not work and subroutine SUBR2 (Fig. B.8) will be called to print the error on the workstation. If all is well, the RPG program will list the two fields, the operation ('A'=add, 'S'=subtract), and the result.

The Assembler subroutine SUBR1 (Figures B.4, B.5, B.6, and B.7) examines the operation code for validity (either an 'A' or an 'S'), verifies that the two fields are numeric and then performs the action specified (either adds Field A to Field B or subtracts Field B from Field A) and stores the answer in the result field. If the operation code is invalid or if one (or both) of the fields is not numeric, then indicator 55 is set on and a message is printed on the workstation identifying the error.

```
V003    *REPORT*PROGRAM*GENERATOR*      SOURCE/DIAGNOSTIC LISTING              09/20/71    PAGE  1

01010H     700
01015F*  THIS PROGRAM AND THE ASSEMBLER SUBROUTINE WITH IT
01016F*  WILL GIVE AN EXAMPLE OF HOW TO USE THE RPG EXIT
01017F*   FACILITY
01018F*  NOTE -- THE 'CORE SIZE' FIELD CONTAINS THE
01019F*   ADDRESS OF THE FIRST TRANSFER VECTOR TO THE EXIT
0101AF*  SUBROUTINE. I.E. 7000.  THE SECOND TRANSFER
0101BF*  VECTOR WILL BE LOCATED 10 POSITIONS HIGHER IN CORE
01020FCARDS   IPE       0080            READER      1
01030FPRINT   O         0132     OB     PRINTER     2
02010ICARDS   AA  01    1NC
02020I                                        1   1 CODE
02030I                                        5  11 FLDA
02040I                                       15  21 FLDB
02050I                                        1  21 CRD
02060I*  THE FOLLOWING SPECIFICATION IS AN EXAMPLE OF THE
02062I*    CATCH-ALL INPUT TECHNIQUE WHICH IS USED TO
02064I*    PREVENT PROGRAM TERMINATION DUE TO AN UNRECOGNIZED
02066I*     INPUT RECORD TYPE
02070I        BB  02   1 C
02075I                                        1  21 ECRD
02080I*
03002C*   FOR A COMPLETE DESCRIPTION OF THE LOGIC OF THIS
03004C*  EXAMPLE SEE THE ASMBLER LISTING FOR THE SUBROUTINE
03006C*
03010C   02                 MOVE 'X'      CODE              INPUT ERROR
03015C   02                 MOVE ECRD     CRD               MOVE ERR TO ARG
03017C*
03020C                      SETOF                      5557  SETOF ERR IND
03030C                      EXIT SUBR1                      CALL SUBR1
03040C                      RLABL         CODE              ARGS ARE: OP
03050C                      RLABL         FLDA                CODE,FLDA,
03060C                      RLABL         FLDB                FLDB,RESULT
03070C                      RLABL         RESULT  80        FIELD,INVALID
03080C                      RLABL         IN55    10          FIELD IND,
03090C                      RLABL         CRD               CARD IMAGE.
03092C*
03100C   55                 GOTO END                          GO AROUND IF
03110C*                                                     ERROR DETECTED
03112C*
03120C         CODE         COMP 'A'                   414140 SET IND FOR
03130C   41    CODE         COMP 'S'                   575741 PRNTING OPER
03140C*                                                IF IND 57 ON,SUBR
03142C*                                                    FAILED TO WORK
03143C*                                                FLDA,FLDB ARE NON-NUM
03144C                      MOVE FLDA     FLDA1   70        MOVE TO NUMERIC
03146C                      MOVE FLDB     FLDB1   70         PRINT FIELDS
03148C*                                                      FOR EDITING.
03149C*
03150C         END          TAG
03160C*    IF SUBROUTINE DID NOT WORK, THEN PRINT ERROR MESSAGE ON
03170C   57                 EXIT SUBR2                      WORKSTATION VIA
03180C                      RLABL         CRD                2ND EXIT, ARGS
```

Figure B.1   SAMPLE RPG OBJECT PROGRAM SHOWING SUBROUTINE EXITS

```
V003   *REPORT*PROGRAM*GENERATOR*     SOURCE/DIAGNOSTIC LISTING              09/20/71   PAGE  2

  54     03190C                          RLABL           CODE              CARD IMAGE,CODE
  55     03200C*
  56     040100PRINT    H  301    0B
  57     040200       OR          1P
  58     040220*    NOTE ABOVE --  OVERFLOW IND IS SPECIFIED 1ST
  59     040300                          UDATE      10
  60     040400                                     56 'EXAMPLE OF RPG EXIT FACI'
  61     040500                                     60 'LITY'
  62     040600                                     74 'PAGE'
  63     040700                          PAGE  Z    80
  64     040800*
  65     040900         D  2      01N55N57
  66     041200                          FLDA1      10 ' ,    ,    -'
  67     041300                   40                17 'PLUS'
  68     041400                   41                18 'MINUS'
  69     041500                          FLDB1      30 ' ,    ,    -'
  70     041600                                     33 '='
  71     041700                          RESULT     46 '  ,    ,    -'
  72   S 040800*  THE FOLLOWING ARE ERROR MESSAGES
  73     041900         D 33      55
  74     042000        OR         57
  75     050100*
  76     050200                  N57                22 'SEE ERROR MESSAGE ON'
  77     050300                  N57                34 'WORKSTATION'
  78     050400                   57                26 'SUBROUTINE FAILED TO DET'
  79     050500                   57                50 'ECT INVALID OP CODE - SE'
  80     050600                   57                70 'E WORKSTATION OUTPUT'
```

Figure B.2   SAMPLE RPG OBJECT PROGRAM SHOWING SUBROUTINE EXITS

```
V003    *REPORT*PROGRAM*GENERATOR*          OBJECT PROGRAM MAP                    09/20/71    PAGE


                                          ****** EXITS ******

       ADDR   EXIT          ADDR   EXIT          ADDR   EXIT          ADDR   EXIT          ADDR   EXIT

       7000   SUBR1         7010   SUBR2

                                          **** INDICATORS ****

       ADDR   IN            ADDR   IN            ADDR   IN            ADDR   IN            ADDR   IN

       2370   L0            2371   LR            2372   L9            2373   L8            2374   L7
       2375   L6            2376   L5            2377   L4            2378   L3            2379   L2
       2380   L1            2381   1P            2382   H1            2383   H2            2384   H3
       2385   MR            2387   0B            2390   01            2391   02            2392   55
       2393   57            2394   41            2395   40

                                          ***** FIELDS *****

       ADDR   FIELD         ADDR   FIELD         ADDR   FIELD         ADDR   FIELD         ADDR   FIELD

       2565   CODE          2566   FLDA          2573   FLDB          2580  *CRD           2601   ECRD
       2622   RESULT        2630   FLDA1         2637   FLDB1         2644   UDATE         2652   PAGE


      ****** ALLOCATION MAP ******

      PROGRAM ENTRY            0340
      FIELD BASE ADDRESS       2370
      COMMUNICATION AREA       2550


      *** COMPILATION STATISTICS ***

      PROGRAM SIZE        3,340
      SPECIFIED SIZE      7,000
      SOURCE FILE = RPGPOL.TEMP
      OBJECT FILE = RPGPOL.RPGOBJ

      NO DIAGNOSTICS LISTED


RPG COMPILATION COMPLETED
```

Figure B.3 SAMPLE RPG OBJECT PROGRAM SHOWING SUBROUTINE EXITS

```
PAGE  001                        ASSMBLER SUBROUTINE FOR RPG PROGRAM
SEQ. LOCN  INSTR/DATA OP   A/R  L I  B/S  L I LINE  LABEL  OPCODE OPERAND(S) AND/OR COMMENTS

     0000                                     0003 ******************************************************************
     0000                                     0004 *                                                                *
     0000                                     0005 *      THIS PROGRAM IS CALLED BY AN RPG PROGRAM. (VIA EXIT )      *
     0000                                     0006 *                                                                *
     0000                                     0007 *      THE PROGRAM WILL TEST THE OP CODE FOR 'A' OR 'S'           *
     0000                                     0008 *        IF NOT 'A' OR 'S' INDICATOR 55 WILL BE SETON AND THE ERROR *
     0000                                     0009 *        RETURN BRANCH IS TAKEN. IF OP CODE CODE IS VALID, FLDA AND *
     0000                                     0010 *        FLDB ARE TESTED FOR NUMERIC VALUES. IF NON-NUMERIC       *
     0000                                     0011 *        INDICATOR 55 IS SETON. IF INDICATOR 55 IS SETON AN ERROR *
     0000                                     0012 *        MESSAGE IS OUTPUT TO THE WORKSTATION IDENTIFYING THE ERROR *
     0000                                     0013 *                                                                *
     0000                                     0014 *      THE CALLING SEQUENCE IS:     (GENERATED BY RPG)           *
     0000                                     0015 *                                                                *
     0000                                     0016 *      BC      31(6),7000(5)  --- (SEE RPG CONTROL CARD CORE SIZE ) *
     0000                                     0017 *      BC      CODE(0),FLDA(0)                                   *
     0000                                     0018 *      BC      FLDB(0),RESULT(0)                                 *
     0000                                     0019 *      BC      IN55(0),CRD(0)                                    *
     0000                                     0020 *                                                                *
     0000                                     0021 *        THE SECOND SUBROUTINE IS DESCRIBED BELOW                *
     0000                                     0022 *                                                                *
     0000                                     0023 ******************************************************************

     0000                                     0025 ******************************************************************
     0000                                     0026 *                                                                *
     0000                                     0027 *      THE METHOD OF EXTRACTING ABSOLUTE ADDRESSES AND DATA AT THOSE*
     0000                                     0028 *          ADDRESSES IS AS FOLLOWS:                              *
     0000                                     0029 *                                                                *
     0000                                     0030 *      ACCESS THE ADDRESS IN REG 3 + THE DISPLACEMENT TO THE     *
     0000                                     0031 *      ARGUMENT DESIRED. FOR EXAMPLE: WE WISH TO ACCESS THE DATA AT *
     0000                                     0032 *      THE ADDRESS REFERENCED BY THE VARIABLE  - RESULT -        *
     0000                                     0033 *                                                                *
     0000                                     0034 *      - RESULT - IS 16 POSITIONS FROM THE ADDRESS IN REG 3 (DUE TO *
     0000                                     0035 *      THE BRANCH AND LINK INSTRUCTION GENERATED BY RPG),THEREFORE *
     0000                                     0036 *      WE WILL EXTRACT THE ADDRESS 16 POSITIONS FROM THE ADDRESS  *
     0000                                     0037 *      IN REG 3 WITH A - MN - INSTRUCTION AND STORE THIS ADDRESS  *
     0000                                     0038 *      IN REG 2 WHICH HAS BEEN SAVED (E.G. MN  16(4,3),IR2  )     *
     0000                                     0039 *                                                                *
     0000                                     0040 *      WE NOW HAVE THE ** RELATIVE ** ADDRESS OF THE DATA WE WANT, *
     0000                                     0041 *      TO GET THE ABSOLUTE ADDRESS WE MUST ADD THE BASE REG(REG 1) *
     0000                                     0042 *      TO THE RELATIVE ADDRESS IN REG 2.  (E.G.  A  IR1,IR2 )     *
     0000                                     0043 *      INDEX REG 2 NOW HAS THE ABSOLUTE ADDRESS OF THE DATA WE WANT *
     0000                                     0044 *                                                                *
     0000                                     0045 ******************************************************************
```

Figure B.4 LISTING OF EXIT SUBROUTINE SUBR1

```
PAGE  002                                  ASSMBLER SUBROUTINE FOR RPG PROGRAM
SEQ. LOCN   INSTR/DATA OP   A/R   L I  B/S   L I LINE  LABEL   OPCODE OPERAND(S) AND/OR COMMENTS

     0000                                        0047  *
     0000                                        0048  *    LABEL INDEX REGISTERS
     0000                                        0049  *
     0000                                        0050          ORG     11
     0011                   0001       0004      0051  IR1     DM      N4
     0015                                        0052  *
     0015                                        0053          ORG     21
     0021                   0001       0004      0054  IR2     DM      N4
     0025                                        0055  *
     0025                                        0056          ORG     31
     0031                   0001       0004      0057  IR3     DM      N4

     0035                                        0059  *
     0035                                        0060  *       DEFINE TRANSFER VECTORS FOR BOTH SUBR1 AND SUBR2
     0035                                        0061  *
     0035                                        0062          ORG     7000
     7000   U7PR000000 11  7020   5 0  0000   0 0 0063          BC      SUBR1(5)
     7010   U7VY000000 11  7690   5 0  0000   0 0 0064          BC      SUBR2(5)


     7020                                        0066  *       BEGIN SUBR1 EXECUTION

     7020   Q001147550 08  0011   1 0  7550   4 0 0068  SUBR1   MC      IR1(14),SAVREG          SAVE REGS 1 & 2
     7030   P00S147541 09  0031   0 0  7541   4 0 0069          MN      IR3,RTN+1               SET RETURN ADDRESS
     7040   P00P1TP021 09  0001   0 3  0021   4 0 0070          MN      1(4,3),IR2   'CODE'     GET RELATIVE ADDRESS OF OP CODE
     7050   4P01140021 04  0011   4 0  0021   4 0 0071          A       IR1,IR2                 ADD BASE ADDR - IR2 NOW HAS
     7060                                        0072  *                                       ABSOLUE ADDRESS OF OP CODE (CODE)
     7060   PPP00Q7564 14  0000   0 2  7564   1 0 0073          C       0(1,2),CA               TEST FOR -A-
     7070   R7QV000000 11  7160   2 0  0000   0 0 0074          BC      DOA(2)                  IF TRUE SET UP TO ADD OPERANDS
     7080   PPP00Q7565 14  0000   0 2  7565   1 0 0075          C       0(1,2),CS               IF FALSE TEST FOR -B-
     7090   R7QX000000 11  7180   2 0  0000   0 0 0076          BC      DOB(2)                  IF TRUE SET UP TO SUB OPERANDS
     7100                                        0077  *                                       IF FALSE SET INDICATOR 55
     7100   P00R1TP021 09  0021   0 3  0021   4 0 0078          MN      21(4,3),IR2  'IN55'     GET REL ADDR OF IN55
     7110   4P01140021 04  0011   4 0  0021   4 0 0079          A       IR1,IR2                 ADD BASE ADDR TO GET ABSOL ADDR
     7120   P760510P00 08  7605   0 0  0000   1 2 0080          MC      ONE,0(,2)               MOVE A ONE(SETON) TO IN55
     7130   076X330001 01  7683   0 0  0001   3 0 0081          W       CR(0),1(3)              CARRIAGE RETURN
     7140   075V610037 01  7566   0 0  0037   1 0 0082          W       ERMSG1(0),37(1)         WRITE ERROR MESSAGE ON WRKST
     7150   U7TW000000 11  7470   5 0  0000   0 0 0083          BC      EXITE(5)                RETURN TO RPG PROGRAM VIA ERROR
     7160                                        0084  *                                       ROUTINE
     7160   P76P317340 09  7603   0 0  7340   1 0 0085  DOA     MN      FIVE,ADDBR              SET BRANCH TO GOTO ADD ROUTINE
     7170   U7QY000000 11  7190   5 0  0000   0 0 0086          BC      GETFLD(5)               GO TO EXTRACT FIELD ADDRESSES
     7180   P76P417340 09  7604   0 0  7340   1 0 0087  DOB     MN      ZERO,ADDBR              NOP BRANCH TO ADD ROUTINE
     7190                                        0088  *

     7190                                        0090  *     HERE WE EXTRACT THE ABSOLUTE ADDRESSES OF FLDA AND FLDB
     7190                                        0091  *       AND TEST THEM FOR NUMERIC

     7190   P00P6TP021 09  0006   0 3  0021   4 0 0093  GETFLD  MN      6(4,3),IR2   'FLDA'     GET REL ADDR OF FLDA
     7200   4P01140021 04  0011   4 0  0021   4 0 0094          A       IR1,IR2                 ADD BASE ADDR TO GET ABSL ADDR
     7210   WP0P0W7606 13  0000   7 2  7606   7 0 0095          FN      0(7,2),WORK             -FN- FLDA DATA TO WORK AREA
     7220   PPP00W7606 14  0000   0 2  7606   7 0 0096          C       0(7,2),WORK             COMPARE - IF EQUAL, FLDA IS NUMRC
     7230   R7RT057420 11  7240   2 0  7420   5 0 0097          BC      *+10(2),ERRN(5)                  IF UNEQUAL - ERROR
     7240   P0000W7613 08  0000   0 2  7613   7 0 0098          MC      0(7,2),FLDA             MOVE DATA TO HOLD AREA FOR FLDA
     7250                                        0099  *
```

