SPERRY RAND

UNIVAC 9400 SYSTEM

REPORT PROGRAM GENERATOR

PROGRAMMERS REFERENCE

This manual is published by the Univac Division of Sperry Rand Corporation in loose leaf format. This format provides a rapid and complete means of keeping recipients apprised of UNIVAC® Systems developments. The information presented herein may not reflect the current status of the product. For the current status of the product, contact your local Univac Representative.

The Univac Division will issue updating packages, utilizing primarily a page-for-page or unit replacement technique. Such issuance will provide notification of hardware or software changes and refinements. The Univac Division reserves the right to make such additions, corrections, and/or deletions as, in the judgment of the Univac Division, are required by the development of its Systems.

PRINTED IN U.S.A.

UNIVAC 9400 System
Report Program Generator Programmers Reference
UP-7707

June 7, 1971

UPDATE PACKAGE "A"

UNIVAC 9400 System P.I.E. Bulletin 23, UP-7593.23, announces the release and availability of Updating Package "A" to the "UNIVAC 9400 System Report Program Generator Programmers Reference," UP-7707, cover and 99 pages plus page i and a 2-page Updating Summary Sheet.

SECTION	DESTROY FORMER PAGES NUMBERED	FILE NEW PAGES NUMBERED
Front Cover	‡	‡
Contents	1 and 2	1 Rev. 1 and 2*
	5 thru 9	5* and 6 Rev. 1
		7 Rev. 1 and 8 Rev. 1
		9 Rev. 1 and 10**
Section 1	3 and 4	3* and 4 Rev. 1
	9 thru 12	9 Rev. 1 and 10*
		11 Rev. 1 and 12 Rev. 1
Section 2	1 and 2	1* and 2 Rev. 1
	7 thru 12	7 Rev. 1 thru 12 Rev. 1
Section 3	1 and 2	1 Rev. 1 and 2 Rev. 1
	7 thru 10	7* and 8 Rev. 1
		9* and 10 Rev. 1
	13 and 14	13 Rev. 1 and 14*
	25 thru 30	25 Rev. 1 and 26*
		27* and 28 Rev. 1
		29 Rev. 1 and 30*
Section 4	5 thru 8	5* and 6 Rev. 1
		7 Rev. 1 and 8*
	17 thru 20	17* and 18 Rev. 1
		19 Rev. 1 and 20 Rev. 1
	23 and 24	23 Rev. 1 and 24*
	31 thru 34	31 Rev. 1 and 32*
		33 Rev. 1 and 34*
	37 and 38	37 Rev. 1 and 38*
Section 5	1 and 2	1* and 2 Rev. 1
	7 thru 10	7 Rev. 1 thru 10 Rev. 1
	13 and 14	13 Rev. 1 and 14*
	17 and 18	17* and 18 Rev. 1

UPDATE PACKAGE "A" Contd.

SECTION	DESTROY FORMER PAGES NUMBERED	FILE NEW PAGES NUMBERED
Section 6	1 and 2	1* and 2 Rev. 1
Section 7	1 and 2	1 Rev. 1 and 2*
Appendix E	1 thru 14	1 Rev. 1 thru 14 Rev. 1
Appendix F	1 thru 6 N. A.	1 Rev. 1 thru 6 Rev. 1 7** thru 11**
Appendix G	1 and 2 N. A.	1 Rev. 1 and 2 Rev. 1 3** thru 6**
Appendix H	N. A.	1** and 2**

‡File new cover and destroy old cover.
*These pages are backups of revised pages and remain unchanged.
**These are new pages.

CONTENTS

APPENDIXES

FIGURES

TABLES

1. INTRODUCTION

1.1. GENERAL

This manual is a reference guide for the preparation of specification forms used by the UNIVAC 9400 Tape and Disc Report Program Generator (RPG). RPG provides an efficient technique for generating programs and processing data as specified by these forms.

The manual assumes some basic familiarity with the principles of RPG programming, such as that provided by the publication *Introduction to RPG, UP-4149* (current version).

1.2. DESCRIPTION OF RPG

The Report Program Generator is based on a series of interrelated specification forms that define the operating parameters of the system for a given processing function. The contents of the six specification forms are keypunched to produce the source program which is introduced into the RPG compiler through standard job control procedures of the Disc or Tape Operating System. From this source program, the RPG generates an object program which is stored in its entirety in the computer processing unit where it controls the internal operation of the computer system to produce a report or any other specified output.

1.3. RPG OPERATION

Through the facilities of the Operating System the object program can be saved on magnetic tape or disc. Thus, the object program can be used for future runs to eliminate unnecessary regeneration of object code. Figure 1-1 illustrates the system flow.

Every record processed by the Report Program Generator goes through the same two-phase cycle of operation. One phase is referred to as detail time and the other phase as total time. All operations specified on the Calculation Specifications and Output Format Specifications forms pass through both phases.

Figure 1-2 shows one complete cycle of processing a simple RPG object program – one input file and one printer output file. The steps in the description are numbered to correspond to the box numbers.

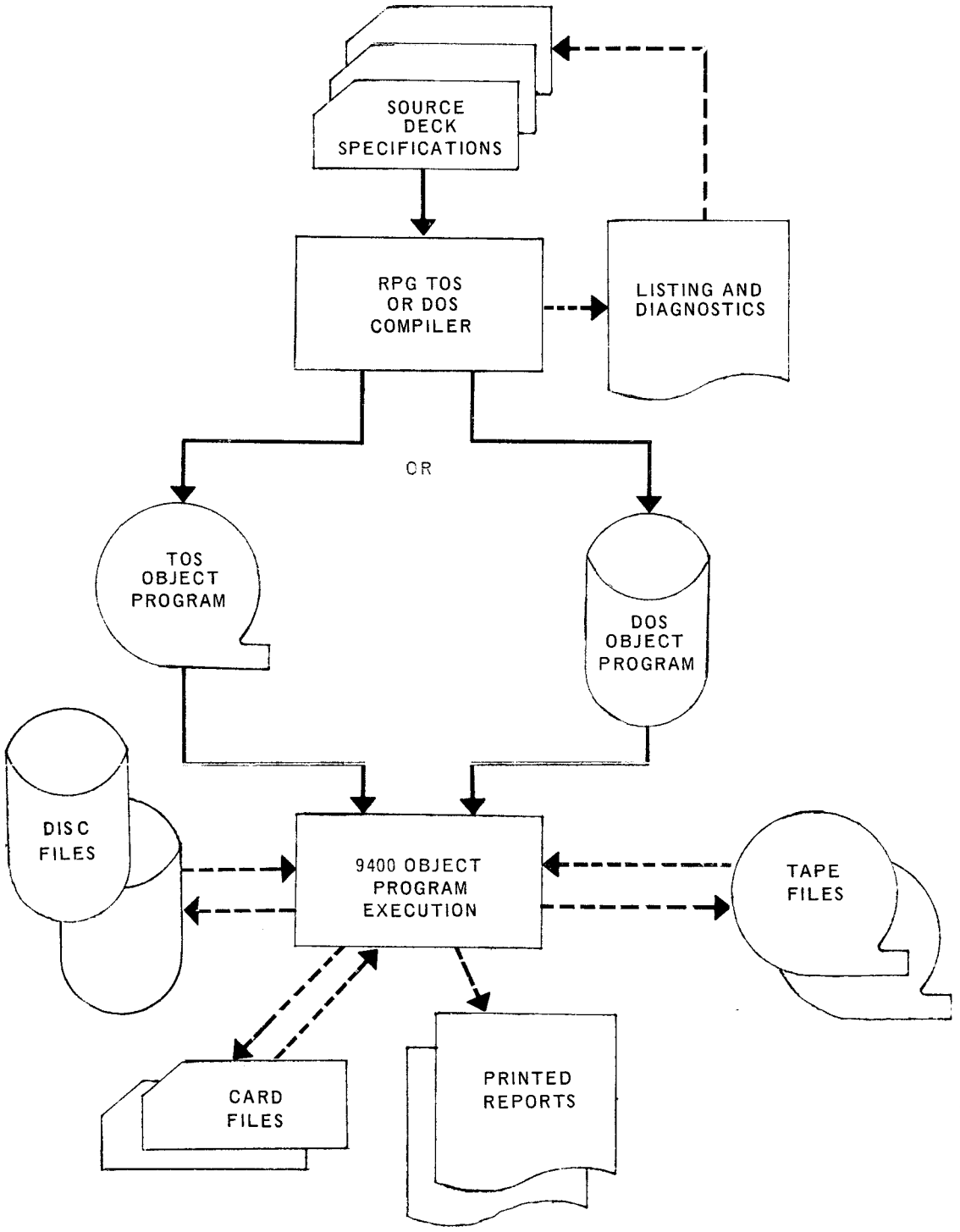


Figure 1–1. Diagram of System Flow

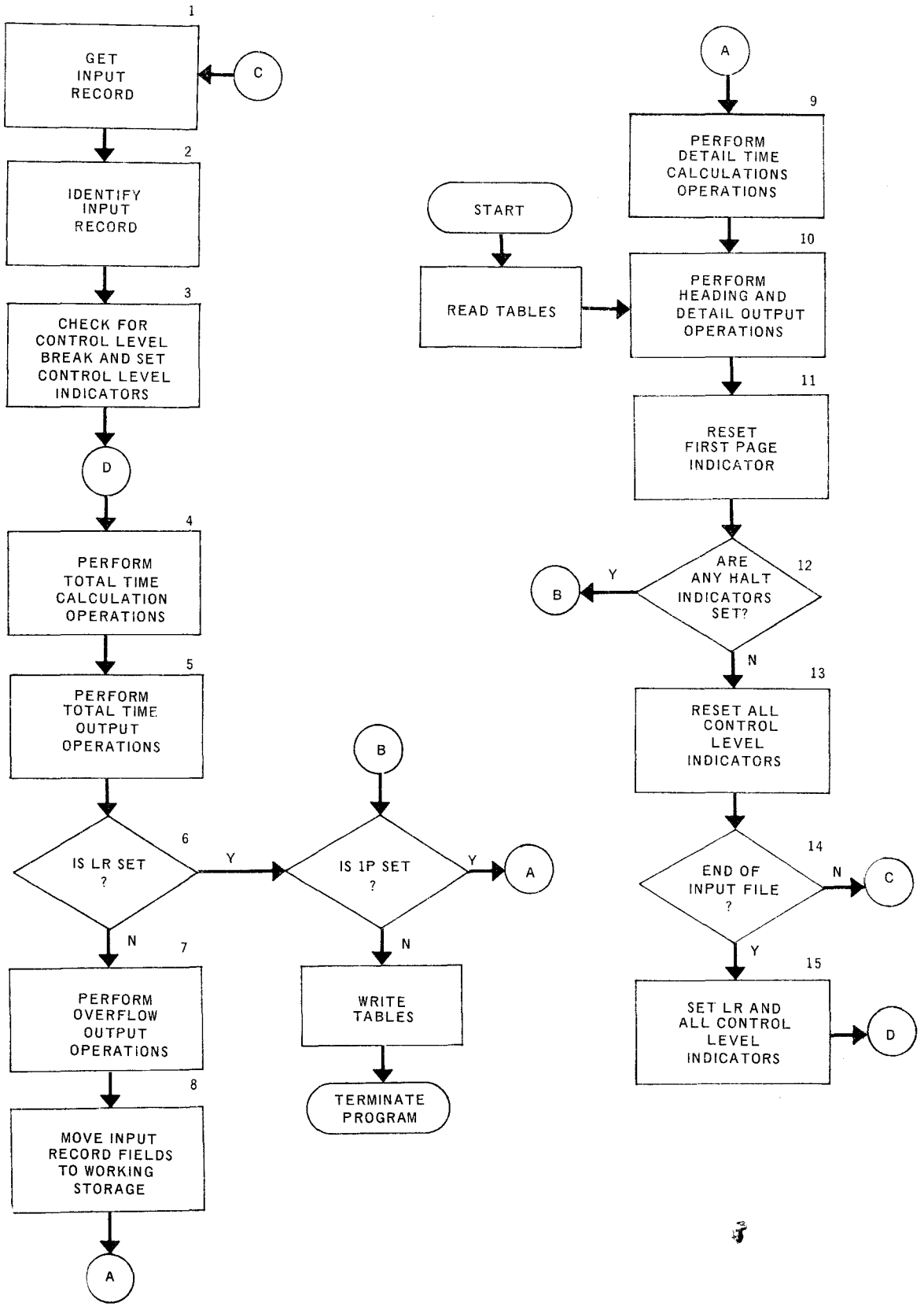


Figure 1–2. RPG Object Program Cycle

STEP	OPERATION
(1)	An input record is read into the system.
(2)	The object program uses the record identification code to identify the record read in step 1 by scanning each record description on the Input Format Specifications form. When the record is identified, the associated record indicator is set.
(3)	The object program checks for control level breaks by using the control level information provided in the record identified in step 2. All fields for this record with a control level entry on the input form are checked for a control break. When a break is found, the program sets the associated indicator and all lower control level indicators.
	NOTE: A control level break for any level occurs the first time that an input record is read which contains a control field for the given level.
(4)	The program performs all calculations appropriate to the control level breaks that have occurred. These calculations prepare for the output of total information to be produced at every control break (ignored on initialization of the object program).
(5)	The total output records are produced (ignored on initialization of the object program). If form overflow has occurred, then an overflow indicator is set.
	NOTE: A control level break for any level occurs the first time that an input record is read which contains a control field for the given level.
(6)	The program tests the last record indicator. If the indicator is not set, record processing continues. If the indicator is set and the first page (1P) indicator is not set, the program ends. However, if the 1P indicator is also set, the program continues with step 9.
(7)	Overflow output lines whose output indicators are satisfied are now produced. Total lines are output before detail lines.
NOTE:	This step is not performed in 9200/9300 mode programs (see 1.5). Instead, overflow output for this mode is produced in steps (5) and (10).
(8)	The fields of the input record are moved to working storage. This is the first time that the values of these fields become available for program processing.
(9)	The detail time calculations appropriate to the input record become available for program processing.
(10)	Heading and detail information is printed. The programmer controls this printing by means of the overflow, 1P, or other appropriate indicators. Headings are produced only at the top of a page.

If an overflow indicator was set previously, it is now reset. This step, in conjunction with step 5, guarantees that after form overflow has occurred the following operations will take place:

(a) Any totals appropriate to the page are printed at the bottom of the page.

(b) Any heading information is printed at the top of the next page.

(c) The overflow indicator remains on for one complete detail time (steps 9 and 10).

(d) If form overflow occurs during detail time output, the overflow indicator is on during one complete total time (steps 4 and 5).

If form overflow has occurred during the detail output time of this program cycle, the appropriate overflow indicators are set.

NOTE: Execution of the RPG object program actually begins at step 10. Thus, heading information may be printed on the first page by means of the 1P indicator. Each detail output operation must be conditioned by an indicator (which will be in the reset state when the object program is initiated) so that no detail output is produced until an input record has been read. All tables that are to be used by the object program are read into memory before any heading information is produced.

(11) The 1P (first page) indicator is reset.

(12) The program tests the halt indicators (H0-H9). If no halt indicators are set, record processing continues. If any halt indicator is set and the first page (1P) indicator is not set, the program ends. If the 1P indicator is also set, the program continues with step 9.

(13) All control level indicators are reset in preparation for the next program operation cycle.

(14) A check is made for the end of the input file; if not detected, control returns to step 1 to begin the next program operation cycle.

(15) If end-of-file condition is detected, the last record indicator and all control level indicators are set because end-of-file condition signifies a control break on all levels. Total time is then performed (steps 4 and 5) to produce the final total output, and the program terminates.

NOTE: 1. After the final total output, any tables which are to be produced as output are written before the program is terminated.

2. When execution of an RPG object program is initiated, all alphanumeric fields defined for the program are set initially to blanks, and all numeric fields to +0.

1.4. DISC FILE RPG CONCEPTS

A number of concepts related to the use of disc files in the RPG are not associated with card and tape processing. This section describes the following areas related to disc:

- Two ways of organizing a file on disc.
- Sequential and nonsequential methods of accessing records.
- Chaining (linking two or more files for processing).
- Updating a disc file.

Disc entries on RPG forms are discussed in the sections of this manual describing the specific RPG forms.

1.4.1. File Organization

Records in a card or a tape file are usually in a physical sequence based on a sorting of a field in the records. This is the sequence in which records in these files can be processed most efficiently.

Records in a disc file can also be set up and processed in the same manner. However, an advantage of using a disc file is the capability of establishing files which permit records to be processed on a nonsequential basis without necessarily processing an entire file. That is, records can be retrieved from randomly selected locations in a file, or records may be retrieved in sequence but within defined limits.

A file on disc is organized in either of two forms: as a sequential file, or as an indexed sequential file. The RPG programmer is not directly concerned with setting up disc files; however the file organization type must be known. This is necessary so that appropriate codes or entries can be specified on the RPG forms. The codes specified inform the RPG compiler of the file type that is to be created on disc or processed if the file is already on disc.

The organization of files is discussed followed by an explanation of how the programmer can select the method of retrieving a record from the file.

1.4.1.1. Sequential File Organization

Records in a file which is organized in a sequential manner are loaded on disc in the physical order in which they were sorted. This file is essentially the same as a card or a tape file. The RPG processor retrieves records in sequence starting with the first record and ending with the last record in the file. (Unsorted disc, tape, or card files may also be processed sequentially by RPG object programs.)

1.4.1.2. Indexed Sequential File Organization

Records in a file which is organized in an indexed sequential manner are loaded on the disc in the physical order in which they were sorted. In addition, an index to the file, created by the processor when the file was loaded, is loaded on the disc. This index is a list of items, each containing the key and the disc address of certain records in the file. (The key is the value in the record field that is used by the sort to position the record in the file.)

To retrieve a record, the processor determines the approximate location of the record by searching the keys in the index. The processor then examines this location to find the precise record required. Thus, the index provides the processor with a technique of locating a record in a file without the necessity of a sequential search starting at the beginning of the file.

1.4.2. Record Retrieval

The following paragraphs describe how records can be accessed for processing. The programmer may select any of the following methods of record retrieval:

- Sequential
- Sequential between specified limits
- Random

An indexed sequential file can be processed by any of the three methods; a sequential file can be processed either sequentially or randomly.

1.4.2.1. Record Retrieval in a Sequential File

Records in a sequential file can be accessed in the sorted or unsorted sequence originally established, or randomly on the basis of some selected field in a record. For random record selection, the file does not have to be physically resequenced. Instead, a special additional file called a *tag file*, created by the user, must be employed. The tag file is a file of disc addresses, each address specifying the location on disc of a given record. The addresses in a tag file are processed in sequence and the associated records are retrieved from random locations on disc. (The tag file is used only with a sequentially organized file, and only one tag file may be used in a program.)

For example, the records in a file originally sorted on employee number can be accessed in order of social security number by creating a tag file based on the social security field in a record.

1.4.2.2. Record Retrieval in an Indexed Sequential File

In addition to sequential accessing, records in an indexed sequential file can be accessed between specified limits within the file or can be accessed randomly. This is accomplished by use of a record address (RA) file created by the user. (This file is used only with an indexed sequential file.) The RA file consists of keys (data fields from records) corresponding to keys in the index created for the indexed sequential file. The field, which makes up the keys, can be on punched cards. Thus, one or more groups of limits between which records are to be accessed or a list of records to be accessed randomly may be specified.

For example, assume that records are organized in ascending order according to department number ranging from 200 through 600. If desired, the programmer can create an RA file specifying 202 and 210 in one record and 400 in a second record. This RA file, specifying limits, indicates that records from departments 202 through 210 and department 400 are to be processed.

An RA file, created for random selection of records also contains keys entered in the records. The keys (as in previous example) may appear in the following order: 600, 310, 205, and 308.

During program execution, the RA file is processed in sequence. The RPG processor compares a key field in the RA file to key fields in the index to find the record. A search then locates the exact record containing the key field.

The following rules apply in using an RA file:

(1) Only one RA file can be used in an RPG program. An RA file is not permitted if a tag file is used in the program.

(2) The file to be accessed on the basis of an RA file must be specified as a primary file.

(3) A numeric RA field cannot be packed.

(4) For specifying limits, only two keys (fields) may appear in a record; the first specifies a lower limit and the second specifies an upper limit. The lower limit must start in the first position of a record and the upper limit must start in a position immediately following the lower limit.

(5) For specifying random access, more than one key may appear in a record. An RA field containing blanks causes the remainder of the fields in that record to be skipped.

(6) The contents of the current RA field are available to the user in a field with the predefined name CONTD. (This field is always treated as an RLABL. See 4.2.4.19.)

See Section 6 for the coding of the RA field specifications in the File Extension form.

1.4.3. Chaining

Records in different files can be made available for concurrent processing by using a technique called chaining. For example, if several files exist, each containing records used in conjunction with records in the other files, and only two files are required for a particular processing application, they can be linked so as to make the record appear as one logical record.

Linking records requires the contents of at least one field in a file to be the same as the key field in the other file. This field is called a chaining field. It can be any field in a record. Chaining is effected by using the contents of the chaining field as the key for locating records in the second file.

The first file to be processed is called a chaining file, the other file is the chained file. A chained file must be an indexed sequential file and is on disc only; a chaining file may be either a sequential or an indexed sequential file. If sequential, it may be on card, tape, or disc.

A chained file may in turn chain to another chained file, and so forth. Also, a single file may chain directly to two or more other files. There is no restriction on the number of chained files, except that the total number of input files is limited to nine.

As an example, chaining can be done between a detail file containing part numbers and a price file containing prices with their associated part number. The field containing the part number in each record has a code, such as C1, specified in an adjacent field on a RPG form. The code allows the processor to link the detail file record to the price file record. Thus, price information is available to the RPG program as well as all other information in the price file record.

Records in a chained file need not be in the same sequence as those in the chaining file. The records in a chained file can be accessed by the RPG processor on a random basis. The accessing of records in a chained file is performed automatically by the processor.

Keys in a chained file may be numeric or alphanumeric. If numeric, they must be in packed format and have valid signs. Numeric chaining fields in a chaining file are packed and altered, if necessary, to suit this requirement before the search of the chained file.

1.4.4. Summary

The following chart shows the relationship between file organization and methods of record retrieval.

RECORD RETRIEVAL	FILE ORGANIZATION	
	SEQUENTIAL	INDEXED SEQUENTIAL
Sequential	Yes	Yes
Sequential Within Limits	No	Yes; with RA File
Random	Yes; with Tag File	Yes; with RA File
Chaining	Yes	Yes
Chained	No	Yes

1.4.5. Updating

The updating of disc files which requires changes only in the contents of existing records does not require processing the entire file. Each record read from a disc file can be changed and then be written back into the file. The only restriction is that the value in the key field must not be changed.

1.5. PROGRAM COMPATIBILITY

UNIVAC 9400 RPG provides an expanded range of features in comparison with UNIVAC 9200 and 9300 RPG. However, UNIVAC 9200 and 9300 RPG programs written for card, tape, or disc systems can be compiled and executed on the UNIVAC 9400. RPG programs for IBM System 360 Model 20 can also be compiled and executed.

In order to minimize any compatibility and conversion problems, the RPG compiler is designed to operate in any one of three modes:

- 9200/9300 mode – for programs originally used with UNIVAC 9200 and UNIVAC 9300.
- IBM 360 Model 20 mode – for programs originally used with IBM System 360 Model 20.
- 9400 mode – recommended for all new programs.

1.6. SUMMARY OF CONTENTS

A brief explanation of the common fields for all RPG specifications forms is included in this section of the manual. The remainder of the manual treats each RPG specification form separately with an additional section on table handling.

Section	Title
2	FILE DESCRIPTION FORM
3	INPUT FORMAT SPECIFICATIONS FORM
4	CALCULATION SPECIFICATIONS FORM
5	OUTPUT FORMAT SPECIFICATIONS FORM
6	FILE EXTENSION SPECIFICATIONS FORM
7	LINE COUNTER SPECIFICATIONS FORM
8	TABLE HANDLING

In each section the individual entries are described in the following format (as applicable):

- Entry

  An entry may be required, optional, or conditional on other factors.

- Purpose

  A brief summary statement of the use and functions of the entry.

- Codes

  Characters used to represent the specifications.

- Rules

  Requirements, constraints, and limitations for completing the entry.

- Examples

  Illustrations using coding sheets and flow charts to represent practical applications of RPG programs to specific problems.

In addition to the eight major sections, the following appendixes are included:

APPENDIX A. SAMPLE PROGRAMS
APPENDIX B. SUMMARY OF MODAL PROCESSING FEATURES
APPENDIX C. SUMMARY OF PROGRAM INDICATORS
APPENDIX D. STERLING NOTATION
APPENDIX E. OPERATING CONSIDERATIONS
APPENDIX F. RPG PROGRAM DIAGNOSTICS LISTING
APPENDIX G. USER-CODED SUBROUTINES AND USE OF RPG AS A SUBROUTINE
APPENDIX H. TAG FILES

1.7. COMMON FIELDS FOR RPG SPECIFICATION FORMS

The six RPG specification forms have certain common fields which have consistent entries within an RPG program. These fields and their respective entries are described here and are not repeated in the section for each specification form.

1.7.1. Page Number (Columns 1-2)

- Entry

  Optional.

- Purpose

  Establishes proper numerical sequence of the pages of the specification forms.

- Rules

  (1) It is recommended that gaps in page numbers be left between the various form types to permit easy insertion of additional pages of a specific type without upsetting page number sequences.

1.7.2. Line Number (Columns 3-5)

- Entry

  Optional (partially preprinted).

- Purpose

  Sequentially numbers each specification line. The two leftmost digits are preprinted. The units position is used by the programmer to insert specifications between two previously written lines.

- Rules

  (1) Inserts should be numbered in ascending order when more than one insert is entered between two previously written lines.

1.7.3. Form Type (Column 6)

- Entry

  Required (except in comments records).

- Purpose

  Identifies each form, and the type of specification being entered. The code is preprinted in Column 6 of each specification form.

- Codes

  H – Header Card

  F – File Description Specifications

  E – File Extension Specifications

  L – Line Counter Specifications

  I – Input Format Specifications

  C – Calculation Specifications

  O – Output Format Specifications

  For compatibility with other RPG compilers, a T in Column 6 is accepted. (T indicates a Table Header Card introducing tables for UNIVAC 9200/9300 RPG.) The first card encountered with T in Column 6 and the following cards in the source program are printed when a listing is made of the specification lines, but otherwise they are ignored by the RPG compiler.

1.7.4. Comments (Column 7)

- Entry

  Optional.

- Purpose

  Allows a comment record to be written at any point in the source stream on any RPG specification form. These entries are ignored by the RPG compiler, but are printed when a listing is made of the specification lines.

- Codes

  * – specifies comments.

- Rules

  (1) Comments may be entered from Column 8-74.

  (2) Column 6 of a comment record is ignored and may contain any EBCDIC character, including blank.

1.7.5. Program Identification (Columns 75-80)

- Entry

  Optional.

- Purpose

  Provides a way of identifying the source program.

- Codes

  One through six alphanumeric characters may be used.

- Example

  Figure 1-3 illustrates the identification of the second page of a program called PAYROL. The page number, line number, and program identification should be keypunched into every card to maintain the proper sequence of the card deck.

  The five blank lines at the bottom of each form allow the programmer to insert additional lines of coding without rewriting the entire page of specifications. To insert two lines of coding between lines 08 and 09, the programmer specifies 081 and 082 under line number and writes in the proper line of coding.

REPORT PROGRAM GENERATOR

INPUT FORMAT SPECIFICATIONS

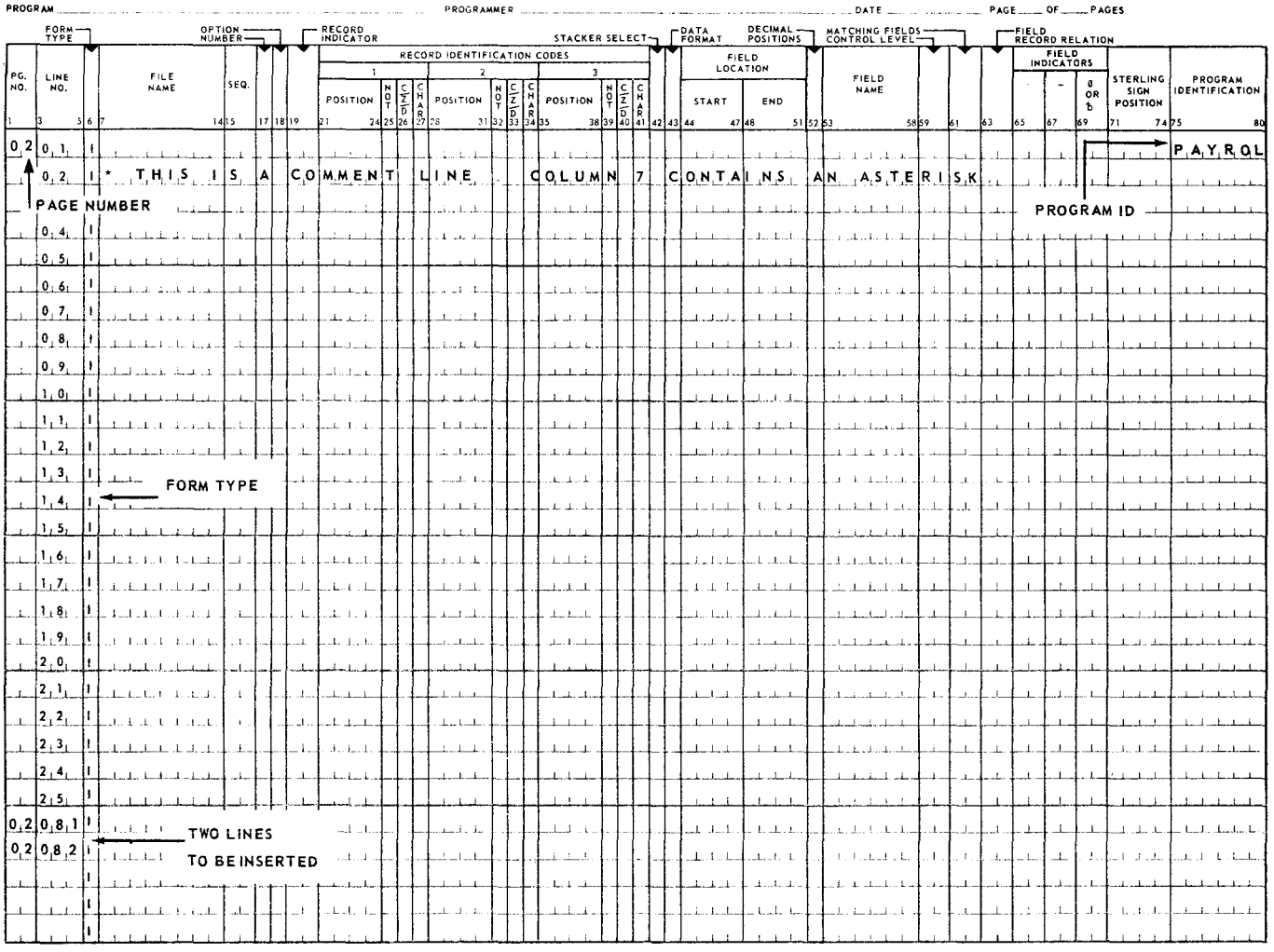


Figure 1-3. Example of Common Field Entries

2. FILE DESCRIPTION FORM

2.1. GENERAL DESCRIPTION

The File Description form (Figure 2-1) is used to assign a unique name to each file associated with an input or output unit used in the program. This form is also used to provide certain basic information about the files that are used by the program.