Figure B.5 LISTING OF EXIT SUBROUTINE SUBR1

```
PAGE  003                           ASSMBLER SUBROUTINE FOR RPG PROGRAM
SEQ. LOCN  INSTR/DATA UP  A/R  L I  B/S  L I LINE  LABEL  OPCODE OPERAND(S) AND/OR COMMENTS

     7250  P00Q1TP021 09 0011  0 3 0021  4 0 0100         MN     11(4,3),IR2  'FLDB'  GET REL ADDR OF FLDB
     7260  4P01140021 04 0011  4 0 0021  4 0 0101         A      IR1,IR2              ADD BASE REG TO GET ABSOLUTE ADDR
     7270  WP0P0W7606 13 0000  7 2 7606  7 0 0102         FN     0(7,2),WORK          -FN- FLDB DATA TO WORK AREA
     7280  PPP00W7606 14 0000  0 2 7606  7 0 0103         C      0(7,2),WORK          COMPARE - IF EQUAL, FLDB IS NUMERC
     7290  R7SP057420 11 7300  2 0 7420  5 0 0104         BC     *+10(2),ERRN(5)                IF UNEQUAL, ERROR
     7300  P0000W7620 08 0000  0 2 7620  7 0 0105         MC     0(7,2),FLDB          MOVE DATA TO HOLD AREA FOR FLDB
     7310                                    0106  *
     7310  P00Q6TP021 09 0016  0 3 0021  4 0 0107         MN     16(4,3),IR2 'RESULT' GET REL ADDR OF RESULT
     7320  4P01140021 04 0011  4 0 0021  4 0 0108         A      IR1,IR2              ADD BASE REG TO GET ABSOLUTE ADDR
     7330  P00R147406 09 0021  0 0 7406  4 0 0109         MN     IR2,MOVRES+6         PUT ADDR OF -RESULT-IN INSTRUCTION
     7340                                    0110  *                                WHICH CPTURES RESULT OF OPERATION
     7340  P7SX000000 11 7380  0 0 0000  0 0 0111  ADDBR  BC     ADDRTN(0)            BRANCH SET BY OP CODE ROUTINE


     7350                                    0113  *   ROUTINE TO SUBTRACT FLDB FROM FLDA

     7350  WW6Q387627 13 7613  7 0 7627  8 0 0115         FN     FLDA(7),WRKA(8)      MOVE FLDA TO RESULT HOLD AREA
     7360  7WVR087627 07 7620  7 0 7627  8 0 0116         S      FLDB,WRKA            SUB FLDB FROM FLDA
     7370  U7TP000000 11 7400  5 0 0000  0 0 0117         BC     MOVRES(5)            GOTO MOVE RESULT
     7380                                    0118  *
     7380  WW6R087627 13 7620  7 0 7627  8 0 0119  ADDRTN FN     FLDB(7),WRKA(8)      MOVE FLDB TO RESULT HOLD AREA
     7390  7W61387627 04 7613  7 0 7627  8 0 0120         A      FLDA,WRKA            ADD FLDA TO FLDB
     7400                                    0121  *
     7400  P762780000 08 7627  0 0 0000  8 0 0122  MOVRES MC     WRKA,0               MOVE RESULT TO ADDRESS FOTTEN FROM
     7410                                    0123  *                                RPG RLABL POINTING FIELD CALLED
     7410                                    0124  *                                - RESULT -

     7410  U7US000000 11 7530  5 0 0000  0 0 0126         BC     EXIT(5)              ROUTINE DONE .. GO TO REURN


     7420                                    0128  *      HERE WHEN FLDA OR FLDB NOT NUMERIC

     7420  P00R1TP021 09 0021  0 3 0021  4 0 0130  ERRN   MN     21(4,3),IR2   'IN55' GET REL ADDR OF IN55
     7430  4P01140021 04 0011  4 0 0021  4 0 0131         A      IR1,IR2              ADD BASE ADDR TO GET ABSOLUTE ADDR
     7440  P760510P00 08 7605  0 0 0000  1 2 0132         MC     ONE,0(,2)            SET ON -IN55-
     7450  076X330001 01 7683  0 0 0001  3 0 0133         W      CR(0),1(3)           CARRIAGE RETURN
     7460  076S510048 01 7635  0 0 0048  1 0 0134         W      ERMSG2(0),48(1)      WRITE ERROR MESSAGE ON WRKST
     7470                                    0135  *
     7470                                    0136  *    ERROR EXIT ROUTINE
     7470                                    0137  *
     7470  076X330001 01 7683  0 0 0001  3 0 0138  EXITE  W      CR(0),1(3)           CARRIAGE RETURN
     7480  P00R6TP021 09 0026  0 3 0021  4 0 0139         MN     26(4,3),IR2  'CRD'   GET REL ADDR OF -CRD-
     7490  4P01140021 04 0011  4 0 0021  4 0 0140         A      IR1,IR2              ADD BASE TO GET ABSOLUTE ADDR
     7500  P00R147511 09 0021  0 0 7511  4 0 0141         MN     IR2,*+11             MOVE ADDR TO WRKST WRITE INST.
     7510  000P010021 01 0000  0 0 0021  1 0 0142  PRTCRD W      0(0),21(1)           WRITE CARD IMAGE
     7520  076X330001 01 7683  0 0 0001  3 0 0143         W      CR(0),1(3)           C. R.
     7530  0755040011 08 7550  1 0 0011  4 0 0144  EXIT   MC     SAVREG,IR1           RESTORE REGS 1 & 2
     7540  U0PP000000 11 0000  5 0 0000  0 0 0145  RTN    BC     0(5)                 RETURN TO RPG


     7550                                    0147  *
     7550                                    0148  *   CONSTANTS AND HOLD AREAS FOR SUBR1
     7550                                    0149  *

     7550                    0001      0014  0151  SAVREG DM     C14
```

Figure B.6   LISTING OF EXIT SUBROUTINE SUBR1

```
PAGE  004                                   ASSMBLER SUBROUTINE FOR RPG PROGRAM
SEQ. LOCN  INSTR/DATA OP   A/R  L I  B/S  L I LINE  LABEL   OPCODE OPERAND(S) AND/OR COMMENTS

     7564  A               0001      0001      0152  CA      DM     C1'A'
     7565  S               0001      0001      0153  CS      DM     C1'S'
     7566  INVALID OP      0001      0037      0154  ERMSG1  DM     C37'INVALID OP CODE (COL 1) IN CARD BELOW'
     7603  5               0001      0001      0155  FIVE    DM     N1'5'
     7604  0               0001      0001      0156  ZERO    DM     N1'0'
     7605  1               0001      0001      0157  ONE     DM     N1'1'
     7606  0               0001      0007      0158  WORK    DM     N7'0'
     7613                  0001      0007      0159  FLDA    DM     N7
     7620                  0001      0007      0160  FLDB    DM     N7
     7627                  0001      0008      0161  WRKA    DM     N8
     7635  FLDA(COL 5      0001      0048      0162  ERMSG2  DM     C48'FLDA(COL 5-11) OR FLDB(COL 15-21) INVALID--BELOW'
     7683  M               0001      0001      0163  CR      DM     C1'M'
```

Figure B.7 LISTING OF EXIT SUBROUTINE SUBR1

```
,PAGE  005                      ASSMBLER SUBROUTINE FOR RPG PROGRAM
 SEQ. LOCN  INSTR/DATA OP   A/R  L I  B/S  L I LINE  LABEL  OPCODE OPERAND(S) AND/OR COMMENTS

      7684                                    0165  ***********************************************************************
      7684                                    0166  *                                                                     *
      7684                                    0167  *      SUBR2  -  THIS SUBROUTINE WILL PRINT A MESSAGE                 *
      7684                                    0168  *                INDICATING THAT SUBR1 DID NOT DETECT                 *
      7684                                    0169  *                AN INVALID OP CODE. MSG OUTPUT TO WORKSTATION        *
      7684                                    0170  *                                                                     *
      7684                                    0171  *           THE CALLING SEQUENCE IS:     (GENERATED BY RPG)           *
      7684                                    0172  *                                                                     *
      7684                                    0173  *      BC     31(6),7010(5)         *NOTE- BRANCH ADDRESS 10 POSITIONS*
      7684                                    0174  *      BC     CRD(0),CODE(0)               HIGHER THAN SUBR1 BRANCH   *
      7684                                    0175  *                                          ADDRESS. ALSO IT IS 10 POS.*
      7684                                    0176  *                                          HIGHER THAN ADDRESS SET IN *
      7684                                    0177  *                                          RPG CONTROL CARD CORE SIZE *
      7684                                    0178  *                                                                     *
      7684                                    0179  ***********************************************************************

      7684                                    0181         ORG    *
      7690  Q001147550 08 0011  1 0 7550  4 0 0182  SUBR2  MC     IR1(14),SAVREG         SAVE REGS 1 & 2
      7700  P00S147541 09 0031  0 0 7541  4 0 0183         MN     IR3,RTN+1              SET RETURN BRANCH
      7710  076X330001 01 7683  0 0 0001  3 0 0184         W      CR(0),1(3)             CARRIAGE RETURN
      7720  P00P6TP021 09 0006  0 3 0021  4 0 0185         MN     6(4,3),IR2   'CODE'    GET REL ADDR OF OP CODE
      7730  4P01140021 04 0011  4 0 0021  4 0 0186         A      IR1,IR2                ADD BASE ADDR TO GET ABSOLUTE ADDR
      7740  P0000Q7819 08 0000  0 2 7819  1 0 0187         MC     0(1,2),MSG3+9          MVOE INVALID CODE TO MESSAGE
      7750  078Q010037 01 7810  0 0 0037  1 0 0188         W      MSG3(0),37(1)          WRITE ERROR MESSAGE ON WRKST
      7760  076X330001 01 7683  0 0 0001  3 0 0189         W      CR(0),1(3)             CR
      7770  P00P1TP021 09 0001  0 3 0021  4 0 0190         MN     1(4,3),IR2   'CARD'    GET REL ADDR OF CARD
      7780  4P01140021 04 0011  4 0 0021  4 0 0191         A      IR1,IR2                ADD BASE REG TO ABSOLUTE ADDR
      7790  P00R147511 09 0021  0 0 7511  4 0 0192         MN     IR2,PRTCRD+1           MOVE ADDR OF CARD TO WRITE INST.
      7800  U7UQ000000 11 7510  5 0 0000  0 0 0193         BC     PRTCRD(5)              WRITE CARD IMAGE ON WRKST
      7810                                    0194  *
      7810  OP CODE -     0001       0037     0195  MSG3   DM     C37'OP CODE - - INVALID. COL 1 CARD BELOW'
      7847                                    0196  *

      7847                                    0198  *
      7847                                    0199  *      END CARD FOLLOWS COMMENTS -*- NOTE: NO 'EXEC' STATEMENT IS
      7847                                    0200  *                                          USED WHEN RPG IS CALLING
      7847                                    0201  *                                          THE PROGRAM
      7847                                    0202  *
      7847                                    0203  *      AFTER ASSEMBLY OF THIS PROGRAM, FILE THE OBJECT INTO THE
      7847                                    0204  *       RPG OUTPUT FILE. THEN COMPILE THE RPG PROGRAM. THE RPG
      7847                                    0205  *       COMPILER WILL THEN CONCATENATE THE RPG OBJECT WITH THE
      7847                                    0206  *       ASSEMBLER OBJECT.
      7847                                    0207  *

      7847                                    0209         END
```

Figure B.8 LISTING OF EXIT SUBROUTINE SUBR2

Appendix C
RPG SOURCE CODE DIAGNOSTICS

APPENDIX C: RPG SOURCE CODE DIAGNOSTICS

DIAGNOSTIC ERROR MESSAGES

The diagnostic error messages produced by the RPG compiler are printed in the compiler output immediately after the line in the source listing where the error occurred.

Format:

```
****ERROR nn****  COLUMN cc  error message
```

where

nn   is the error reference number and

cc   is the column number at or near the RPG language element which was found to be in error.

A typical diagnostic is shown below.

```
2     01020FCSTLST  IPEA      94           DISC   RPGTST CSTLS1         SA0135
3     01030FPRNT    0        0132     OF     PRINTER     2              SA0100
 **** ERROR 68 ****  COLUMN 33  INVALID INDICATOR SPECIFIED
4     02010ICSTLST  AAN 01   1 CI   2 CM   3 CN                        SA0100
 **** ERROR 00 ****  COLUMN 17  SYNTAX REQUIRES THIS COLUMN TO BE BLANK
```

Figure C.1   EXAMPLE OF SOURCE CODE DIAGNOSTICS

NOTE: In the above example, 'Error 68' results because the Overflow Indicator 'OF' in columns 33-34 had a zero rather than an alphabetic 'O' in column 33.

APPENDIX C: RPG SOURCE CODE DIAGNOSTICS

The full list of compiler error messages and their meanings is as follows.

Table C-1 RPG SOURCE CODE DIAGNOSTIC ERROR MESSAGES

REFERENCE NUMBER	SC	ERROR MESSAGE	MEANING
00	W	SYNTAX REQUIRES THIS COLUMN TO BE BLANK	Nonblank data was found in a column which should be blank.
01	T	HEADER CARD OMITTED	The header card was omitted or out of order.
02	T	INPUT SPECIFICATION CARDS OMITTED	No input specification cards were found for some File Description cards.
03	W	INVALID CORE SIZE	The Header card contained invalid characters in the program size field.
04	W	INVALID OVERFLOW LINE NUMBER	The header card contained invalid characters in the overflow line field or specified line 0.
05	T	OUTPUT CARDS OMITTED	The Output Format Specification cards were omitted or out of order.
06	T	MULTIPLE READER FILES ENCOUNTERED	Only one card reader input file is permitted.
07	T	MULTIPLE PRINTER FILES ENCOUNTERED	Only one line printer output file is permitted.
08	T	MULTIPLE PUNCH FILES ENCOUNTERED	Only one card punch output file is permitted.
09	T	DUPLICATE FILE NAME	Two or more File Description cards contained the same file name.

Note: "SC" stands for Severity Code which has two possible values:

W - Warning

The object program should execute; however, the result of execution may not be consistent with the source program specifications.

T - Termination

The object program will not execute. If execution is attempted, the results are unpredictable.

If the "compile-and-go" option has been specified (available with 10K compiler only), all diagnostic messages <u>except Reference No. 00</u> will suppress execution of the object program.

Table C-1 RPG SOURCE CODE DIAGNOSTIC ERROR MESSAGES (continued)

REFERENCE NUMBER	SC	ERROR MESSAGE	MEANING
10	T	FILE DESIGNATION INVALID OR OMITTED	The File Designation column on the File Description card was left blank or a character other than 'P' or 'S' was found.
11	W	MULTIPLE PRIMARY FILES ENCOUNTERED	The File Description cards designated more than one primary file or the first Input-File Description card was not designated as primary.
12	W	INVALID END OF FILE CHARACTER	The End-Of-File column on the File Description Card did not contain a blank or 'E'.
13	W	FILE SEQUENCE INVALID OR OMITTED	The sequence column of the File Description was blank or did not contain an 'A' or 'D' when matching fields were specified.
14	T	RECORD LENGTH INVALID OR OMITTED	The Record Length Field of the File Description card was blank, the number was not right-justified, contained illegal characters or it was specified as zero.
15	T	FILE TYPE INVALID OR OMITTED	The File Type field of the File Description card was blank or contained other than an 'O' or 'I' or I and O types were mixed.
16	T	RECORD LENGTH IS INVALID FOR THE SPECIFIED DEVICE	The length specified exceeds the maximum allowed for the device or was specified as zero.
17	T	MAIL CODE OMITTED	The file description card device entry specified common and a required mail code was not found.
18	T	INPUT FILE DESCRIPTION CARDS OMITTED	No input files were described by the File Description cards or the File Description cards were out of order, i.e., they did not immediately follow the Header card.
19	W	OUTPUT FILE DESCRIPTION CARDS OMITTED	The File Description cards did not describe at least one output file.
20	T	DEVICE TYPE INVALID OR OMITTED	The device specified on the File Description card contained invalid characters, was not left-justified, is not supported by RPG, or the field was left blank.
21	T	POOL NAME INVALID OR OMITTED	The pool name field on the File Description card was left blank, contained illegal characters or was not left-justified in the field.
22	T	FILE NAME INVALID OR OMITTED	The file name field was left blank, contained illegal characters or was not left-justified in the field.
23	T	UNIT NUMBER INVALID OR OMITTED IOC DEVICE	The unit number field on the File Description card was left blank or contained an illegal character.
24	T	FILE DESCRIPTION INCORRECT OR OMITTED	The file name encountered when processing an input specification or an output-format card had an incorrect file designation, i.e., input specifications were found for a file designated as output, or the file description cards for this file were omitted.

"SC" means "Severity Code".

W - Warning

T - Termination

Table C-1 RPG SOURCE CODE DIAGNOSTIC ERROR MESSAGES (continued)

REFERENCE NUMBER	SC	ERROR MESSAGE	MEANING
25	W	RECORD SEQUENCE INVALID OR OMITTED	The sequence field on the input specification card was left blank or the sequence field was numeric and 01 was not encountered first, numeric specification was out of sequence (not ascending) or the field contained alphabetic characters after a numeric specification was encountered.
26	W	INVALID NEGATE CODE	A 'not' field was not blank and did not contain 'N'.
27	W	RECORD NUMBER INVALID OR OMITTED	The number field on the input specification card was required and left blank or contained a character other than '1' or 'N'.
28	W	INVALID RECORD OPTION	The option field on the input specification card was not blank and did not contain 'O'.
29	W	INCORRECT INDICATOR USAGE	The usage of the specified indicator is incorrect. (See indicator usage table).
30	W	INVALID HOLLERITH CODE	The Hollerith indicator field of the Input Specifications card was not blank or did not contain an 'H'.
31	W	INVALID DECIMAL POSITION SPECIFICATION	The Decimal Positions Field did not contain a blank or 0-9.
32	W	ILLEGAL HOLLERITH CONVERSION	Hollerith conversion was specified for a field defined as alphanumeric.
33	T	FIELD LOCATION INVALID OR OMITTED	The 'From' and/or 'To' field on the Input Specifications card was left blank, not right-justified, contained illegal characters or the fields were not in ascending order.
34	T	INVALID FIELD LENGTH	*Implicit or explicit length specification exceeds 100 for an alphanumeric field or 18 for a numeric field. The length specification on the Calculation Card contains invalid characters or is not right-justified.
35	T	INVALID MATCH FIELD	The Matching Field columns on the Input Specification card was not blank and contained characters other than M1-M9.
36	T	DUPLICATE MATCH FIELD	The same Match Field Identifier was specified for more than one field within the same Record Identification specification.
37	T	FIELD NAME INVALID OR OMITTED	The field name columns on the Input Specification card or Output-Format card were left blank when the field location fields or end position columns defined a field, the field name was not left-justified or the field name contained illegal characters. The factor/result columns on the calculation specification card were left blank, and were required by the operation specified, the field name was not left-justified or contained illegal characters.
38	T	DUPLICATE FIELD NAME	The same field name was specified for more than one field within the same Record Identification specification.

*'IMPLICIT LENGTH' refers to the lenght of a numeric field after alignment.

"SC" means "Severity Code".

W - Warning

T - Termination

Table C-1 RPG SOURCE CODE DIAGNOSTIC ERROR MESSAGES (continued)

REFERENCE NUMBER	SC	ERROR MESSAGE	MEANING
39*	T	*PLACE SPECIFICATION INVALID OR OUT OF SEQUENCE	No fields or constants preceded the *PLACE specification or the fields preceding the *PLACE had errors.
40	W	INCORRECT MATCH FIELD USAGE	The concatenation of the Match Fields for each record defined resulted in different lengths or the resultant length exceeded 100 characters.
41	W	DUPLICATE TAG	The Factor 1 columns of the Calculation card specified a TAG name that was used on a previous 'TAG'.
42	W	RECORD IDENTIFICATION CHARACTER INVALID OR OMITTED	The Character column on the Input Specifications card was left blank or contained other than a 'C'.
43	T	UNDEFINED FIELD	The Field Name encountered on the Calculation Specifications card or the Output-Format card references a field which was not defined by an input specification or a previous Result Field.
44	W	INVALID HALF ADJUST CODE	The Half-Adjust Column on the Calculation Specifications card was not blank and contains a character other than an 'H'.
45	W	ILLEGAL HALF ADJUST SPECIFIED	Half-Adjust was specified for a field defined as alphanumeric.
46	T	INVALID EXIT NAME	The EXIT operation was specified and the result columns on the Calculation Specifications card were left blank, the name was not left-justified, or it contained illegal characters.
47	T	RLABL OUT OF SEQUENCE	The RLABL operation(s) was not preceded by an EXIT operation.
48	W	RESULTING INDICATORS OMITTED	The Resulting Indicators field of the Calculation Specifications were left blank for an operation which requires a Resulting Indicator specification.
49	T	TAG NAME INVALID OR OMITTED	A TAG operation was specified on the calculation specification card and the Factor 1 field was left blank, the name was not left-justified in the field or the name contained illegal characters.
50	T	ILLEGAL TAG NAME SPECIFIED	The name defined as a TAG was previously defined as a field.
51*	W	DSPLY DEVICE NUMBER INVALID OR OMITTED	Factor 2 of a DSPLY operation was left blank or did not contain a valid IOC device number.
52	T	ILLEGAL LENGTH FOR MULTIPLY	Factor 1 or Factor 2 length is greater than 10.
53	T	ILLEGAL LENGTH FOR DIVIDE	The length of Factor 2 is greater than 10 or the length of Factor 1 after adjustment exceeds 18.

* 10K Compiler Message Only

"SC" means "Severity Code"

W - Warning

T - Termination

Table C-1 RPG SOURCE CODE DIAGNOSTIC ERROR MESSAGES (continued)

REFERENCE NUMBER	SC	ERROR MESSAGE	MEANING
54	T	OPERATION REQUIRES A NUMERIC FACTOR	An alphanumeric field was specified for Factor 1/Factor 2 or Result field for an arithmetic operation.
55	T	OPERATION INVALID OR OMITTED	The operation field of a Calculation Specification card was not left-justified, contains illegal characters or is not supported. (See Table of RPG operations.)
56	T	FORMAT TYPE INVALID OR OMITTED	The Type column on the Output-Format card did not contain an 'H', 'D', or 'T'.
57	W	INVALID SPACE VALUE	The Space Before/After column was not left blank and did not contain a numeric value.
58	W	INVALID SKIP VALUE	The Skip Before/After column was not left blank and did not contain a numeric value.
59	T	END POSITION INVALID OR OMITTED	The End Position columns on the Output-Format card was left blank, the end position was not right-justified in the field or the field contained non-numeric data.
60	T	END POSITION EXCEEDS RECORD LENGTH	The end position columns on the Output-Format card specified an end position that was greater than the Record Length specified for the file on the File Description card.
61	T	INVALID EDIT CODE	The Edit Code column on the Output-Format card was not blank and did not contain a 'Z'.
62	T	ILLEGAL EDIT SPECIFICATION	The Edit Code column specified a 'Z' and an edit word was also specified or the Field Name columns of the Output-Format specification card specified a field defined as alphanumeric and an Edit operation was specified.
63	W	INVALID BLANK AFTER	The Blank After column on the Output-Format specification card was not left blank and contained a character other than 'B'.
64	T	INVALID CONSTANT OR LITERAL	The constant specified was not left-justified in the field or a beginning or ending apostrophe was omitted.
65	T	INVALID CONSTANT OR EDIT WORD	The constant columns on the Output-Format specification card contained unbalanced apostrophes, or the Constant or Edit Word was not left-justified in the field.
66	T	CARD TYPE SPECIFICATION INVALID OR OUT OF SEQUENCE	The Card Type column was left blank or contained a character other than 'H', 'F', 'I', 'C', or 'O'. The RPG program cards were not in standard RPG order of occurrence.

"SC" means Severity Code

W - Warning

T - Termination

Table C-1 RPG SOURCE CODE DIAGNOSTIC ERROR MESSAGES (continued)

REFERENCE NUMBER	SC	ERROR MESSAGE	MEANING
67	T	INCORRECT CALCULATION CARD SEQUENCE	The Calculation Specifications must specify detail calculations before total calculations.
68	T	INVALID INDICATOR SPECIFIED	An invalid RPG indicator was specified.
69	T	CONFLICTING MATCH FIELD LENGTH	The length for the specified Match Field was different than the length of the identical Match Field defined for a previous record.
70	W	CONFLICTING MATCH FIELD DATA TYPE	The Field Data Type attribute was different than the Data Type attribute specified for the identical match field for a previous record.
71*	W	INVALID FIELD REDEFINITION	A field had a length or decimal specification which was not consistent with the field as originally defined.
72	T	CONFLICTING CONTROL BREAK LENGTH	The control level indicator used for this field was used for a field of different length in a previous record.
73	T	CONFLICTING CONTROL BREAK DATA TYPE	The control level indicator used for this field was used for a field of a different data type in a previous record.
74*	T	END POSITION SPECIFIED IS LESS THAN FIELD LENGTH	The length of the field exceeded the end position specified in the output-format specification resulting in overlapping the left end of the output buffer.
75*	T	FACTOR 1 AND/OR RESULT MUST BE SPECIFIED.	Factor 1 and the result field of a DSPLY operation were left blank.
76*	W	FIELDS WITH NO FIELD RECORD RELATION SHOULD COME FIRST	In the preceding record description, a field without a field record relation indicator was encountered after fields with field-record relation were processed.
77*	T	TOTAL MATCH/CONTROL FIELD LENGTH EXCEEDS 100	In the preceding record description, the sum of the lengths of all match fields (M1-M9) exceeded 100 characters, or the sum of the lengths of a split control break field, e.g., L1, exceeded 100 characters.
99	T	UNDEFINED DETAIL TAG TOTAL	A GOTO operation specified a 'tag name' which was not found, or an attempt was made to jump from the detail calculations to the total calculations or vice versa.
99*	T	UNDEFINED TAG	A GOTO operation specified a "tag name" which was not found.

* 10K Compiler Message Only

"SC" means "Severity Code".

W - Warning

The object program should execute; however, the result of execution may not be consistent with the source program specifications.

T - TERMINATION

The object program will not execute. If execution is attempted, the results are unpredictable.

Appendix D
COMMON CORE CONVENTIONS

APPENDIX D: COMMON CORE CONVENTIONS

The System Ten core memory includes up to twenty partition areas and one common area.

The core memory in each partition is hardware protected from all other partitions. The core memory of common is accessible to all partition programs as each gets its timed share of control of the computer. Since there is no hardware protection of common core, orderly control of this area is obtainable only through a set of Common Core Conventions. The System Ten RPG compiler conforms with these conventions.

The areas in common are allocated as follows:

AREA	COMMON LOCATIONS	SIZE	USE
1	0300 - 0399	100	System Constants.
2	0400 - 0559	160	Reserved for System Ten software use only.
3	0560 - 0579	20	Partition Status Flags.
4	0580 - 0999	420	System Mailbox.
5	1000 - 9999	0000-9000*	User and Application Defined Area.

*The size of the "user" area varies with the size of common core. A minimum configuration of 1K will provide zero characters of user area while a maximum common configuration of 10K will provide a capacity of 9000 characters.

Of particular interest to the RPG programmer are the System Constants area (0300 - 0399) and System Mailbox (0580 - 0999).

System Constants

The layout of the System Constants area is as follows:

Location	Format	Description
0300-0305	yymmdd	Current Calendar Date (record storage use).
0306-0313	mm/dd/yy	Current Calendar Date (display us).
0314-0316	jjj	Julian Date (file header use).
0317	d	Day of Week (1-7).
0318-0399		Reserved

System Mailbox

This area provides the system with a buffer which can be used to "mail" records from one partition to another. This buffer may be used by any software or application program. Control of the buffer is maintained automatically by RPG programs and/or standard Assembler language routines (see the coding sheets at the end of this appendix). These Assembler routines are designed so that the entire buffer area may be filled and emptied without control passing from one partition to another. Thus, the responsibility for keeping the buffer "open" is placed on both the sending and receiving programs.

Only two RPG programs can use the mailbox at one time. Also, only one record can be sent via the mailbox at one time. The RPG program sending records through the common mailbox will automatically send a common end-of-file (@@@@@@) when its LR Indicator is turned on and it closes its files.

The first five locations of the mailbox (0580 - 0584) contain the following:

Locations	Contents
0580	Mailbox Flag A = Available B = Busy
0581-0582	Mail To Address The number or symbolic name of the partition to which the record will be sent.
0583-0584	Mail From Address The number or symbolic name of the partition from which the record is received.

It is the user's responsibility to insure that the Common Mailbox locations are properly initialized to allow transfer of data between partitions and that the partitions that are to participate in the data exchange are properly loaded with the requisite programs. This is best accomplished via the System Ten multipartition loader facility.

The RPG object program will appear to be "idle" if the proper communication is not available as defined by the file description cards describing the common devices.

To initialize the Common Mailbox Flag, the user should do the following:

1. Obtain a load condition on device 0.

2. Enter '0058P10001'

3. The device will then be selected for input.

4. Enter the single character 'A'.

5. The Common Mailbox Flag will now be set to 'A' (available).

When an RPG program specifies COMMON as the input or output device for a file (see columns 40-46 of the File Description Specifications Form), the Mail To address must be entered in columns 51-52 and the Mail From address must be entered in columns 58-59 of that form.

The remainder of the Common Mailbox area, locations 0585-0999, may be used for transmitting records with a record length up to 415 characters.

The routines shown at the end of this appendix illustrate how the buffer is to be accessed and released in Assembler language programs. The sequence of instructions shown there must be followed to avoid uncontrolled partition switching.