REPORT PROGRAM GENERATOR
FILE DESCRIPTION

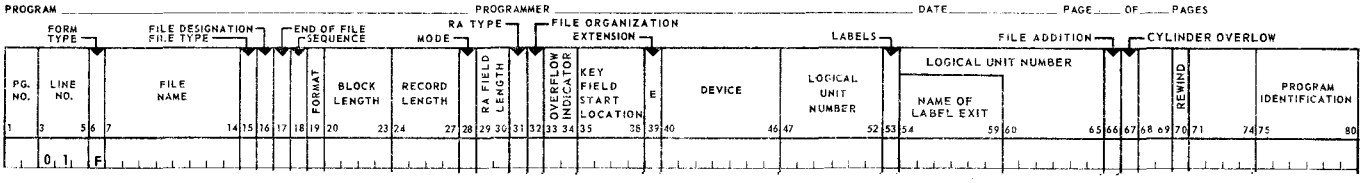


Figure 2-1. File Description Form

A maximum of 10 files may be used. There are additional restrictions on the number of various specific types of files as illustrated in Table 2-1.

TYPE OF FILE		MAXIMUM NUMBER
Input	Primary	1
	Secondary	8
	Record Address or Tag	1
	Chained	9
	Table	8
Output	Report Files	8
	Other Output Files	9
Update	Primary	1
	Secondary	8
	Chained	9
Combined	Primary	1
	Secondary	1

Table 2-1. Number of Files Supported

2.2. FORM ENTRIES

The following descriptions explain each entry used on the File Description form and demonstrate their functions in relation to the RPG program.

2.2.1. File Name (Columns 7-14)

- Entry

  Required.

- Purpose

  Assigns a unique name to each input and output file. The file names entered in these columns must also be entered on the Input and Output Format Specifications form. This makes all the information in the File Description form relating to this file available to the RPG program.

- Rules

  (1) The file name must be left-justified (to Column 7) and the first character must be alphabetic.

  NOTE: For this manual, the alphabetic characters are considered to be the letters A through Z, $, #, @, ?, and %. The numeric characters are 0 through 9. All other characters, other than blank, are special characters.

  (2) The remaining characters may be alphabetic or numeric, but not special characters or spaces.

  (3) A maximum of eight characters may be used for a file name, but the first seven identify the file. In 9200/9300 mode, all eight characters identify the file. For example, WORKFIL1 and WORKFIL2 are considered the same file name. However, the entire eight-character file name entry is checked for syntactic validity.

- Example

  Figure 2–2 illustrates file name entries on a File Description form.

REPORT PROGRAM GENERATOR
FILE DESCRIPTION

PROGRAM _____ PROGRAMMER _____ DATE _____ PAGE ____ OF ____ PAGES

PG. NO.	LINE NO.	FORM TYPE	FILE NAME	FILE DESIGNATION / FILE TYPE	END OF FILE SEQUENCE	FORMAT	BLOCK LENGTH	RECORD LENGTH	RA MODE	RA FIELD LENGTH	FILE ORGANIZATION / RA TYPE	OVERFLOW INDICATOR	KEY FIELD START LOCATION	EXTENSION E	DEVICE	LABELS / LOGICAL UNIT NUMBER	FILE ADDITION / LOGICAL UNIT NUMBER / NAME OF LABEL EXIT	CYLINDER OVERLOW	REWIND	PROGRAM IDENTIFICATION
	01	F	* THE FOLLOWING ENTRIES SPECIFY FILENAMES, FILE TYPE, FILE																	
	02	F	* DESIGNATION, END OF FILE AND FORMAT.																	
	03	F	READPNCHCS		A	F														
	04	F	PRINTOUTO			V														
	05	F	TAPEIN	I	S	E	A	F												
	06	F	CARDIN	I	P	E	A	F												

Figure 2–2. Examples of Various File Description Entries

2.2.2. File Type (Column 15)

- Entry

  Required.

- Purpose

  Specifies the type of file defined on the line.

- Codes

  I – Input. An input file consists either of data records or of table, record address, or tag files read from a specified input device during object program execution.

  O – Output. An output file may either be a data file or a table file which is written, printed, or punched.

  C – Combined. A combined file consists of a card file read into the system and subsequently punched as output into the same cards that have been read or into blank cards inserted into the combined file. In addition to being punched, the blank cards must be identified as an input record type (see 3.2.5).

  U – Update. The update file is disc-oriented and is an input and output file. The file is an update file if the object program alters the data in one or more fields of each record in the file. The revision made to these fields must not alter the nature of the data, the field length, or the field location.

  A chained file may be updated at detail and total time; all other files can be updated at detail time only.

- Rules

  (1) Every RPG program must have at least one input file.

  (2) Files of different types may be listed in any order.

  (3) Blank or prepunched cards may be specified as an output file on the Type 0604 Row Punch with read/punch feature. Prepunched information cannot be read when the cards are designated as an output file.

  NOTE: Stacker select entries for records of a combined file that are read only should be made on the Input Format Specifications form. Stacker select entries for records that are punched only should be made on the Output Format Specifications form.

- Example

  Figure 2–2 illustrates various combinations of file type entries.

2.2.3. File Designation (Column 16)

- Entry

  Conditional; used with file types I, C, and U.

- Purpose

  Indicates whether an input file is a table file or a data file. If a data file, indicates whether that file is primary, secondary, or chained.

- Codes

  T – Table file.

  P – Primary data file.

  S – Secondary data file.

  R – Record address or tag file.

  C – Chained (disc chained) file.

- Rules

  (1) Column 16 is left blank for all output files.

  (2) A maximum of nine input files can be specified, including at least one primary file.

- Example

  Refer to Figure 2–2 for an illustration of file designation entries.

2.2.4. End of File (Column 17)

- Entry

  Optional.

- Purpose

  Indicates which files are to be checked for an end of file condition in order to turn on the last record (LR) indicator. Should only be used when all end of file conditions are not present to complete processing.

- Codes

  E – Indicates end of file conditions to be checked.

- Rules

  (1) If Column 17 is left blank for all input files, the LR indicator is turned on when all input files have been processed.

  (2) An E entry for a single file indicates that all data records for this file must be read and processed before the LR indicator is turned on.

  (3) An E entry for multiple files indicates that the last data record of each specified file must be processed before the LR indicator is turned on.

  (4) The object program detects the end of a card file by reading a card containing a slash (0-1) punched in Column 1 and asterisk (11-4-8) punched in Column 2.

- Example

  See Figure 2–2.

2.2.5. Sequence (Column 18)

- Entry

  Conditional.

- Purpose

  Indicates the sequence of the record files so that the matching operations in the program are performed correctly (see 3.3.6).

- Codes

  A – Ascending Sequence.

  D – Descending Sequence.

- Rules

  (1) Sequence entries apply only to input files.

  (2) If the sequence entry for an input file is left blank and matching fields are specified in the Input Format Specifications, the file is assumed to have ascending sequence.

- Example

  See Figure 2–2.

2.2.6. File Format (Column 19)

- Entry

  Conditional; not required for printer and card files in 9200/9300 mode and IBM 360 Model 20 mode programs.

- Purpose

  Defines the record format of input and output files.

- Codes

  F – Fixed-Length Records.

  V – Variable-Length Records.

- Rules

  (1) Fixed-length records must be specified for disc files with indexed sequential organization (I in Column 32).

  (2) Variable-length records must be specified for all files with line counter specifications (L in Column 39).

  (3) In 9400 mode programs, if the file format entry for a file is blank, F is assumed for indexed sequential files, V for all others. In other programs, F is assumed for card, printer, and indexed sequential files, V for all others.

- Example

  See Figure 2–3.

REPORT PROGRAM GENERATOR
FILE DESCRIPTION

PROGRAM _____ PROGRAMMER _____ DATE _____ PAGE ____ OF ____ PAGES

PG. NO.	LINE NO.	FORM TYPE	FILE NAME	FILE DESIGNATION	FILE TYPE	END OF FILE SEQUENCE	FORMAT	BLOCK LENGTH	RECORD LENGTH	MODE	RA TYPE	RA FIELD LENGTH	OVERFLOW INDICATOR	FILE ORGANIZATION	KEY FIELD START LOCATION	EXTENSION	DEVICE	LABELS	LOGICAL UNIT NUMBER	FILE ADDITION	CYLINDER OVERLOW	REWIND	PROGRAM IDENTIFICATION
																E			NAME OF LABEL EXIT				
1	3 5	6	7 14	15	16	17 18	19	20 23	24 27	28	29 30	31	32	33 34	35 38	39	40 46	47 52	53 59	60 65	66	67 68	69 70	71 74	75 80
	0 1	F	T A P E I N 1	I	P	E	A	F 1 0 0 0	1 0 0								T A P E								
	0 2	F	T A P E I N 2	I	S	E	A	F 1 0 0 0	1 0 0								T A P E								
	0 3	F	T A P E I N 3	I	T			V 5 0 0	1 8							E	T A P E								
	0 4	F	T A P O U T 1	O				F 1 0 0 0	1 0 0								T A P E								
	0 5	F	T A P O U T 2	O				F 1 0 0 0	1 0 0								T A P E								
	0 6	F	C A R D I N	I	S	E	A	F 8 0	8 0								R E A D E R								
	0 7	F	N E W D I S C	U	S	E		F 3 4 5	1 1 5								D I S C								

Figure 2–3. Completed File Description Form

2.2.7. Block Length (Columns 20-23)

- Entry

  Conditional; optional for printer and card files.

- Purpose

  Indicates the number of data characters in a block of an input or output file.

- Codes

1-9999 – Specifies the number of bytes in the block. The maximum and minimum block length by type of file is:

TYPE OF FILE	BLOCK LENGTH	
	Minimum	Maximum
Card	1	80
Printer	1	132
Tape – VI-C	18 (F format) 10 (V format)	4095
Tape – 12, 16	18 (F format) 10 (V format)	9999
Disc – 8411	1	3625 (F format) 3617 (V format)
Disc – 8414	1	7294 (F format) 7286 (V format)

- Rules

(1) The size specifications must be right-justified; leading zeros may be omitted.

(2) If record length is fixed, enter the size of the tape block. This entry must be some multiple of the record length entry (Columns 24-27).

(3) If record length is variable, enter the size of the largest block in the file. This entry should be the largest possible sum of the lengths of the data characters of the records contained in any block (see Appendix E).

(4) For unblocked, indexed sequential files, the block length specified must include the length of the key.

- Example

See Figure 2-3.

2.2.8. Record Length (Columns 24-27)

- Entry

Conditional; optional for printer and card files.

- Purpose

Indicates the length of the logical record contained in the file.

- Codes

The codes for record length are the same as those for block length (see 2.2.7).

- Rules

  (1) The size specification must be right-justified; leading zeros may be omitted.

  (2) If record length is fixed, enter the record length. For all indexed sequential files, blocked or unblocked, the record length given must include the key length.

  (3) If record length is variable, enter the number of data characters in the largest record (see Appendix E).

- Example

  See Figure 2-3.

2.2.9. File Processing Mode (Column 28)

- Entry

  Conditional; used with disc files only.

- Purpose

  Indicates the method or mode by which the record is processed.

- Codes

  Blank – Indicates sequential processing of a file. The file organization (Column 32) may be either sequential, tag, or indexed sequential.

  L – Indicates sequential processing of a portion of an indexed sequential file using a record address (RA) file. The user supplies the record addresses which provide the upper and lower limits of the file to be processed.

  R – Indicates random processing of a sequential file using a tag file or of an indexed sequential file using an RA file or a chaining file. The file must be a data file.

- Rules

  (1) This column is blank for tape, card, or printer files.

2.2.10. Key Length or Record Address Field Length (Columns 29-30)

- Entry

  Conditional; for use with indexed sequential, RA, and tag files.

- Purpose

  Specifies the number of positions that each entry in an RA or tag file occupies or the length of the key for an indexed sequential file.

- Codes

  01-99

- Rules

  (1) The length specification must be right-justified; a leading zero may be omitted.

  (2) This field is blank for tape, card, or printer files.

  (3) For a tag file, a field length of 10 is assumed if this field is blank.

  (4) For an RA file, a field length of 8 is assumed if this field is blank.

2.2.11. Record Address Type (Column 31)

- Entry

  Conditional.

- Purpose

  Identifies the address type.

- Codes

  Blank – Indicates a sequential file, accessed sequentially.

  K – Indicates an indexed sequential file which will be processed using a record key.

  I – Indicates a sequential file whose records will be retrieved by a tag file containing track addresses (see Appendix H).

  R – Indicates a sequential file whose records will be retrieved by a tag file containing relative block numbers (see Appendix H).

- Rules

  (1) This field is blank for tape, card, or printer files.

  (2) Files of record address type I or R must be data files.

2.2.12. File Organization Type (Column 32)

- Entry

  Conditional.

- Purpose

  Identifies the type of file organization.

- Codes

  Blank – Designates a sequential file, accessed sequentially. Unbuffered (demand mode) input and output are provided.

  1-9 – Designates a sequential file, accessed sequentially. Buffered input and output are provided.

  D – Indicates a sequential file, the records of which are accessed randomly using a tag file. This file must be a data file.

  T – Indicates a tag file (see Appendix H).

  I – Indicates an indexed sequential file to be created or processed.

- Rules

  (1) Placing any digit in the range 1 through 9 in Column 32 causes buffered input and output to be provided for sequential files, where buffering is appropriate. Except for card reader files which are always buffered, input and output normally are provided unbuffered.

2.2.13. Overflow Indicator (Columns 33-34)

- Entry

  Conditional; used for report files only.

- Purpose

  Identifies the overflow indicator associated with the file.

- Codes

  The overflow indicators are:

  OA, OB, OC, OD, OE, OF, OG, OV

- Rules

  (1) This field is blank except for report files.

  (2) Generally one overflow indicator is used to control the output of each report file.

  (3) In 9200/9300 mode programs, OF is assumed for the printer file. (No more than one printer file should be specified with blanks in columns 33 and 34.)

2.2.14. Key Field Starting Location (Columns 35-38)

- Entry

  Conditional; for indexed sequential files only.

- Purpose

Indicates the starting location of the key field within a data record. This specification is provided so that the key field may be located anywhere within the data record.

- Codes

Blank or 0000 – Indicates an unblocked indexed sequential file to be created or processed.

0001-7229 – Indicates a blocked indexed sequential file to be created or processed.

- Rules

(1) This entry is required for blocked indexed sequential files only.

(2) The entry must be right-justified; leading zeros may be omitted.

(3) In unblocked indexed sequential files, the key is always treated as though it started in position 1 of the record. However, an entry of 1 (or higher) should never be specified for unblocked files.

		MODE OF PROCESSING (COLUMN 28)		RECORD ADDRESS TYPE (COLUMN 31)		TYPE OF FILE ORGANIZATION (COLUMN 32)		KEY FIELD STARTING LOCATION (COLUMNS 35-38)
		CONTENTS	REMARKS	CONTENTS	REMARKS	CONTENTS	REMARKS	CONTENTS
DISC FILE TYPE	Sequential	(Blank)	The entire file will be processed.	(Blank)	Not applicable	(Blank)	Sequential	Blank
	Indexed Sequential	(Blank) (L) (R)	The entire file will be processed. A segment of the file will be processed. The limits to be processed are supplied by a record address (RA) file. The records will be processed randomly. The addresses are supplied: (a) by an RA file, or (b) by the data contained in the chaining field of an input record.	(K)	Record Key	(I)	Indexed Sequential	Key Field Starting Location
	Tag File Processing	(Blank) (R)	Tag (Address) File Data File	(Blank) (1 or R)	Tag (Address) File Data File	(T) (D)	Tag (Address) File Data File	Blank

Table 2-2. Summary of Columns 28, 31, 32, 35 – 38 for Disc Files

2.2.15. Extension Code (Column 39)

- Entry

Conditional; used with chaining, RA, tag, and table files.

- Purpose

Indicates to the RPG processor that additional information about the file is coded on a File Extension form (see Section 6) or Line Counter Specifications form (see Section 7).

- Codes

  E – A File Extension Specifications form is present for the file.

  L – A Line Counter Specifications form is present for the file.

- Rules

  (1) L can be coded only for sequential output files. Column 19 of the File Description form must contain a V.

- Example

  See Figure 2–3.

2.2.16. Device (Columns 40-46)

- Entry

  Required.

- Purpose

  Relates a file to a specific type of input or output unit during compilation time.

- Codes

Entry	Input/Output Device
READER, READ01, READ20, READ40, READ42	600 CPM Reader, Type 0711
PUNCH, PUNCH42, PUNCH20, ROWPNCH, READ20, READ40, READ42	Row Punch, Type 0604
CRP, CRP20, RRP, READ20, READ40, READ42	Row Punch, Type 0604 with read/punch feature
PRINTER, PRINT63, PRINT16, PRINTLF, PRINTUF, PRINTDR	Drum Printer, Type 0768-00 or 0767-01
TAPE, TAPE7	UNISERVO 12, 16, VI-C tape units
TAPE6C	UNISERVO VI-C tape units
DISC, DISK11F, DISK11	Disc Unit, Type 8411
DISC14, DISK14	Disc Unit, Type 8414

NOTE: Programs for card controller, multifunction card machine or similar device, or for a direct access storage device other than the UNIVAC 8411/8414 or IBM 2311 cannot be compiled or executed.

- Rules

  (1) Each file must be assigned one of the input/output devices.

  (2) The code must be left-justified.

  (3) The entry TAPE6C is permitted for files on UNISERVO VI C tape units, but is not required.

  (4) The various alternative codes listed in the table are provided for compatibility with other RPG compilers.

- Example

  See Figure 2–3.

2.2.17. Logical Unit Number (Columns 47-52)

In the 9400 RPG this field is not used. For the sake of compatibility with other RPG compilers, however, any EBCDIC characters or blanks may appear here.

2.2.18. Labels (Column 53)

- Entry

  Conditional; used only with magnetic tape or discs.

- Purpose

  Specifies the type of label to be checked by the program during label processing.

- Codes

  Blank – The file is unlabeled.
  S – The file has standard labels.
  N – The file has nonstandard labels.
  E – The file has standard labels followed by user standard labels.

- Rules

  (1) Label processing is provided by the RPG program. It is, however, the programmer's responsibility to supply any required label information in Job Control (LBL) cards at the time the object program is loaded for execution.

  (2) For E entries, the standard label is processed. In addition, an automatic exit is made to the user routine specified in Columns 54-59 to process the user standard label.

(3) Standard labels are required for indexed sequential disc files.

(4) Nonstandard labels are not permitted for sequential disc files.

(5) A tape mark must precede the data on an unlabeled output file.

- Example

Figure 2–3 illustrates sample entries for tape files.

2.2.19. Label Exit Name (Columns 54-59)

- Entry

Conditional; used if E or N appears in Column 53.

- Purpose

Gives the name of the routine written by the user to process user standard labels or nonstandard labels.

- Rules

(1) The name of the user routine must be left-justified and may contain a maximum of six characters. The first character must be alphabetic; the remainder, alphabetic or numeric.

(2) In 9200/9300 mode programs, Columns 54-59 are ignored.

(3) See Appendix F for a discussion of label exit routines.

2.2.20. Columns 60-65

These columns are not used, and may contain any EBCDIC characters or blanks.

2.2.21. File Addition (Column 66)

- Entry

Conditional; used with indexed sequential files only.

- Purpose

Indicates the addition of new records.

- Codes

A – New records are to be added.

Blank – New records are not to be added.

- Rules

  (1) This column is blank for tape, card, or printer files.

  (2) The file addition specification applies *only* to indexed sequential files and is defined according to the file type specified in Column 15, as well as the mode of processing. Table 2-3 indicates when an indexed sequential file may specify a file addition (A).

FILE TYPE	PROCESSING MODE	FILE ADDITION
Output File	Creating or extending file	blank
Output File	Adding records	A
Input File	Processing without updating and without adding records	blank
Input File	Processing and adding records	A
Update File	Processing and updating records	blank
Update File	Processing, updating, and adding records	A

Table 2–3. Summary of File Addition

2.2.22. Number of Tracks for Cylinder Overflow (Column 67)

- Entry

  Conditional; used with indexed sequential files only.

- Purpose

  Contains the number of tracks to be reserved for cylinder overflow.

- Codes

  0-8

- Rules

  (1) This column is blank for all other specifications.

2.2.23. Columns 68-69

These columns are not used, and may contain any EBCDIC characters or blanks.

2.2.24. Tape Rewind Option (Column 70)

- Entry

  Optional.

- Purpose

  This option is used during the execution of the object program and greatly simplfies tape handling procedures.

- Codes

9400 MODE	9200/9300 MODE	IBM 360/20 MODE	ACTION AT START OF PROCESSING	ACTION AT END OF PROCESSING
R	--	Blank	Rewind, No Lock	Rewind, No Lock
U or Blank	Blank	U	Rewind, No Lock	Rewind, Lock
N	--	--	No Rewind	No Rewind
K	--	N	Rewind, No Lock	No Rewind
L	--	--	No Rewind	Rewind, No Lock
M	N	--	No Rewind	Rewind, Lock

- Rules

  (1) This column is used for tape files only. For all other types of files this column must be left blank.

2.2.25. Columns 71-74

These columns are not used, and may contain any EBCDIC characters or blanks.

3. INPUT FORMAT SPECIFICATIONS FORM

3.1. GENERAL DESCRIPTION

The Input Specifications form (Figure 3-1) is used by the RPG program to do the following:

- Specify input records within an input file;
- Define input record sequence;
- Describe data field format and locations within the input record.

All Input Specifications for a given input file must be grouped together.

The entries for this form are divided into two categories:

(1) Record Identification (Columns 7-42) – These entries identify the input record by assigning the record type indicators, by specifying the input record codes, and by indicating the relationship of one record to another. A minimum of one record identification entry is required in each RPG program. Each record identification entry must be entered on a separate line above the field description entries for that record.

(2) Field Description (Columns 43-70) – These entries describe the individual fields of the input record that are used by the object program. Each entry is entered on a separate line below the applicable record identification entry.

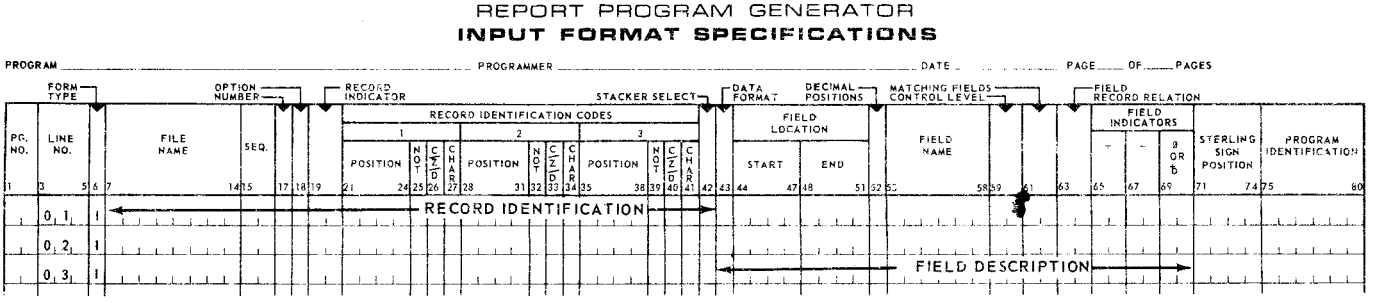


Figure 3-1. Input Format Specifications Form

3.2. RECORD IDENTIFICATION ENTRIES

The following descriptions explain the entries for the record identification portion of the Input Format Specifications form.

3.2.1. File Name (Columns 7-14)

- Entry

  Required.

- Purpose

  Assigns an identifying name to each input file. The file name entry used on the Input Format Specifications form must correspond to the file name entry on the File Description form. This enables the RPG program to gather all the pertinent information about a specific file.

- Rules

  (1) A file name must be specified for each input file, and it must correspond with the file name assigned on the File Description form.

  (2) The file name must be left-justified and the first character must be alphabetic. (Alphabetic characters are defined in 2.2.1.)

  (3) The remaining characters may be alphabetic or numeric.

  (4) A maximum of eight characters may be used for a file name but in 9400 mode and IBM 360 Model 20 mode, the first seven identify the file. For example, WORKFIL1 and WORKFIL2 are considered the same file name.

- Example

  Figure 3-2 illustrates file name entries.

3.2.2. Sequence (Columns 15-16)

- Entry

  Required.

- Purpose

  This field is used when the input records must follow a specified sequence in an input file. Figure 3-3 illustrates such a sequence. An out-of-sequence condition causes the halt indicator H0 to be turned on. The object program halts unless this indicator is turned off by the SETOF instruction (see 4.2.4.13). Records in a tape or disc file are treated the same way as records in a card file.

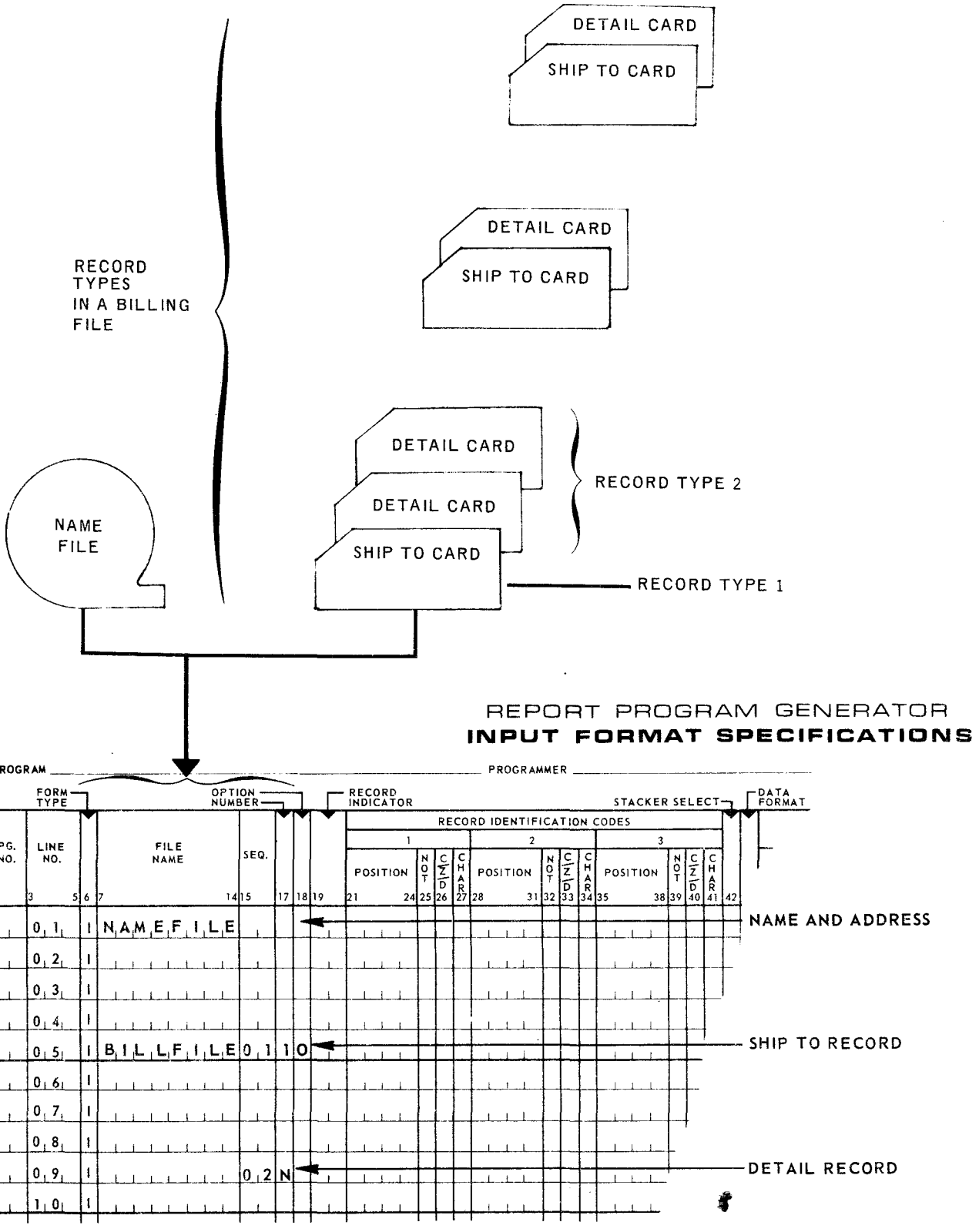


Figure 3–2. Example of File Name, Sequence, Number, and Option Entries

- Codes

  01-99 – Consecutive sequence of records.

  Two alphabetic characters – Indicates records do not have to be in sequence.