ASSEMBLER CODING FORM

PROGRAM: SYSTEM TEN MAILBOX DEVICE REFERENCES

PROGRAMMER: | EXTENSION: | DATE: | PUNCHING INSTRUCTIONS: GRAPHIC / PUNCH | PAGE OF

```
*      SYSTEM MAILBOX DEVICE CONSTANTS
       COMMON
       ORG    0580C
MBFLAG DM     C1          MAILBOX FLAG
MBTO   DM     C2          MAILBOX 'TO' ADDRESS
MBFROM DM     C2          MAILBOX 'FROM' ADDRESS
MBREC  DM     C415        MAILBOX RECORD




       NORMAL
FLAGA  DM     C'A'               VALUE OF 'A'
FLAGB  DM     C'B'               VALUE OF 'B'
IDR    DM     C'BSW'             OWN IDENTIFICATION FOR READ
IDW    DM     C'BSWNN'           IDENTIFICATION FOR WRITE
```

Figure D.1 SYSTEM MAILBOX DEVICE CONSTANTS

ASSEMBLER CODING FORM

PROGRAM: SYSTEM TEN MAILBOX DEVICE READ WRITE ROUTINES

PROGRAMMER | EXTENSION | DATE | PUNCHING INSTRUCTIONS: GRAPHIC / PUNCH | PAGE OF

```
*                      SINGLE PURPOSE MAILBOX READ ROUTINE
READRT BC     *+10(8)                    READ RETRY - SWITCH & FALL THRU
READMB C      IDR(3),    MBFLAG          IS MAIL FOR ME
       BC     READRT(3), READRT(1)       N - GO TO READ RETRY
       MC     MBREC(N),  OWNMEM          Y - GET RECORD (NNN CHAR)
       MC     FLAGA(1),  MBFLAG            - RELEASE MAILBOX




*                      SINGLE PURPOSE MAILBOX WRITE ROUTINE
WRITMB C      FLAGA(1),  MBFLAG          IS MAILBOX AVAILABLE
       BC     WRITRT(1), WRITRT(3)       N - GO TO WRITE RETRY
       MC     IDW(5),    MBFLAG          Y - SET MAIL ADDRESSES
       MC     OWNREC(N), MBREC               PUT RECORD (NNN CHAR)
       BC     *+10(8)                        SWITCH & FALL THRU
       C      IDW(5),    MBFLAG          WAS RECORD READ
       BC     NXTRTN(3), NXTRTN(1)       Y - GO TO NEXT ROUTINE
       MC     FLAGA(1),  MBFLAG          N - RELEASE MAILBOX
WRITRT BC     WRITMB(8)                      SWITCH, GO TO WRITE AGAIN
```

Figure D.2 SYSTEM MAILBOX DEVICE READ/WRITE ROUTINES

Appendix E
SAMPLE RPG PROGRAM

APPENDIX E: SAMPLE RPG PROGRAM

The following pages show the RPG specifications, source program listing, data cards and output for a simple card listing program which totals the balance owed by a number of fictitious "customers". Ten data cards were submitted. The seventh card (belonging to "Alexander the Great") lacks a proper identification code, an 'A' in column 1, causing a message to be printed:

THIS CARD HAS AN INVALID IDENTIFICATION CODE.

The amount on that card is not included in the total.

RPG CONTROL CARD AND FILE DESCRIPTION SPECIFICATIONS

Date ____________

Program ____________

Programmer ____________

Punching Instruction	Graphic							
	Punch							

Page (1 2) ____

Program Identification (75 76 77 78 79 80) ____

Control Card Specifications

Line (3-5)	Form Type (6)	NOT USED (7-11)	Core Size to Execute (12-14)	NOT USED (15-26)	Printer Line Count for Object Program (if other than 60) (27-28)	Comments (29-74)
010	H		000			

File Description Specifications

Line (3-5)	Form Type (6)	Filename (7-12)	NOT USED (13-14)	I/O File Type (15)	P/S File Designation (16)	End of File (17)	A/D Sequence (18)	NOT USED (19-23)	Record Length (24-27)	NOT USED (28-32)	Overflow Indicator (33-34)	NOT USED (35-39)	Device (40-46)	Pool Name (Disc) / Mail to (Common) / Device Number (47-52)	NOT USED (53)	File Name (Disc) / Mail From (Common) / (Leave blank for all other devices) (54-59)
020	F	IN		I	P	E			80				READER	1		
030	F	ØUT		Ø					132		ØV		PRINTER			
04	F															
05	F															
06	F															
07	F															

011 * SAMPLE PRØGRAM - READ A CARD INCLUDING CUSTØMER'S

012 * BALANCE. LIST INFØ ØN CUSTØMER AND TØTAL BALANCE.

FORM 40-344 (OBSOLETES FORM 40-254)

SINGER BUSINESS MACHINES

Figure E.1 SAMPLE RPG PROGRAM -- CONTROL CARD AND FILE DESCRIPTION SPECIFICATIONS

RPG INPUT SPECIFICATIONS

10K Compiler Feature

Date ____________
Program ____________
Programmer ____________

Punching Instruction	Graphic							
	Punch							

Page 1 2

Program Identification 75 76 77 78 79 80

Line	Form Type	Filename	AND / OR: Sequence	Number (1/N)	Option (O)	Record Indentifying Indicators	Record Identification Codes 1: Position	1: Not (N)	1: Type (C/Z/D)	1: Character	2: Position	2: Not (N)	2: Type (C/Z/D)	2: Character	3: Position	3: Not (N)	3: Type (C/Z/D)	3: Character	Hollerith Indicator	Field Location: From	Field Location: To	Decimal Positions	Field Name	Control Level (L1-L9)	Matching Fields	Field-Record Relation	Field Indicators: Plus	Field Indicators: Minus	Field Indicators: Zero or Blank
3 4 5	6	7 8 9 10 11 12	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	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
010	I	IN	AA			01	1		C	A																			
020	I																			2	5	0	ACCTNØ						
030	I																			6	30		NAME						
040	I																			31	50		ADDRES						
050	I																			51	65		CITY						
060	I																			66	70		STATE						
070	I																			71	75	0	ZIP						
080	I																			76	80	2	AMTØWD						
090	I		BB			02	1	N	C	A																			
10	I																												
11	I																												
12	I																												
13	I																												
14	I																												
15	I																												
	I																												
	I																												
	I																												
	I																												
	I																												

FORM 40-342 (OBSOLETES FORM 40-253)

SINGER
BUSINESS MACHINES

Figure E.2 SAMPLE RPG PROGRAM -- INPUT SPECIFICATIONS

RPG CALCULATION SPECIFICATIONS

10K Compiler Feature

Date ______
Program ______
Programmer ______

Punching Instruction	Graphic							
	Punch							

Page 1 2 [ ][ ]

Program Identification 75 76 77 78 79 80 [ ][ ][ ][ ][ ][ ]

Line (3-5)	Form Type (6)	Control Level (L0-L9,LR) (7-8)	Indicators (9-17)	Factor 1 (18-27)	Operation (28-32)	Factor 2 (33-42)	Result Field (43-48)	Field Length (49-51)	Decimal Positions (52)	Half Adjust (H) (53)	Resulting Indicators: Arithmetic Plus / Compare High F1>F2 (54-55)	Minus / Low F1<F2 (56-57)	Zero / Equal F2=F1 (58-59)	Comments (60-74)
010	C		01	AMTØWD	ADD	TØTAL	TØTAL	7	2					TØTAL AMT ØWED
02	C													
03	C													
04	C													
05	C													
06	C													
07	C													
08	C													
09	C													
10	C													
11	C													
12	C													
13	C													
14	C													
15	C													
	C													
	C													
	C													
	C													
	C													

FORM 40-343 (OBSOLETES FORM 40-252)

SINGER BUSINESS MACHINES

Figure E.3 SAMPLE RPG PROGRAM -- CALCULATION SPECIFICATIONS

3/72

RPG OUTPUT - FORMAT SPECIFICATION

10K Compiler Feature

Date ____________
Program ____________
Programmer ____________

Punching Instruction	Graphic	
	Punch	

Page 1 2

Program Identification 75 76 77 78 79 80

Line (3-5)	Form Type (6)	Filename (7-12)	NOT USED (13)	AND/OR (14)	Type (H/D/T/E) (15)	(16)	Space Before (17)	Space After (18)	Skip Before (19-20)	Skip After (21-22)	Output Indicators (23-31)	Field Name (32-37)	Edit Codes (38)	Blank After (B) (39)	End Position in Output Record (40-43)	Hollerith Indicator (44)	Constant or Edit Word (45-70)
010	Ø	ØUT			H			3	03		ØV						
020	Ø			Ø	R						1P						
030	Ø														61		'S I M P L E C A R D'
040	Ø														77		'LISTING'
050	Ø				H			3			ØV						
060	Ø			Ø	R						1P						
070	Ø														9		'ACCT NØ.'
080	Ø														23		'NAME'
090	Ø														56		'ADDRESS'
100	Ø														77		'CITY'
110	Ø														94		'STATE'
120	Ø														102		'ƵIP'
130	Ø														121		'BALANCE ØWED'
140	Ø				D			2			01						
150	Ø											ACCTNØ			7		
160	Ø											NAME			39		
170	Ø											ADDRES			64		
180	Ø											STATE			94		
190	Ø											ƵIP			104		
200	Ø														110		'$'

Edit Codes: Ƶ = Zero Suppress; b̸ = Use Edit Word or No Editing

FORM 40-341 (OBSOLETES FORM 40-251)

SINGER BUSINESS MACHINES

Figure E.4 SAMPLE RPG PROGRAM -- OUTPUT FORMAT SPECIFICATIONS

RPG OUTPUT - FORMAT SPECIFICATION

▒ 10K Compiler Feature

Date ______
Program ______
Programmer ______

Punching Instruction	Graphic							
	Punch							

Page 1 2

Program Identification 75 76 77 78 79 80

Line (3-5)	Form Type (6)	Filename (7-12)	NOT USED (13)	AND/OR Type (H/D/T/E) (14-16)	Space Before (17)	Space After (18)	Skip Before (19-20)	Skip After (21-22)	Output Indicators (23-31)	Field Name (32-37)	Edit Codes (38)	Blank After (B) (39)	End Position in Output Record (40-43)	Hollerith Indicator (44)	Constant or Edit Word (45-70)
010	Ø									AMTØWD			122		' 0.00-'
020	Ø			D		2			02						
030	Ø												18		'THIS CARD HAS AN'
040	Ø												43		'INVALID IDENTIFICATION'
050	Ø												49		'CØDE.'
060	Ø			T		2			LR						
070	Ø												122		'---------------'
080	Ø			T				01	LR						
090	Ø												110		'TØTAL BALANCE ØWED $'
100	Ø									TØTAL			122		' , 0.00-'
11	Ø														
12	Ø														
13	Ø														
14	Ø														
15	Ø														
005	Ø									CITY			84		
	Ø														
	Ø														
	Ø														
	Ø														

z = Zero Suppress
b = Use Edit Word or No Editing

FORM 40-341 (OBSOLETES FORM 40-251)

SINGER BUSINESS MACHINES

Figure E.5 SAMPLE RPG PROGRAM -- OUTPUT FORMAT SPECIFICATIONS

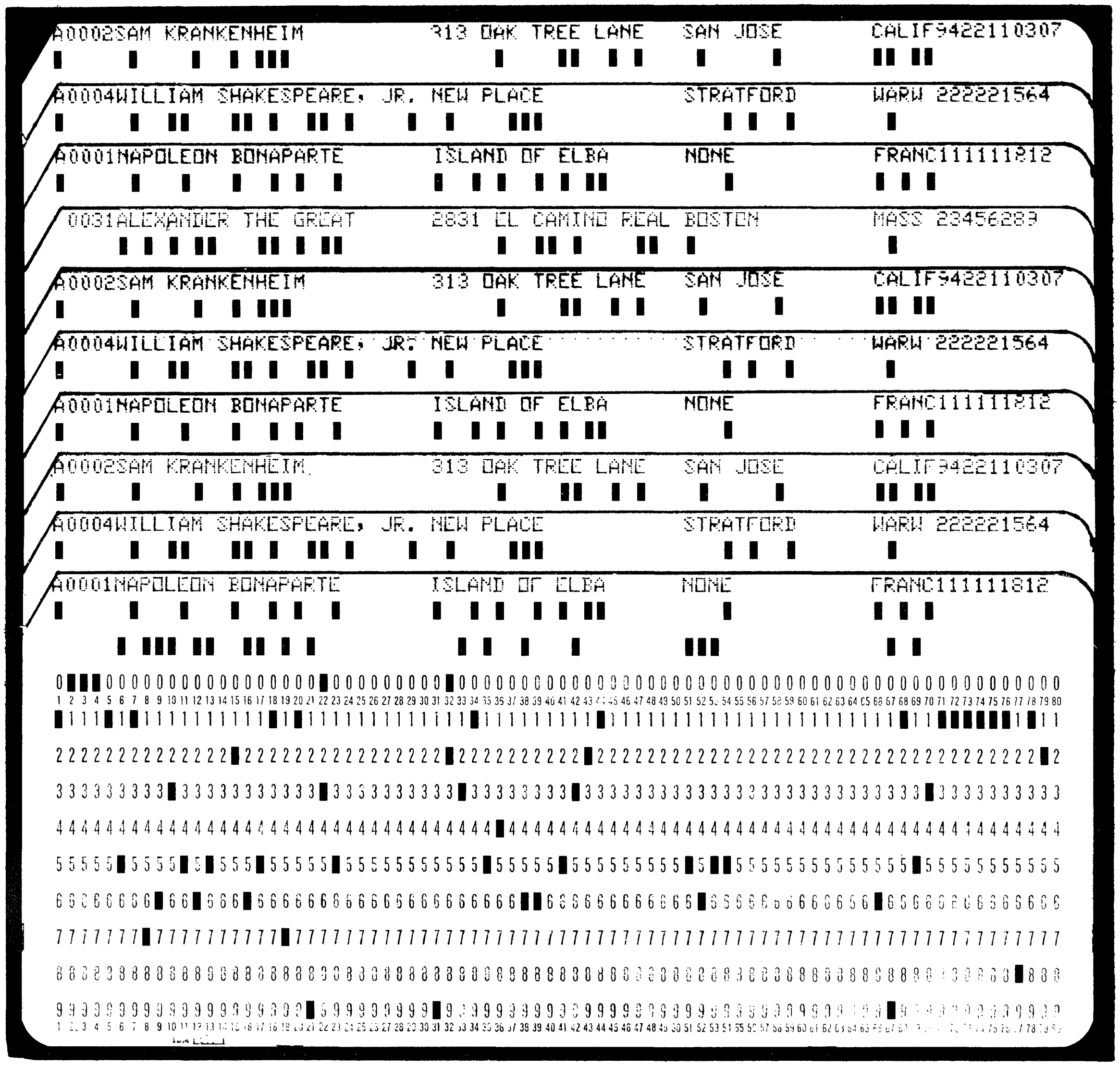


Figure E.6 SAMPLE RPG PROGRAM – INPUT DATA CARDS

```
V003    *REPORT*PROGRAM*GENERATOR*      SOURCE/DIAGNOSTIC LISTING      MHR001  09/20/71   PAGE  1

   1     01010H      000                                                                  MHR001
   2     01011F* SAMPLE PROGRAM - READ A CARD INCLUDING CUSTOMER'S                        MHR001
   3     01012F* BALANCE. LIST INFO ON CUSTOMER AND TOTAL BALANCE.                        MHR001
   4     01020FIN        IPE         80            READER      1                          MHR001
   5     01030FOUT       O          132      OV    PRINTER     2                          MHR001
   6     02010IIN        AA   01     1 CA                                                 MHR001
   7     02020I                                            2   50ACCTNO                   MHR001
   8     02030I                                            6   30 NAME                    MHR001
   9     02040I                                           31   50 ADDRES                  MHR001
  10     02050I                                           51   65 CITY                    MHR001
  11     02060I                                           66   70 STATE                   MHR001
  12     02070I                                           71   750ZIP                     MHR001
  13     02080I                                           76   802AMTOWD                  MHR001
  14     02090I          BB   02     1NCA                                                 MHR001
  15     03010C    01       AMTOWD     ADD  TOTAL      TOTAL   72       TOTAL AMT OWED    MHR001
  16     040100OUT       H  303     OV                                                    MHR001
  17     040200        OR           1P                                                    MHR001
  18     040300                                      61 ' S I M P L E    C A R D'         MHR001
  19     040400                                      77 '    LISTING'                     MHR001
  20     040500          H   3      OV                                                    MHR001
  21     040600        OR           1P                                                    MHR001
  22     040700                                       9 'ACCT NO.'                        MHR001
  23     040800                                      23 'NAME'                            MHR001
  24     040900                                      56 'ADDRESS'                         MHR001
  25     041000                                      77 'CITY'                            MHR001
  26     041100                                      94 'STATE'                           MHR001
  27     041200                                     102 'ZIP'                             MHR001
  28     041300                                     121 'BALANCE OWED'                    MHR001
  29     041400          D   2      01                                                    MHR001
  30     041500                           ACCTNO      7                                   MHR001
  31     041600                           NAME       39                                   MHR001
  32     041700                           ADDRES     64                                   MHR001
  33     041800                           STATE      94                                   MHR001
  34     041900                           ZIP       104                                   MHR001
  35     042000                                     110 '$'                               MHR001
  36     050050                           CITY       84                                   MHR001
  37     050100                           AMTOWD    122 '   0.00-'                        MHR001
  38     050200          D   2      02                                                    MHR001
  39     050300                                      18 'THIS CARD HAS AN'                MHR001
  40     050400                                      43 'INVALID   IDENTIFICATION'        MHR001
  41     050500                                      49 'CODE.'                           MHR001
  42     050600          T   2      LR                                                    MHR001
  43     050700                                     122 '--------------'                  MHR001
  44     050800          T       01 LR                                                    MHR001
  45     050900                                     110 'TOTAL BALANCE OWED  $'           MHR001
  46     051000                           TOTAL     122 '  ,   0.00-'                     MHR001
```

Figure E.7 SAMPLE RPG PROGRAM – SOURCE/DIAGNOSTIC LISTING

```
V003    *REPORT*PROGRAM*GENERATOR*          OBJECT PROGRAM MAP        MHR001  09/20/71    PAGE  2


                                   **** INDICATORS ****

          ADDR   IN            ADDR   IN            ADDR   IN            ADDR   IN            ADDR   IN

          2080   L0            2081   LR            2082   L9            2083   L8            2084   L7
          2085   L6            2086   L5            2087   L4            2088   L3            2089   L2
          2090   L1            2091   1P            2092   H1            2093   H2            2094   H3
          2095   MR            2097   OV            2100   01            2101   02

                                   ***** FIELDS *****

          ADDR   FIELD         ADDR   FIELD         ADDR   FIELD         ADDR   FIELD         ADDR   FIELD

          2275   ACCTNO        2279   NAME          2304   ADDRES        2324   CITY          2339   STATE
          2344   ZIP           2349   AMTOWD        2354   TOTAL


         ****** ALLOCATION MAP ******

         PROGRAM ENTRY            0340
         FIELD BASE ADDRESS       2080
         COMMUNICATION AREA       2260


         *** COMPILATION STATISTICS ***

         PROGRAM SIZE        3,090
         SPECIFIED SIZE     10,000
         SOURCE FILE - RPGPOL.TEMP
         OBJECT FILE - RPGPOL.RPGOBJ

         NO DIAGNOSTICS LISTED


RPG COMPILATION COMPLETED
```

Figure E.8 SAMPLE RPG PROGRAM – OBJECT PROGRAM MAP

S I M P L E C A R D LISTING

ACCT NO.	NAME	ADDRESS	CITY	STATE	ZIP	BALANCE OWED
0001	NAPOLEON BONAPARTE	ISLAND OF ELBA	NONE	FRANC	11111	$ 18.12
0004	WILLIAM SHAKESPEARE, JR.	NEW PLACE	STRATFORD	WARW	22222	$ 15.64
0002	SAM KRANKENHEIM	313 OAK TREE LANE	SAN JOSE	CALIF	94221	$ 103.07
0001	NAPOLEON BONAPARTE	ISLAND OF ELBA	NONE	FRANC	11111	$ 18.12
0004	WILLIAM SHAKESPEARE, JR.	NEW PLACE	STRATFORD	WARW	22222	$ 15.64
0002	SAM KRANKENHEIM	313 OAK TREE LANE	SAN JOSE	CALIF	94221	$ 103.07
THIS CARD HAS AN INVALID IDENTIFICATION CODE.						
0001	NAPOLEON BONAPARTE	ISLAND OF ELBA	NONE	FRANC	11111	$ 18.12
0004	WILLIAM SHAKESPEARE, JR.	NEW PLACE	STRATFORD	WARW	22222	$ 15.64
0002	SAM KRANKENHEIM	313 OAK TREE LANE	SAN JOSE	CALIF	94221	$ 103.07

				TOTAL BALANCE OWED		$ 410.49

Figure E.9 SAMPLE RPG PROGRAM – GENERATED OUTPUT REPORT

Appendix F

RPG DEBUGGING EXAMPLES

APPENDIX F: RPG DEBUGGING EXAMPLES

This section contains examples of some typical debugging procedures used to analyze RPG execution time halts. These examples are intended not as a comprehensive description of System Ten debugging techniques, but as a guide to the RPG object tables. The object tables, which are presented in Section 8, help the user to isolate the source of execution time errors involving file and record manipulation.

Each example is accompanied by a copy of the source diagnostic listing, the object program map, and the core dump for the attempted execution.

EXAMPLE 1. UNIDENTIFIED RECORD ENCOUNTERED

Refer to Figure F.1, F.2, and F.3.

Step 1.

Determine the address of the Halt message from the Error Register, Locations 41-44 (A in Fig. F.3). This is 19Y1 or 1991. Subtract 11.

1991 - 11 = 1980

The Halt message starting in location 1980 is

HALT 4 (See Fig. F.3, B)

Referring to the Halt Error Code Summary (Table 8-1), we find the error description is "An unidentifiable record was encountered." Another indication of the presence of an unidentified record may be obtained from the Communication Area. Referring to the Object Program Map (Fig. F.2) we find the address of the Communication Area is 2170. Characters 4-7 of the Communication Area (C in Fig. F.3) contain zeros. If we check the Communication Area Format (Table 8-2), we find that zeros in characters 4-7 mean "the record type was not identified."

Step 2.

To identify the record in error, locate the Input Buffer via the Program Entry Address. Referring to Figure F.2, we find

Program Entry = 0340

The left-hand end of the input buffer is 20 less than this address.

340 - 20 = 320 = L.H.E. of Input Buffer

Thus we can locate the Input Buffer and the record that had been read at the time of the program halt (D in Figure F.3).

Step 3.

We inspect position 80 of the input record and discover an 'S'. Referring to the source program (Fig. F.1), we find that the only permitted record identification codes are an '&', '-', or '0' in position 80 of the input record. Thus 'S' in position 80 is an unspecified identification code, and causes a program halt.

```
VOOB    *REPORT*PROGRAM*GENERATOR*      SOURCE/DIAGNOSTIC LISTING              08/18/71    PAGE  1

     01010H      000
     01020FCARDIN  IPEA      0080              READER      1                  TEST-1
     01030FLIST    O         0132              PRINTER     2                  TEST-1
   S 01020F*  TEST 1 - TEST RESULTS WHEN AN UNIDENTIFIED RECORD
     01030F*             IS ENCOUNTERED
     01040F*                                                                  TEST-1
     02010ICARDIN  011 01   80 C&                                             TEST-1
     0202 I                                      1    6 EMPNR L1              TEST-1
     0203 I                                      9  142EARN                   TEST-1
     0204 I        021 02   80 C-                                             TEST-1
     0205 I                                      1    6 EMPNR L1              TEST-1
     0206 I                                      9  152YTDERN                 TEST-1
     0207 I        031 03   80 C0                                             TEST-1
     0208 I                                      1    6 EMPNR L1              TEST-1
     0209 I                                      7    7 FNAM                  TEST-1
     0210 I                                      8    8 MNAM                  TEST-1
     0211 I                                      9   19 SNAM                  TEST-1
     0301 OLIST    H  301    1P                                               TEST-1
     0305 O                                31 'TEST 1 - TEST RESULTS'         TEST-1
     0306 O                                55 'WHEN AN UNIDENTIFIED RE'       TEST-1
     0307 O                                74 'CORD IS ENCOUNTERED'           TEST-1
   S 03   O        D 03      03                                               TEST-1
   S 03   O                          EMPNR      6                             TEST-1
   S 03   O                          FNAM      10                             TEST-1
   S 03   O                          MNAM      12                             TEST-1
     040100                          SNAM      24                             TEST-1
     0402 O        T 3                                                        TEST-1
     0403 O                                10 'END OF JOB'                    TEST-1
```

Figure F.1 EXAMPLE 1: UNIDENTIFIED RECORD ENCOUNTERED - SOURCE/DIAGNOSTIC LISTING

```
VOOB    *REPORT*PROGRAM*GENERATOR*          OBJECT PROGRAM MAP                 08/18/71    PAGE  2


                                       **** INDICATORS ****

          ADDR   IN           ADDR   IN           ADDR   IN           ADDR   IN           ADDR   IN

          1990   L0           1991   LR           1992   L9           1993   L8           1994   L7
          1995   L6           1996   L5           1997   L4           1998   L3           1999   L2
          2000   L1           2001   1P           2002   H1           2003   H2           2004   H3
          2005   MR           2010   01           2011   02           2012   03

                                       ***** FIELDS *****

          ADDR   FIELD        ADDR   FIELD        ADDR   FIELD        ADDR   FIELD        ADDR   FIELD

          2185   EMPNR        2191  *EARN         2197   (L1)         2203  *YTDERN       2210   FNAM
          2211   MNAM         2212   SNAM

        ****** ALLOCATION MAP ******

        PROGRAM ENTRY              0340
        FIELD BASE ADDRESS         1990
        COMMUNICATION AREA         2170


        *** COMPILATION STATISTICS ***

        PROGRAM SIZE        2,720
        SPECIFIED SIZE     10,000
        SOURCE FILE - RPGPOL.TEMP
        OBJECT FILE - RPGPOL.RPGOBJ

        NO DIAGNOSTICS LISTED


RPG COMPILATION COMPLETED
```