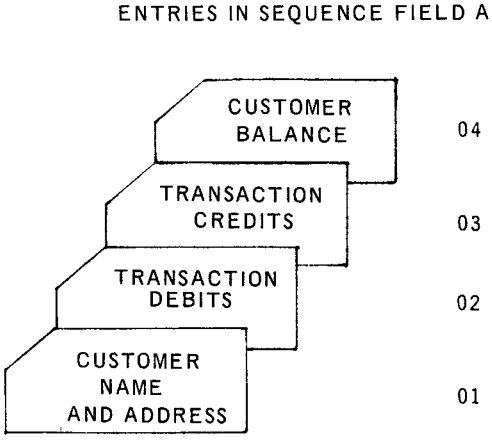


Figure 3–3. Card File Sequence

- Rules

  (1) Alphabetic characters must be entered in this field when the input records do not have to be in sequence; any two alphabetic characters may be used.

  (2) Records not required to be in sequence must be entered prior to those that must be in sequence.

  (3) Input records required to be in sequence should be listed in the same order as they appear in the input file.

  (4) The numeric sequence entries for each file requiring sequencing must be consecutive beginning with 01.

- Examples

  See Figure 3-2.

3.2.3. Number (Column 17)

- Entry

  Conditional; must be used when the sequence field contains a numeric entry.

- Purpose

  This field indicates whether one or more records of a specific type may appear together. Figure 3-4 provides an example of such a record type.

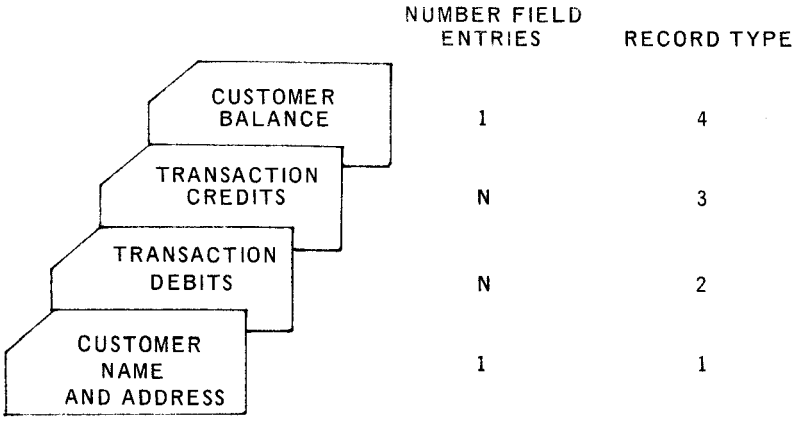


Figure 3–4. Record Types

- Codes

  1 – Indicates that only one of that record type can appear.

  N – Indicates that one or more of that record type may appear together.

  Blank – Indicates that the sequence field contains alphabetic characters.

- Rules

  (1) An entry must be made in this field when the corresponding sequence field contains numeric data.

  (2) This field must be blank when the sequence field contains alphabetic data.

- Example

  Figure 3-2 illustrates entries in this field.

3.2.4. Option (Column 18)

- Entry

  Conditional; used only when sequenced records are numeric.

- Purpose

  Indicates whether or not a specific record type must be present in the sequence of record types.

- Codes

  Blank – Indicates record must be present.

  0 – Indicates record need not be present.

- Rules

  (1) This column must be left blank if the record type must be present.

  (2) The letter O is entered when the presence of the record type within the control group is not mandatory or if the records are nonsequential.

- Examples

  Figure 3-2 illustrates the sample problem described in the following text.

  A billing file contains two different record types: ship to, and details. The ship to card is not required if the address is the same as the one specified in the name and address file. Figure 3-2 illustrates a typical portion of the card file and the record type entries required on the Input Format Specifications form. The file name BILLFILE is written only once in Columns 7-14. The two record types are specified on subsequent lines in the sequence in which they are to be read.

  A ship to card may or may not be present. It must be the first card, and is limited to one card per customer. The 01 in the sequence field indicates the order of sequence, the 1 in the number field specifies the number of record types allowable, and the letter O in the option field indicates that the record may or may not be present.

  The detail card is of the second record type, determined by the 02 entry in the sequence field. The N in the number field specifies that one or more of this record type may appear. The blank option entry indicates that this record type must be present.

3.2.5. Record Indicator (Columns 19-20)

- Entry

  Required; except in an AND relationship and optional in an OR relationship.

- Purpose

  This specification has the following functions:

  (1) To establish a two-digit code for each input record type. It is used in conjunction with the record identification codes entry (Columns 21-41). Record indicators simplify references to a particular record by supplying a two-digit code to take the place of the normal record identification.

  (2) To set (or turn on) an indicator in the object program each time a particular input record is read into the system. The object program processes certain specifications of the Calculation and Output Format forms, depending on the on or off condition of this indicator.

  This function is similar in operation to that of selectors on punch card machines. A record indicator permits certain operations indicated on the Calculation and Output Specifications forms to be performed.

These operations will be performed only if the particular record indicator written on the specification line is in the on state. To put this indicator in the on state, the system must read in that particular record type which has the same indicator specified in Columns 19-20 as its record indicator. Specifications associated with other record indicators in the off state are not performed.

- Codes

  01-99 – These numbers do not have to be assigned in sequence and are mutually exclusive.

- Rules

  (1) Each indicator code must have a unique number, from 01 through 99.

  (2) The numbers do not have to be in any sequence.

  (3) Normally, only one record indicator is on at one time; when a particular record indicator is turned on, all other record indicators are automatically turned off. However, when chaining files are being processed, resulting indicators may be turned on for the chaining file (also turned on for some or all of the chained files).

- Example

  Figure 3-5 illustrates record indicator entries.

REPORT PROGRAM GENERATOR
INPUT FORMAT SPECIFICATIONS

PROGRAM _____

FORM TYPE	PG. NO.	LINE NO.	FILE NAME	SEQ.	OPTION NUMBER		RECORD INDICATOR	RECORD ID POSITION	NOT	C/Z/D	CHAR	POS	
1	3	5 6	7 14	15	17	18	19	21 24	25	26	27	28	
F		01	NAMEFILE				10						RECORD INDICATOR 10 IS ASSIGNED TO THE NAME AND ADDRESS CARD.
F		02											
F		03	BILLFILE	01	1		11	80		C	E		SHIP TO RECORD
F		04											
F		05		02	N		12	80		D	5		DETAIL RECORD; THE 02 SPECIFIES THE SEQUENCE OF THIS INPUT FILE. N INDICATES ONE OR MORE RECORD TYPES MAY APPEAR TOGETHER. 12 IS THE ASSIGNED RECORD INDICATOR WHICH WILL BE TURNED ON IF THE NUMERIC PORTION IN COLUMN 80 IS EQUAL TO 5.
F		06											
F		07											
F		08											
F		09											
F		10											
F		11											
F		12											
F		13											

Figure 3-5. Example of Record Indicator and Record Identification Codes Entries

3.2.6. Record Identification Codes (Columns 21-41)

- Entry

Conditional; it is recommended that entries be made for each record type. However, if the user desires, the last record type described need only have sequence and record indicator specified. In particular, this means that:

(1) If all input records are to be processed alike, the record identification codes may be left blank.

(2) If the input file consists of many different record types, only some of which are to be processed, the record types to be processed can be listed first with record identification codes. All remaining record types can be grouped under one record indicator by not specifying record identification codes for this last indicator.

(3) If an input record is read and not identified, halt indicator H0 is turned on. The object program will stop unless this indicator is turned off by the SETOF instruction (see 4.2.4.13).

- Purpose

This section of the Input Format Specifications form identifies the different record types referenced by the record indicator. After identifying this record, all future references to the record are made using the record indicator.

- Codes

Three identifying codes per line are provided on the form, as indicated by the Columns 21 through 41 and headed 1, 2, and 3. More than three identifying codes may be specified on successive lines by using AND relationships, as described in 3.2.7. Since the three sets of entries are identical, only the first set of entries (Columns 21-27) is described. The entries consist of the following fields:

POSITION
NOT
C/Z/D
CHAR

3.2.6.1. POSITION (Columns 21-24)

- Entry

Required.

- Purpose

Indicates the card column or tape or disc record position that contains the identifying code for a specific record type.

- Codes

  0001-9999

- Rules

  (1) The entry must be right-justified.

  (2) Leading zeros may be omitted.

3.2.6.2. NOT (Column 25)

- Entry

  Conditional; see codes.

- Purpose

  This entry specifies whether the identifying code must be present or absent in the record.

- Codes

  N – Indicates that the specified position in the record must not contain the code to satisfy the identification requirement.

  Blank – Indicates that the position must contain the code to satisfy the requirement.

3.2.6.3. C/Z/D (Character, Zone, Digit) (Column 26)

- Entry

  Required.

- Purpose

  Specifies the portion of the identification code indicated in the position entry that must be examined for identification purposes.

- Codes

  C – The entire character is used for comparison and identification.

  Z – The zone portion of the character is used for comparison and identification.

  D – The digit portion of the character is used for comparison and identification.

- Method of Zone and Digit Testing

The method for zone and digit testing by the RPG object program depends on the mode of compilation selected in the RPG control card, Column 7 (see Appendix E).

(1) 9400 mode – *zones* and *digits*, as those words are used in punched-card processing systems, are compared, without reference to internal representation. The possible zones are:

12 zone (characters A-I, &, plus zero)
11 zone (characters J-R,–, minus zero)
0 zone (characters S-Z)
no zone (characters 0-9, blank)

All other characters are treated as having the same zone and the same digit.

(2) 9200/9300 mode – characters to be tested are translated to 9200/9300 compressed code. Zone testing is performed on bits 6 and 7 of the input and identification character; digit testing is performed on bits 0 through 5.

(3) IBM 360 Model 20 mode – EBCDIC zones and digits are compared, except for &, –, and blank, which are considered to have the same zones as A, J, and 1, respectively.

3.2.6.4. CHAR (Character) (Column 27)

- Entry

Required.

- Purpose

Identifies the character on the input specifications card to be compared with the specified columns of the input record.

- Rules

(1) The information written on the Input Specifications form is punched into cards. At record identification time in the object program operating cycle, a portion of Column 27 on the input specification card is compared with the same portion of the specified column of the input record. The portion that is involved in the comparison is a function of the C/Z/D specification and the mode of compilation selected in the RPG control card (see 3.2.6.3).

3.2.7. AND Relationship (Columns 14-16)

- Entry

Conditional.

- Purpose

  Allows for more than three codes to be used to identify any one record.

- Code

  AND – Placed in Columns 14-16 on the secondary record identification line.

- Rules

  (1) Columns 17-20 must be blank.

  (2) As many AND lines as necessary may be used to identify a record.

  (3) An AND line cannot be used if the preceding line contains no record identifications.

3.2.8. OR Relationship (Columns 14-15)

- Entry

  Conditional.

- Purpose

  The OR relationship is used to reduce the writing of specifications for records that are similar. The use of the relationship will be clearer after reviewing the example given for the field record relation field. An OR relationship can be applied under the following conditions:

  (1) When two different record types have identical fields located in the same positions, two record identification entries can use the same field description entries to describe the field.

  (2) When two records have the same fields but their field locations are not identical, then an OR relationship can be used in conjunction with a record indicator specified in the field record relation field (Columns 63-64). If the fields are not located in the same columns in both record types, they must be specified twice. The appropriate record indicator is indicated in the field record relation field.

  (3) When two records have common fields and different fields they can be related in an OR relationship. In doing this, each field unique to a record must be so identified in the field record relation entry. It is useful to use an OR relationship if the records contain more common fields than different fields.

- Code

  OR – Placed in Columns 14-15 on the secondary record identification lines.

- Rules

  (1) The record identification codes for one record in the OR relationship appear before the OR line on the input form; those for the other record appear on the OR statement line.

  (2) Columns 16-18 must be blank.

  (3) All records in an OR relationship must have the same sequence characteristics.

  (4) Two or more records may be specified in an OR relationship.

- Example

  See Figures 3-5 and 3-6. Figure 3-6 illustrates how the AND, OR, and NOT relationship may be used.

REPORT PROGRAM GENERATOR
INPUT FORMAT SPECIFICATIONS

PROGRAM _____ PROGRAMMER _____

PG. NO.	LINE NO.	FORM TYPE	FILE NAME	SEQ.	OPTION NUMBER		RECORD INDICATOR	STACKER SELECT	RECORD IDENTIFICATION CODES 1 POSITION	1 NOT	1 C/Z/D	1 CHAR	2 POSITION	2 NOT	2 C/Z/D	2 CHAR	3 POSITION	3 NOT	3 C/Z/D	3 CHAR	
1	3 5	6	7 14	15	17	18	19	21	24	25	26	27	28 31	32	33	34	35 38	39	40	41	42 43
	01	I	INVFILE	AA			14	5		C	1										
	02	I	OR					1		Z	-										
	03	I																			
	04	I																			
	05	I																			
	06	I		BB			10	80	N	D	4										
	07	I																			
	08	I																			
	09	I		CC			11	1	N	Z	&		2		C	D		3		C	E
	10	I	AND					4	C	D											
	11	I																			
	12	I																			
	13	I																			
	14	I																			
	15	I																			
	16	I																			

RECORD INDICATOR 14 IS TURNED ON IF EITHER A 1 IS PUNCHED IN COLUMN 5 OR AN 11 PUNCH (XZONE PUNCH) APPEARS IN COLUMN 1.

RECORD INDICATOR 10 WILL BE TURNED ON IF A 4 IS NOT PRESENT IN COLUMN 80.

INDICATOR 11 WILL BE TURNED ON IF THE FOLLOWING FOUR CONDITIONS ARE SATISFIED:
NO. 12 PUNCH
THE CHARACTER D IN COLUMN 2
THE CHARACTER E IN COLUMN 3
THE CHARACTER D IN COLUMN 4

Figure 3-6. Examples of AND, OR, and NOT Relationships

3.2.9. Stacker Select (Column 42)

- Entry

  Optional.

- Purpose

  Permits cards read on the Type 0704 Read/Punch (if used as an input file) to be selected into stackers.

- Codes

  2 – Select stacker.
  1 or blank – Normal stacker.

- Rules

  (1) A 2 entry causes cards of this type to be directed to the select stacker.

  (2) A blank, a 1 entry, or any other entry causes cards of this type to be directed to the normal stacker.

3.3. FIELD DESCRIPTION ENTRIES

The following descriptions explain the entries for the field description portion of the Input Format Specifications form. Field description specifications are always written on the lines immediately below the associated record descriptions.

3.3.1. Data Format (Column 43)

- Entry

  Conditional.

- Purpose

  Indicates to the system that the input data is in packed decimal format. The RPG program automatically converts numeric input data from unpacked format to packed decimal format.

- Codes

  P – Indicates packed input format.
  Blank – Indicates an unpacked input format (normal entry).

- Rule

  Column 43 must be blank for an alphanumeric field.

3.3.2. Field Location (Columns 44-51)

- Entry

  Required.

- Purpose

  Defines the location of each field in the input record. The field location entry is divided into two sections:

  (1) Start (Columns 44-47) – This specification contains the location of the first position of the field.

  (2) End (Columns 48-51) – This specification contains the location of the last position of the field.

- Codes

  0001-9999

- Rules

  (1) Both entries must be right-justified; leading zeros may be omitted.

  (2) The fields may be specified in any sequence.

  (3) The maximum length for numeric fields is 15 digits.

  (4) The maximum length for alphanumeric fields is 256 characters.

  (5) In all indexed sequential files, the key is considered part of the input record. All or part of the key may be specified as a field for processing. In unblocked, indexed sequential files the record key always starts in location 1 and the record data starts in 1 + KL, where KL is the Key Length specified in the File Description (Columns 29-30).

3.3.3. Decimal Position (Column 52)

- Entry

  Conditional; required only for numeric fields.

- Purpose

  Determines for the RPG program the number of decimal positions contained in the field location entry. Also causes the zone punches within the card or tape field specified to be removed, except in the rightmost position.

- Codes

  Blank – Indicates an alphanumeric field.
  0-9 – Indicates the number of decimal positions.

- Rules

  (1) The decimal position must be indicated if the field specified is to have calculations, edit functions, or zero suppression in the output.

  (2) A zero must be entered for all numeric fields that do not contain decimal positions.

  (3) If an entry is made in this column, the RPG considers the associated field to be numeric; if no entry is made, the field is considered alphanumeric.

3.3.4. Field Name (Columns 53-58)

- Entry

  Required.

- Purpose

Assigns a field name to a defined field so that the programmer does not have to remember the field positions of input records. All references to a field must be made by using its field name.

- Rules

(1) The name may contain a maximum of six characters; only alphabetic (A-Z, $, #, @, ?, %) and numeric characters are permitted.

(2) The first character of the name must be alphabetic and left-justified.

(3) Fields in two different records not having an OR relationship may use the same name but only if they have the same length and number of decimal positions.

(4) The contents of a defined and named field are replaced by new data when a record using the same name is read. Data that must be retained for the entire control group must be given a unique name.

3.3.5. Control Level (Columns 59-60)

- Entry

Conditional; must contain an entry if this field is used for control purposes.

- Purpose

Provides a convenient and simple method for specifying control functions. Nine control levels may be specified from low to high L1, L2, L3, ..., L9. An indicator is associated with each control level.

When control levels are used, a control break occurs. The value of the field causing the break is retained in a storage field associated with the level on which the break occurred. When a record containing a control field is read, the value of that field is compared with the value stored in the level storage associated with the level of control for the field. If the two are not equal, a control break on that level has occurred, and the level indicators for that level and all lower levels are set. The setting of these level indicators can be used to control calculation and output at both total and detail time. The level indicators are not reset until after detail time for the record causing the control break. However, total time is not executed until a second control break occurs.

- Codes

L1, L2, L3, ..., L9 – Indicate control levels from low to high.

- Rules

(1) Control fields may be either numeric or alphabetic; when numeric, zone bits are not taken into consideration in comparisons to determine a control break.

(2) Numeric control fields may be in packed or unpacked format.

(3) Control field specifications are permitted for all primary, secondary, and chained files.

3.3.5.1. Split Control Fields

- Entry

  Optional.

- Purpose

  Permits several fields in an input record to be specified as one control field called a split control field. The same control level may be specified for split or nonsplit control fields in various record types by use of the field record relation specification in conjunction with control level indicator entries.

- Codes

  L1, L2, L3, ..., L9 – Indicate control levels from low to high.

- Rules

  (1) The fields of a split control field do not have to occupy contiguous positions.

  (2) The following rules apply when using field record relation indicators with split control field specifications:

  – Split control fields of any one level are arranged in storage in the same sequence as they appear on the Input Specifications form.

  – The sum of the lengths of the split control fields must not exceed 256 bytes. For numeric fields, the packed length is used to calculate the sum of the lengths.

  – When a record with split control fields is specified with field record relation indicators, then only these fields are used whenever that record indicator is turned on.

  – The overall field length for each control level must be the same in each group. For example, if control level (L1) is composed of two fields with an overall length of 20 positions, then the same control level (L1) used with a field record relation indicator and composed of only one field must be 20 positions in length.

- Example

  (1) Figure 3-7 is an example of a split control field.

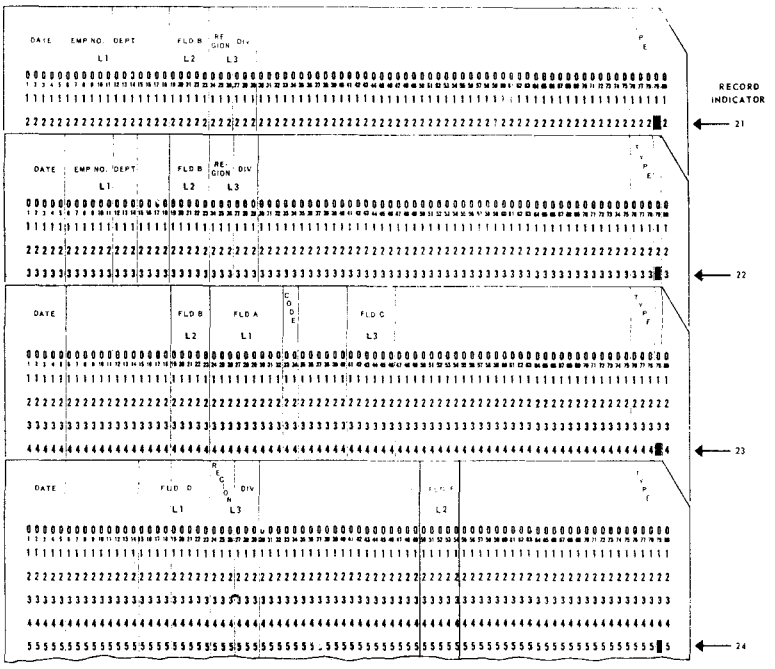
  (2) Figure 3-8 is an example of a multiple split control field.

REPORT PROGRAM GENERATOR
INPUT FORMAT SPECIFICATIONS

PROGRAM ____ PROGRAMMER ____ DATE ____ PAGE ____ OF ____ PAGES

PG. NO.	LINE NO.	FORM TYPE	FILE NAME	SEQ.	OPTION NUMBER	RECORD INDICATOR	STACKER SELECT	RECORD IDENTIFICATION CODES 1: POSITION	1: NOT	1: C/Z/D	1: CHAR	2: POSITION	2: NOT	2: C/Z/D	2: CHAR	3: POSITION	3: NOT	3: C/Z/D	3: CHAR	DATA FORMAT	FIELD LOCATION: START	FIELD LOCATION: END	DECIMAL POSITIONS	MATCHING FIELDS CONTROL LEVEL	FIELD NAME	FIELD RECORD RELATION	FIELD INDICATORS	STERLING SIGN POSITION	PROGRAM IDENTIFICATION
	01	I	UNIFILE AA			1		79			C2																		
	02	I																		1	3			DIVNO	L3				
	03	I																		4	6			DEPT	L2		SPLIT		
	04	I																		10	12			OFFICE	L2		CONTROL		
	05	I																		23	25			SECT	L2		FIELD		
	06	I																		32	37			EMPNO	L1				
	07	I																		52	55			CODE					
	08	I																		56	59	2		RATE					
	09	I	* THE THREE FIELDS DEPT, OFFICE AND SECT ARE NOT IN CONTIGUOUS																											
	10	I	* POSITIONS IN THE INPUT RECORD BUT ARE SPECIFIED WITH THE SAME																											
	11	I	* CONTROL LEVEL (L2). THUS, THE THREE FIELDS ARE HANDLED AS ONE																											
	12	I	* CONTROL FIELD.																											

Figure 3–7. Example of a Split Control Field



REPORT PROGRAM GENERATOR

INPUT FORMAT SPECIFICATIONS

PROGRAM _____ PROGRAMMER _____ DATE _____ PAGE ____ OF ____ PAGES

PG. NO.	LINE NO.	FORM TYPE	FILE NAME	OPTION NUMBER SEQ.	RECORD INDICATOR	STACKER SELECT	RECORD IDENTIFICATION CODES 1 POSITION	1 NOT	1 C/Z/D	1 CHAR	2 POSITION	2 NOT	2 C/Z/D	2 CHAR	3 POSITION	3 NOT	3 C/Z/D	3 CHAR	DATA FORMAT	FIELD LOCATION START	FIELD LOCATION END	DECIMAL POSITIONS	MATCHING FIELDS CONTROL LEVEL	FIELD NAME	FIELD RECORD RELATION	FIELD INDICATORS	STERLING SIGN POSITION	PROGRAM IDENTIFICATION
1	3 5	6	7 14	15 17 18	19		21 24	25	26	27	28 31	32	33	34	35 38	39	40	41	42	43 44 47	48 51	52	53 58	59 61	63	65 67 69	71 74	75 80
	01	I	PAYFILE AA		21		79		C	2																		
	02	I	OR		22		79		C	3																		
	03	I	OR		23		79		C	4																		
	04	I	OR		24		79		C	5																		
	05	I																		1	5		DATE					
	06	I																		6	11		EMPNO	L1				
	07	I																		12	14		DEPT	L1				
	08	I																		19	23		FLDB	L2				
	09	I																		24	26		REGION	L3				
	10	I																		27	29		DIV	L3				
	11	I																		24	32		FLDA	L1	23			
	12	I																		33	34		CODE		23			
	13	I																		41	46		FLDC	L3	23			
	14	I																		15	23		FLDD	L1	24			
	15	I																		50	54		FLDE	L2	24			
	16	I																		76	78		TYPE					
	17	I	*THE SAME CONTROL LEVEL MAY BE SPECIFIED FOR SPLIT OR NONSPLIT																										
	18	I	*CONTROL FIELDS IN VARIOUS RECORD TYPES BY USING THE FIELD																										
	19	I	*RECORD RELATION SPECIFICATION IN CONJUNCTION WITH CONTROL LEVEL																										
	20	I	*INDICATORS.																										

Figure 3–8. Example of Multiple Split Control Field Specifications

3.3.6. Matching Fields (Columns 61-62)

- Entry

  Optional.

- Purpose

  Provides the object program with the ability to match or to chain the records of one file with those of another file and to check sequence if only one file is specified.

- Codes

  M1, M2, ..., M9 – Nine matching field indicators designate low to high order.

- Rules

  (1) A maximum of nine fields may be matched in one operation.

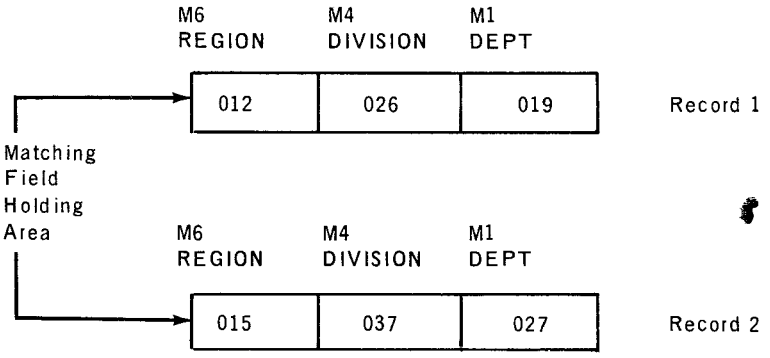
  (2) The fields to be matched do not have to occupy the same position in all files or all record types within a file. Fields designated by the same matching fields indicator need not have identical field lengths in all files.

  (3) The total length of matching fields must be identical for all record types of a file, and must not exceed 256 bytes.

  (4) If a record with matching fields is specified with and without field record relation indicators, then only the fields conditioned by a field record relation indicator are used when that record indicator is turned on. The matching fields without a field record relation indicator are ignored. These fields may be specified in any order on the Input Specifications form.

  (5) Matching fields may be packed or unpacked numeric or alphabetic; when numeric, zones or signs are not taken into consideration to determine match or sequence.

  (6) When sequence checking, the data of the fields are moved to the matching field holding area. The compare operation is made on all fields at the same time. The fields are placed in adjacent locations, starting with the highest numbered field (M9, if present) and continuing with the successively lower-numbered fields.



(7) If the sequence of the matching fields is not the same as the internal collating sequence of the 9400, the RPG program can provide an automatic exit to an external user subroutine with the predefined label ALTSEQ that translates the matching fields to the proper collating sequence. This option is selected by entering an A in Column 26 of the RPG control card (see Appendix E).

- Example

Figure 3-9 is an example of matching fields.

REPORT PROGRAM GENERATOR
INPUT FORMAT SPECIFICATIONS

PROGRAM ____ PROGRAMMER ____ DATE ____ PAGE ___ OF ___ PAGES

PG. NO.	LINE NO.	FORM TYPE	FILE NAME	OPTION NUMBER SEQ.	RECORD INDICATOR	RECORD IDENTIFICATION CODES: 1 POSITION	1 NOT	1 C/Z/D	1 CHAR	2 POSITION	2 NOT	2 C/Z/D	2 CHAR	3 POSITION	3 NOT	3 C/Z/D	3 CHAR	STACKER SELECT	DATA FORMAT	FIELD LOCATION START	FIELD LOCATION END	DECIMAL POSITIONS	MATCHING FIELDS CONTROL LEVEL	FIELD NAME	FIELD RECORD RELATION	FIELD INDICATORS	STERLING SIGN POSITION	PROGRAM IDENTIFICATION
1	3 5	6	7 14	15 17	18 19	21 24	25	26	27	28 31	32	33	34	35 38	39	40	41	42	43	44 47	48 51	52	53 58	59 61	63	65 67 69	71 74	75 80
	01		SSBCARD	AA	10	80		C	1																			
	02		OR		11	80		C	2																			
	03																			2	6		DATE					
	04																			7	27		DESCR					
	05																			28	35		PARTNO L1 M1					
	06																			36	38		PREFIX L2 M2					
	07																			39	41		RUN L2 M2					
	08																			50	55		AMT					
	09		ISSUE	BB	12	80		C	3																			
	10																			2	6		DATE					
	11																			7	27		DESCR					
	12																			28	35		PARTNO L1 M1					
	13																			60	62		PREFIX L2 M2					
	14																			50	55		AMT					
	15		RECEIPT	CC	13	80		C	4																			
	16																			2	6		DATE					
	17																			7	27		DESCR					
	18																			28	35		PARTNO L1 M1					
	19																			60	62		PREFIX L2 M2					
	20																			50	55		AMT					
	21																											
	22		* THREE INPUT FILES ARE USED ABOVE, AND THE FIELD PREFIX IS																									
	23		* LOCATED IN DIFFERENT POSITIONS ON TWO CARDS AND THE FIELD																									
	24		* LENGTHS ARE THE SAME.																									

MASTER STOCK STATUS BALANCE CARD

DETAIL ISSUE CARD

DETAIL RECEIPT CARD

Figure 3–9. Example of Matching Fields

3.3.6.1. Matching Record Indicator – Multifile Input

- Purpose

The function of matching records of one file, called primary, against the records of another file, called secondary, has an associated internal indicator called the matching record (MR) indicator. The MR indicator is used to control operations specified on the Calculation and Output Format Specifications forms.

The MR indicator is turned on for a control group when a secondary file record matches a primary file record. All primary records of a matched control group are processed before the first secondary record. The MR indicator is on during the complete processing of the group and is turned off at the completion of total time for that group.