Figure F.2 EXAMPLE 1: UNIDENTIFIED RECORD ENCOUNTERED – OBJECT PROGRAM MAP

	00	10	20	30	40	50	60	70	80	90
0000	P0PP05040P	1990/////	/1306RPG0B	J1306SFTL	19Y100000	0000050000	0020000015	R0PX050010	P02946001 5	PPR0010037
0100	Q0QU030060	P02P110130	20250135	R020700300	U0PV000000	P020140176	PPP3510038	R0PP050000	0350000000	250000260
0200	S00002720			P0PP000000	00B 08/18	/71		0040		//////
0300	U1YR000000	U2TY000000	COMPAC 00	0531						S
0400	END OF JOB									
0500				3510596	P00S140691	Q0010PP230	P00P1TP031	Q0000PP610	P061010790	P061010620
0600	V0XQ150700	204P010132	201W03P002	1P1205PPQ5	P02S050235	V0XQ150700	PPP15UPPQ0	S0VX050690	P01201P0Q8	U1VT000000
0700	P01301P1U4	RP2S6201U6	R0WX000000	5PPQ5RP1U6	Q0WU050780	QP1R05PPQ5	P01591P1U4	U0WQ000000	2P238201U6	201U43P005
0800	2P1565PPQ5	U0VV000000	PPS9910P01	Q0SP030300	P0031401X4	U0SP000000	P0320601Y5	VP3R8602P1	U0SP000000	PPS20602P7
0900	R0YS000000	P0320602P7	P01201P0Q0	U0SP000000	PPS9910P02	Q0SP030300	P0031401X4	U0SP000000	P0320601Y5	WP3R8702Q3
1000	U0SP000000	PPS20602P7	R1PU000000	P0320602P7	P01201P0Q0	U0SP000000	PPS9910P00	Q0SP030300	P0031401X4	U0SP000000
1100	P0320601Y5	P0326102R0	P0327102R1	Q0328102R2	U0SP000000	PPS20602P7	R1QY000000	P0320602P7	P01201P0Q0	U0SP000000
1200	1032010080	S1RR051230	P01201P1Y4	U0SP000000	0000200820	0860089000	0000002109	4009801010	0000000022	1060110011
1300	5000001240	1200126212	4000000300	01204P0101	3200060000	08TR100000	P01191P400	P040000401	S040020501	U0SP000000
1400	PPQ201P0Q1	Q0SQ030310	V0SP151360	R0003Q0410	R0024S0432	Q0047Y0455	V0PS150540	P1SS200540	P0PP200100	U0SQ000000
1500	PPQ201P0R2	Q0SQ030310	V0SP151360	P01956P400	P02201P409	P02211P411	Q02221P413	V0PS150540	P1SS200540	P0PP200000
1600	U0SQ000000	V0SP151360	Q0066P0400	V0PS150540	P1SS200540	P0PP000003	U0SQ000000	PPP16T0P12	Q0SQ000000	PPP16T0P08
1700	R1XR000000	P0016T0031	2P1924PP08	PPP08T0021	Q1WV000000	P0012T0P08	PPP16T0P08	R0SQ000000	[illegible]008T0031	PPP01QP1S0
1800	R1XQ051710	HALT 2	P0008T0031	PPP00QP1S0	R1XU050310	HALT 3	P0020R01W4	P0000T0031	P01304P1X4	P01884P021
1900	P0006TQ916	V0SP151060	P01804P021	TP1X44PP16	R1YU050310	2P1924P031	1PQR02P1W4	S1X[illegible]00000	HALT 4	1000000000
2000	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000
2100	000000000	1111111111	0000000000	0000000000	490P0/00KR	@@@@@@+LLA	MJ00000000	1306000022	23220COMPA	C005736COM
2200	PAC0006397		0&-TEST	1 - TEST	RESULTSWHE	N AN UNIDE	NTIFIED RE	CORD IS EN	COUNTEREDE	ND OF JOBO
2300	V0SQ151400	V0SQ151500	P00181P0Q7	P01301P0Q8	PPQ201P0P1	R2UU022550	PPQ303P0Q2	R2SY000000	HALT 5	P01804P021
2400	P0004T2416	V0SP151200	Q01304P0P1	P00193P0R0	PPQ941P1R0	Q2TX032480	Q01200P0P1	U2UU002550	V0SQ151860	V0SQ151670
2500	P00P2TR516	P01201P0R1	TP0Q4TR536	V0SP131010	P2UW002570	P01884P021	V0SQ151610	TR5S642536	P0PP002610	P01S06R580
2600	P26P012540	PPP011P1R0	R2WP022700	P00161P0Q5	P01804P021	P0016T0031	P00Q0TR676	V0SP150980	P01884P021	U2SP002300
2700	QP1S00P000	U0PP000000	0128	NO DIAGNO	STICS LIST	ED 2719276	1/0325RPG	COMPILATIO	N COMPLETE	D2795/0033
2800	COMPILATIO	N TERMINAT	ED ABNORMA	LLY2837/02	37 *	** COMPILA	TION STATI	STICS ***R	MJ 2879N0	2894/0235
2900	****	** ALLOCAT	ION MAP **	****/0130R	PG OBJECT	PROGRAM EX	CEEDS 10K2	9342977/03	61INSUFFIC	IENT AVAIL
3000	ABLE SECTO	RS TO CONT	AIN THE RP	G OBJECT F	ILERPG V00	B 08/18/7	1246725632	5872700266	72433/////	/3QP053030
3100	PPW3510654	R4RP000000	P033533048	P060583053	P052040678	V2TT152360	V2SU152320	V2RU152190	PPR0011491	R3RP053250
3200	P065540201	P069340205	Q304380235	P308560855	P06U513270	X020000761	V2RY152260	P3QV053280	P034460015	0076100015
3300	R3SQ051560	VP7W960015	S3SS053410	0020000015	R3SU051560	VP2Y460779	S3SW053390	P077960015	U3SS000000	P083360294
3400	U3TU000000	P082760015	0020000015	R3TT051560	P083360224	002P000015	R3TW051560	P081560015	0020000015	R3UP051560
3500	P046860236	4PWQ460254	002P000015	R3UT051560	P289040031	V1WQ151580	Y087920880	P065410875	P086140031	P306140021
3600	P065540011	TQ4Y24P909	R3VV000000	R0000T0885	V1WQ151580	P065510875	1P66640011	2S08140021	PPP2143065	S3WP053610
3700	P078562454	P079762461	P307740021	P065540011	PRT541P728	R3WX000000	P24541PP00	1P65440021	1P65440011	PPP1142450
3800	S3XU053740	P080962454	P082162461	P307340021	V3XP153730	PP69252605	PP68252638	5PVY250682	Q3YP053890	P264482645
3900	P283340031	V1WQ151580	P306940021	Y087920880	P065410875	P086140031	S0000S0885	V1WQ151580	P065510875	2S08340021
4000	PPP2143073	S4PR053960	PPV9250654	S4PT054060	P296940031	V1WQ151580	PP51042728	S4PY000000	P288822731	1PVU440510
4100	R4QQ054120	P274582744	P275340031	V1WQ151580	P275740031	V1XP151720	P288440031	V1XP151720	Q065500000	U0PP000000
4200	P297340031	V1WQ151580	P279142757	U4QT000000	V2WU152700	PPR0910655	R4RR000000	P0203600P6	P0210400P0	V2VV152460
4300	U4RR000000	RP6W840678	V2TT152360	V4YS154910	V4VW154620	TP2P340737	R4SW054470	Q441003720	V3WU153680	V4VW154620
4400	U4SX000000	P020705Y50	Q0207PT520	V4YP154680	2P65440031	2PVU440678	V4VW124620	PPV8020654	S4TR024420	P06X114505
4500	P0207VT520	V4UW654530	U2SP002300	2P68044576	2PVU420680	P06X114575	V4VW154620	P020744526	V4YP154680	1T57540031
4600	1TUW520680	R4TV054470	V2SU152320	PPR0011491	R2WV000000	RP2P140678	QP6U540031	U4TP000000	PTU2010721	R4XU000000
4700	Q45R000745	Q06U504520	PTU2520655	R4WT054780	PPW4640654	Q4WX000000	4PWS740746	4P69740746	PTU2720655	R4XP054840
4800	PPW5140654	Q4XT000000	4PWS740751	4P69740751	Q07T504520	PPV4840652	V2QX111940	Q452005Y50	2P65440021	2PVU440648
4900	U3RQ000000	UP6X740697	4P02140697	U4ST000000	Q587003720	P065510736	QTYP005740	V5WT155670	QTYP005740	U2WV000000
5000	QP6W830031	1PV6630031	P12Y4TU031	U5XY000000	P020005Y50	Q504003720	R043600356	P037240678	V1YS151860	VP3V260362
5100	R2WV000000	PPS6440678	Q5QS055140	P036440678	V3WU153680	4PVW860362	R2WV055080	V4YS154910	P069741496	U2WV000000
5200	V4YS154910	P069741492	U2WV000000	P086540011	P06U512500	R100400984	P115440978	R041600356	P037240031	V1YS151810
5300	VP3V260362	Q2VW000000	P0200RP0P6	P0202TP0P0	V2VV152450	U5RY055290	Q587003720	P086540011	R037600356	P037240031
5400	R102400984	P06U512500	P103050978	V1YS151810	VP3V260362	Q2WV000000	PUU1644156	Q5UP000000	Q586005660	U5US000000
5500	P0206TP741	PPW4141170	S5WV000000	P0210TP678	P065510736	PPR16QP654	R5UW055580	P072810736	P0200VP0P6	P0206TP0P0
5600	PPR18QP654	R5VS000000	P0722100P5	V1YS151810	VP3V260362	Q5VV055670	P06U515750	PPV4840678	V2QX111940	P073615Y50
5700	1PVU440678	V3WU133680	1P65440021	1PVU440648	V2VV152450	P5TV052670	PPPS145516	P0724400P6	P9998RP0P7	P0000TP0P0
5800	P0004TP678	P072810736	2P65440031	PPPS145516	U5VW000000	Q5WV052670	P06U515750	P5950P5Y51	1PVU440520	V3YQ153840
5900	V2RU152190	P149150761	V2RY152260	P033860015	U0PV000000	V0SQ151400	V0SQ151500	P00181P0Q7	P01301P0Q8	PPQ201P0P1

Figure F.3 EXAMPLE 1: UNIDENTIFIED RECORD ENCOUNTERED — CORE DUMP

EXAMPLE 2. MULTIPLE RECORDS ENCOUNTERED WHEN ONLY ONE IS PERMITTED

Refer to Figures F.4, F.5, and F.6.

Step 1.

Determine the address of the Halt message from the Error Register, Locations 41-44 (A in Fig. F.6). This is 18W1 or 1871. Subtract 11.

    1871 - 11 = 1860

The Halt message starting in location 1860 (see B in Fig. F.6) is

    HALT 3

Referring to the Halt Error Code Summary in Table 8-1, we find this message means that multiple records were encountered for a particular record type in a group of sequenced record types when only one record of that type was allowed per group.

Step 2.

Obtain the address of the Communication Area from the Object Program Map (Fig. F.5).

    A 'Communication Area' = 2180

From characters 4-7 of the Communication Area (Fig. F.6, C) obtain the address of the entry in the Record Type Table for the current record.

    A ' Current Record Type Entry' = 1284

From characters 0-3 of the Communication Area (Fig. F.6, D) obtain the address of the current File Control Block (FCB).

    A ' Current FCB' = 1306

Characters 0-3 of the FCB (E in Fig. F.6) give the address of the Record Type Table for this file.

    A 'Record Type Table' = 1240

The entries in the Record Type Table are 22 characters long and the entries appear in the table in the same order as the record types in a sequenced group. Thus the relative position in the Record Type Table of the current entry can be calculated by

    (A'C' - A'E') ÷ 22 = n

where A'C' = Address of Record Type Table entry for the current record

A'E' = Address of the Record Type Table

n = relative position of Record Type Table entry (n=0 for the first record type, n=1 for the second record type, n=2 for the third record type.)

In this case, we have

$$(1284 - 1240) \div 22 = 2$$

or the current record is of the third type. This can be verified by examining the current record in the input buffer, which starts 20 positions lower than Program Entry, or at 0320, (see F in Fig. F.6). Column 80 contains a '0' which is the Record Identification Code for the third type of record (see line 13 of the source listing, Fig. F.4). Inspecting the FCB again, we find that characters 8-11 of the FCB (E in Fig. F.6) contain the address of the Record Type Table entry for the previous record. We see that this address is 1284, which is identical to the Record Type Table entry for the current record. That is, two records of the same type were encountered in succession, where only one record of each type per sequenced group is allowed by the Input Specifications.

The duplicate record condition can also be seen by comparing the address in positions 8-11 and positions 16-19 of the FCB which give the Record Type Table entries for the previous record and current record, respectively. These are both 1284. (Refer to H and G in Fig. F.6.)

```
VOOB    *REPORT*PROGRAM*GENERATOR*     SOURCE/DIAGNOSTIC LISTING              08/18/71    PAGE  1

      01010H      000
      01020FCARDIN  IPEA      0080            READER      1                      TEST=1
      01030FLIST    O         0132            PRINTER     2                      TEST=1
      01040F*  TEST 2 = TEST RESULTS WHEN MULTIPLE RECORDS ARE
      01050F*         ENCOUNTERED AND ONLY 1 IS ALLOWED
    S 01040F*                                                                    TEST=1
      02010ICARDIN  011 01   80 C&                                               TEST=1
      0202 I                                           1    6 EMPNR L1           TEST=1
      0203 I                                           9  142EARN                TEST=1
      0204 I        021 02   80 C=                                               TEST=1
      0205 I                                           1    6 EMPNR L1           TEST=1
      0206 I                                           9  152YTDERN              TEST=1
      0207 I        031 03   80 C0                                               TEST=1
      0208 I                                           1    6 EMPNR L1           TEST=1
      0209 I                                           7    7 FNAM               TEST=1
      0210 I                                           8    8 MNAM               TEST=1
      0211 I                                           9   19 SNAM               TEST=1
      0301 OLIST    H  301    1P                                                 TEST=1
      0308 O                                      31 'TEST 2 = TEST RESULTS'     TEST=1
      0309 O                                      55 'WHEN MULTIPLE RECORDS A'   TEST=1
      0310 O                                      78 'RE ENCOUNTERED AND ONLY'   TEST=1
      0311 O                                      91 '1 IS ALLOWED'              TEST=1
    S 03   O        D 03      03                                                 TEST=1
    S 03   O                            EMPNR      6                             TEST=1
    S 03   O                            FNAM      10                             TEST=1
    S 03   O                            MNAM      12                             TEST=1
      040100                            SNAM      24                             TEST=1
      0402 O        T 3                                                          TEST=1
      0403 O                                      10 'END OF JOB'                TEST=1
```

Figure F.4 EXAMPLE 2: MULTIPLE RECORDS ENCOUNTERED WHEN ONLY ONE IS PERMITTED – SOURCE/DIAGNOSTIC LISTING

```
VOOB    *REPORT*PROGRAM*GENERATOR*          OBJECT PROGRAM MAP                    08/18/71    PAGE  2


                                        **** INDICATORS ****

        ADDR   IN            ADDR   IN            ADDR   IN            ADDR   IN            ADDR   IN

        2000   L0            2001   LR            2002   L9            2003   L8            2004   L7
        2005   L6            2006   L5            2007   L4            2008   L3            2009   L2
        2010   L1            2011   1P            2012   H1            2013   H2            2014   H3
        2015   MR            2020   01            2021   02            2022   03

                                        ***** FIELDS *****

        ADDR   FIELD         ADDR   FIELD         ADDR   FIELD         ADDR   FIELD         ADDR   FIELD

        2195   EMPNR         2201  *EARN          2207   (L1)          2213  *YTDERN        2220   FNAM
        2221   MNAM          2222   SNAM

       ****** ALLOCATION MAP ******

       PROGRAM ENTRY              0340
       FIELD BASE ADDRESS         2000
       COMMUNICATION AREA         2180


       *** COMPILATION STATISTICS ***

       PROGRAM SIZE        2,750
       SPECIFIED SIZE     10,000
       SOURCE FILE - RPGPOL.TEMP
       OBJECT FILE - RPGPOL.RPGOBJ

       NO DIAGNOSTICS LISTED


    RPG COMPILATION COMPLETED
```