If there is no match of the secondary file to the primary, the MR indicator is not turned on. Thus, the MR can be used to control operations indicated on the Calculation and Output Format Specifications forms. It can be used to select processing on unmatched primary and secondary cards. If no match is found on the matching fields, the object program processes the next record in the sequence.

- Rules

(1) The matching fields specification may still be used if only some of the record types in the file contain the fields necessary for matching. The record types without matching fields contain blanks in the matching fields specification on the input form. The blank field indicates to the RPG that these records are not to be checked for a matching field. The MR indicator is turned off when these record types are processed.

(2) Nine input files may be used with the matching fields specification. This multifile input is accomplished through the use of a combination of input devices: card reader, read/punch, tape units, and disc units. The relative priority of the secondary files is determined by their sequence on the File Description form. The MR indicator is turned on when a record in a secondary file matches a record from the primary file.

(3) The total length of matching fields must be identical for all record types of a file.

(4) Matching fields may be packed or unpacked numeric or alphabetic; when numeric, zones or signs are not taken into consideration to determine match or sequence.

(5) If all the matching record fields of a given record contain decimal or binary zeros (represented by a 12-0-9-8-1 multipunch on a punched card), there is a possibility that the MR indicator may be set when it is not expected to be.

(6) Secondary files must appear in the same order on the Input Format Specifications form as they appear on the File Description form.

- Examples

(1) Card File to Card File Matching

Figure 3-10 illustrates an inventory application in which a master file of stock status balance cards (SSBC) is being matched against a detail transaction deck for updating. Each group of detail cards (transactions) must have a matching master card (stock status balance card) in order to be processed. Nonmatched detail cards (issue cards coded 32) are to be selected for special processing.

The input specifications for this inventory application were shown in Figure 3-9. The file that is listed first on the form (the SSBCARD file) is called the primary file. The matching record indicators, M1 and M2, cause each transaction card read to be compared with the master card just read.

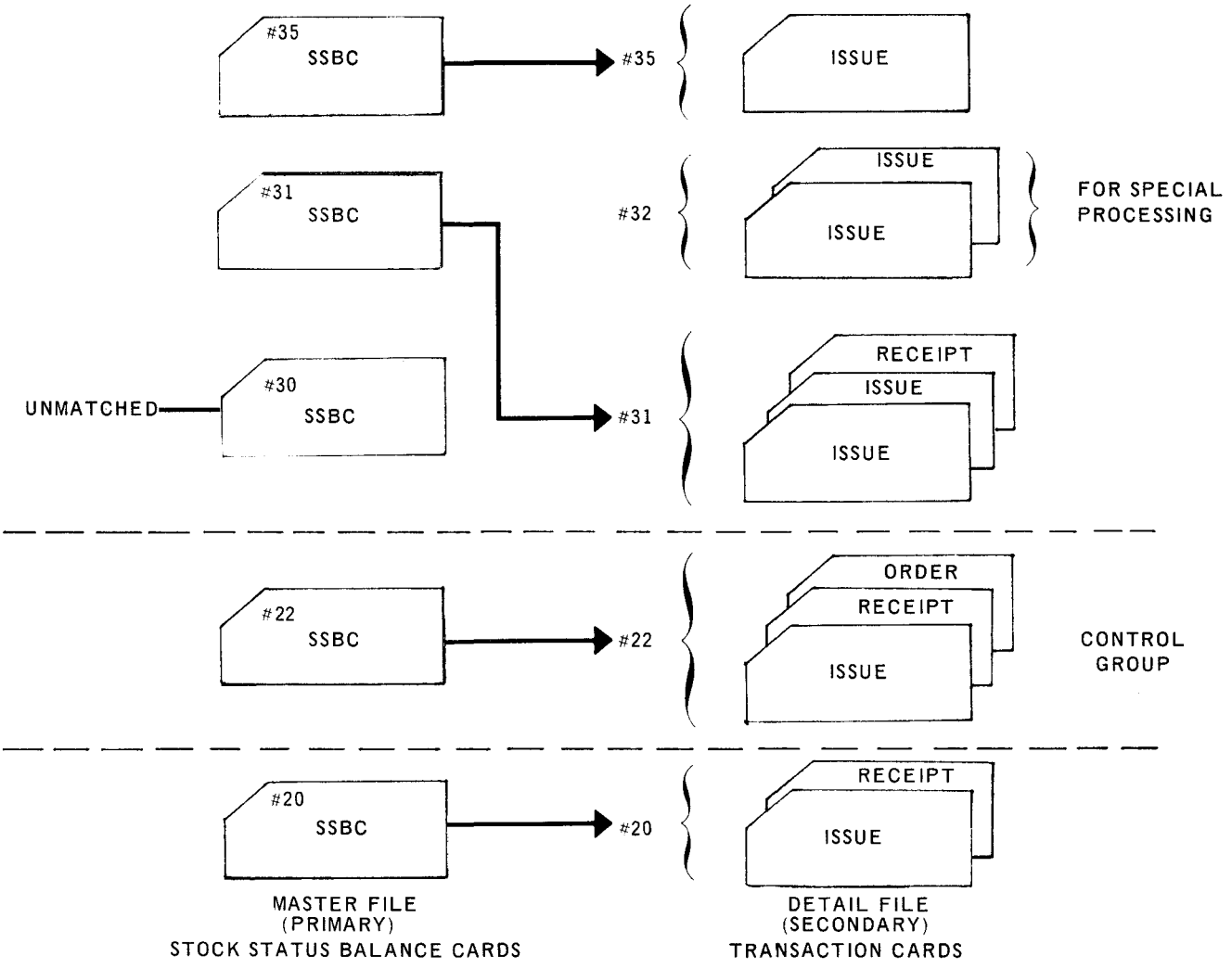


Figure 3-10. Example of Card File to Card File Matching

(2) Tape File/Disc File to Card File Matching

Figure 3–11 illustrates an inventory application in which a master file of stock status balance records (SSBR) is being matched against a detail transaction card deck for updating. Each group of detail cards (transactions) must have a matched master record in order to be processed. Unmatched detail cards are to be selected from the file as noted in the example by #32.

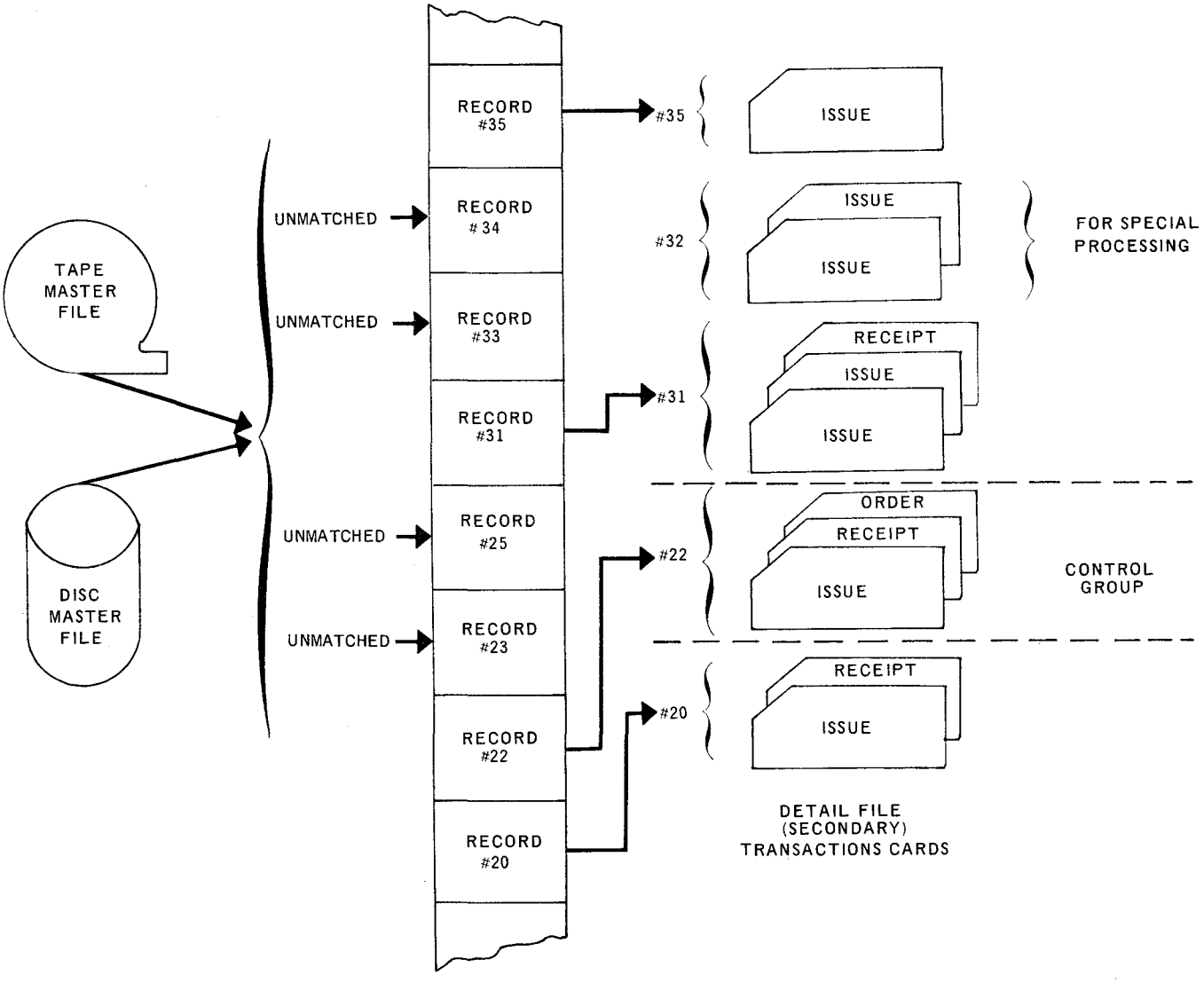


Figure 3–11. Example of Tape File/Disc File to Card File Matching

3.3.6.2. Sequence Checking

- Purpose

The matching fields indicator may be used to indicate sequence checking when one input file is specified. This function is generally associated with card files. The specifications of M1 through M9 cause each field of a record so designated by this indicator to be compared for sequence with the field previously read. If an out-of-sequence condition occurs (the sequence is specified in Column 18 of the File Description form), halt indicator H0 is turned on and processing stops unless the indicator is turned off by the SETOF operation (see 4.2.4.13).

- Example

Figure 3-12 is an example of sequence checking three input fields.

REPORT PROGRAM GENERATOR
INPUT FORMAT SPECIFICATIONS

PROGRAM _____ PROGRAMMER _____ DATE _____ PAGE ____ OF ____ PAGES

PG. NO.	LINE NO.	FORM TYPE	FILE NAME	SEQ.	OPTION NUMBER	RECORD INDICATOR	RECORD IDENTIFICATION CODES 1: POSITION	1: NOT	1: C/Z/D	1: CHAR	2: POSITION	2: NOT	2: C/Z/D	2: CHAR	3: POSITION	3: NOT	3: C/Z/D	3: CHAR	STACKER SELECT	DATA FORMAT	DECIMAL POSITIONS	FIELD LOCATION START	FIELD LOCATION END	MATCHING FIELDS CONTROL LEVEL	FIELD NAME	FIELD RECORD RELATION	FIELD INDICATORS +	FIELD INDICATORS −	FIELD INDICATORS 0 OR b	STERLING SIGN POSITION	PROGRAM IDENTIFICATION
	01	I																													
	02	I																				2	6		DATE						
	03	I																				7	27		DESCR						
	04	I																				28	35		PARTNO	M1					
	05	I																				36	38		PREFIX	M2					
	06	I																				39	41		RUN	M3					
	07	I																				42	46		ONHAND						
	08	I																				47	51		ORDER						
	09	I																				52	56		MIN						
	10	I																				57	61		MAX						
	11	I																													
	12	I	*, M3 SPECIFIES RUN, THE HIGH ORDER FIELD; PREFIX IS THE INTER-																													
	13	I	*, MEDIATE FIELD M2, AND PARTNO IS THE LOW ORDER FIELD TO BE																													
	14	I	*, CHECKED FOR SEQUENCE. A COMPARE OPERATION IS PERFORMED ON ALL																													
	15	I	*, THREE FIELDS AT THE SAME TIME.																													

Figure 3-12. Example of Sequence Checking with Matching Fields Specification

3.3.6.3. Chaining Fields

- Entry

Conditional; used with chaining files.

- Purpose

Permits chaining fields for multiple input files. This allows a user to link or chain a record of one file to a record of another file. Up to nine chaining fields may be specified.

- Codes

  C1, C2, C3, ..., C9 – Identifies a chaining field in a record of one file with a chained field of a record in an indexed sequential file. Each code is associated with one field in a record.

- Rules

  (1) A field used as both a matching field and a chaining field must be defined twice using two different field names.

  (2) A chaining field can be composed of more than one field in a record (split chaining field); the same chaining code is specified for each such field. The fields need not be adjacent.

3.3.7. Field Record Relation (Columns 63-64)

- Entry

  Conditional:

  – Used in conjunction with entries for records in an OR relationship;

  – Used with split control fields;

  – Used in conjunction with chaining fields.

- Codes

  01-99 – These numbers do not have to be assigned in sequence and are mutually exclusive.

- Purpose

  Designates the record containing this field by referencing the appropriate record indicator. Used when input records in an OR relationship do not have identical data or control field locations. The entries are more completely described in 3.3.5.1, and 3.2.8.

  With chaining fields (C1 to C9), this field is used to selectively control chaining operations. If a chaining field is specified on a field description line and the field record relation is blank, the chained record is obtained when the record type is present. However, if a resulting indicator is placed in field record relation, then the chained record is obtained only if the record type is present. A further use of field record relation is to control two chained files with one chaining field. The chaining field should be specified twice on the Input Specifications form, each line having a separate resulting indicator. The chained record obtained depends on the on or off status of the resulting indicator.

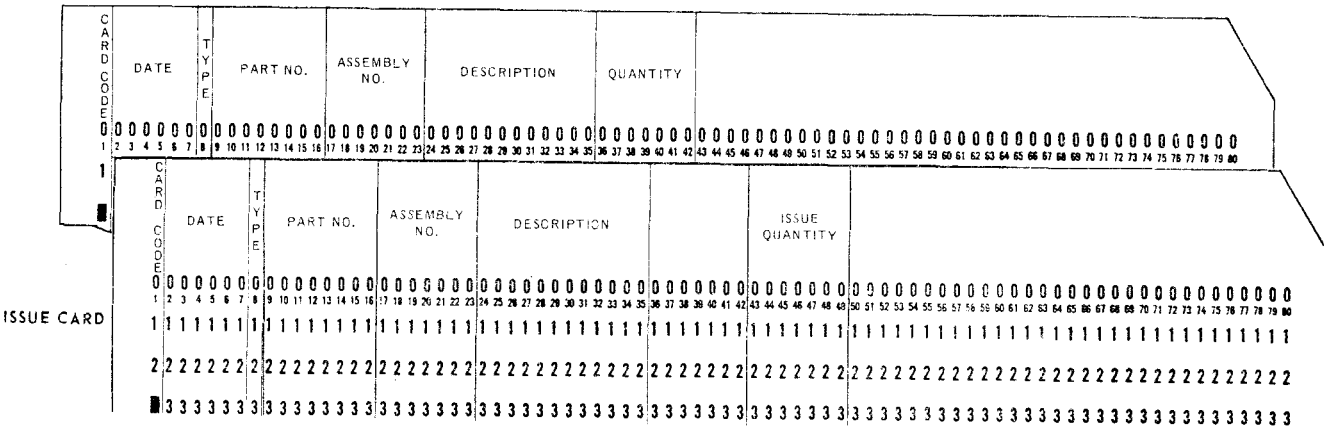
- Rules

  (1) When two records have identical fields, but the locations of the fields in the records are not identical, only one of the two sets of field specifications needs to be modified by a field record relation indicator.

- Example

Consider an inventory application which uses a receipt card and an issue card. Both cards contain the date, type, part number, assembly number, and description in Columns 2-35. The only distinction between the two cards is that the receipt card contains the field QUANTITY in Columns 36-42, and the issue card contains the field ISSUE QUANTITY in Columns 43-49. Figure 3-13 illustrates how the OR relationship is used in conjunction with the field record relation field to simplify input specifications. First, a record indicator is specified for each card. The OR relationship allows the fields to be described once for both cards. To relate a field to one specific card, the record indicator of that card is entered in the field record relation column on the same line with that field. In this example, the QTY field belongs only to the field record relation column on the line with QTY. Similarly, 30 in the field record relation column relates ISSUE to the issue card.

RECEIPT CARD



REPORT PROGRAM GENERATOR
INPUT FORMAT SPECIFICATIONS

PROGRAM _____ PROGRAMMER _____ DATE _____ PAGE ____ OF ____ PAGES

PG. NO.	LINE NO.	FORM TYPE	FILE NAME	SEQ.	OPTION NUMBER	RECORD INDICATOR	RECORD IDENTIFICATION CODES 1: POSITION	1: NOT	1: CHAR	2: POSITION	2: NOT	2: CHAR	3: POSITION	3: NOT	3: CHAR	STACKER SELECT	DATA FORMAT	DECIMAL POSITIONS	FIELD LOCATION START	FIELD LOCATION END	MATCHING FIELDS CONTROL LEVEL	FIELD NAME	FIELD RECORD RELATION	FIELD INDICATORS +	FIELD INDICATORS −	FIELD INDICATORS & OR b	STERLING SIGN POSITION	PROGRAM IDENTIFICATION
	01	I	INV			29	1		C2																			
	02	I	OR			30	1		C3																			
	03	I																	2	7		DATE						
	04	I																	8	8		TYPE						
	05	I																	9	16		PARTNO						
	06	I																	17	23		ASSEMB						
	07	I																	24	35		DESC						
	08	I																	36	42		QTY		29				
	09	I																	43	49		ISSUE		30				
	10	I																										
	11	I	* LINE 8 IS THE RECEIPT RECORD AND RELATES TO LINE 1.																									
	12	I	* LINE 9 IS THE ISSUE RECORD AND RELATES TO LINE 2.																									

Figure 3–13. Example of Field Record Relation Entries

3.3.8. Field Indicators (Columns 65-70)

- Entry

  Optional.

- Purpose

  Allows testing the status of a field when it is read into the system. Depending on the results of the test (positive, negative, blank, or zero status), a field indicator is turned on that can be used to control calculation and output specifications or stop processing the object program. Each field indicator is generally related to a record type. The setting of these indicators (on or off) is not altered until a record of the same type is subsequently read or until the indicator number is defined in some other specification. Several field indicators can be on at the same time.

NOTE: Defining an indicator means to specify it on the Input form in the record indicator or field indicator columns. *Using* an indicator means to specify it in the field output indicators field on the Output form or in the indicators field (Columns 9-17) on the Calculation form. When an indicator is defined, it is set or turned on, if certain conditions are present. An indicator is used when the condition of that indicator is tested and the results of the test are used to control further processing of the object program.

- Codes

There are three types of field status codes and two types of field indicator codes.

(1) The field status indicator codes are as follows:

Plus – The value of a numeric field is greater than zero.

Minus – The value of a numeric field is less than zero.

Zero or Blank – The value of a numeric field is all zeros; for an alphabetic field it is all blanks. Unpunched columns of unpacked numeric fields are considered zeros. In 9400 and IBM 360 Model 20 modes of compilation, any zero or blank indicator is turned on at the start of execution of the object program.

(2) The field indicator codes are as follows:

Numeric indicators – A two-digit numeric value, 01-99, may be assigned. It is used to control program calculations and output. This operation is accomplished by testing the selected indicator to determine its state. These indicators can be defined more than once on a form. The second specification of the indicator resets it from its previous state.

Halt indicators – These indicators are designated from H0 through H9 and halt the processing of the object program when error conditions (as determined by the programmer) are detected. Halt indicators can also be used to control calculation and output specifications. If one of these indicators has been turned on during processing of a record, then the object program is stopped when the processing of that record is completed. Processing is not interrupted if a halt indicator that was on is turned off. If a halt indicator appears in columns 69 and 70 in 9400 or IBM 360 Model 20 mode programs, the program stops after the first header or detail output and before the first input record is read.

The H0 indicator is also used by the RPG program for internal operations.

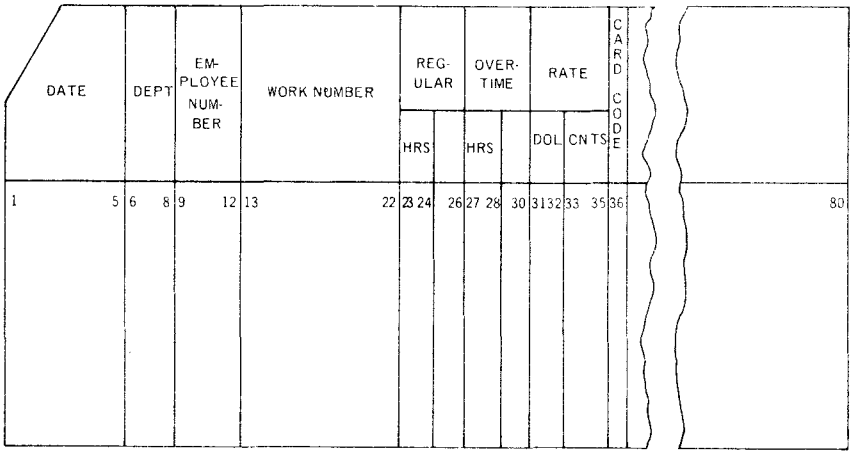
- Rules

(1) The plus and minus fields of the field indicators specification must be blank for alphanumeric fields.

(2) The halt indicator must be turned off or the program terminates before the next input record is read. (See Columns 54-59 on the Calculation Specifications form.)

- Example

Figure 3-14 represents the field identification entries necessary to describe completely the card layout shown. Each field is designated by a name written in Columns 53-58. The location of card fields is specified in Columns 44-51. Decimal positions must be specified for all numeric fields. Regular hours and overtime hours have two decimal positions. Rate has three decimal positions. Department, work number, and employee number are specified as numeric with no decimal positions.



REPORT PROGRAM GENERATOR
INPUT FORMAT SPECIFICATIONS

PROGRAM _____ PROGRAMMER _____ DATE _____ PAGE ___ OF ___ PAGES

PG. NO.	LINE NO.	FORM TYPE	FILE NAME	OPTION NUMBER SEQ.	RECORD INDICATOR	RECORD IDENTIFICATION CODES 1 POSITION	1 NOT	1 C/Z/D	1 CHAR	2 POSITION	2 NOT	2 C/Z/D	2 CHAR	3 POSITION	3 NOT	3 C/Z/D	3 CHAR	STACKER SELECT	DATA FORMAT	FIELD LOCATION START	FIELD LOCATION END	DECIMAL POSITIONS	FIELD NAME	MATCHING FIELDS CONTROL LEVEL	FIELD RECORD RELATION	FIELD INDICATORS +	FIELD INDICATORS -	FIELD INDICATORS ∅ OR b	STERLING SIGN POSITION	PROGRAM IDENTIFICATION
1	3 5	6	7 14	15 17	18 19	21 24	25	26	27	28 31	32	33	34	35 38	39	40	41	42	43	44 47	48 51	52	53 58	59	61	63	65	67	69	71 74	75 80
01	I		JOBCARD	AA	10	35		C1																							
02	I																			1	5		DATE							
03	I																			6	8	0	DEPT							
04	I																			9	12	0	EMPNO	L1	M1					
05	I																			13	22	0	WORKNO							
06	I																			23	26	2	REGHRS							
07	I																			27	30	2	OTHRS						34	
08	I																			31	35	3	RATE							
09	I																			36	36		CARD							
10	I	*	THE JOB CARDS SHOULD BE IN SEQUENCE BY EMPLOYEE NUMBER (EMPNO).																													
11	I	*	THIS IS SPECIFIED BY WRITING M1 UNDER MATCHING FIELDS ON THE																													
12	I	*	ENTRY LINE FOR EMPNO. BY SPECIFYING L1 UNDER CONTROL LEVEL ON																													
13	I	*	THE SAME LINE, THE INDICATOR L1 IS TURNED ON WHEN A CHANGE IN																													
14	I	*	EMPLOYEE NUMBER OCCURS. L1 ONLY INDICATES THAT A CHANGE HAS																													
15	I	*	OCCURED IN THIS CONTROL FIELD (EMPNO). AN M1 IS ENTERED IN																													
16	I	*	MATCHING FIELDS TO INDICATE THAT A SEQUENCE CHECK ON EMPLOYEE																													
17	I	*	NUMBER IS DESIRED. FIELD INDICATOR 34 WILL BE TURNED ON IF THE																													
18	I	*	OTHRS FIELD IS ZERO. INDICATOR 34 CAN BE USED TO ELIMINATE																													
19	I	*	OVERTIME PROCESSING WHERE THERE IS NO ENTRY IN THE OTHRS FIELD.																													

Figure 3-14. Example of Field Identification Entries

3.3.9. Sterling Sign Position (Columns 71–74)

- Entry

  Conditional; must be used for programs processing sterling currency amounts.

- Purpose

  Alerts the RPG program when currency amounts are expressed in sterling, and provides a means to convert these amounts. The user should read Appendix D before using this specification.

- Codes

  S – Entered in Column 74 if the sterling input field is signed as specified in the Format 2 section of Appendix D.

  If the sterling input field does not conform to the sign specifications as described in Appendix D, then the position of the sign must be entered right-justified in Columns 71-74.

- Rules

  (1) Decimal position (Column 52) must be specified with one of the digits, 0, 1, 2, or 3 to indicate the number of required decimal positions.

  (2) The same field name may not be used for both a decimal and a sterling field.

  (3) Columns 71–74 must be left blank if sterling currency is not being processed.

4. CALCULATION SPECIFICATIONS FORM

4.1. GENERAL DESCRIPTION

The Calculation Specifications forms specify the operations, either computational or logical, that must be performed by the object program on the input data or other data previously generated by the program. The general rules for preparing Calculation Specifications forms are as follows:

(1) Only one operation may be entered on each specifications line.

(2) All detail calculations must be entered before total calculations.

(3) Operations must be entered in the sequence in which they are to be executed.

The Calculation Specifications form (Figure 4-1) is divided into the following three sections:

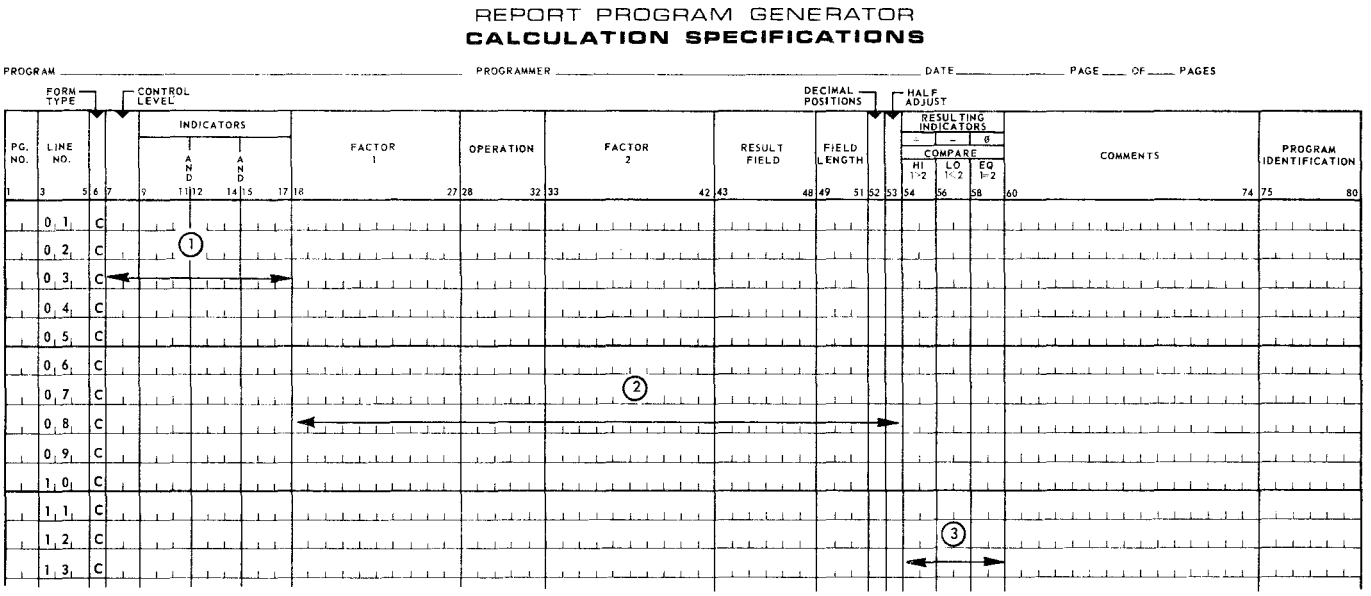


Figure 4-1. Calculation Specifications Form

(1) Control indicators (Columns 7-17) define the record types on which the calculations are to be performed and under which conditions they are to be performed. If calculations are to be performed on the basis of a control break, this specification determines the exact control level at which the calculation will be executed.

(2) Columns 18-53 determine the type of calculations to be performed, such as add, subtract, multiply, divide, and compare. The names of fields to be operated upon are specified (factor 1 and factor 2) along with the name of the result field where the answer is to be stored.

(3) Columns 54-59 permit the results of calculations to be tested, thereby controlling subsequent calculations or output specifications.

4.2. FORM ENTRIES

The following descriptions explain the entries for the calculation specifications and their relation to the RPG program.

4.2.1. Control Level (Columns 7-8)

- **Entry**

  Optional.

- **Purpose**

  Calculations may be executed at either detail time or total time. The control level specification indicates that a calculation line is to be executed at total time if the specified control level indicator is on. An entry in the control level field governs when the computation indicated in Columns 18-59 of the specification line is to be executed.

  A test for a control break occurs after a record is read into the system. If a control break has occurred, total calculations are performed before the card causing the control break is processed. When a control break occurs, the indicator for the new control level and all lower order control levels are turned on at the same time. A control level indicator which is turned on by a control break is on during total time and remains on for the following detail time, including computation and output of the detail record.

- **Codes**

  L1-L9–Control break indicators 1 through 9, from lowest level to highest level.