Figure F.5 EXAMPLE 2: MULTIPLE RECORDS ENCOUNTERED WHEN ONLY ONE IS PERMITTED – OBJECT PROGRAM MAP

```
        00         10         20         30         40         50         60         70         80         90
0000    POPP05040P  2000///// /1306RPGOB J1284SFTL  18W100000 0000050000 0020000015 ROPX050010 P029460015 PPR0010037
0100    Q0QU030060 P02P110130 P020250135 R020700300 U0PV000000 P020140176 PPP3510038 ROPP050000 0350000000 250000260
0200    S00002750                        POPP200002 008  08/18 /71                         0041                  //////
0300    U1YS000000 U2US000000 COMPAC   00 6363                                                                         0
0400    COMPAC        006937
0500                                     3510596 P00S140691 Q0010PP230 P0OP1TP031 Q0000PP610 P061010790 P061010620
0600    V0XQ150700 204P010132 201W03P002 1P1205PPQ5 P02S050235 V0XQ150700 PPP15UPPQ0 S0VX050690 P01201P0Q8 U1UY000000
0700    P01301P1U4 RP2S62011U6 R0WX000000 5PPQ5RP1U6 Q0WU050780 QP1R05PPQ5 P01591P1U4 U0WQ000000 2P238201U6 201U43P005
0800    2P1565PPQ5 U0VV000000 PPS9910P01 Q0SP030300 P0031401X4 U0SP000000 P0320601Y5 VP3R8602P1 U0SP000000 PPS20602P7
0900    R0YS000000 P0320602P7 P01201P0Q0 U0SP000000 PPS9910P02 Q0SP030300 P0031401X4 U0SP000000 P0320601Y5 WP3R870203
1000    U0SP000000 PPS20602P7 R1PU000000 P0320602P7 P01201P0Q0 U0SP000000 PPS9910P00 Q0SP030300 P0031401X4 U0SP000000
1100    P03206[illegible]Y5 P032[illegible]02R0 P0[illegible]7102R1 Q0328102R2 U0SP000000 PPS20602P7 R1QY000000 P0320602P7 P01201P0Q0 U0SP000000
1200    1032010080 S1RR051230 P01201P1Y4 U0SP000000 0000200820 0860089000 0000002109 4009801010 0000000022 1060110011
1300    5000001240 1200128412 4012840300 01204P0101 3200060000 15TR100000 P01191P400 P040000401 S040020501 U0SP000000
1400    PPQ201P0Q1 Q0SQ030310 V0SP151360 R0003Q0410 R0024S0432 R0047S0455 Q0070R0479 V0PS150540 P1SS200540 P0PP200100
1500    U0SQ000000 PPQ201P0R2 Q0SQ030310 V0SP151360 P01956P400 P02201P409 P02211P411 Q02221P413 V0PS150540 P1SS200540
1600    P0PP200000 U0SQ000000 V0SP151360 Q0082P0400 V0PS150540 P1SS200540 [illegible]PP000003 U0SQ000000 PPP16T0P12 Q0SQ000000
1700    PPP16T0P08 R1XS000000 P0016T0031 2P1924PP08 PPP08T0021 Q1WW000000 P0012T0P08 PPP16T0P08 R0SQ000000 P0008T0031
1800    PPP01QP1S0 R1XR051720 HALT 2     P0008T0031 PPP00QP1S0 R1XV050310 HALT 3                P0020R01W4 P0000T0031 P01304P1X4
1900    P01884P021 P0006TQ926 V0SP151060 P01804P021 TP1X44PP16 R1YV050310 2P1924P031 1PQR02P1W4 [illegible]YP000000 HALT 4
2000    1000000000 0000000000 0000000000 0000000000 0000000000 0000000000 0000000000 0000000000 0000000000 0000000000
2100    0000000000 000000000  1111111111 0000000000 0000000000 490P0/02KR @@@@@@+LLA MJ00010000 1306128422 33220COMPA
2200    C005736COM PAC0006397   006937      06-TEST  2 - TEST  RESULTSWHE N MULTIPLE  RECORDS A RE ENCOUNT ERED AND O
2300    NLY1 IS AL LOWEDEND O F JOB00015 V0SQ151400 V0SQ151510 P00181P0Q7 P01301P0Q8 PPQ201P0P1 R2UX022580 PPQ303P0Q2
2400    R2TR000000 HALT 5     P01804P021 P0004T2446 V0SP151200 Q01304P0P1 P00193P0R0 PPQ941P1R0 Q2UQ032510 Q01200P0P1
2500    U2UX002580 V0SQ151870 V0SQ151680 P00P2TR546 P01201P0R2 TP0Q4TR566 V0SP131150 P2VP002600 P01884P021 V0SQ151620
2600    TR5V642566 P0PP002640 P01S06R610 P26S012570 PPP011P1R0 R2WS022730 P00161P0Q5 P01804P021 P0016T0031 P00Q0TR706
2700    V0SP151100 P01884P021 U2SS002330 QP1S00P000 U0PP000000 ED 2719276 1/0325RPG  COMPILATIO N COMPLETE D2795/0033
2800    COMPILATIO N TERMINAT ED ABNORMA LLY2837/02 37         * ** COMPILA TION STATI STICS ***R MJ  2879N0 2894/0235
2900          **** ** ALLOCAT ION MAP ** ****/0130R PG OBJECT  PROGRAM EX CEEDS 10K2 9342977/03 61INSUFFIC IENT AVAIL
3000    ABLE SECTO RS TO CONT AIN THE RP G OBJECT F ILERPG V00 B  08/18/7 1246725632 5872700266 72433///// /3QP053030
3100    PPW3510654 R4RP000000 P033533048 P060583053 P052040678 V2TT152360 V2SU152320 V2RU152190 PPR0011491 R3RP053250
3200    P065540201 P069340205 Q304380235 P308560855 P06U513270 X020000761 V2RY152260 P3QV053280 P034460015 0076100015
3300    R3SQ051560 VP7W960015 S3SS053410 0020000015 R3SU051560 VP2Y460779 S3SW053390 P077960015 U3SS000000 P083360294
3400    U3TU000000 P082760015 0020000015 R3TT051560 P083360224 002P000015 R3TW051560 P081560015 0020000015 R3UP051560
3500    P046860236 4PWQ460254 002P000015 R3UT051560 P289040031 V1WQ151580 Y087920880 P065410875 P086140031 P306140021
3600    P065540011 TQ4Y24P909 R3VV000000 R0000T0885 V1WQ151580 P065510875 1P66640011 2S08140021 PPP2143065 S3WP053610
3700    P078562454 P079762461 P307740021 P065540011 PRT541P728 R3WX000000 P24541PP00 1P65440021 1P65440011 PPP1142450
3800    S3XU053740 P080962454 P082162461 P307340021 V3XP153730 PP69252605 PP68252638 5PVY250682 Q3YP053890 P264482645
3900    P283340031 V1WQ151580 P306940021 Y087920880 P065410875 P086140031 S0000S0885 V1WQ151580 P065510875 2S08340021
4000    PPP2143073 S4PR053960 PPV9250654 S4PT054060 P296940031 V1WQ151580 PP51042728 S4PY000000 P288822731 1PVU440510
4100    R4QQ054120 P274582744 P275340031 V1WQ151580 P275740031 V1XP151720 P2X8440031 V1XP151720 Q065500000 U0PP000000
4200    P297340031 V1WQ151580 P279142757 U4QT000000 V2WU152700 PPR0910655 R4RR000000 P0203600P6 P0210400P0 V2VV152460
4300    U4RR000000 RP6W840678 V2TT152360 V4YS154910 V4VW154620 TP2P340737 R4SW054470 Q441003720 V3WU153680 V4VW154620
4400    U4SX000000 P020705Y50 Q0207PT520 V4YP154680 2P65440031 2PVU440678 V4VW124620 PPV8020654 S4TR024420 P06X114505
4500    P0207VT520 V4UW654530 U2SS002330 2P68044576 2PVU420680 P06X114575 V4VW154620 P020744526 V4YP154680 1T57540031
4600    1TUW520680 R4TV054470 V2SU152320 PPR0011491 R2WV000000 RP2P140678 QP6U540031 U4TP000000 PTU2010721 R4XU000000
4700    Q45R000745 Q06U504520 PTU2520655 R4WT054780 PPW4640654 Q4WX000000 4PWS740746 4P69740746 PTU2720655 R4XP054840
4800    PPW5140654 Q4XT000000 4PWS740751 4P69740751 Q07T504520 PPV4840652 V2QX111940 Q452005Y50 2P65440021 2PVU440648
4900    U3RQ000000 UP6X740697 4P02140697 U4ST000000 Q587003720 P065510736 QTYP005740 V5WT155670 QTYP005740 U2WV000000
5000    QP6W830031 1PV6630031 P12Y4TU031 U5XY000000 P020005Y50 Q504003720 R043600356 P037240678 V1YS151860 VP3V260362
5100    R2WV000000 PPS6440678 Q5QS055140 P036440678 V3WU153680 4PVW860362 R2WV055080 V4YS154910 P069741496 U2WV000000
5200    V4YS154910 P069741492 U2WV000000 P086540011 P06U512500 R100400984 P115440978 R041600356 P037240031 V1YS151810
5300    VP3V260362 Q2VW000000 P0200RP0P6 P0202TP0P0 V2VV152450 U5RY055290 Q587003720 P086540011 R037600356 P037240031
5400    R102400984 P06U512500 P103050978 V1YS151810 VP3V260362 Q2WV000000 PUU1644156 Q5UP000000 Q586005660 U5US000000
5500    P0206TP741 PPW4141170 S5WV000000 P0210TP678 P065510736 PPR16QP654 R5UW055580 P072810736 P0200VP0P6 P0206TP0P0
5600    PPR18QP654 R5VS000000 P0722100P5 V1YS151810 VP3V260362 Q5VV055670 P06U515750 PPV4840678 V2QX111940 P073615Y50
5700    1PVU440678 V3WU133680 1P65440021 1PVU440648 V2VV152450 P5TV052670 PPPS145516 P0724400P6 P9998RP0P7 P0000TP0P0
5800    P0004TP678 P072810736 2P65440031 PPPS145516 U5VW000000 Q5WV052670 P06U515750 P5950P5Y51 1PVU440520 V3YQ153840
5900    V2RU152190 P149150761 V2RY152260 P033860015 U0PV000000 V0SQ151400 V0SQ151510 P00181P0Q7 P01301P0Q8 PPQ201P0P1
```

Figure F.6 EXAMPLE 2: MULTIPLE RECORDS ENCOUNTERED WHEN ONLY ONE IS PERMITTED – CORE DUMP

EXAMPLE 3. MATCHED FIELDS OUT OF SEQUENCE

Refer to Figures F.7, F.8, and F.9.

Step 1.

Determine the address of the Halt message from the contents of the Error Register, Locations 41-44 (A in Fig. F.9). The address 41T1 is equivalent to 4141. Subtract 11.

4141 - 11 = 4130

The Halt message starting in location 4130 (B in Fig. F.9) is

HALT 6

Referring to the Halt Error Code Summary (Table 8-1), we find this means "the match fields were found to be out of specified sequence."

Step 2.

To determine which file is being processed, obtain the address of the Communication Area (3360) from the Object Program Map (Fig. F.8). Obtain the address of the current FCB from characters 0-3 of the Communication Area (Fig. F.9, C).

A 'FCB' = 1842

Now locate the address 1842 in the FCB Address Table (D in Fig. F.9). The FCB Address Table begins at location 2970 (see Object Program Map, Fig. F.8) and consists of three four-character addresses of the FCBs for each of the input files. The addresses appear in the table in the same order as the input files are specified. Thus, we find '1842' in characters 0-3 of the FCB Address Table, signifying that the file being processed at the time of the execution error was the first input file specified, or the primary file. If we refer to the source listing (Fig. F.7) we see that this file is named CARDIN.

Step 3.

To locate the record being processed, find the Program Entry address from the Object Program Map (Fig. F.8)

A 'Program Entry' = 1160

Subtracting 20, we find the left-hand end of the input buffer is at 1140 (E in Fig. F.9). We also note that the record's first five characters which comprise the match field, are 'YYYYY'.

Step 4.

We will compare the contents of the <u>current</u> match field hold area with the hold area for the <u>previous</u> matched field. The hold area is located by adding 26 of the address of the current FCB. (Refer to the FCB format in Table 8-4.) In this case,

1842 + 26 = 1868

The length of the match field hold area is determined by adding the lengths of all match fields for this record type. (In this case we have one match field of five characters, so the hold area has a length of five characters.) Examining the match field hold area of the FCB (F in Fig. F.9), we find the contents are 'YYYYY', identical to the first five characters of the current input record. The previous match field is always moved to location 200 by the RPG object program. Examining locations 200-204 (G in Fig. F.9), we find that the previous match field is 'ZZZZZ'. Since the current match field 'YYYYY' is "less than" the previous match field and the sequence of match fields was specified in the File Description Specifications as ascending (Fig. F.7, line 4, 'A' in column 18), a program halt is the result.

```
VOOB   *REPORT*PROGRAM*GENERATOR*     SOURCE/DIAGNOSTIC LISTING              08/18/71   PAGE  1

   1      01010H
   2      01020F*  TEST 3 - TEST RESULTS WHEN MATCH FIELDS ARE OUT OF
   3      01030F*           SEQUENCE
   4      01050FCARDIN  IPFA     0080             READER     1                      TEST-1
   5      01060FDISC1   ISEA     0080             DISC   SYSPOL CBTEST              TEST-1
   6      01070FDISC2   ISEA     0080             DISC   SYSPOL DBGT                TEST-1
   7      01080FPRINTR  O        0025     OF      PRINTER    2                      TEST-1
   8      0212 ICARDIN  AA  01   80NC                                               TEST-1
   9      0213 I                                       1    5 SEQP     M1          TEST-1
  10      0214 I                                      11   15 TYPEP                TEST-1
  11      0215 IDISC1   AA  02   80NC                                               TEST-1
  12   S  02   I                                       1    5 SEQS1    M1          TEST-1
  13   S  02   I                                      11   15 TYPES1               TEST-1
  14   S  02   IDISC2   AA  03   80NC                                               TEST-1
  15   S  02   I                                       1    5 SEQS2    M1          TEST-1
  16   S  02   I                                      11   15 TYPES2               TEST-1
  17      0312 OPRINTR  H  201    OF                                                TEST-1
  18      0313 O       OR         1P                                                TEST-1
  19      0404 O*                                                                   TEST-1
  20      0406 O                          PAGE  Z 0004                              TEST-1
  21      0407 O                        MR           20 'MR'                        TEST-1
  22   S  0405 O        D 1       01                                                TEST-1
  23      0408 O                          SEQP        5                             TEST-1
  24      0409 O                          TYPEP      10                             TEST-1
  25      0410 O                        MR           20 'MR'                        TEST-1
  26      0411 O        D 1       02                                                TEST-1
  27      0412 O                          SEQS1       5                             TEST-1
  28      0413 O                          TYPES1     10                             TEST-1
  29      0414 O                        MR           20 'MR'                        TEST-1
  30      0415 O        D 1       03                                                TEST-1
  31   S  04   O                          SEQS2       5                             TEST-1
  32   S  04   O                          TYPES2     10                             TEST-1
  33   S  04   O                        MR           20 'MR'                        TEST-1
```

Figure F.7 EXAMPLE 3: MATCHED FIELDS OUT OF SEQUENCE – SOURCE/DIAGNOSTIC LISTING

```
VOOB    *REPORT*PROGRAM*GENERATOR*          OBJECT PROGRAM MAP                 08/18/71    PAGE  2


                                           **** INDICATORS ****

          ADDR  IN            ADDR  IN            ADDR  IN            ADDR  IN            ADDR  IN

          3180  L0            3181  LR            3182  L9            3183  L8            3184  L7
          3185  L6            3186  L5            3187  L4            3188  L3            3189  L2
          3190  L1            3191  1P            3192  H1            3193  H2            3194  H3
          3195  MR            3197  OF            3200  01            3201  02            3202  03

                                           ***** FIELDS *****

          ADDR  FIELD         ADDR  FIELD         ADDR  FIELD         ADDR  FIELD         ADDR  FIELD

          3389  SEQP          3394  TYPEP         3399  SEQS1         3404  TYPES1        3409  SEQS2
          3414  TYPES2        3419  PAGE


         ****** ALLOCATION MAP ******

         PROGRAM ENTRY             1160
         FIELD BASE ADDRESS        3180
         FCB ADDRESS TABLE         2970
         COMMUNICATION AREA        3360
         MULTIFILE I/O POINTERS    3375


         *** COMPILATION STATISTICS ***

         PROGRAM SIZE        4,640
         SPECIFIED SIZE     10,000
         SOURCE FILE = RPGPOL.TEMP
         OBJECT FILE = RPGPOL.RPGOBJ

         NO DIAGNOSTICS LISTED


RPG COMPILATION COMPLETED
```