  L0 (Level Zero)–The level zero indicator is always on during the operation of the program, unless it is turned off with a SETOF instruction (see 4.2.4.13).

  Even though a control break has not occurred, total computations may be performed by specifying L0 (level zero) for the control level entry. The L0 condition may be used to produce end of page totals even though no control break has occurred.

  LR (Last Record)–The last record indicator is turned on after the last input record has been processed and after all applicable outputs have been printed and punched. The control level indicators L1-L9 are also turned on at this time.

- **Rules**

  (1) If no control level is indicated (blanks in Columns 7-8), the calculation is executed at detail time.

  (2) If the program turns off indicator L0, it is not turned on again when a level break occurs. It must be turned on, if required, by a SETON operation.

■ Examples

Figure 4–2 illustrates the functions of control level entries.

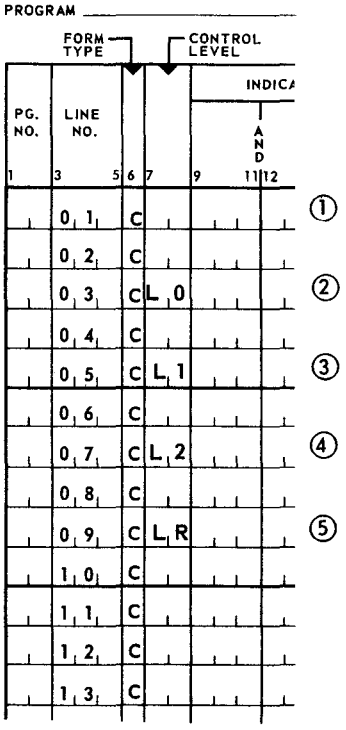


Figure 4–2. Example of Control Level Entries

① Blanks in the control level columns indicate this calculation will be performed at detail time provided the proper indicators are set.

② An L0 control level indicates that the calculation is to be performed at total time. (L0 is always set.)

③ An L1 control level indicates that the calculation is to be performed at total time when control level 1 is set.

④ An L2 control level indicates that the calculation is to be performed at total time when control level 2 is set. Note that if L2 is set, so is L1.

⑤ An LR control level indicates that the calculation is to be performed at total time for the last record in the input file. When the LR indicator is set, so are level indicators L1–L9.

4.2.2. Indicators (Columns 9-17)

- Entry

  Optional.

- Purpose

  Indicates the conditions that control the calculation to be performed.

- Codes

  Indicator specifications are grouped into the following categories:

  (1) If the control level and indicator specifications are blank, the calculation indicated on that specification line is executed for each detail record.

  (2) A record indicator code (01-99) indicates the record type on which the specified calculation is made. The calculation cannot be performed on any other record type.

  (3) A field indicator code (01-99) controls the calculation based on the status of an input field. A field indicator is turned on in the input section based on a plus, minus, zero, or blank field.

  (4) A resulting indicator code (01-99) governs computation on a current specification line based on conditions that occurred in a previous calculation.

  (5) A control level indicator (L1-L9) specified in conjunction with a record indicator permits computation to be executed only at the detail time of the record which caused the control break. This permits a calculation to occur only once, on the first record, after the specified control break.

  (6) The halt indicators (H0-H9), when specified on a calculation line without an N prefix, will cause the termination of a program or suppress a calculation when an error has been detected. The halt indicators may also be used with the N prefix.

  (7) The overflow indicators (OA, OB, OC, OD, OE, OF, OG, and OV) permit a calculation specification line to be executed only if a page overflow has occurred.

  (8) The matching record indicator code (MR) indicates that the calculation is performed only if there is a matching record in a secondary input file.

  (9) The indicator L0 indicates that the calculation indicated on that specification line is executed for each detail or total record.

- Rules

(1) Indicators can be specified at either detail time or total time.

(2) All total calculations are bypassed until the first control break has occurred.

(3) A maximum of three indicators plus any level indicator may be defined for a single calculation specification line. The indicators are tested in an AND relationship, which means that all conditions set up by the indicator entries for a given specification line must be met before the indicated calculation is performed.

(4) If columns 7 through 17 are blank (spaces), the associated calculation is performed at every detail time.

(5) A calculation specification line may contain a combination of the various indicators. If the control of an operation is determined by an indicator not being on, the indicator code must be preceded by the letter N.

(6) To avoid rewriting the indicators when several calculations are to be performed under the same conditions, an asterisk (*) may be entered in column 11. A line with an asterisk in Column 11 is executed if the conditions were met for the preceding line. Also, once an asterisk condition has been set up for a series of lines, more stringent conditions can be put on any remaining subset of those lines.

REPORT PROGRAM GENERATOR
CALCULATION SPECIFICATIONS

PROGRAM _____ PROGRAMMER _____ DATE _____ PAGE ___ OF ___ PAGES

PG. NO.	LINE NO.	FORM TYPE	CONTROL LEVEL	INDICATORS: AND	INDICATORS: AND	FACTOR 1	OPERATION	FACTOR 2	RESULT FIELD	FIELD LENGTH	DECIMAL POSITIONS	HALF ADJUST	RESULTING INDICATORS: COMPARE HI 1>2	RESULTING INDICATORS: COMPARE LO 1<2	RESULTING INDICATORS: COMPARE EQ 1=2	COMMENTS	PROGRAM IDENTIFICATION
1 3	5	6	7	9 11	12 14	15 17	18 27	28 32	33 42	43 48	49 51	52	53	54 56	56 58	60 74	75 80
	01	C		10													
	02	C		*													
	03	C	*	LINES 1 AND 2 WILL			BE EXECUTED IF	INDICATOR	10	IS ON							
	04	C		*	11												
	05	C		*													
	06	C	*	LINES 4 AND 5 WILL			BE EXECUTED IF	INDICATORS	10 AND 11	ARE ON							

(7) Indicators are tested only when they are explicitly specified. An asterisk in Column 11, therefore, does not cause indicators to be retested.

REPORT PROGRAM GENERATOR
CALCULATION SPECIFICATIONS

PROGRAM _____ PROGRAMMER _____ DATE _____ PAGE ___ OF ___ PAGES

PG. NO.	LINE NO.	FORM TYPE	CONTROL LEVEL	INDICATORS AND	INDICATORS AND	FACTOR 1	OPERATION	FACTOR 2	RESULT FIELD	FIELD LENGTH	DECIMAL POSITIONS	HALF ADJUST	RESULTING INDICATORS +	RESULTING INDICATORS −	RESULTING INDICATORS 0	COMMENTS	PROGRAM IDENTIFICATION
													COMPARE HI 1>2	COMPARE LO 1<2	COMPARE EQ 1=2		
1	3 5	6	7	9 11	12 14	15 17 18 27	28 32	33 42	43 48	49 51	52	53	54	56	58	60 74	75 80
	0 1	C		1 0													
	0 2	C		*													
	0 3	C		*													
	0 4	C		*													
	0 5	C	*	I N D I C A T O R	1 0 I S	T E S T E D	O N L Y	P R E P A R A T O R Y	T O	E X E C U T I N G	L I N E	1 .					
	0 6	C	*	R P G	A S S U M E S	T H A T	I F	L I N E	1 I S	E X E C U T E D , ,	T H E N	L I N E S	2 , 3 ,	A N D	4		
	0 7	C	*	S H O U L D	B E	E X E C U T E D	A L S O	A N D	P R O D U C E S	O B J E C T	C O D E	T H A T	D O E S	S O			
	0 8	C	*	W I T H O U T	F U R T H E R	T E S T I N G	O F	T H E	I N D I C A T O R S .								

(8) The asterisk convention cannot be used on a total time line if the preceding line is a detail time line.

(9) The asterisk convention may also be used with level indicators. If a preceding line is identified as a total time line, placing an asterisk in Column 11 of that line usually is functionally equivalent to respecifying the level indicator.

(10) The asterisk should not be used on a TAG (see 4.2.4.17), on the specification following a TAG, or on the first detail calculation specification.

■ Example

Figure 4–3 illustrates the conditions that must be present for the associated calculations to be performed.

PG. NO.	LINE NO.	FORM TYPE	CONTROL LEVEL	INDICATORS AND	INDICATORS AND		
1	3 5	6	7	9 11	12 14	15 17	18
	0 1	C		1 0				①
	0 2	C						
	0 3	C		1 0	N 1 1			②
	0 4	C						
	0 5	C		2 0	L 2			③
	0 6	C						
	0 7	C		3 5	N H 1			④
	0 8	C						
	0 9	C		0 F				⑤
	1 0	C						
	1 1	C	L 0	1 7				⑥
	1 2	C						
	1 3	C	L 1	4 1				⑦
	1 4	C						
	1 5	C	L 2					⑧
	1 6	C						
	1 7	C	L 3	M R	1 2			⑨
	1 8	C		*				⑩

Figure 4--3. Example of Indicator Entries

Calculations are performed at detail time as follows:

(1) When indicator 10 is set,

(2) When indicator 10 is set and indicator 11 is not set,

(3) When indicator 20 and control level L2 are set,

(4) When indicator 35 is set and halt indicator H1 is not set, and

(5) When the overflow indicator is set.

Calculations are performed at total time as follows:

(6) When control level L0 is set (this is always set) and indicator 17 is set,

(7) When control level L1 and indicator 41 are set,

(8) When control level L2 is set (no indicators have to be set on or off),

(9) When control level L3, the matching record indicator, and indicator 12 are set, and

(10) The asterisk continues the condition from the preceding line.

4.2.3. Factor 1 (Columns 18-27) and Factor 2 (Columns 33-42)

- Entry

Required.

- Purpose

Identifies the field name (see 4.2.5) or *literal* that is to be processed.

A literal is actual data, rather than the assigned name which represents the address or location of the data. There are three types of literals:

(1) Numeric Literals – A numeric literal may contain only digits. Alphabetics and special characters are not permitted. One algebraic sign, plus (+) or minus (–), and one decimal point (period or comma) may appear in the literal. The following rules apply to numeric literals:

(a) Spaces are not permitted.

(b) If the literal does not include an algebraic sign, it is assumed to be positive. If a plus or minus sign is included, it must be the leftmost character in the literal.

(c) The decimal point (period or comma), if used, may appear anywhere in the literal.

(d) Numeric literals must not be enclosed in apostrophes.

(e) Maximum length is ten digits.

(f) If the inverted (European) print option is indicated on the RPG control card (I in Column 21), a comma must be used to represent a decimal point. Otherwise, a period must be used.

(2) Alphanumeric Literals - An alphanumeric literal is any group of characters or spaces enclosed in apostrophes. Alphanumeric literals are never used in computations. The following rules apply to alphanumeric literals:

(a) Any character may be used in forming an alphanumeric literal. Spaces are considered valid characters in the body of the literal. If an apostrophe is required within a literal, it must be indicated as two consecutive single apostrophes.

(b) Alphanumeric literals must be enclosed in apostrophes.

(c) Maximum length is eight characters, not including the identifying apostrophes.

(3) Hexadecimal Literals - A hexadecimal literal consists of an X followed by a maximum of seven hexadecimal digits enclosed in apostrophes.

Each hexadecimal digit represents a half byte of information (four bits). The hexadecimal digits and their corresponding values are:

0 – 0000	8 – 1000
1 – 0001	9 – 1001
2 – 0010	A – 1010
3 – 0011	B – 1011
4 – 0100	C – 1100
5 – 0101	D – 1101
6 – 0110	E – 1110
7 – 0111	F – 1111

Examples of hexadecimal literals and their values are as follows:

X'D' =		00001101
X'101' =	00000001	00000001
X'3FFF' =	00111111	11111111

In all operations, an RPG generated program treats a hexadecimal literal according to the same rules as an alphanumeric literal.

■ Rules

(1) The three types of literals apply to factor 1 and factor 2.

(2) The factor specification must be a field name or a literal.

(3) A field name cannot exceed six characters, a numeric literal cannot exceed ten characters, an alphanumeric literal cannot exceed eight characters, and a hexadecimal literal cannot exceed seven characters.

(4) Entries in factor 1 and factor 2 must be left-justified.

(5) When a factor entry is a field name, it must be entered in Columns 53-54 of the Input Format Specifications form or defined as a result field (Column 43-48) on the Calculation Specifications form.

■ Examples

Figure 4–4 illustrates several types of factor entries.

REPORT PROGRAM GENERATOR
CALCULATION SPECIFICATIONS

PROGRAM _____ PROGRAMMER _____

PG. NO.	LINE NO.	FORM TYPE	CONTROL LEVEL	INDICATORS AND	INDICATORS AND	FACTOR 1	OPE
1	3 5	6	7	9 11	12 14	15 17 18	27 28
	01	C				0,1,2,3,4,5,6,7,8,9	
	02	C				-5,5,5,5,5,5.,5,5	
	03	C				+,6,6,.,6,6	
	04	C				7.,7,7	
	05	C				0,0,8	
	06	C				9	
	07	C				X'ABC'	
	08	C				X'1,0,4,8,6,A'	
	09	C				'JAN.,15,,'	
	10	C				'TUESDAY'	
	11	C				'O''GRADY'	
	12	C				FIELD4	
	13	C				FLD3	
	14	C				DEPT	

① NUMERIC LITERALS

② HEXADECIMAL LITERALS

③ ALPHANUMERIC LITERALS

④ FIELD NAME

Figure 4–4. Example of Factor 1 Entries

4.2.4. Operation (Columns 28-32)

■ Entry

Required.

■ Purpose

Specify the calculations to be performed using factor 1 and factor 2.

■ Codes

Table 4–1 shows all the operation codes.

OPERATION CATEGORY	OPERATION CODE	DESCRIPTION
ARITHMETIC	ADD	Add
	Z-ADD	Zero and Add
	SUB	Subtract
	Z-SUB	Zero and Subtract
	MULT	Multiply
	DIV	Divide
	MVR	Move Remainder
MOVE	MOVE	Move
	MOVEL	Move Leftmost Characters
	MLLZO, MLHZO MHLZO, MHHZO	Move Zones
TEST	COMP	Compare
	TESTZ	Test Zone
SET INDICATORS	SETOF	Set Indicators Off
	SETON	Set Indicators On
TABLE	LOKUP	Table Lookup
BRANCH	GOTO	Branch
	TAG	Tag
	EXIT	Exit to Subroutine
	RLABL	Result Label
	ULABL	User Label

Table 4–1. Operation Codes

4.2.4.1. Add (ADD)

The ADD operation causes the contents of the field or literal specified in factor 1 to be algebraically added to the contents of the field or literal specified in factor 2 and the sum to be stored in the area specified by the result field. The two factors involved in the ADD operation may contain only numeric data.

REPORT PROGRAM GENERATOR
CALCULATION SPECIFICATIONS

PROGRAM _____ PROGRAMMER _____ DATE _____

PG. NO.	LINE NO.	FORM TYPE	CONTROL LEVEL	INDICATORS AND	INDICATORS AND	FACTOR 1	OPERATION	FACTOR 2	RESULT FIELD	FIELD LENGTH	DECIMAL POSITIONS	HALF ADJUST	RESULTING INDICATORS +	RESULTING INDICATORS −	RESULTING INDICATORS 0	COMPARE HI 1>2	COMPARE LO 1<2	COMPARE EQ 1=2
1	3 5	6	7	9 11	12 14	15 17	18 27	28 32	33 42	43 48	49 51	52	53	54	56	58			
	01	C					00438	ADD	FOX	SUM1	70								
	02	C					DOG	ADD	CAT	MIX	60								
	03	C					MIX	ADD	−45	DIFF	60								
	04	C					00438	ADD	50	SUM2	50								
	05	C																	

LITERALS 00438
−45
50

LABELS FOX = 0004936
SUM1 = 0005374
DOG = 0635
CAT = 0421
MIX = 001056
DIFF = 001011
SUM2 = 00488

4.2.4.2. Zero and Add (Z-ADD)

The Z-ADD operation causes the specified result field to be set to zeros and the contents of the field or literal specified by factor 2 to be added to the result field. Factor 1 is not involved in this operation.

REPORT PROGRAM GENERATOR
CALCULATION SPECIFICATIONS

PROGRAM _____ PROGRAMMER _____ DATE _____

PG. NO.	LINE NO.	FORM TYPE	CONTROL LEVEL	INDICATORS AND	INDICATORS AND	FACTOR 1	OPERATION	FACTOR 2	RESULT FIELD	FIELD LENGTH	DECIMAL POSITIONS	HALF ADJUST	RESULTING INDICATORS +	RESULTING INDICATORS −	RESULTING INDICATORS 0	COMPARE HI 1>2	COMPARE LO 1<2	COMPARE EQ 1=2
1	3 5	6	7	9 11	12 14	15 17	18 27	28 32	33 42	43 48	49 51	52	53	54	56	58			
	01	C					①	Z-ADD	0519	FOX	60								
	02	C					②	Z-ADD	CAT	JOE	60								
	03	C					③	Z-ADD	+00000000	TOM	80								
	04	C																	

① FOX(before) = 004936, FOX(after) = 000519.

② CAT = 056381
JOE(before) = 012345, JOE(after) = 056381.

③ TOM(before) = 000569, TOM(after) = 000000.

4.2.4.3. Subtract (SUB)

The SUB operation causes the contents of the field or literal specified in factor 2 to be algebraically subtracted from the contents of the field or literal specified in factor 1 and the difference to be stored in the area specified by the result field. The two factors involved in the subtract operation may contain only numeric data.

REPORT PROGRAM GENERATOR
CALCULATION SPECIFICATIONS

PROGRAM _____ PROGRAMMER _____ DATE _____

PG. NO.	LINE NO.	FORM TYPE	CONTROL LEVEL	INDICATORS AND	INDICATORS AND	FACTOR 1	OPERATION	FACTOR 2	RESULT FIELD	FIELD LENGTH	DECIMAL POSITIONS	HALF ADJUST	RESULTING INDICATORS +	RESULTING INDICATORS -	RESULTING INDICATORS Ø	COMPARE HI 1>2	COMPARE LO 1<2	COMPARE EQ 1=2
	01	C				JOE	SUB	421	FOX	60								
	02	C				521	SUB	45	DIFF	30								
	03	C				DOG	SUB	CAT	PEN	40								
	04	C				0069830	SUB	DIFF	AMT	70								
	05	C																

LITERALS 421
521
0069830

LABELS
JOE = 824
FOX = 000403
DIFF = 476
DOG = 0621
CAT = 0050
PEN = 0571
AMT = 0069354

4.2.4.4. Zero and Subtract (Z-SUB)

The Z-SUB operation causes the specified result field to be set to zeros and the contents of the field or literal specified by factor 2 to be subtracted from the result field. Factor 1 is not involved in this operation.

NOTE: In the Z-ADD and Z-SUB operations, a hyphen is required after the letter Z.

REPORT PROGRAM GENERATOR
CALCULATION SPECIFICATIONS

PROGRAM _____ PROGRAMMER _____ DATE _____

PG. NO.	LINE NO.	FORM TYPE	CONTROL LEVEL	INDICATORS AND	INDICATORS AND	FACTOR 1	OPERATION	FACTOR 2	RESULT FIELD	FIELD LENGTH	DECIMAL POSITIONS	HALF ADJUST	RESULTING INDICATORS +	RESULTING INDICATORS -	RESULTING INDICATORS Ø	COMPARE HI	COMPARE LO	COMPARE EQ
1	3 5	6	7	9 11	12 14	15 17	18 27	28 32	33 42	43 48	49 51	52	53	54	55	56	57	58
	01	C					(1)	Z-SUB	CAT	REST	6	0						
	02	C					(2)	Z-SUB	50	FOX	4	0						
	03	C					(3)	Z-SUB	-75	JOE	4	0						
	04	C																

(1) CAT = +000630
REST(before) = +000982, REST(after) = -000630.
(2) FOX (before) = +1500, FOX (after) = -0050.
(3) JOE (before) = +4500, JOE (after) = +0075.

4.2.4.5. Multiply (MULT)

The MULT operation causes the contents of the field or literal specified in factor 1 to be algebraically multiplied by the contents of the field or literal specified in factor 2 and the results stored in the area specified by the result field. The two factors involved in the multiply operation can contain only numeric data.

The number of decimal positions specified in Column 52 for the result field determines the number of decimal places that are retained in the product. Any excess digits at the low order end of the results are dropped. If, at this point, the result is larger than 15 positions, all digits in excess of 15 are dropped from the high order end of the result. The following formula can be used to determine if the most significant positions of the result are to be truncated (no error condition is indicated by the RPG in this case):

$$L_1 - D_1 + L_2 - D_2 + D_r \geq 15$$

where:

L_1 = length of factor 1 (multiplicand) (number of positions)
D_1 = number of decimal positions in factor 1
L_2 = length of factor 2 (multiplier) (number of positions)
D_2 = number of decimal positions in factor 2
D_r = number of decimal positions in the result (product)

REPORT PROGRAM GENERATOR
CALCULATION SPECIFICATIONS

PROGRAM _____ PROGRAMMER _____ DATE _____

PG. NO.	LINE NO.	FORM TYPE	CONTROL LEVEL	INDICATORS AND	INDICATORS AND	FACTOR 1	OPERATION	FACTOR 2	RESULT FIELD	FIELD LENGTH	DECIMAL POSITIONS	HALF ADJUST	RESULTING INDICATORS +	RESULTING INDICATORS −	RESULTING INDICATORS ø	COMPARE HI	COMPARE LO	COMPARE EQ
	01	C		①		CAND	MULT	MIER	PROD	6	0							
	02	C		②		FOX1	MULT	FOX2	FOX3	8	3							

① CAND = 006398 ∧ , Developed Product = 000007895132 ∧
MIER = 001234 ∧ , Stored Product = 895132 ∧

NOTE: If the length of the result field is less than the developed product, digits on the left are truncated.

② FOX1 = 004 ∧ 63 , Developed Product = 0000055 ∧ 565556
FOX2 = 012 ∧ 0012 , Stored Product = 00055 ∧ 565

NOTE: The length of the result field and the decimal positions specified may cause truncation on either end or both ends of the result.

REPORT PROGRAM GENERATOR
CALCULATION SPECIFICATIONS

PROGRAM _____ PROGRAMMER _____ DATE _____

PG. NO.	LINE NO.	FORM TYPE	CONTROL LEVEL	INDICATORS AND	INDICATORS AND	FACTOR 1	OPERATION	FACTOR 2	RESULT FIELD	FIELD LENGTH	DECIMAL POSITIONS	HALF ADJUST	RESULTING INDICATORS +	RESULTING INDICATORS −	RESULTING INDICATORS ø	COMPARE HI	COMPARE LO	COMPARE EQ
	01	C		①		GROSS	MULT	14.5	TAX	6	2	H						
	02	C		②		HOURS	MULT	2.2325	EXT	6	2	H						
	03	C		③		512.4216	MULT	RATE	AMT	8	3	H						

① GROSS = 0156 ∧ 32 , Developed Product Tax Plus Rounding
Multiplier = 14 ∧ 5 , Factor = 022266 ∧ 645
, Stored Product = 2266 ∧ 64

NOTE: The product will be half adjusted and the positions to the right of the number of decimal positions specified are dropped.

② HOURS = 048 ∧ 25 , Developed Product 107 ∧ 718125
Multiplier = 2 ∧ 2325 , Rounded Product 107 ∧ 723125
, Stored Product 0107 ∧ 72

③ Multiplicand = 512 ∧ 4216 , Developed Product = 632 ∧ 615210496
RATE = 1 ∧ 23456 , Rounded Product = 632 ∧ 615710496
, Stored Product = 00632 ∧ 615

4.2.4.6. Divide (DIV)

The DIV operation causes the contents of the field or literal specified in factor 1 to be algebraically divided by the contents of the field or literal specified in factor 2 and the quotient to be stored in the area specified by the result field. The two factors involved in the divide operation may contain only numeric data. The remainder, if any, from the DIV operation is lost unless a move remainder (MR) operation is specified on the following line. Factor 2 (divisor) must not be equal to zero.

To determine the highest number of positions (or decimal positions) available in any of the DIV factors, the programmer may apply the following formulas. These formulas can be applied to check in advance whether the field lengths to be used are such that a given division can be performed. As in the MULT operation, a maximum of 15 positions are available.

(1) $L_2 + D_1 - D_2 - D_r \leq 15$

(2) $L_1 - D_1 + D_2 + D_r \leq 15$, if not half adjusted

(3) $L_1 - D_1 + D_2 + D_r \leq 14$, if half adjusted

where:

L_1 = length of factor 1 (dividend) (number of positions)
D_1 = number of decimal positions in factor 1
L_2 = length of factor 2 (divisor) (number of positions)
D_2 = number of decimal positions in factor 2
D_r = number of decimal positions in the result (quotient)

The number of decimal positions in the dividend, divisor, and quotient of a pertinent division should satisfy the following conditions:

$$A = D_r - D_1 + D_2 = 0$$

If this condition is not satisfied, either the dividend or the divisor is adjusted (padded) by adding zeros to the right, depending on the following conditions:

$A > 0$ adjust or pad the dividend
$A < 0$ adjust or pad the divisor

The number of zeros added is equal to the absolute value of A.

NOTE: The dividend value of a division may be zero, but a divisor of zero is not permitted. If a divisor of zero is used, the object program is terminated immediately.

4.2.4.7. Move Remainder (MVR)

The MVR operation causes the remainder of a division (DIV) operation to be moved into the result field after resetting the result field to zero. The remainder is aligned to the implied decimal point. The MVR operation, if performed, must immediately follow the pertinent DIV operation, and is meaningful only if the quotient of this preceding divide operation was not half adjusted. The MVR line must contain all control level and indicator conditions (Columns 7-17) established for the preceding divide operation. A higher control level may be specified. Both the DIV and associated MVR operations must be executed at either detail or total time. Factor 1 and factor 2 cannot be specified with an MVR operation. The result field of an MVR operation must be numeric. The remainder is defined as follows:

Remainder = Dividend − (Divisor x Quotient)

The following examples illustrate various decimal point settings for the dividend and divisor, and the resulting quotient and remainder, as specified by the field length and decimal positions on the Calculation Specifications form.

REPORT PROGRAM GENERATOR
CALCULATION SPECIFICATIONS

PROGRAM _____ PROGRAMMER _____ DATE _____

PG. NO.	LINE NO.	FORM TYPE	CONTROL LEVEL	INDICATORS (AND)	INDICATORS (AND)	FACTOR 1	OPERATION	FACTOR 2	RESULT FIELD	FIELD LENGTH	DECIMAL POSITIONS	HALF ADJUST	RESULTING INDICATORS: +	−	0	COMPARE: HI 1>2	LO 1<2	EQ 1=2
1	3 5	6	7	9 11	12 14	18 27	28 32	33 42	43 48	49 51	52	53	54	56	58			
	01	C		(1)		QTY	DIV	26	QUOT	5	2							
	02	C					MVR		REM	5	3							
	03	C																
	04	C		(2)		123.456	DIV	PERCT	QUOT	6	3							
	05	C					MVR		REM	4	3							

(1) Given a dividend QTY equal to 461., to be divided by the literal 26. We wish to produce a quotient of 5 positions with 2 decimal places and a remainder of 5 positions with 3 decimal places.
461.÷26., = QUOT 017.73, REM 00.020

(2) Given the literal 123.456 to be divided by PERCT, which is equal to 12.3 , we wish to produce a quotient of 6 positions with 3 decimal places and a remainder of 4 positions with 3 decimal places.
123.456÷12.3, = QUOT 10.037, REM 0.109

The following can be used to determine the number of positions to the right and left of the decimal point in the dividend, divisor, quotient and remainder.

FACTOR 1	FACTOR 2	RESULT	REMAINDER (MVR)
I_1 . D_1	I_2 . D_2	I_Q . D_Q	I_R . D_4
Decimal ↑ Point	Decimal ↑ Point	Decimal ↑ Point	Decimal ↑ Point
DIVIDEND	DIVISOR	QUOTIENT	REMAINDER

where:

I_1 = Positions to the left of the decimal point in factor 1
I_2 = Positions to the left of the decimal point in factor 2
I_Q = Positions to the left of the decimal point in quotient
I_R = Positions to the left of the decimal point in remainder
D_1 = Positions to the right of the decimal point in factor 1
D_2 = Positions to the right of the decimal point in factor 2
D_Q = Positions to the right of the decimal point in quotient
D_R = Positions to the right of the decimal point in remainder

The maximum number of significant positions to the left of the decimal point in the quotient result (I_Q) is determined by the following:

$$I_Q = I_1 + D_2$$

The maximum number of significant positions to the right of the decimal point in the remainder (D_R) can be determined by the following:

$$D_R = D_1, \text{ if } D_1 \geq D_Q + D_2$$

or

$$D_R = D_Q + D_2, \text{ if } D_1 < D_Q + D_2$$

4.2.4.8. Move (MOVE)

The MOVE operation causes the contents of the field or literal specified in factor 2 to be stored right-justified in the area specified in the result field. Factor 1 is not involved in this operation. The contents of the field or literal specified in factor 2 may be numeric or alphanumeric data. The MOVE operation operates according to the following rules:

(1) If the field or literal specified in factor 2 is shorter than the area specified in the result field, factor 2 is moved to the rightmost positions of the result field. The leftmost positions of the result field located beyond the length of factor 2 are undisturbed.

(2) If the field or literal specified in factor 2 is longer than the area specified in the result field, factor 2 is left truncated and the excess characters of factor 2 are disregarded.

(3) A numeric field may be moved to an alphanumeric field and vice versa. This may be used for unpacking and packing data.

(4) When an alphanumeric field is moved into a numeric field, the sign of the numeric field is automatically checked and, if necessary, corrected to a valid UNIVAC 9400 sign. The correction made depends on the mode of compilation selected in the RPG control card (see Appendix E).

For 9400 mode programs, signs 6_{16} and B_{16} are changed to D_{16}. D_{16} is unchanged, and all other signs are changed to C_{16}.

For 9200/9300 mode programs, no checking or correction is done. Extreme care should be observed when moving data from an alphanumeric to a numeric field.

For IBM 360 Model 20 mode programs, signs C_{16} through F_{16} are unchanged, sign 6_{16} is changed to D_{16} if the accompanying digit is 0_{16}, and all other signs (including 6_{16} with other digits) are changed to C_{16}.