Figure F.8 EXAMPLE 3: MATCHED FIELDS OUT OF SEQUENCE – OBJECT PROGRAM MAP

	00	10	20	30	40	50	60	70	80	90
0000	P0PP05040P	318000257	71842PEN	1820SFTL1	:41T100000	0000050000	0020000015	R0PX050010	P029460015	PPR0010037
0100	Q0QU030060	P02P110130	P020250135	T020720867	U0PV000000	P020140176	PPP3510038	R0PP050000	0001000000	250000260
0200	ZZZZZ	JJJJJ		P0PP000000				J	0000000000	0000//////
0300	U4Q0000000	U4PU000000	U0VQ000000	V0VP150520	VP0U0VPPU0	S0SV050420	V0VP150440	P00Q9QP400	P00R0QP405	P0013TP406
0400	X020001300	U0UY000000	P0425101Y4	U0UY000000	PPU5060PT4	R0TW050460	P0044VPPS8	P0050VPPT4	P0044VP015	V0WY150730
0500	P029460PU0	U0VP000000	P00S140591	P00P1TP031	U0VP000000	//////3000	P055610PR5	P0025QP586	HALT 70	U2SR000000
0600	U0SW000000	V0VP150520	PPU5060PU0	R0UV000000	V0VP150440	V0WY150680	1P66060PU6	U0UY000000	P00R0QP715	P00Q9QP710
0700	P0013TP711	P000010200	U0WV000000	P078010783	P075810757	U0WX000112	P076010783	P075910757	0020000015	R0UP040820
0800	P075710PR5	U0UW000000	P001510898	009P830001	008X010028	009P830001	0090910001	U0WX000000	A)RFADY DI	SC UNIT /,
0900	TYPE R.MA	DY DISC N.	0003540000	60009695SY	SP0LDBGT	1210////10	20MM100000	0PW1960731	R0YY051080	0020000192
1000	P112660025	U1PS000000	P112060025	P113220035	P019260015	0020000015	R1PW051110	P005060015	P022460180	Q110000000
1100	U0PV051110	HALT 1	OPEN CLOS	E SF000000	YYYYY	FFFFF				
1200		W	ZZZ7Z	CCCCC						1
1300	ZZZ7Z	JJJJJ						J	ZZ77ZEEEEE	MR
1400	P744	P00S141561	Q0010PP230	P00P1TP031	Q0000PQ480	P148011660	P148011490	V1VX151570	213X010025	201W03P002
1500	1P1205PPQ5	P02S050235	V1VX151570	PPP15UPPQ0	S1UU051560	P01201P0Q8	U2WP000000	P01301P1U4	RP2S6201U6	R1VU000000
1600	5PPQ5RP1U6	Q1VR051650	QP1R05PPQ5	P01591P1U4	U1UX000000	2P238201U6	201U43P0[illegible]5	2P1565PPQ5	U1US000000	PQR1910P01
1700	R0SP020300	P0031401X4	U0SP000000	P1140502P9	P1150502Q4	U0SP000000	P114050R26	U0SP000000	1114010080	S1XP051810
1800	P01201P1Y4	U0SP000000	0000201690	1730000017	6018201780	1842184218	20010111YY	YYYQ041530	PQR9910P01	R0SP020300
1900	P0031401X4	U0SP000000	P1220502Q9	P1230502R4	U0SP000000	P122050P26	U0SP000000	0000026002	5551220008	00000
2000	0000////00	2569002570	//////0000	00////////	/////0S0768	V0PS150330	P1YW000330	U0SP000000	0000211880	1920000019
2100	5020802050	2102210220	80010111ZZ	ZZZT412145	PQS7910P01	R0SP020300	P0031401X4	U0SP000000	P1300502R9	P1310502S4
2200	U0SP000000	P130050P26	U0SP000000	0000026002	5591300008	00000	0000////00	2576002577	//////0000	00////////
2300	////410735	V0PS150330	P2RS000330	U0SP000000	0000222140	2180000022	1023402310	2362236223	40010111ZZ	ZZZ213X010
2400	0250006000	053X030031	P01191Q380	R138051381	U0SP000000	PPQ201P0Q7	Q2TX032480	U2UQ002510	U0SQ000000	PPQ201P0Q1
2500	Q0SQ030310	V0SP152420	1P1204P2S9	PP2394Q380	PPQ201P0Q5	Q2UW032570	P0002R1398	V0PS151410	P2SY301410	P0PP100100
2600	U0SQ000000	PPQ201P0R0	Q0SQ030310	V0SP152420	P02095Q380	P02145Q385	PPQ201P0Q5	Q2VY032690	P0002R1398	V0PS151410
2700	P2SY301410	P0PP000001	U0SQ000000	PPQ201P0R1	Q0SQ030310	V0SP152420	P02195Q380	P02245Q385	PPQ201P0Q5	Q2XQ032810
2800	P0002R1398	V0PS151410	P2SY301410	P0PP000001	U0SQ000000	PPQ201P0R2	Q0SQ030310	V0SP152420	P02295Q380	P02345Q385
2900	PPQ201P0Q5	Q2YS032930	P0002R1398	V0PS151410	P2SY301410	P0PP000001	U0SQ000000	1842210223	62Z7ZZZFIC	P297440021
3000	P297840031	PPP26UPP26	Q3PS053040	P003140021	U0SQ000000	P0020R01W4	P0000T0031	P01304P1X4	P01884P021	P0006TS106
3100	V0SP151690	P01804P021	TP1X44PP16	R3QT050310	2P1924P031	1PQR02P1W4	[illegible]3PX000000	HALT 4	1000000000	000011000
3200	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000	0000000000	000000000
3300	1111111111	0000000000	0000000000	490P0/00KR	@@@@@@•LLA	MJ00010000	1842182034	2322029702	98203031•Z	ZZZZEEEEEY
3400	YYYYBBBBBY	YYYYIIIII0	0010 MR015	V0SQ152490	V0SQ152610	V0SQ152730	V0SQ152850	P00181P0Q7	P01301P0Q8	PPQ201P0P1
3500	R4TP024400	PPQ303P0Q2	R3UT000000	HALT 5	U3XX000000	P02032P1W2	P01954P1Y9	P01994P021	P0000T01X0	P01804P021
3600	PPP23Q01R0	R3XR000000	P0004T3636	V0SP152310	PPQ941P1R0	R3WS000000	PPP24Q01R0	Q3YQ033910	V0SQ153050	P0016T0031
3700	P0018TS716	V0SP152210	U3XQ000000	P01201PP22	P01301P1Y4	PPP25Q01R0	R3WW053790	1PQR02P2P5	R4PR024020	P02081PP26
3800	P0026T0P27	P01201PP23	1P1504P1Y9	1PQR02P1W2	R3XU053570	P38W013540	P01301P0Q1	U4QT004140	P01804P021	P0004T3906
3900	V0SP151780	Q01304P0P1	P00193P0R0	[illegible]Q941P1R0	R3YU054040	P01301P1Y4	P02081PP26	P0026T0P27	PPP25Q01R0	R4PP054010
4000	1PQR02P2P5	Q4QT034140	Q01200P0P1	U4TP004400	V0SQ153050	P0016T0031	PPP18TP1S0	R4SU024350	P0026U0200	P0018TT106
4100	V0SP151760	PPP26U0200	Q4QS054140	HALT 6	PPR071P1R0	R4QV054210	PPP26U2982	R4SR000000	V0SQ152990	PPP26U2982
4200	R4SR054220	V0SQ152990	P01954P031	P0000TP031	PPP26U0PR6	R4RY014270	P003140021	P01301P0Q6	U4SS000000	P01201P2P7
4300	P0026UR982	P003140021	P01201P0Q6	P0021401X0	P0016T0031	P00P2TT366	P01201P0R0	TP0Q4TT386	V0SP130000	U4TQ004410
4400	P01884P021	TT3X644386	R4TU024450	P01S06T420	P44T014390	PPP011P1R0	R4VR024620	PPP171P1S0	R4TY054500	P00181P0Q7
4500	P01301P0Q8	PPP171P1S0	R4UU024550	P01884P021	V0SQ152450	P00161P0Q5	P01804P021	P0016T0031	P00Q0TT596	V0SP151730
4600	P01884P021	U3TS003430	QP1S00P000	U0PP000000	R2WV000000	RP2P140678	QP6U540031	U4TP000000	PTU2010721	R4XU000000
4700	Q45R000745	Q06U504520	PTU2520655	R4WT054780	PPW4640654	Q4WX000000	4PWS740746	4P69740746	PTU2720655	R4XP054840
4800	PPW5140654	Q4XT000000	4PWS740751	4P69740751	Q07T504520	PPV4840652	V2QX111940	Q452005Y50	2P65440021	2PVU440648
4900	U3RQ000000	UP6X740697	4P02140697	U4ST000000	Q587003720	P065510736	QTYP005740	V5WT155670	QTYP005740	U2WV000000
5000	QP6W830031	1PV6630031	P12Y4TU031	U5XX000000	P020005Y50	Q504003720	R043600356	P037240678	V1YS151860	VP3V260362
5100	R2WV000000	PPS6440678	Q5QS055140	P036440678	V3WU153680	4PVW860362	R2WV055080	V4YS154910	P069741496	U2WV000000
5200	V4YS154910	P069741492	U2WV000000	P086540011	P06U512500	R100400984	P115440978	R041600356	P037240031	V1YS151810
5300	VP3V260362	Q2VW000000	P0200RP0P6	P0202TP0P0	V2VV152450	U5RY055290	Q587003720	P086540011	R037600356	P037240031
5400	R102400984	P06U512500	P103050978	V1YS151810	VP3V260362	Q2WV000000	PUU1644156	Q5UP000000	Q586005660	U5US000000
5500	P0206TP741	PPW4141160	S5WV000000	P0210TP678	P065510736	PPR16QP654	R5UW055580	P072810736	P0200VP0P6	P0206TP0P0
5600	PPR18QP654	R5VS000000	P0722100P5	V1YS151810	VP3V260362	Q5VV055670	P06U515750	PPV4840678	V2QX111940	P073615Y50
5700	1PVU440678	V3WU133680	1P65440021	1PVU440648	V2VV152450	P5TV052670	PPPS145516	P0724400P6	P9998RP0P7	P0000TP0P0
5800	P0004TP678	P072810736	2P65440031	PPPS145516	U5VW000000	Q5WV052670	P06U515750	P5950P5Y51	1PVU440520	V3YQ153840
5900	V2RU152190	P149150761	V2RY152260	P033860015	U0PV000000	V0SQ152490	V0SQ152610	V0SQ152730	V0SQ152850	P00181P0Q7

Figure F.9 EXAMPLE 3: MATCHED FIELDS OUT OF SEQUENCE — CORE DUMP

Appendix G
CONVERSION TABLE FOR NUMERIC CONTROL FIELDS

APPENDIX G: CONVERSION TABLE FOR NUMERIC CONTROL FIELDS

Numeric control fields are compared without regard for algebraic sign or decimal position. The zone bits are stripped off all digits in the control field prior to comparison of control fields to determine whether a control break has occurred. This may result in the storage of what appears to be non-numeric data in the control field hold areas. The following table permits a translation from the printed characters appearing in the control field hold area to their actual numeric values. (The numeric portions of the character codes for these characters are identical to the numeric parts of the codes for the digits 0-9.)

Table G-1 CONVERSION TABLE FOR NUMERIC CONTROL FIELDS

Printed Character	Numeric Value
(space)	0
!	1
"	2
#	3
$	4
%	5
&	6
'	7
(	8
)	9

Appendix H

SUMMARY OF DIFFERENCES BETWEEN 9K AND 10K COMPILERS

APPENDIX H: SUMMARY OF DIFFERENCES BETWEEN 9K AND 10K COMPILERS

SOURCE INPUT

The 9K compiler requires that source input be from a disc file.

The 10K compiler can have input either from a disc file or a card reader.

OBJECT OUTPUT

The 9K compiler must place the object program on a disc file.

The 10K compiler may place the object output on a disc file or punch it out as an object deck.

HARDWARE REQUIREMENTS

The 9K compiler requires at least a 9K partition to compile.

The 10K compiler requires at least a 10K partition to compile.

SOFTWARE REQUIREMENTS

The 9K compiler requires DMF, OPEN and CLOSE.

The 10K compiler requires DMF, R_OPEN and CLOSE.

FILE DESCRIPTION SPECIFICATIONS FORM

Record Length (Columns 24-27)

The 9K compiler allows a maximum record length for disc of 94 characters (one sector).

The 10K compiler allows multi-sector disc records of up to 940 characters.

INPUT SPECIFICATIONS FORM

Type (C/D/Z) (Columns 26, 33, and 40)

The 9K compiler only allows the use of the complete character (C option) in specifying record identification codes.

The 10K compiler allows the use of the digit or zone portions of characters as well as the complete character (C, D or Z options) in specifying record identification codes.

Hollerith Indicator (Column 43)

The 9K compiler recognizes an 11-zone punch for negative numbers in Hollerith punched card input when an H is present in column 43 of the specifications form. It does not, however, recognize the 12-zone punch for an explicitly positive number and will enter any such numbers into the system as zeros.

The 10K compiler will recognize both the 11-zone punch for negative numbers and the 12-zone punch for positive numbers in Hollerith punched card input when an H is present in column 43 of the specifications form. Both positive and negative numbers will be properly entered into the system.

Field-Record Relation (Columns 63-63)

The 9K compiler does not provide this feature.

The 10K compiler provides this feature which simplifies the writing of input specifications when there are two types of input records with only minor differences.

Field Indicators (Columns 65-70)

The 9K compiler does not provide this feature.

The 10K compiler permits the testing of contents of fields when the input data is entered. Numeric fields are tested for plus, minus or zero. Alphanumeric fields are tested for zeros or blanks. Specified indicators are turned on according to the results of the test.

CALCULATION SPECIFICATIONS FORM

Use of AND and OR with Indicators (Columns 7-8)

The 9K compiler does not provide this feature.

With the 10K compiler, more than one line of indicators can be used to control whether an operation will be done. Both the logical AND and OR relationships are allowed.

Z-ADD Operation

With the 9K compiler, an attempt to Z-ADD a field to itself is ignored.

With the 10K compiler, Z-ADD of a field to itself can be used to test the sign of the contents of the field and turn on an appropriate indicator.

GOTO Operation

The 9K compiler permits branching within detail calculations and within total calculations, but not between detail and total calculations.

The 10K compiler permits branching between detail and total calculations as well as within either type of calculation.

DSPLY Operation

The 9K compiler does not provide this feature.

The 10K compiler allows for displaying the contents of a field on a workstation, displaying the contents of a field and entering new contents to replace them from the workstation, or displaying two fields and entering new contents for one of the fields.

EXCPT Operation

The 9K compiler does not provide this feature.

The 10K compiler allows output lines designated with an E in column 15 to be printed while calculations are in progress.

Duplicate Specification of Field Lengths

With the 9K compiler, the field length and decimal positions can only be written once in the specifications: that is, when the field is first defined.

With the 10K compiler, the field length and decimal positions for a defined field may be written not only when the field is first defined in the input or calculation specifications, but subsequently whenever the field is used as a result field.

OUTPUT FORMAT SPECIFICATIONS FORM

Exception Record

The 9K compiler does not provide this feature.

With the 10K compiler, a record may be specified as an Exception Record by entering an E in column 15. Then every time an EXCPT operation is performed, all records so identified will be printed.

*PLACE

Not available with the 9K compiler.

The 10K compiler employs this symbol to signify the repeated placement of a field or group of fields across an output line.

Hollerith Output Code (Column 44)

The 9K compiler cannot generate the Hollerith punched card code for a negative number (11-zone punch).

The 10K compiler will generate Hollerith code for punched card output of negative numbers (11-zone punch), if an H is entered in column 44.

COMPILATION

Installation of Compiler

The 9K compiler has default values for the parameter input device and the printer device. These can be changed at installation time or the user can override them at compilation time.

The 10K compiler has default values installed not only for the parameter input device and the printer device, but options for the source input device, object output device and compile-and-go. Default values are set for these options when the compiler is installed and the user can override them at compilation time from the parameter input device.

Compile-and-Go

With the 10K compiler only, the user can specify the immediate execution of a successfully compiled RPG program by entering GO as an input parameter. The "compile-and-go" option is only valid when the object program is written to a disc file.

Compiler Error Messages

The 10K compiler provides several additional error messages not pertinent to the 9K compiler.

OBJECT PROGRAM

Halt Messages

The 9K compiler gives the Halt Code for execution time errors in the core dump.

The 10K compiler gives the Halt Code for execution time errors not only in the core dump but displays it on the workstation.

INDEX

INDEX

A

B

C

G

H

I

S

T

U

W

Z

RPG REFERENCE MANUAL
EDITION B
Publication No. 40-226-1

We produce manuals for you, and we want you to find them useful and informative. That's our job.

So we're asking you to help us furnish you with the best possible publications. Please take a few minutes to answer the following questions. Add any comments you wish. If you desire a reply to any question, be sure to include your name and address.

Thank you.

DETACH HERE

- Does this manual meet your needs? Yes ☐ No ☐
  If not, what additional information would be of help to you?

- Can you find what you're looking for quickly and easily? Yes ☐ No ☐
  How can the organization be improved?

- Is the material easy to read and to understand? Yes ☐ No ☐
  Are there enough illustrations to support the text? Yes ☐ No ☐
  Comments

- Did you find any errors or ambiguities in the manual? Yes ☐ No ☐
  If yes, please cite page, line, and/or figure number with your comments.

- Other comments.

- What is your relationship to the product described?
  - ☐ Operator.
  - ☐ Programmer.
  - ☐ Other (please specify)

SINGER
BUSINESS MACHINES

STAPLE

STAPLE

FOLD BACK

FIRST CLASS
PERMIT No. 320
San Leandro, Calif.

BUSINESS REPLY MAIL
No postage stamp necessary if mailed in the United States

POSTAGE WILL BE PAID BY

SINGER BUSINESS MACHINES
2350 Washington Ave.
San Leandro, California 94577

Attn: Customer Technical Publications,
Department 753

FOLD BACK

SINGER
BUSINESS MACHINES

PUBLICATION NO. 40-226-1
CONTROL NO. B241PB-1