(5) Resulting indicators can be specified on a MOVE operation. For an alphanumeric result field, only a blank indicator (columns 58 and 59) is meaningful.

REPORT PROGRAM GENERATOR
CALCULATION SPECIFICATIONS

	OPERATION (27 28–32)	FACTOR 2 (33–42)	RESULT FIELD (43–48)	FIELD LENGTH (49–51)	DECIMAL POSITIONS (52)	HALF ADJUST (53)	RES INDI (54)	FACTOR 2	RESULT FIELD BEFORE MOVE	RESULT FIELD AFTER MOVE
①	MOVE	1234.98	FOX	10	2			123498	AAAAAAAAAA	AAAA123498
②	MOVE	98	PER	6	2			98	BBBBBB	BBBB98
③	MOVE	.623.41	AMT	5	3			62341	CCCCC	62341
④	MOVE	62341	AMT2	5	2			62341	DDDDD	62341
⑤	MOVE	'ABCDEF'	NUM	8	0			ABCDEF	99999999	99123456
⑥	MOVE	123456	CAT	3	0			123456	FFF	456
⑦	MOVE	X'123456'	DOG	8	0			123456	11111111	11123456
⑧	MOVE	'ABCDEF'	SAME	6				ABCDEF	222222	ABCDEF
⑨	MOVE	NUM	ALPH	10				99123456	3333333333	339912345F

Lines 1, 2, 5, 7, 9 – If the receiving area (RESULT) is larger than length of factor 2, the excess leftmost positions of result are unchanged.

Line 3 – Decimal positions are not aligned during the MOVE operation.

Line 5 – Packing data – zones (except the least significant zone) are stripped off.

Line 6 – If the receiving area is smaller than the length of factor 2, truncation takes place to the left.

Line 9 – Unpacking data – zones are restored to the result except the least significant zone; this receives the same zone as the original position.

4.2.4.9. Move Leftmost Characters (MOVEL)

The MOVEL operation causes the contents of the field or literal specified in factor 2 to be moved into the specified result field, left-justified. The rules for the MOVEL operation are as follows:

(1) The MOVEL operation does not use factor 1 or half adjust.

(2) Both alphanumeric and numeric data may be moved into either type field. If alphanumeric data is moved into numeric field, then all zones are stripped off except the rightmost zone position. Decimal alignment is not performed.

(3) Sign checking is performed during this operation (see Rule 4 in 4.2.4.8).

(4) Field length and decimal positions specifications (if result field is numeric) are required in this operation if they have not been defined in the program for the result field.

Examples of the results of the MOVEL operation are illustrated in Figure 4–5.

REPORT PROGRAM GENERATOR
CALCULATION SPECIFICATIONS

	OPERATION	FACTOR 2	RESULT FIELD	FIELD LENGTH	DECIMAL POSITIONS	HALF ADJUST	FACTOR 2	RESULT FIELD BEFORE MOVE	RESULT FIELD AFTER MOVE
	27 28 32	33 42	43 48	49 51	52	53 54			
①	MOVEL	-2,3,4,.5,6	AMOUNT	8	3		23456̄	88888888	2345688
②	MOVEL	-2345,6,78	RESULT	5	2		2345678̄	33333	23456̄
③	MOVEL	RESULT	FINAL	7			23456̄	AAAAAAA	23456̄AA
④	MOVEL	FLDA	FLDX	5			2345678̄	AAAAA	23456
⑤	MOVEL	'HERE'	FLDY	7			HERE	AAAAAAA	HEREAA

Figure 4–5. Examples of MOVEL Operation

(1) If the result field is longer than factor 2, the excess (righthand) position of the result field remains untouched (see Figure 4-5, line 1).

(2) If the result field is shorter than factor 2, the excess (righthand) positions of factor 2 are not moved (see line 2).

(3) If the factor 2 is shorter than the (numeric) result field, then the sign of factor 2 is not moved (see line 1).

(4) If factor 2 is equal to or longer than the (numeric) result field, then the sign of factor 2 is moved into the rightmost position of the result field (see line 2).

(5) If a numeric factor 2 field is moved into an alphanumeric result field that is equal to or longer than factor 2, then the sign of factor 2 is moved into the position containing the rightmost position of factor 2 (see line 3).

(6) If an alphanumeric result field is shorter than the numeric factor 2 field, then the sign is not moved and is lost (see line 4).

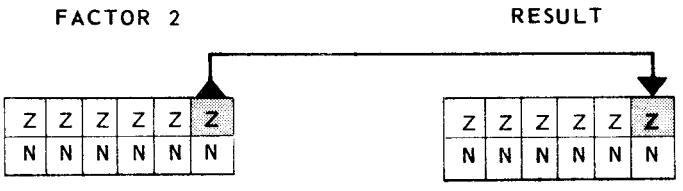
(7) Resulting indicators can be specified on a MOVEL operation. For an alphanumeric result field, only a blank indicator (columns 58 and 59) is meaningful.

4.2.4.10. Move Zone

The move zone operation causes the zone portion of the specified position, either low order (L) or high order (H), of the field indicated in factor 2 to be moved to the specified position, either low order (L) or high order (H), of the result field. The zone position of a numeric field is the sign, and is in the rightmost position of the field. Factor 1 and resulting indicators cannot be specified in any of the move zone operations.

4.2.4.10.1. Move Low to Low (MLLZO)

The zone located at the least significant position of factor 2 is moved to the least significant position of the result field. Factor 2 and the result field can be either numeric or alphanumeric.



REPORT PROGRAM GENERATOR
CALCULATION SPECIFICATIONS

PROGRAM _____ PROGRAMMER _____ DATE _____

PG. NO.	LINE NO.	FORM TYPE	CONTROL LEVEL	INDICATORS AND	INDICATORS AND	FACTOR 1	OPERATION	FACTOR 2	RESULT FIELD	FIELD LENGTH	DECIMAL POSITIONS	HALF ADJUST	RESULTING INDICATORS +	RESULTING INDICATORS −	RESULTING INDICATORS Ø	COMPARE HI 1>2	COMPARE LO 1<2	COMPARE EQ 1=2
1	3 5	6	7	9 11	12 14	15 17	18 27	28 32	33 42	43 48	49 51	52	53	54	56	58		
	0 1	C						MLLZO	LOW 1									
										LOW 2								

When factor 2 is alphanumeric and the result field is numeric, the sign of the result is automatically checked and corrected, if necessary, to a valid UNIVAC 9400 sign. The correction made depends on the mode of compilation selected in the RPG control card (see Appendix E).

For 9400 mode programs, signs 6_{16} and B_{16} are changed to D_{16}, D_{16} is unchanged, and all other signs are changed to C_{16}.

For 9200/9300 mode programs, signs 6_{16} and 9_{16} are changed to D_{16}, A_{16} through F_{16} are unchanged, and all other signs are changed to C_{16}.

For IBM 360 Model 20 mode programs, signs C_{16} through F_{16} are unchanged, sign 6_{16} is changed to D_{16} if the accompanying digit is 0_{16}, and all other signs (including 6_{16} with other digits) are changed to C_{16}.

4.2.4.10.2. Move Low to High (MLHZO)

The zone located at the least significant position of factor 2 is moved to the most significant position of the result field. Factor 2 can be either numeric or alphanumeric, but the result field can only be alphanumeric.

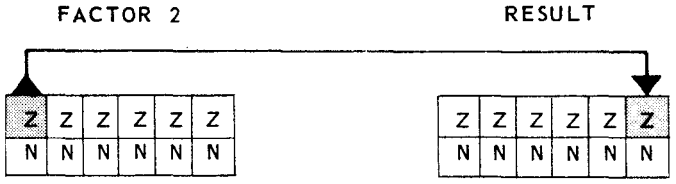
REPORT PROGRAM GENERATOR
CALCULATION SPECIFICATIONS

PROGRAM _____ PROGRAMMER _____ DATE _____

PG. NO.	LINE NO.	FORM TYPE	CONTROL LEVEL	INDICATORS AND	INDICATORS AND	FACTOR 1	OPERATION	FACTOR 2	RESULT FIELD	FIELD LENGTH	DECIMAL POSITIONS	HALF ADJUST	RESULTING INDICATORS +	RESULTING INDICATORS −	RESULTING INDICATORS ø	COMPARE HI >2	COMPARE LO <2	COMPARE EQ =2
1	3 5	6	7	9 11	12 14	15 17	18 27	28 32	33 42	43 48	49 51	52	53	54	55	56	57	58
	01	C						MLHZO	LOW1									
										HI2								

4.2.4.10.3. Move High to Low (MHLZO)

The zone located at the most significant position of factor 2 is moved to the least significant position of the result field. Factor 2 can only be alphanumeric while the result field can be either numeric or alphanumeric. Sign checking is performed during this operation (see 4.2.4.10.1).



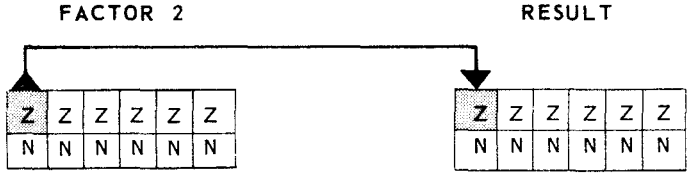
REPORT PROGRAM GENERATOR
CALCULATION SPECIFICATIONS

PROGRAM _____ PROGRAMMER _____ DATE _____

PG. NO.	LINE NO.	FORM TYPE	CONTROL LEVEL	INDICATORS AND	INDICATORS AND	FACTOR 1	OPERATION	FACTOR 2	RESULT FIELD	FIELD LENGTH	DECIMAL POSITIONS	HALF ADJUST	RESULTING INDICATORS +	RESULTING INDICATORS −	RESULTING INDICATORS Ø	COMPARE HI 1>2	COMPARE LO 1<2	COMPARE EQ 1=2
1	3 5	6	7	9 11	12 14	15 17 18 27	28 32	33 42	43 48	49 51	52	53	54	56	58			
	0 1	C					M H L Z O	H I 1										
									L O W 2									

4.2.4.10.4. Move High to High (MHHZO)

The zone located at the most significant position of factor 2 is moved to the most significant position of the result field. Factor 2 and the result field must be alphanumeric.



REPORT PROGRAM GENERATOR
CALCULATION SPECIFICATIONS

PROGRAM _____ PROGRAMMER _____ DATE _____

PG. NO.	LINE NO.	FORM TYPE	CONTROL LEVEL	INDICATORS AND	INDICATORS AND	FACTOR 1	OPERATION	FACTOR 2	RESULT FIELD	FIELD LENGTH	DECIMAL POSITIONS	HALF ADJUST	RESULTING INDICATORS +	RESULTING INDICATORS −	RESULTING INDICATORS Ø	COMPARE HI 1>2	COMPARE LO 1<2	COMPARE EQ 1=2
1	3 5	6	7	9 11	12 14	15 17 18 27	28 32	33 42	43 48	49 51	52	53	54	56	58			
	0 1	C					M H H Z O	H I 1										
									H I 2									

4.2.4.11. Compare (COMP)

The COMP operation causes the contents of the field or literal specified in factor 1 to be compared with the contents of the field or literal specified in factor 2. An indicator is turned on as specified in the resulting indicators specification. All numeric comparisons are algebraic; positive numbers and zero are treated as greater than negative numbers regardless of their bit configuration in storage. The specified indicator is turned on under the following conditions:

(1) If factor 1 is greater than factor 2, the indicator specified as HIGH is turned on.

(2) If factor 1 is less than factor 2, the indicator specified as LOW is turned on.

(3) If factor 1 is equal to factor 2, the indicator specified as EQUAL is turned on.

A result field is not specified in a compare operation. The purpose of the compare operation is to turn on an indicator which can be used to modify subsequent calculations. Procedures of the COMP operation include the following:

(1) The maximum field length for alphanumeric fields involved in compare operations is 256 characters; for numeric fields, maximum length is 15 digits.

(2) In a comparison of numeric fields, the two factors are aligned according to their implied decimal points before the comparison is executed.

(3) When numeric fields in a compare operation are of unequal length, the missing digits of the shorter field are assumed to be zeros.

(4) When alphanumeric fields in a COMP operation are of unequal length, the fields are aligned to their leftmost character, and missing characters are assumed to be blank.

(5) All numeric comparisons are algebraic. If a comparison of the absolute values of numeric fields is desired, it can be done by means of a user-coded RPG routine.

■ Example – Numeric Comparisons

REPORT PROGRAM GENERATOR
CALCULATION SPECIFICATIONS

PROGRAM _____ PROGRAMMER _____ DATE _____ PAGE ___ OF ___ PAGES

PG. NO.	LINE NO.	FORM TYPE	CONTROL LEVEL	INDICATORS AND	INDICATORS AND	FACTOR 1	OPERATION	FACTOR 2	RESULT FIELD	FIELD LENGTH	DECIMAL POSITIONS	HALF ADJUST	RESULTING INDICATORS +	RESULTING INDICATORS −	RESULTING INDICATORS COMPARE Ø	RESULTING INDICATORS COMPARE HI 1>2	RESULTING INDICATORS COMPARE LO 1<2	RESULTING INDICATORS COMPARE EQ 1=2	COMMENTS	PROGRAM IDENTIFICATION
	01	C				FOX	COMP	-.463										16		
	02	C	*	IF	CONTENTS	OF FOX	IS	A NEGATIVE	463,	THEN	SET						20	21	INDICATOR 16.	
	03	C				CTR	COMP	66									20	21		
	04	C	*	USE	INDICATOR 20	AS	A	LOOP	CONTROL	UNTIL	THE						20	21	COUNTER (CTR)	
	05	C	*	CONTAINS	66;	WHEN	THE	CTR	IS	EQUAL	TO	66,	USE				20	21	INDICATOR 21	
	06	C	*	TO	EXIT	THE	LOOP.													

■ Example – Absolute Comparisons

REPORT PROGRAM GENERATOR
CALCULATION SPECIFICATIONS

PROGRAM _____ PROGRAMMER _____ DATE _____ PAGE ___ OF ___ PAGES

PG. NO.	LINE NO.	FORM TYPE	CONTROL LEVEL	INDICATORS AND	INDICATORS AND	FACTOR 1	OPERATION	FACTOR 2	RESULT FIELD	FIELD LENGTH	DECIMAL POSITIONS	HALF ADJUST	RESULTING INDICATORS +	RESULTING INDICATORS −	RESULTING INDICATORS COMPARE Ø	RESULTING INDICATORS COMPARE HI 1>2	RESULTING INDICATORS COMPARE LO 1<2	RESULTING INDICATORS COMPARE EQ 1=2	COMMENTS	PROGRAM IDENTIFICATION
	01	C					Z-ADD	SUMB	TWS	72							18			
	02	C		18			Z-SUB	SUMB	TWS											
	03	C				TWS	COMP	500								19	20	21		
	04	C	*	THE	NUMERIC	FIELDS	SUMB	HAS	A	FIELD	LENGTH	OF	7	WITH	2	DECIMAL				
	05	C	*	PLACES.	THE	ABSOLUTE	VALUE	OF	SUMB	IS	TO	BE	COMPARED	AGAINST						
	06	C	*	500.																

4.2.4.12. Test Zone (TESTZ)

This operation is used to test the high order zone position of the alphanumeric field entered in the result field. One of the resulting indicators is turned on depending on the test results. The resulting indicators may be used to condition subsequent calculations and/or output operations.

(1) If the result of the test is a 12-zone (&, A through I, plus zero) the indicator specified in Columns 54-55 is turned on.

(2) If the result of the test is an 11-zone (–, J through R, minus zero), the indicator specified in Columns 56-57 is turned on.

(3) Any other zone turns on the indicator specified in Columns 58-59.

REPORT PROGRAM GENERATOR
CALCULATION SPECIFICATIONS

PROGRAM _____ PROGRAMMER _____ DATE _____ PAGE ___ OF ___ PAGES

PG. NO.	LINE NO.	FORM TYPE	CONTROL LEVEL	INDICATORS AND	INDICATORS AND	FACTOR 1	OPERATION	FACTOR 2	RESULT FIELD	FIELD LENGTH	DECIMAL POSITIONS	HALF ADJUST	RESULTING INDICATORS COMPARE +	RESULTING INDICATORS COMPARE −	RESULTING INDICATORS COMPARE 0	COMMENTS	PROGRAM IDENTIFICATION
													HI 1>2	LO 1<2	EQ 1=2		
	0 1	C					TESTZ		AMT								
	0 2	C	*	TESTZ	WILL BE PERFORMED	ON THE FIELD AMT.	IF	THE	HIGH ZONE								
	0 3	C	*	POSITION	HAS A 12-ZONE,	INDICATORS IN COLUMNS 54-55			WILL BE								
	0 4	C	*	TURNED ON.													

If the 9200/9300 mode of compilation is selected in the RPG control card (Column 7), the TESTZ operation cannot be used.

4.2.4.13. Set Indicators Off (SETOF)

This operation causes the indicators specified to be placed in the off condition. A maximum of three indicators may be specified in Columns 54-55, and 58-59. The headings of Columns 54-59 have no relation to the indicators specified in these columns by the SETOF operation. The SETOF operation may be used to turn off any RPG indicators. It also may be made a conditional statement through the use of control levels and indicators.

4.2.4.14. Set Indicators On (SETON)

This operation causes the indicators specified to be placed in the on condition. A maximum of three indicators may be specified in Columns 54-55, 56-57, and 58-59. The headings of Columns 54-59 have no relation to the indicators specified in these columns by the SETON operation. The SETON operation may be used to set any RPG indicators. It may also be made a conditional statement through the use of control levels and indicators.

REPORT PROGRAM GENERATOR
CALCULATION SPECIFICATIONS

PROGRAM _____ PROGRAMMER _____ DATE _____ PAGE ___ OF ___ PAGES

PG. NO.	LINE NO.	FORM TYPE	CONTROL LEVEL	INDICATORS AND	INDICATORS AND	FACTOR 1	OPERATION	FACTOR 2	RESULT FIELD	FIELD LENGTH	DECIMAL POSITIONS	HALF ADJUST	RESULTING INDICATORS COMPARE +	RESULTING INDICATORS COMPARE −	RESULTING INDICATORS COMPARE 0	COMMENTS	PROGRAM IDENTIFICATION
													HI 1>2	LO 1<2	EQ 1=2		
	0 1 (1)	C					SETOF						10	11	25		
	0 2	C															
	0 3 (2)	C		16			SETOF						20	21			
	0 4	C															
	0 5 (3)	C					SETON						13				
	0 6	C															
	0 7 (4)	C		23	28		SETON						75				

(1) Indicators 10, 11, and 25 are turned OFF by this line of coding.

(2) Indicators 20 and 21 are turned OFF only if indicator 16 is on.

(3) Indicator 13 is turned ON by this line of coding.

(4) Indicator 75 is turned ON only if both indicator 23 and 28 are on.

4.2.4.15. Table Lookup (LOKUP)

The LOKUP causes the RPG to search a table in storage and secure from it specific data needed in subsequent calculations. The LOKUP operation is used in conjunction with the File Extension Specifications form and is described in complete detail in Section 8.

4.2.4.16. Branch (GOTO)

The GOTO statement causes the RPG to transfer control to some operation other than the next in sequence. This operation may be used as either a conditional or an unconditional branch operation. The line or operation to which RPG control is transferred must be a tag line followed by the calculation or subroutine to be performed. This operation enables the programmer to address and write subroutines, to incorporate loops of iterative coding, and to bypass segments of undesired coding.

The rules for the GOTO operation are as follows:

General:

(1) A control level entry (L0-L9) must be specified when a GOTO statement is used in total calculations.

(2) A control level entry (L0-L9) must be specified in Columns 7-8 of the tag line of the total calculation operation.

Unconditional GOTO Statements:

(3) The mnemonic GOTO must be entered in Columns 28-31 of the operation field and a name, the transfer address, in factor 2; all other columns with the possible exception of control level must be blank.

(4) The name in factor 2 must be unique and may consist of one through six alphanumeric characters.

(5) The first character of the name must be alphabetic and left-justified; embedded blanks may not be used.

Conditional GOTO Statements:

(6) The GOTO must be entered in Columns 28-31 of the operation field, a name in factor 2, a maximum of three indicators in Columns 9-17, and a control level entry may be present in Columns 7 and 8. All other columns must be left blank.

(7) The name specified in factor 2 is the address to which control is transferred if the conditions indicated in Columns 9-17 are satisfied.

Figure 4-6 illustrates various entries for the GOTO operation. Note that the tag line must contain the following:

(1) Factor 1 (Columns 18-27) is the name of the routine as specified by the GOTO statement.

(2) Operation field (Columns 28-32) must be the constant TAG left-justified.

REPORT PROGRAM GENERATOR
CALCULATION SPECIFICATIONS

PROGRAM _____ PROGRAMMER _____ DATE _____ PAGE ___ OF ___ PAGES

PG. NO.	LINE NO.	FORM TYPE	CONTROL LEVEL	INDICATORS: AND	INDICATORS: AND	FACTOR 1	OPERATION	FACTOR 2	RESULT FIELD	FIELD LENGTH	DECIMAL POSITIONS	HALF ADJUST	RESULTING INDICATORS: COMPARE +	COMPARE −	COMPARE 0	HI 1-2	LO 1-2	EQ 1-2	COMMENTS	PROGRAM IDENTIFICATION
	01	C					GOTO	ISSUE											UNCONDITIONAL	
	02	C				ORDER	TAG													
	03	C																	} CALCULATIONS	
	04	C																		
	05	C					GOTO	ONHAND											UNCONDITIONAL	
	06	C				ISSUE	TAG													
	07	C																	} CALCULATIONS	
	08	C																		
	09	C	17	14			GOTO	ORDER											CONDITIONAL	
	10	C				ONHAND	TAG													
	11	C																		
	12	C																	} CALCULATIONS	
	13	C																		
	14	C	L1			USAGE	TAG													
	15	C																	} CALCULATIONS	
	16	C																		
	17	C	L1	12	N10		GOTO	ADDUP											CONDITIONAL	

Figure 4-6. Examples of GOTO Operations

The GOTO statement in line 01 transfers control unconditionally to the tag line (ISSUE) in line 06.

When indicators 17 and 14 are on, the GOTO statement in line 09 will transfer control to the ORDER line (02). If both indicators are not on, the next sequential operation will be executed. In this example it is the ONHAND coding. This operation is also the last of the detail calculations.

If L1 and indicator 12 are on and indicator 10 is not on at total time, control will be transferred by the GOTO statement in line 17 to ADDUP.

4.2.4.16.1. Detail-to-Total Branch

RPG permits a branch from detail calculations to total calculations. This causes the program to repeat certain specifications, causing a detail-to-total loop. A detail-to-total loop allows the programmer to put out one or more total lines for each input record.

The detail-to-total loop is accomplished by the specification GOTO in detail calculations and TAG in total calculations. An indicator conditions the GOTO specification and is set on to initiate the branch.

The detail-to-total loop includes the following processing:

(1) Detail calculations, which may include a GOTO

(2) Total calculations

(3) Total output

(4) Move data in current input record to data fields

(5) Return to detail calculations (1)

Note that:

(1) Unless the GOTO is conditioned by indicators that are reset by the programmer when the loop is to be ended, the loop may be repeated indefinitely with no new input data being read.

(2) Data fields for the record type are reinitialized from the input record during each cycle through the loop. Therefore, if the programmer wishes to change data field values and keep their new contents, the new contents should be moved during the detail or total calculations to a different field which is not reinitialized. Such a field may be specially defined for this purpose in the Calculation Specifications.

4.2.4.17. Tag (TAG)

The operation TAG provides a name to which the program can branch. The name is entered in factor 1 and TAG in the operation field. The TAG entry has been further defined in 4.2.4.16.

4.2.4.18. Exit to Subroutine (EXIT)

This EXIT operation enables the programmer to transfer control from the RPG program to a user subroutine. A subroutine of this type may compute such things as trigonometric functions or logarithms. The subroutine, written in assembler language, COBOL, FORTRAN, or RPG, is performed and control is transferred back to the main program.

EXIT is placed in the operation field and the name of the entry point of the subroutine is entered in factor 2. The name of the subroutine cannot exceed 6 alphanumeric characters left-justified; the first character must be alphabetic. The EXIT operation can be a conditional operation by specifying a control level or indicators (Columns 7-17). If indicators are not used, the EXIT occurs every time the detail calculations are performed.

The user may specify an EXIT operation anywhere in the program. The following facts should be noted regarding the placement of the EXIT operation in four specific positions.

(1) If the EXIT operation is the first detail calculation, control is transferred to the indicated subroutine after an input record is read into the system.

(2) If the EXIT operation is the last detail calculation, control is transferred to the indicated subroutine prior to printing or punching heading and detail records.

(3) If the EXIT operation is the first total calculation, control is transferred to the indicated subroutine after the input record has been identified and the control field has been tested.

(4) If the EXIT operation is the last total calculation, control is transferred to the indicated subroutine prior to printing or punching totals.

Figure 4-7 shows an example of the EXIT operation usage and format.

REPORT PROGRAM GENERATOR
CALCULATION SPECIFICATIONS

PROGRAM _____ PROGRAMMER _____ DATE _____ PAGE ___ OF ___ PAGES

PG. NO.	LINE NO.	FORM TYPE	CONTROL LEVEL	INDICATORS AND	INDICATORS AND	FACTOR 1	OPERATION	FACTOR 2	RESULT FIELD	FIELD LENGTH	DECIMAL POSITIONS	HALF ADJUST	RESULTING INDICATORS +	RESULTING INDICATORS −	RESULTING INDICATORS Ø	COMPARE HI 1>2	COMPARE LO 1<2	COMPARE EQ 1=2	COMMENTS	PROGRAM IDENTIFICATION
	01	C	L1	16	N10		EXIT	SUBR												
	02	C	*	THE	INDICATORS	ARE	OPTIONAL.	NOTE	THAT	THE	NAME	OF	THE							
	03	C	*	SUBROUTINE	IS	ONLY	FOUR	CHARACTERS												

Figure 4–7. Format of EXIT Operation

4.2.4.19. Result Label (RLABL)

The RLABL statement is used to provide result field and indicator linkage between the RPG program and a subroutine. It allows the subroutine to process data generated in the RPG program and to return it for further processing. (The RLABL statement is similar to the ENTRY assembler directive.)

RLABL is placed in the operation field and the name of the field is indicated in the result field. The name of the field cannot exceed six alphanumeric characters. The first character must be alphabetic and must be left-justified.

Figures 4-8 and 4-9 illustrate RLABL formats with field defined and field not defined elsewhere.

REPORT PROGRAM GENERATOR
CALCULATION SPECIFICATIONS

PROGRAM _____ PROGRAMMER _____ DATE _____ PAGE ___ OF ___ PAGES

PG. NO.	LINE NO.	FORM TYPE	CONTROL LEVEL	INDICATORS AND	INDICATORS AND	FACTOR 1	OPERATION	FACTOR 2	RESULT FIELD	FIELD LENGTH	DECIMAL POSITIONS	HALF ADJUST	RESULTING INDICATORS +	RESULTING INDICATORS −	RESULTING INDICATORS Ø	COMPARE HI 1>2	COMPARE LO 1<2	COMPARE EQ 1=2	COMMENTS	PROGRAM IDENTIFICATION
	01	C					RLABL		FLDA	010	2									

Figure 4–8. Format of RLABL Operation, Field Not Defined Elsewhere

REPORT PROGRAM GENERATOR
CALCULATION SPECIFICATIONS

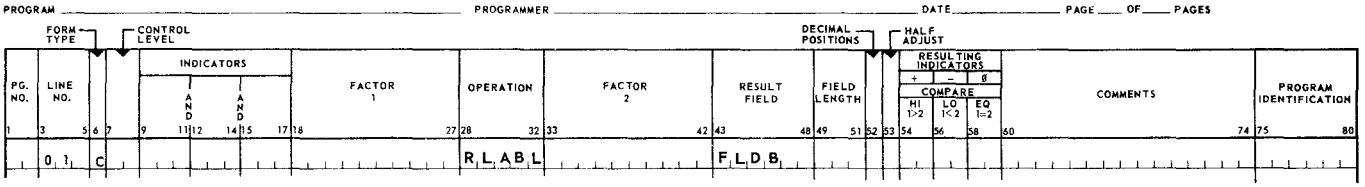


Figure 4–9. Format of RLABL Operation, Field Defined Elsewhere

If the condition of RPG result indicators must be known in a subroutine, then they must be defined in the RLABL statement. An entry in Columns 43-46 of the result field consists of the letters IN followed by the indicator number or letters.

Figure 4-10 illustrates the entry for an indicator definition in an RLABL statement.

REPORT PROGRAM GENERATOR
CALCULATION SPECIFICATIONS

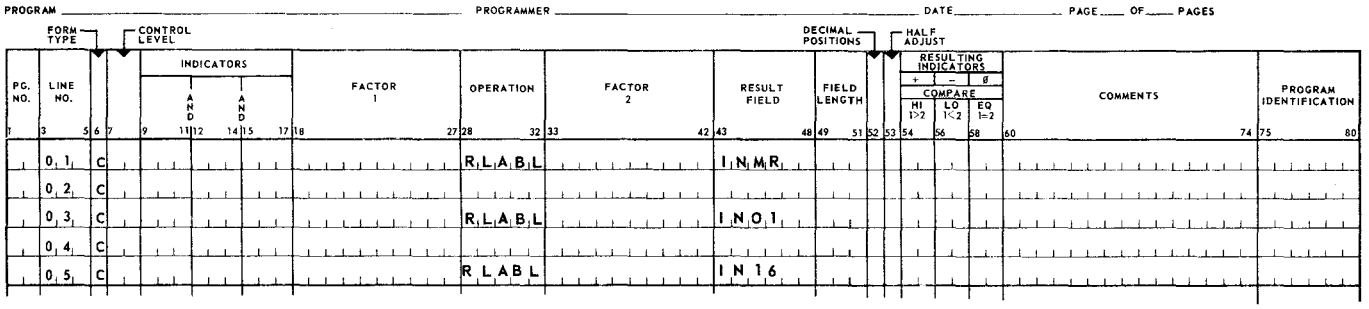


Figure 4–10. Result Label Statement Indicator Definitions

The following are general rules when using RLABL:

(1) Columns 7-27, 33-42, and 53-59 must be blank.

(2) Field length (Columns 49-51) and decimal positions (Column 52) must be specified if the result field has not been defined elsewhere in the program. If the field has been defined in the Input Format Specifications form or on the Calculation Specifications form (as a result field), then the field length and decimal position entries may be left blank.

(3) A field name used in a TAG statement must not be used in a subroutine. Therefore, an RLABL statement must not contain a name used in a tag line.

(4) Each resulting indicator defined in an RLABL statement must be used in at least one other RPG operation. The subroutine may refer to the indicator as the data located at INxx. A name or indicator used in an RLABL statement may be used in several routines.

(5) RLABL statements may be specified on any line on the Calculation Specifications form.

NOTE: The hexadecimal representation of indicator conditions is F0 for on and 00 for off. For example, a Compare Immediate instruction (in the assembler language) would use the hexadecimal representation of F0 or the eight bits of 11110000 to test for an on condition of an indicator.

4.2.4.20. User Label (ULABL)

The ULABL statement enables the RPG program to reference a field contained in a user subroutine. The name of the field is entered in the result field (left-justified). The name may not exceed six alphanumeric characters and the first character must be alphabetic. The field length and decimal position must be defined. Indicators, factor 1, and factor 2 should not be used.

4.2.4.20.1. Coding of Subroutines

Figure 4-11 illustrates the coding necessary for an RPG porgram and an associated subroutine that uses SUBA and SUBC fields as defined in the RPG program. For more detailed information on the coding of subroutines, see Appendix G.

The number of user subroutines and user labels permitted in a program is limited; the maximum number depends on the types of input and output files used in the program, the number of label processing exits specified (see 2.2.19), and the presence or absence of an ALTSEQ subroutine (see Appendix G3). For most programs, the maximum number is 200.

REPORT PROGRAM GENERATOR
CALCULATION SPECIFICATIONS

PROGRAM _____ PROGRAMMER _____ DATE _____ PAGE ___ OF ___ PAGES

PG. NO.	LINE NO.	FORM TYPE	CONTROL LEVEL	INDICATORS AND	INDICATORS AND	FACTOR 1	OPERATION	FACTOR 2	RESULT FIELD	FIELD LENGTH	DECIMAL POSITIONS	HALF ADJUST	RESULTING INDICATORS +	RESULTING INDICATORS −	RESULTING INDICATORS 0	COMPARE HI 1>2	COMPARE LO 1<2	COMPARE EQ 1=2	COMMENTS	PROGRAM IDENTIFICATION
	01	C																		
	02	C																		
	03	C																		
	04	C					EXIT	CALC												
	05	C																		
	06	C																		
	07	C					RLABL		SUBA		8	2								
	08	C																		
	09	C																		
	10	C																		
	11	C					RLABL		SUBC		10	3								

ASSEMBLER CODING FORM

PROGRAM _____ PROGRAMMER _____ DATE _____ PAGE ___

LABEL	OPERATION	OPERAND	COMMENTS
CALC	START		
	EXTRN	SUBA	
	EXTRN	SUBC	
		INSTRUCTIONS	
	BR	14	← EXIT
	DC	A(SUBA)	
	DC	A(SUBC)	

Figure 4–11. Sample Coding RPG and Associated Subroutine

4.2.5. Result Field (Columns 43-48)

- Entry

  As required by the operation.

- Purpose

  Sets up the storage area where the result of a calculation will be stored.

- Rules

  (1) The name of the result field may be alphanumeric, but the first character must be alphabetic. Embedded spaces are not allowed.

  (2) The field name may contain up to six characters and must be left-justified.

  (3) The sign for arithmetic fields is always stored in the units position of the result field.

  (4) The same field name may be used more than once in different calculations if the length of the field and the number of decimal locations are the same.

  (5) The date in the Supervisor may be accessed by the name DATE which is automatically defined by RPG as a six-position alphanumeric field.

- Examples

  Figure 4-12 shows examples of result field entries.

REPORT PROGRAM GENERATOR
CALCULATION SPECIFICATIONS

PROGRAM _____ PROGRAMMER _____ DATE _____ PAGE ____ OF ____ PAGES

PG. NO.	LINE NO.	FORM TYPE	CONTROL LEVEL	INDICATORS: AND	INDICATORS: AND	FACTOR 1	OPERATION	FACTOR 2	RESULT FIELD	FIELD LENGTH	DECIMAL POSITIONS	HALF ADJUST	RESULTING INDICATORS: COMPARE HI 1-2	RESULTING INDICATORS: COMPARE LO 1-2	RESULTING INDICATORS: COMPARE EQ 1-2	COMMENTS	PROGRAM IDENTIFICATION
	01	C							FIELD2								
	02	C							FLD3								
	03	C							AA04								
	04	C							TOTAL1								
	05	C							Z945								
	06	C							M46R4								

Figure 4-12. Example of Result Field Entries

4.2.6. Field Length (Columns 49-51)

- Entry

  Optional.

- Purpose

  Specifies the number of storage positions to be reserved for the contents of the result field.

■ Rules

(1) The maximum field lengths for RPG operations are as follows:

decimal fields	– 15 digits
alphanumeric fields	– 256 characters

(2) If the same field name is specified more than once, the field length and decimal position specifications need not be repeated.

(3) If a decimal result generated is larger than the specified result field, the result is aligned on the decimal point and is truncated to fit the field. Truncation may occur on either or both ends of the number.

NOTE: When the half adjust specification is used, the field length entry refers to the length of the result field after half adjustment.

■ Examples

Figure 4-13 shows examples of field length.

4.2.7. Decimal Position (Column 52)

■ Entry

Optional.

■ Purpose

Indicates the number of decimal places required in the result field (the decimal alignment of input fields is specified on the Input Format Specifications form).

■ Rules

(1) An entry must be made in this field for all arithmetic operations, if not previously defined. When the result field of an arithmetic calculation has no fractional positions, a zero must be entered.

(2) A maximum of nine decimal places may be specified.

(3) Decimal alignment for calculated numeric results is automatically handled by the RPG.

(4) When the result field is alphanumeric, the decimal position field must remain blank.

■ Examples

Figure 4-13 shows examples of decimal position specifications.

4.2.8. Half Adjust (Column 53)

- Entry

  Optional.

- Purpose

  Causes the contents of the result field to be rounded. The digit 5 is added to the position immediately to the right of the last decimal position retained in the result field.

- Code

  H – Half adjust takes place only when this code is entered in the half adjust field.

- Rule

  This specification must be blank for all alphanumeric fields.

- Examples

  Figure 4-13 shows examples of half adjust.

REPORT PROGRAM GENERATOR
CALCULATION SPECIFICATIONS

PROGRAM _____ PROGRAMMER _____ DATE _____

PG. NO.	LINE NO.	FORM TYPE	CONTROL LEVEL	INDICATORS AND	INDICATORS AND	FACTOR 1	OPERATION	FACTOR 2	RESULT FIELD	FIELD LENGTH	DECIMAL POSITIONS	HALF ADJUST	RESULTING INDICATORS COMPARE HI 1:2	RESULTING INDICATORS COMPARE LO 1:2	RESULTING INDICATORS COMPARE EQ 1:2
1	3 5	6	7	9 11	12 14	15 17 18 27	28 32	33 42	43 48	49 51	52	53	54	56	58
	01	C				(1)			DOG	62					
	02	C				(2)			COW	100					
	03	C				(3)			SUM	152					
	04	C				(4)			GROS	82		H			
	05	C				(5)			ALPH	25					

Figure 4–13. Field Length, Decimal Positions, and Half Adjust

(1) The operand labeled DOG is specified as 6 numeric positions in length with 2 decimal places.

(2) The operand labeled COW is specified as 10 numeric positions in length with no decimal places.

(3) The operand labeled SUM is specified as 15 numeric positions in length with 2 decimal places.

④ The operand labeled GROS is specified as 8 numeric positions in length with 2 decimal places remaining in the rounded result.

⑤ The operand labeled ALPH is specified as 25 alphanumeric positions.

4.2.9. Resulting Indicators (Columns 54-59)

■ Entry

Optional.

■ Purpose

This specification allows testing the value of a result field after the operation is completed. Based on the results of this test, an indicator may be turned on and used to control subsequent calculations and output operations. The resulting indicators specification may be used as follows:

(1) To test whether the result of an arithmetic operation is plus, minus, or zero, and to turn on the specified indicator.

(2) To test the result of a compare operation and to set the associated indicators according to the following:

Factor 1>Factor 2---Set HI indicator (Columns 54-55)
Factor 1<Factor 2---Set LO indicator (Columns 56-57)
Factor 1=Factor 2---Set EQ indicator (Columns 58-59)

When a resulting indicator is specified and the operation with which it is associated is executed, the indicator is set if its condition is met; it is reset if the condition is not met.

(3) To define the type of LOKUP operation to determine whether the table is searched for an entry next higher, next lower, or equal to, the search argument specified is factor 1 (see Section 8).

(4) To define what type of zone is to be tested in a TESTZ operation.

(5) To define which indicators are to be turned off or on in a SETOF and SETON operation.

■ Codes

01-99, H0-H9 - May be used for all arithmetic operations, compare, TESTZ, and LOKUP operations.

00-99, H0-H9, L0-L9, MR, OA, OB, OC, OD, OE, OF, OG, and OV - May be specified for SETON and SETOF operations.

- Rules

(1) More than one indicator may be on at any given time.

(2) A resulting indicator (01-99) may be defined more than once on the form. Any subsequent redefining of a resulting indicator may alter the indicator setting. If it is not redefined, the indicator is not altered until that calculation line is executed again.

NOTE: The term *defining* as used in conjunction with indicators refers to the setting of an indicator to its on or off condition. During calculation, indicators are defined in the resulting indicators specification. Indicators are also defined on the Input Format Specifications form under the record indicator specification to identify the input card type and under field indicators to record the status of a field (plus, minus, zero, or blank). All other references to indicators on the various specification forms test the specified indicators to determine their status, but the indicator is not altered. This includes the indicators specification (Columns 9-17) on the Calculation Specifications form which controls the execution of a calculation, and the output indicators specification of the Output Format Specifications form which controls the execution of output operations.

(3) Resulting indicators are not permitted with the move zone operation.

(4) Resulting indicators are not permitted with any of the branch and exit operations.

4.2.10. Comments (Columns 60-74)

- Entry

Optional.

- Purpose

The comments field is not required by the program. However, the programmer may enter any useful remarks that will be of assistance in program checking and testing.

■ Example

Figure 4-14 illustrates this entry.

REPORT PROGRAM GENERATOR
CALCULATION SPECIFICATIONS

PROGRAM _____ PROGRAMMER _____ DATE _____ PAGE ___ OF ___ PAGES

PG. NO.	LINE NO.	FORM TYPE	CONTROL LEVEL	INDICATORS AND	INDICATORS AND	FACTOR 1	OPERATION	FACTOR 2	RESULT FIELD	FIELD LENGTH	DECIMAL POSITIONS	HALF ADJUST	RESULTING INDICATORS COMPARE HI 1-2	RESULTING INDICATORS COMPARE LO 1-2	RESULTING INDICATORS COMPARE EQ 1-2	COMMENTS	PROGRAM IDENTIFICATION
	01	C				GROSS	MULT	+3.625	FICA	5	2	H				COMPUTE FICA	
	02	C															
	03	C				GROSS	MULT	+18	WHTAX	6	2	H				COMPUTE TAX	
	04	C															
	05	C				FICA	ADD	WHTAX	DED	6	2					TOTAL DEDUCTION	
	06	C															
	07	C				GROSS	SUB	DED	NET	6	2					NET PAY	

Figure 4–14. Comments Entry

4.3. EXAMPLE—COMPLETED CALCULATION SPECIFICATIONS FORM

Figure 4-15 is an example of a completed Calculation Specifications form.

Assume that FIELDA through FIELDN have been defined on the Input Format Specifications form as follows: FIELDN is alphanumeric; all others are numeric; FIELDA, FIELDB, and FIELDC each have three decimal places.

REPORT PROGRAM GENERATOR
CALCULATION SPECIFICATIONS

PROGRAMMER _____ DATE _____

TORS AND	FACTOR 1	OPERATION	FACTOR 2	RESULT FIELD	FIELD LENGTH	DECIMAL POSITIONS	HALF ADJUST	RESULTING INDICATORS COMPARE HI 1-2	RESULTING INDICATORS COMPARE LO 1-2	RESULTING INDICATORS COMPARE EQ 1-2	
	FIELDA	ADD	FIELDB	FIELD1	8	2	H			65	①
N65	FIELDC	ADD	FIELD1	FIELD1			H				②
	FIELDD	SUB	FIELDE	FIELD3	6	0			98	99	③
		Z-SUB	FIELDF	FIELDG							④
	FIELDH	MULT	3625	FIELD4	12	3					⑤
	FIELDJ	DIV	FIELDK	FIELD5	150						⑥
	FIELDL	COMP	4800000					75		75	⑦
		MOVE	FIELDM	FIELD6	150						⑧
		MOVE	FIELDN	FIELD7	80						⑨

Figure 4–15. Example of a Completed Calculation Specifications Form

(1) FIELDA is added to FIELDB, and the result is stored in FIELD1 (defined as eight numeric positions with two decimal places). The H entry in Column 53 specifies half adjustment of the answer. The zero resulting indicator is used to set indicator 65 when the result of the addition is zero.

(2) This calculation is performed only if indicator 65 is not set (previous result not zero). FIELDC is added to FIELD1, and the result is stored in FIELD1. Field length and decimal positions are not defined for FIELD1 because it is not necessary to repeat them. However, the H entry is repeated to specify half adjustment. (FIELDC has three decimal places.)

(3) FIELDE is subtracted from FIELDD, and the difference is stored in FIELD3. Indicators 98 and 99 are set when the result is negative or zero, respectively.

(4) FIELDF is stored with reversed sign in FIELDG, as defined on the Input Format Specifications form.

(5) FIELDH is multiplied by the numeric literal 3625, and the result is stored in FIELD4, containing three decimal places.

(6) FIELDJ is divided by FIELDK and the result is stored in FIELD5. Note required entry of zero decimal places.

(7) FIELDL is compared to the literal 480000. If FIELDL is algebraically greater than or equal to 480000, indicator 75 is set.

(8) A numeric field is moved from FIELDM to FIELD6.

(9) An alphanumeric field is moved from FIELDN to FIELD7. (No entry in the decimal positions specification indicates an alphanumeric field.)

5. OUTPUT FORMAT SPECIFICATIONS FORM

5.1. GENERAL DESCRIPTION

The Output Format Specifications form (Figure 5-1) defines the types of output which are to be produced on the printer, punch, disc, or tape units, and also specifies the location of the data fields in the output reports and records. The specifications for this form are divided into the following two areas:

- File Identification and Control

  The file identification and control specifications (Columns 7-31) identify the printed reports, tapes, discs, and punch card files that are to be produced. The specifications govern the spacing on printed reports and stacker selection of cards processed through the punch. This section also controls the conditions under which the various outputs are to be produced.

- Field Description

  The field description specifications (Columns 23-70) define the location of each individual field within an output file and the controlling conditions under which output fields are printed, punched, or written on tape or disc. Each file identification line entered on the Output Format Specifications form is followed by one field description line for each output field.

  Note that the output indicator specification applies to both file identification and field description. This facility permits selective control of each individual field within a file as well as selective control of files.

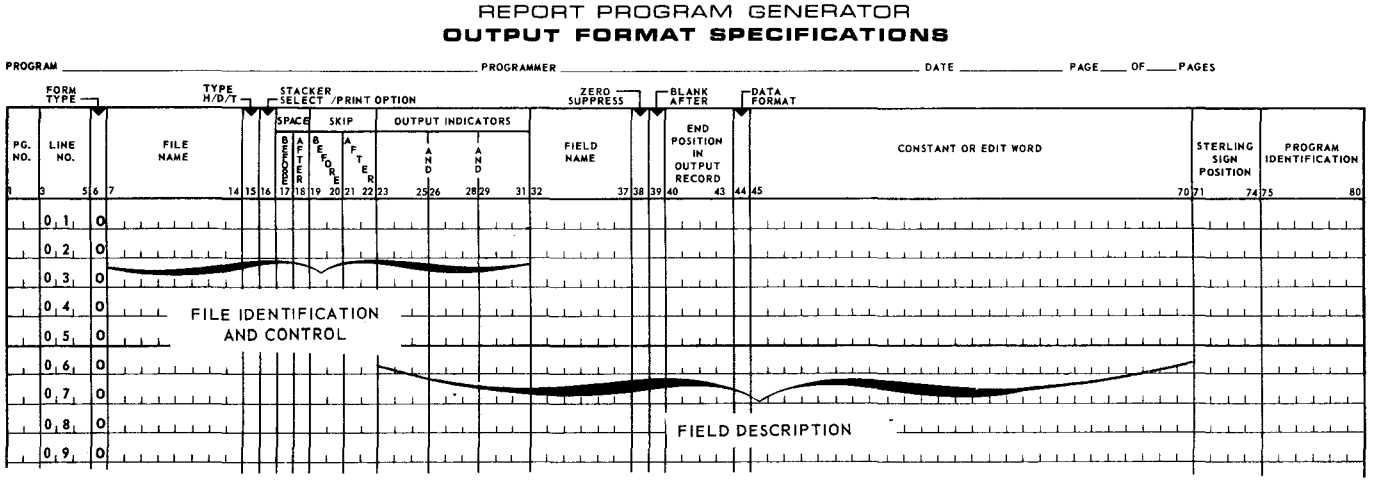


Figure 5–1. Output Format Specifications Form

– Specification Order

The sequence of entries for a given file (described in Figure 5-2) are as follows. The normal occurrence of events governs the sequence in which output specifications are written. The specifications for a typical printed report must list the information for headings first, followed by the specifications for the detail lines, and finally the required specifications for all control levels. Files must be listed in the order in which they are to be produced. All information pertinent to the output operation indicated by a file identification line must be listed with that line. This entry must specify stacker selection on the punch, or spacing or skipping on the printer, and must be followed by successive lines describing the fields involved in the output operation.

– Output Unit Specification

Since the output unit applicable to each file is specified on the File Description form, it is not necessary to indicate it on the Output Format Specifications form. Each output file is named on the output form, and is related through this name to its specific output unit.

REPORT PROGRAM GENERATOR
OUTPUT FORMAT SPECIFICATIONS

PROGRAM _____ PROGRAMMER _____ DATE _____ PAGE ___ OF ___ PAGES

PG. NO.	LINE NO.	FORM TYPE	FILE NAME	TYPE H/D/T	STACKER SELECT	SPACE BEFORE	SPACE AFTER	SKIP BEFORE	SKIP AFTER	OUTPUT INDICATORS AND	OUTPUT INDICATORS AND	FIELD NAME	ZERO SUPPRESS	BLANK AFTER	END POSITION IN OUTPUT RECORD	DATA FORMAT	CONSTANT OR EDIT WORD	STERLING SIGN POSITION	PROGRAM IDENTIFICATION
	0 1	O	File Identification and Control																
	0 2	O								Field Description							First Header Record		
	0 3	O								Field Description									
	0 4	O	File Identification and Control																
	0 5	O								Field Description							Second Header Record		
	0 6	O								Field Description									
	0 7	O								Field Description									
	0 8	O	File Identification and Control																
	0 9	O								Field Description									
	1 0	O								Field Description									
	1 1	O								Field Description									
	1 2	O								Field Description							Detail Record		
	1 3	O								Field Description									
	1 4	O								Field Description									
	1 5	O	File Identification and Control																
	1 6	O								Field Description									
	1 7	O								Field Description							Detail Record		
	1 8	O								Field Description									
	1 9	O								Field Description									
	2 0	O	File Identification and Control																
	2 1	O								Field Description									
	2 2	O								Field Description							Total Record		
	2 3	O								Field Description									

Figure 5–2. Sequence of File Entries

5.2. FILE IDENTIFICATION AND CONTROL FORM ENTRIES

The following descriptions explain the file identification and control portion of the Output Specifications form.

5.2.1. File Name (Columns 7-14)

- Entry

  Required (except as described in Rule 3).

- Purpose

  Assigns a name to each output file. A file name must be entered on the specification form for each different output file involved in the printing, punching, disc, and tape operations specified on the File Description form. Consequently, one file name would be assigned to the printing operation, another file name to the punching operation, and still another file name to the tape or disc operation.

- Rules

  (1) The file name should correspond exactly to that in the File Description Specification.

  (2) The file name must be left-justified.

  (3) If several lines of the same file are specified in sequence, it is not necessary to repeat the file name on each line. The file name is entered on the first specification line only.

- Example

Figure 5-3 illustrates several file name entries.

REPORT PROGRAM GENERATOR
OUTPUT FORMAT SPECIFICATIONS

PROGRAM _____ PROGRAMMER _____ DATE _____ PAGE ___ OF ___ PAGES

PG. NO.	LINE NO.	FO-I TYPE	FILE NAME	TYPE H/D/T	STACKER SELECT	SPACE AFTER	SPACE BEFORE	SKIP AFTER	OUTPUT INDICATORS AND	OUTPUT INDICATORS AND	FIELD NAME	ZERO SUPPRESS	BLANK AFTER	END POSITION IN OUTPUT RECORD	DATA FORMAT	CONSTANT OR EDIT WORD	STERLING SIGN POSITION	PROGRAM IDENTIFICATION
	01	O	*THE FOLLOWING ARE EXAMPLES OF FILENAME ENTRIES.															
	02	O	PUNHCRD															
	03	O																
	04	O	PRNTRPT															
	05	O																
	06	O	WRITAPE															

Figure 5–3. Examples of File Name Entries

5.2.2. Type (Column 15)

- Entry

  Required.

- Purpose

  Identifies the type of entry being specified, that is, heading, detail, or total records.

- Codes

  H – Heading. These records usually contain heading information for printed output. This information is usually constant, although it may be from an input source as well, including the current input record.

  D – Detail. Generally the data in the detail records comes from the input record, or it is calculated within the program at detail time.

  T – Total. Before detail input fields are operated on, it is preceded by a test for a control break. If there is a control break, total time calculations are performed, total lines are printed, and summary cards are punched. An input record which forces a control break is not reflected in the totals that result from the control change.

- Rules

(1) The following sequence is required for entries:
- heading entries
- detail entries
- total entries

(2) The same file name must be assigned to all *type* entries that refer to that file. Thus, heading, detail, and total type entries would be used for a specific file name that references the printer. All file name entries must be identical for each output unit, since the File Description form relates a file name directly to a printer, punch, tape, or disc.

- Example

Figure 5–4 illustrates the use of the record type column.

REPORT PROGRAM GENERATOR
OUTPUT FORMAT SPECIFICATIONS

PROGRAM _____ PROGRAMMER _____ DATE _____ PAGE ___ OF ___ PAGES

PG. NO.	LINE NO.	FORM TYPE	FILE NAME	TYPE H/D/T	STACKER SELECT	SPACE BEFORE	SPACE AFTER	SKIP BEFORE	SKIP AFTER	OUTPUT INDICATORS AND	OUTPUT INDICATORS AND	FIELD NAME	ZERO SUPPRESS	BLANK AFTER	END POSITION IN OUTPUT RECORD	DATA FORMAT	CONSTANT OR EDIT WORD	STERLING SIGN POSITION	PROGRAM IDENTIFICATION
	01	O	PRNTRPT	H															
	02	O	* HEADING LINE																
	03	O	PRNTRPT	D															
	04	O	* DETAIL LINE																
	05	O	PRNTRPT	T															
	06	O	* TOTAL LINE																
	07	O	PRNTRPT	T															
	08	O	* TOTAL LINE																
	09	O	PUNHCRD	D															
	10	O	* DETAIL CARD																
	11	O	PUNHCRD	T															
	12	O	* TOTAL CARD																

Figure 5–4. Example of Type H/D/T Entries

5.2.3. Stacker Select (Column 16)

- Entry

Optional.

- Purpose

The stacker select specification permits cards on the punch to be selected into the select stacker.

- Codes

Blank or 1 – Cards to normal stacker.

2 – Cards to select stacker.

- Rules

  (1) The stacker select is used only for output files or combined files. The stacker selection of input files is specified in the Input Format Specifications form.

  (2) For compatibility with other RPG compilers, N is permitted in Column 16 and is treated as a blank.

- Example

  Figure 5-5 illustrates entries for this option.

REPORT PROGRAM GENERATOR
OUTPUT FORMAT SPECIFICATIONS

PROGRAM _____ PROGRAMMER _____ DATE _____ PAGE ___ OF ___ PAGES

PG. NO.	LINE NO.	FORM TYPE	FILE NAME	TYPE H/D/T	STACKER SELECT	SPACE BEFORE	SPACE AFTER	SKIP BEFORE	SKIP AFTER	OUTPUT INDICATORS AND	OUTPUT INDICATORS AND	FIELD NAME	ZERO SUPPRESS	BLANK AFTER	END POSITION IN OUTPUT RECORD	DATA FORMAT	CONSTANT OR EDIT WORD	STERLING SIGN POSITION	PROGRAM IDENTIFICATION
	01	O	PUNHCRD	D															
	02	O	* DETAIL	CARD	TO NORMAL STACKER														
	03	O	PUNHCRD	T	2														
	04	O	* TOTAL	CARD	TO SELECT STACKER														

Figure 5–5. Example of Entries for Stacker Select Option

5.2.4. Space (Columns 17-18)

- Entry

  Conditional.

- Purpose

  Controls the spacing of printed reports.

- Codes

  Space Before (Column 17):
  0 – No space
  1 – Single space
  2 – Double space
  3 – Triple space

  Space After (Column 18):
  0 – No space
  1 – Single space
  2 – Double space
  3 – Triple space

- Rules

  (1) If Column 17 is left blank, no spacing before printing is provided.

  (2) A blank in Column 18 provides single spacing after printing a line.

- Example

  Figure 5-6 presents samples of valid spacing entries.

REPORT PROGRAM GENERATOR
OUTPUT FORMAT SPECIFICATIONS

PROGRAM _____ PROGRAMMER _____ DATE _____ PAGE ____ OF ____ PAGES

PG. NO.	LINE NO.	FORM TYPE	FILE NAME	TYPE H/D/T	STACKER SELECT	SPACE BEFORE	SPACE AFTER	SKIP BEFORE	SKIP AFTER	OUTPUT INDICATORS AND	OUTPUT INDICATORS AND	FIELD NAME	ZERO SUPPRESS	BLANK AFTER	END POSITION IN OUTPUT RECORD	DATA FORMAT	CONSTANT OR EDIT WORD	STERLING SIGN POSITION	PROGRAM IDENTIFICATION
1	3 5,6	7	14	15,16	17	17	18	19,20	21,22,23	25,26	28,29,31	32	37	38	39,40,43	44,45	70	71	75,80
	01	O	PRNTRPT	H				07	04										
	02	O	* SKIP TO CHANNEL 7 BEFORE PRINTING & TO CHANNEL 4 AFTER PRINTING.																
	03	O	PRNTRPT	D			1												
	04	O	* SINGLE SPACE AFTER PRINTING.																
	05	O	PRNTRPT	T		2	2												
	06	O	* DOUBLE SPACE BEFORE & DOUBLE SPACE AFTER PRINTING.																
	07	O	PRNTRPT	T		2													
	08	O	* DOUBLE SPACE BEFORE & SINGLE SPACE AFTER PRINTING.																

Figure 5-6. Example of Space and Skip Entries

5.2.5. Adding Records to an Indexed Sequential File (Columns 16-18)

- Entry

  Conditional; used if the output record is to be added to an indexed sequential file.

- Purpose

  Indicates a record is to added to an output file.

- Codes

  ADD – Output record is to be added to an indexed sequential file.

  Blank – Output record is used to update, load, or extend an indexed sequential file.

5.2.6. Skip (Columns 19-22)

- Entry

  Conditional.

- Purpose

  Provides for output formatting and directly relates to the 12 codes (usually referred to as channels), which may be punched in the printer carriage control tape. When a specified skip is executed, the printer form advances until the channel indicated is detected in the paper loop.

- Codes

  Valid codes are shown in Table 5-1.

  An entry in Columns 19-20 causes the form to advance to the specified channel before printing occurs.

  An entry in Columns 21-22 causes the form to advance to the specified channel after printing occurs.

- Rules

  (1) Spacing and skipping is performed in the following order:

  Skip before
  Space before
  Skip after
  Space after

  (2) In an OR relationship, if space and skip specifications are entered in the first line, they need not be entered in the subsequent lines, provided the specifications in the first line are applicable for all subsequent lines.

  (3) The channel codes that may be used are listed in Table 5-1 and depend on the mode of compilation chosen in the RPG control card (see Appendix E).

  The compiler treats UNIVAC 9200/9300 channel numbers and IBM 360/20 channel numbers internally as the corresponding UNIVAC 9400 channel number shown. The printer loop for the execution of the compiled object program must be prepared in accordance with this table.

	UNIVAC 9400	UNIVAC 9200/9300	IBM 360 Model 20
Overflow Channel	9	1	12
Home Paper Channel	14 15	7 15	1 15
Other Channels	4 – 6	4 – 6	4 – 6
	7	–	7
	8	–	8 – 9
	10 – 11	–	10 – 11
	12 – 13	2 – 3	2 – 3

Table 5–1. Valid Channel Codes

(4) Channels 8 and 9 of IBM 360/20 programs are both translated as Channel 8. Programs that make use of both channels must be modified.

(5) Channel 15 may be specified in any mode, and when punched in the printer carriage control tape causes printer spacing at the rate of 8 lines per inch instead of 6 lines per inch.

■ Examples

Figure 5-6 presents samples of skip entries.

5.2.7. Output Indicators (Columns 23-31)

■ Entry

Required; at least one indicator must be designated for each file identification and control entry.

■ Purpose

Determines when a specific output operation is to be performed. The kinds of indicators entered as output indicators and the significance of the output produced as a consequence are listed below.

(1) A record indicator (of a specific input record type) specifies output to be produced only when this record type is being processed in an operation cycle. The operation is not performed during operation cycles in which other input record types identified by other record indicators are being processed. Record indicators are turned on in the input section to identify a specific input record.

(2) A field indicator (turned on because a specific input field has a specific value) designates output to be produced only when the input field has this specified value.

(3) A resulting indicator (turned on because a specific calculation produces a result field with a specific value) designates output to be produced only when the result field has this specified value.

(4) A control level indicator (turned on because a control break has occurred) specifies output to be produced only at this control break. Control level indicators are L0, L1 through L9, and LR.

(5) A halt indicator (usually turned on because an error is detected during an input or calculation operation) is generally used in conjunction with an "N" prefix to suppress the production of output.

(6) The first page, 1P, indicator (turned on when the RPG object program is initiated and permanently turned off at the end of detail time of the first processing cycle of the object program) specifies output to be produced during the first processing cycle of the program. The 1P indicator is generally used to produce headings on the first page to be printed by the printer.

(7) The MR indicator causes the output operation to be performed only if there is a matching record in a secondary input file.

(8) The overflow indicators, OA, OB, OC, OD, OE, OF, OG and OV, specify output to be produced only when the end of the page (form overflow) has occurred. The overflow indicators are generally used to produce headings on the report. (Normally, one indicator is used to control output for each file.) An overflow punch in the printer carriage control tape signals form overflow after one or two additional printed lines. When form overflow is detected, the fact is noted, and at the end of this output time (total or detail), the overflow indicator is turned on. If this occurred during detail time, the overflow indicator remains on for one complete cycle and is turned off following the next detail time. If it occurred during total time, the overflow indicator is on only for the detail time which follows, and it is then reset. If the programmer has not specified any output specification lines conditioned by the overflow indicators, automatic skipping to home paper position is provided whenever the overflow condition is detected.

Note that in 9400 mode and IBM 360 Model 20 mode programs an overflow line will never be output during the normal header-detail or total output times in the processing cycle, even if all conditions for output are satisfied. An overflow line can only be output during overflow output time. An overflow line is defined as one which has an overflow indicator specified on the first source specification line used to describe the line. In other words, an overflow indicator on an AND line or a field description line will not cause the output line to be considered an overflow line.

Form overflow is detected one or two lines after the specified overflow line is reached. The number of lines depends on the entries in Format (column 19) and File Organization Type (column 32) of the File Description Form. If an F is entered in column 19 *and* a blank in column 32, form overflow is detected after one additional line; otherwise, form overflow is detected after two additional lines. (For the purposes of this rule, each line is defined as a separate printer operation. Note that each "line" requested in the Output-Format Specifications may generate as many as four separate print operations. A printed line with or without a Space After or Skip After request generates a single print operation. Additional space and skip requests for the same printed line generate one additional print operation each.)

If a UNIVAC 9200/9300 or an IBM 360 Model 20 program is being compiled which includes a skip to the overflow channel, it may have to be revised to operate properly on the UNIVAC 9400.

An output specification line may contain a combination of the above indicators. If the control operation is to be based on an indicator not being on, the indicator is preceded by the letter N.

- Codes

OA, OB, OC, OD, OE, OF, OG, OV – Overflow indicators

1P – First page

L0 – Level zero indicator

L1-L9 – Control level indicators

LR – Last record

00-99 – Record indicators, field indicators, resulting indicators

MR – Matching record indicator

H0-H9 – Halt indicators

- Rules

(1) A maximum of three indicators may be entered on a line. If no output indicators are specified for a detail line, the line is put out every time heading and detail lines are processed. When more than one indicator has been entered on an output specification line, then all such indicators must have been previously turned on before the output specification line is executed. This is referred to as an AND relationship.

(2) If more than three indicators in an AND relationship are required, then the word AND is written in Columns 14-16 of the specification line, and the additional indicators are entered in the output indicator area of this line. Columns 17-22 must be blank. Each AND line, except the last, must have three entries for indicators.

(3) An output specification line may be executed based on an OR relationship. In this case, OR is written in Columns 14 and 15 of the following specification line with other indicators entered on the same line. An OR relationship is one in which either the output indicators of one specification line or the other must have been previously turned on for that specification line to be executed. When an OR line is specified on a print file, the printer control functions (Columns 17-22) may all be left blank, in which case those of the preceding line are implemented. They may, however, differ from the preceding line, if required. This implementation does not occur if stacker selection is requested, because a blank is a valid stacker select code.

(4) Any number of additional specification lines may be written for AND or OR conditions. Each additional line must begin in Column 14 with the word AND or OR.

(5) An N in Columns 23, 26, or 29 signifies an off condition for the indicator. A blank signifies an on condition.

■ Example

Figure 5-7 presents examples of output indicator entries.

REPORT PROGRAM GENERATOR
OUTPUT FORMAT SPECIFICATIONS

PROGRAM _____ PROGRAMMER

PG. NO.	LINE NO.	FORM TYPE	FILE NAME	TYPE H/D/T	STACKER SELECT /PRINT OPTION	SPACE BEFORE	SPACE AFTER	SKIP BEFORE	SKIP AFTER	OUTPUT INDICATORS	AND	AND
	01	O	PRNTRPT	H				07	14	0F		
	02	O	0	R						1P		
	03	O										
	04	O	PUNHCRD	0						10	N21	N22
	05	O	AND							N23		
	06	O	PRNTRPT	D		1				10		
	07	O										
	08	O										
	09	O	PUNHCRD	T	2					L1		
	10	O										
	11	O										
	12	O	PRNTRPT	T		2	2			L1		
	13	O										
	14	O	PRNTRPT	T		2			14	LR		
	15	O										
	16	O	PRNTRPT	T		2	2			MR	16	
	17	O										
	18	O										
	19	O										
	20	O										

1. Printing occurs when there is an overflow condition or when the 1P (first page) indicator is on.
2. Punching occurs only when indicator 10 is on and indicators 21, 22 and 23 are off.
3. Printing occurs when indicator 10 is on.
4. Punching occurs only when the level –1(L1) indicator is on.
5. Printing occurs only when the level–1(L1) indicator is on.
6. Printing occurs only when the Last Record (LR) indicator is on.
7. Printing occurs only when the Matching Record (MR) indicator and indicator 16 are on.

Figure 5–7. Examples of Output Indicator Entries

NOTE: Many times several output lines are written with identical indicators specified in Columns 23-31; that is, several output lines are to be performed under the same conditions. To avoid rewriting the indicators, an asterisk (*) may be entered in Column 25. A line with an asterisk in Column 25 will be performed if the conditions were met for the preceding line of the same (H/D/T) type. Once an asterisk condition has been set up for a series of lines, more stringent conditions can be put on any remaining subset of those lines. The instances in which this asterisk convention cannot be used are as follows:

(1) The first line of a group of field description specifications.

(2) A field description specifying a page numbering field (PAGE,... PAGE7; see 5.3.2(6)) or a field description specification immediately following such a specification.

(3) A total time line if the preceding line is a detail line.

(4) An overflow line if the preceding line is a nonoverflow line, and vice versa.

(5) An AND specification.

(6) When there are more than 150 lines using the same set of conditions in continuous processing, the asterisk cannot be used on line 151. Line 151 must redefine the indicators. Line 152 may again use the asterisk convention.

The asterisk convention also applies to lines in an OR relationship.

Figure 5–8 illustrates the asterisk convention.

REPORT PROGRAM GENERATOR
OUTPUT FORMAT SPECIFICATIONS

PROGRAM _____ PROGRAMMER _____ DATE _____ PAGE ___ OF ___ PAGES

PG. NO.	LINE NO.	FORM TYPE	FILE NAME	TYPE H/D/T	STACKER SELECT	SPACE BEFORE	SPACE AFTER	SKIP BEFORE	SKIP AFTER	OUTPUT INDICATORS AND	OUTPUT INDICATORS AND	FIELD NAME	ZERO SUPPRESS	BLANK AFTER	END POSITION IN OUTPUT RECORD	DATA FORMAT	CONSTANT OR EDIT WORD	STERLING SIGN POSITION	PROGRAM IDENTIFICATION
	01	O	PRINT							10									
	02	O										FIELDA							
	03	O										FIELDB							
	04	O	*EXECUTED IF INDICATOR 10 IS ON																
	05	O																	
	06	O	PRINT							*									
	07	O										FIELDC							
	08	O										FIELDD							
	09	O	*EXECUTED IF INDICATOR 10 IS ON																
	10	O																	
	11	O	PRINT							*	11								
	12	O										FIELDE							
	13	O										FIELDF							
	14	O	*EXECUTED IF INDICATOR 10 AND 11 ARE ON																
	15	O																	
	16	O	PRINT							*									
	17	O	*EXECUTED IF INDICATOR 10 AND 11 ARE ON																
	18	O																	
	19	O																	
	20	O																	
	21	O	PRNTLNE							10									
	22	O	OR							12									
	23	O								*									
	24	O	*LINE 23 WILL BE EXECUTED IF EITHER INDICATOR 10 OR 12 IS ON.																

Figure 5-8 Example of Asterisk Convention

5.3. FIELD DESCRIPTION

The field description entries include specifications about the control of the individual fields of a record and the output format of individual fields within the output file. The field description entries are written starting on the line following the file identification line. Each individual field description is entered on a separate line and Columns 7-22 must always be blank.

NOTE: In general, the sign of a numeric field is located in the rightmost position of that field. A positive sign is indicated by a 12 overpunch; a negative value is designated by an 11 overpunch. If a signed field is printed or punched, then the rightmost position contains one of the letters A-R or a blank. Amount fields may be zero suppressed or edited before printing to remove the algebraic sign from the unit position. An algebraic sign also may be removed in UNIVAC 9200/9300 mode programs by executing an MLLZO or MHLZO operation, specifying an alphanumeric literal of zero without sign in factor 2.

5.3.1. Output Indicators (Columns 23-31)

- Entry

  Optional.

- Purpose

Specifies the point during output processing at which a field is written on tape, disc, printed, or punched. The same types of indicators previously specified for file identification may also be specified in defining individual fields.

5.3.2. Field Name (Columns 32-37)

- Entry

Required; unless the output is to be constant.

- Purpose

Identifies the specific field of the record that is to be written on tape, disc, or is to be printed, or punched.

- Rules

(1) Only names previously defined in the Input Format Specifications or Calculation Specifications forms may be used.

(2) The fields may be listed in any desired sequence, since the entry in Columns 40-43 (end position in output record) specifies the exact print or punch positions.

(3) The field name is blank when a constant is specified in Columns 45-70.

(4) In blocked records, when all records are alike, only one record need be described.

(5) The date in the Supervisor may be accessed by the name D?TE, which is automatically defined by the RPG as a six-position alphanumeric field.

(6) The names PAGE, PAGE1...PAGE7 are restricted. They are used by the RPG to indicate output page numbering.

- Example

Figure 5–9 illustrates field names.

REPORT PROGRAM GENERATOR
OUTPUT FORMAT SPECIFICATIONS

PROGRAM _____ PROGRAMMER _____ DATE _____ PAGE ___ OF ___ PAGES

PG. NO.	LINE NO.	FORM TYPE	FILE NAME	TYPE H/O/T	STACKER SELECT	SPACE BEFORE	SPACE AFTER	SKIP BEFORE	SKIP AFTER	OUTPUT INDICATORS AND	OUTPUT INDICATORS AND	FIELD NAME	ZERO SUPPRESS	BLANK AFTER	END POSITION IN OUTPUT RECORD	CONSTANT OR EDIT WORD	STERLING SIGN POSITION	PROGRAM IDENTIFICATION
	01	O	TAPEOUT							10								
	02	O										DEPT			4			
	03	O										DIVN			7			
	04	O										EMPNO			13			
	05	O										NAME			38			
	06	O										SHIFT			39			
	07	O										RATE			44			
	08	O										PREM		B	48			
	09	O										STATS			50			
	10	O										SECTN			54			
	11	O										INCTV			58			
	12	O										LOAD			62			
	13	O										GROS			69			
	14	O										TAX			74			
	15	O										FICA			79			
	16	O										INS		B	84			
	17	O										BOND		B	89			
	18	O										PENS		B	94			
	19	O										CRUN		B	100			
	20	O										SAVE		B	105			
	01	O	DAILYRPTH					06		NOF								
	02	O	OR					04		05								
	03	O								NOF		ORDRNO			112			
	04	O								NOF		DATE			119			
	05	O								NOF		SALESH			138			
	06	O										CUST			135			
	07	O										INVOICZ			46			
	08	O										PAGE			68			

Figure 5-9. Example of Asterisk Convention

5.3.2.1. PAGE Field Name

Page numbering may be accomplished automatically for a report file by indicating, on the print specification line, the word PAGE in the field name area, and the rightmost print position of the desired printing area in the end position of the output record.

- PAGE is a restricted name. It may be used as a field name only to indicate page numbering.

- Pages are numbered from 1-9999. The RPG object program allocates a four-position decimal working storage area to the PAGE field. Each time the value of the PAGE field is to be an output, the object program increases its value by one before the output operation.

- Page numbering starts automatically with page 1. If it is desired to begin page numbering with a page other than 1, the programmer may define a header card on the Input Format Specifications form. This header card contains the desired starting page number, less one, because the page number is always increased by 1 before printing occurs. In such a situation the record indicator of the header card should be used in lieu of the 1P indicator.

- Any output indicators specified in a PAGE line are checked before printing. If all indicators are on, the page counter is reset to 1, instead of being incremented by 1.

- The PAGE field should be zero suppressed when it is printed so the sign of the field is eliminated.

- If more than one report file is being produced, any of seven additional page number fields can be used (PAGE1, PAGE2, PAGE3, PAGE4, PAGE5, PAGE6, PAGE7). The additional fields are initialized, incremented, and reset as described for PAGE.

5.3.3. Zero Suppress (Column 38)

■ Entry

Conditional; may not be used if an edit word is specified in Columns 45-70.

■ Purpose

Eliminates zeros that appear to the left of the first significant digit of a field and removes zone bits from the units position of the field.

■ Code

Z – Specifies zero suppression.

■ Rules

(1) All amount fields should be zero suppressed or edited before printing to remove the algebraic sign from the units positions. If the zone bits (algebraic sign) are not removed, the units position prints as an alphabetic letter or blank. These same fields, when specified for punching, generally should not specify zero suppression so that the sign position and leading zeros are punched into the card.

(2) The zero suppress column must be blank if an edit word is specified for a field.

■ Example

See Figure 5–9.

5.3.4. Blank After (Column 39)

- Entry

  Operational.

- Purpose

  Resets alphabetic output fields to blanks or numeric fields to zeros after an output operation has been completed.

  This specification also has an additional feature. If the output field being reset by the blank after specification is also being tested for zero or blank on the Input or Calculation Specifications, the corresponding indicator is turned on after the output field is reset. However, the associated plus or minus indicators are not turned off. If a given field is specified for output at more than one position in the *same* record, the blank after entry is made only on the last specification where the field is used. Resetting of calculation and input indicators is never done in 9200/9300 mode programs.

- Code

  B – Specifies that the field is to be reset to blanks or zeros.

  NOTE: If the blank after specification is used on a constant line, then the constant will be blanked and lost for use in subsequent operations. The RPG stores constants only upon program initialization; it also eliminates duplicates.

- Example

  See Figure 5–9.

5.3.5. End Position in Output Record (Columns 40-43)

- Entry

  Required.

- Purpose

  Defines the print position, card column, or record position of the rightmost character of the output field.

- Codes

  0001-9999

- Rules

  (1) Entries are right-justified; leading zeros may be omitted.

(2) In all indexed sequential files, the key is considered part of the output record. Data must always be provided for the key field locations in the record. (For update records, the output key must be the same as the input key.) In unblocked indexed sequential files, the record key always starts in location 1 and the record data starts in 1 + KL, where KL is the Key Length specified in the File Description (Columns 29-30).

- Example

See Figure 5–9.

5.3.6. Data Format (Column 44)

- Entry

Optional; used only for packed numeric format on tape, disc, or punched output.

- Purpose

Specifies that a field is to be punched or written in packed format.

- Codes

P – Packed format. (Required for numeric key fields of an indexed sequential file that will later be accessed – by another RPG program – as a chained file.)

Blank – Unpacked format.

- Rules

(1) Individual fields may be specified as either packed or unpacked format, as required.

(2) If an edit control word or sterling entry is specified for the field, this column must be blank.

5.3.7. Constant or Edit Word (Columns 45-70)

- Entry

Optional.

- Purpose

Provides for the creation of constants that may be printed or punched or permits editing of numeric fields on printed reports.

- Rule

Constants and edit words must be left-justified.

5.3.7.1. Constant

- Purpose

Permits the programmer to specify an actual value that is to appear in the output. This provides a way of placing values in the RPG object program that do not change from one running of the object program to the next. This mechanism is typically used to set up headings.

- Rules

(1) The field name must be blank.

(2) An alphanumeric constant may be a maximum of 24 characters in length and is always enclosed in apostrophes. Any character, including space and apostrophe, may be used to form a constant. If a single apostrophe is required within a constant, it must be specified as two consecutive single apostrophes.

(3) A hexadecimal constant consists of a string of a maximum of 22 hexadecimal digits preceded by X' and followed by an apostrophe sign ('). The hexadecimal digits and their values are as follows:

0 – 0000	8 – 1000
1 – 0001	9 – 1001
2 – 0010	A – 1010
3 – 0011	B – 1011
4 – 0100	C – 1100
5 – 0101	D – 1101
6 – 0110	E – 1110
7 – 0111	F – 1111

Each hexadecimal digit represents a half byte of information.

Some examples of hexadecimal constants and their values are:

Hexadecimal Representation	Binary Value
X'D'	00001101
X'101'	00000001 00000001
X'3FFF'	00011111 11111111

5.3.7.2. Edit Word

- Purpose

Permits the programmer to punctuate a print field with commas, a decimal point, and a preceding dollar sign; to suppress leading zeros; and to express signed condition of the field. The edit word consists of two parts: body and status.

(1) The body is that portion which begins with the leftmost character of the edit word following the leading apostrophe and continues to the right to include the character controlling the transfer of the units position of the data field.

(2) The status of an edit word is the portion continuing to the right from the body including all characters preceding the closing apostrophe. The credit symbol (CR) or minus sign (–), if present, may appear anywhere within the status. The letters CR or the minus symbol in the status portion of the edit word are undisturbed if the sign in the data field is minus. If the sign is plus, they are blanked out.

- Rules

(1) An edit word must be enclosed in apostrophes.

(2) The first zero or asterisk encountered by RPG, scanning from left to right, is the significance start character. The significance start character is also a digit select character. If the significance start character is a zero, the fill character is a space; if the significance start character is an asterisk, so is the fill character.

(3) A blank in the edit word is replaced with the digit from the corresponding position of the data field specified in field name (digit select). A blank position is referred to as a digit position.

(4) A zero is used to stop zero suppression. It is put in the rightmost position of the edit word where zero suppression is to take place and is replaced with the character from the corresponding position of the data field, unless that character is zero.

(5) An asterisk in the body of a control word is used for asterisk protection and zero suppression. It is put in the rightmost position of the edit word where zero suppression is to take place. It is replaced with the character from the corresponding position of the data field unless that character is a zero and there is no significant digit to its left. Each zero that is suppressed is replaced by an asterisk.

(6) Asterisks to the right of the CR or minus symbol are undisturbed. They are normally used to indicate a specific class of totals.

(7) The body of the edit word may not contain a minus sign.

(8) Decimals and commas are printed in the same position in which they were written unless they are to the left of the significant characters in which case they are replaced by an asterisk or blanked out.

(9) An edit word can contain a maximum of 15 digit positions (except for a sterling field for which the maximum is 16 digits).

(10) An ampersand causes a space in the edited field. There is no way to obtain an ampersand in the edited field.

(11) A dollar sign to the left of the zero suppression code causes the dollar sign to be inserted immediately preceding the first significant digit. This is called the floating dollar sign. If the dollar sign is to be printed when all digits are significant, the edit word must start with an ampersand to allow a space in which it can be printed.

(12) If a dollar sign is entered immediately after the leading apostrophe, it is printed in this location and is called a fixed dollar sign. For example: in the edit word '$0bb', the dollar sign is considered to be fixed. The fixed dollar sign feature is available for the UNIVAC 9400 and IBM 360/20 mode of compilation. In the UNIVAC 9200/9300 mode of compilation, the dollar sign is always floated.

- Example

Figure 5-10 illustrates the use of edit words and Figure 5-11 presents examples of constants and edit words in various situations.

5.3.8. Sterling Sign Position

- Entry

Conditional; used only when sterling currency is being processed.

- Purpose

The user should read Appendix D before using this specification.

- Codes

S – Entered in Column 74 for output with the sign in standard position (see Appendix D).

0001-9999 – Entered right-justified in Columns 71-74 to indicate the location of a sign in nonstandard position.

- Rules

(1) If a field is defined as sterling on input but is not defined as sterling on output specifications, then the output is in pence format.

(2) Unedited printer representation allocates one print position to each pound position, two print positions for shillings, two print positions for integral pence, and one print position for each fractional pence position of the shillings and pence fields. The sign is associated with the least significant pense position (integral or fractional).

(3) The rules governing the use of edit control words for sterling notation are identical to those for decimal fields.

REPORT PROGRAM GENERATOR
OUTPUT FORMAT SPECIFICATIONS

PROGRAM ..... PROGRAMMER ..... DATE ..... PAGE ..... OF ..... PAGES

PG. NO.	LINE NO.	FORM TYPE	TYPE H O T	FILE NAME	STACKER SELECT	SPACE AFTER	SPACE BEFORE	SKIP BEFORE	SKIP AFTER	OUTPUT INDICATORS AND	OUTPUT INDICATORS AND	FIELD NAME	ZERO SUPPRESS	BLANK AFTER	END POSITION IN OUTPUT RECORD	CONSTANT OR EDIT WORD	STERLING SIGN POSITION	PROGRAM IDENTIFICATION
	01	O										TOTAMT			25	' $0 &CR TOTAL		
	02	O	* IF TOTAMT IS							000045	98	EDITED		OUTPUT		WILL BE $45.98 TOTAL		
	03	O	*							000045	98	EDITED		OUTPUT		WILL BE $45.98CR TOTAL		
	04	O	*							000000	00	EDITED		OUTPUT		WILL BE $.00 TOTAL		
	05	O	*							012436	05	EDITED		OUTPUT		WILL BE $12,436.05CR TOTAL		
	06	O																
	07	O										AMT			45	' CR **		
	08	O	* IF AMT IS							001234	56	EDITED		OUTPUT		WILL BE 1,234.56 **		
	09	O	*							001234	56	EDITED		OUTPUT		WILL BE 1,234.56CR **		
	10	O	*							000000	00	EDITED		OUTPUT		WILL BE **		
	11	O																
	12	O										FINTOT			80	' FINAL 0		
	13	O	* IF FINTOT IS							000216	30	EDITED		OUTPUT		WILL BE FINAL 216.30		
	14	O								000216	30	EDITED		OUTPUT		WILL BE FINAL 216.30		
	15	O	*							000000	05	EDITED		OUTPUT		WILL BE FINAL .05		
	16	O																
	17	O										TOTAL			90	' CR BALANCE		
	18	O	* IF TOTAL IS							000006	78	EDITED		OUTPUT		WILL BE 6.78 BALANCE		
	19	O	*							000123	45	EDITED		OUTPUT		WILL BE 123.45CR BALANCE		
	20	O	*							000000	00	EDITED		OUTPUT		WILL BE BALANCE		
	21	O	*							000000	00	EDITED		OUTPUT		WILL BE CR BALANCE		
	22	O																
	23	O										SUBTOT			80	' $0 .		
	24	O	* IF SUBTOT IS							02625		EDITED		OUTPUT		WILL BE $ 26.25		
	25	O	*													$26.25 (9300 MODE)		

Figure 5–10. Examples of Edit Words

REPORT PROGRAM GENERATOR
OUTPUT FORMAT SPECIFICATIONS

PROGRAM _____ PROGRAMMER _____ DATE _____ PAGE ___ OF ___ PAGES

PG. NO.	LINE NO.	FORM TYPE	FILE NAME	TYPE H/D/T	STACKER SELECT	SPACE BEFORE	SPACE AFTER	SKIP BEFORE	SKIP AFTER	OUTPUT INDICATORS AND	OUTPUT INDICATORS AND	FIELD NAME	ZERO SUPPRESS	BLANK AFTER	END POSITION IN OUTPUT RECORD	DATA FORMAT	CONSTANT OR EDIT WORD	STERLING SIGN POSITION	PROGRAM IDENTIFICATION
1	3 5	6	7 14	15	16	17	18	19 20	21 22	23 25	26 28	29 31	32 37	38	39 40 43	44	45 70	71 74	75 80
01	01	O	PRNTRPT H					14	04	OF									
	02	O	OR							1P									
	03	O													18		'FIELD ONE HEADER'		
	04	O													27		'FIELD 2'		
	05	O													36		'FIELD 3'		
	06	O													49		'FIELD 4'		
	07	O													64		'FIELD 5'		
	08	O													82		'FIELD SIX'		
	09	O													98		'FIELD 7'		
	10	O													116		'FIELD 8'		
	11	O	* THIS HEADER LINE SHOWS									EXAMPLES OF					CONSTANT SPECIFICATIONS. THE		
	12	O	* CONSTANT FIELD ONE HEADER IS									PRINTED IN					THE OUTPUT RECORD ENDING		
	13	O	* AT PRINT POSITION 18. SIMILAR									PRINT					INSTRUCTION INFORMATION IS		
	14	O	* INCLUDED IN ALL THE OTHER									COLUMN TITLES. THE FIELD					NAME SPECIFI-		
	15	O	* CATION MUST BE BLANK									FOR CONSTANTS.					COLUMN 44 IS BLANK TO		
	16	O	* INDICATE UNPACKED									FORMAT.							
	17	O																	
	18	O	PRNTRPT D		1					10									
	19	O										FIELD1			20				
	20	O										FIELD2			28				
	21	O										FIELD3			38				
	22	O										FIELD4	Z		52				
	23	O										FIELD5	Z		68				
	24	O										FIELD6			85		'. $0. &CR'		
	25	O										FIELD7			102		'. $0. &-'		
		O										FIELD8			120		'. $0. &CR'		
		O	* THIS DETAIL PRINT LINE									HAS 8 SPECIFIED					FIELDS. THE FIRST 3 ARE		
		O	* ALPHANUMERIC AND PRINTED									RIGHT-JUSTIFIED					AS INDICATED BY THE		
		O	* COLUMN IN THE END									POSITION FIELD.					THE NEXT 2 ARE NUMERIC AND ARE		
		O	* ZERO SUPPRESSED.									THE NEXT 3 ARE					NUMERIC AND HAVE EDIT WORDS.		

Figure 5–11. Example of Constants and Edit Words

6. FILE EXTENSION SPECIFICATIONS FORM

6.1. GENERAL DESCRIPTION

The File Extension form (Figure 6-1) provides information to RPG about record address, chaining and tag files, and tables used in the object program. For additional information on tables, refer to Section 8.

REPORT PROGRAM GENERATOR
FILE EXTENSION SPECIFICATIONS

PROGRAM _____ PROGRAMMER _____ DATE _____ PAGE ___ OF ___ PAGES

PG. NO.	LINE NO.	FORM TYPE	RECORD SEQUENCE	CHAINING FIELD NUMBER	FROM FILE NAME	TO FILE NAME	TABLE NAME	NUMBER OF TABLE ENTRIES PER RECORD	NUMBER OF TABLE ENTRIES PER TABLE	LENGTH OF TABLE ENTRY	PACKED	NUMERIC	SEQUENCE	TABLE NAME	LENGTH OF TABLE ENTRY	PACKED	NUMERIC	SEQUENCE	COMMENTS	PROGRAM IDENTIFICATION
1	3 5	6	7 8	9 10	11 18	19 26	27 32	33 35	36 39	40 42	43	44	45	46 51	52 54	55	56	57	58 74	75 80

Figure 6-1. File Extension Specifications Form

6.2. FORM ENTRIES

The entries for the File Extension Specifications form and their relation to the RPG program are explained in the following paragraphs.

6.2.1. Record Sequence of Chaining Files (Columns 7-8)

- Entry

  Required for chaining files; in the UNIVAC 9200/9300 mode of compilation, this field should be blank.

- Purpose

  Specifies the sequence of the input files. This entry is the same as that made for chaining files on the Input Specifications form Columns 15-16.

- Codes

  01-99 – Consecutive sequence of records. Two alphabetic characters – indicates records do not have to be in sequence.

6.2.2. Chaining Field Number (Columns 9-10)

- Entry

  Conditional; for chaining files only.

- Purpose

  Defines a meaningful chaining field.

- Codes

  C1-C9 – Identifies the number of the chaining fields.

- Rules

  (1) The code entered must be the same code entered in Columns 61-62 of the Input Specifications form.

  (2) This field is blank if the file is not a chaining file.

6.2.3. From File Name (Columns 11-18)

- Entry

  Required.

- Purpose

  Defines the relationship between files for the RPG program. This field is used in conjunction with the to file name entry. See Table 6-1 for a summary of this relationship.

- Codes

  Any file name specified and defined in the File Description Specifications form.

■ Rules

(1) The file name must be left-justified.

(2) For table files:
- A file may contain one or more tables. If read at object time, it is expected to conform to the file description referenced by the specified file name. If produced at object time, it is produced in conformance with the file description.
- A table produced at object time may be subsequently processed as a file. In such a case, record length for the file is equal to the length of table entry (Columns 40-42) of the File Extension form plus the length of table entry (Columns 52-54), and block length is equal to record length multiplied by the number of table entries per record (Columns 33-35) of the File Extension form.
- If a table file is to be read from an input unit, the file name of the input unit is written in the from file name entry.
- If a table file is to be written on an output unit, the file name of the output unit is written in the to file name entry.

FILE TYPE	FROM FILE NAME (Columns 11-18)	TO FILE NAME (Columns 19-26)
Record Address (RA) Files	The name of the record address (RA) file is entered here.	The name of the file containing the data records is entered here.
Chaining Files	The name of the chaining file defined in Columns 7-14 on the File Description Specifications form is entered here. This file has the data record containing the chaining field.	The name of the chained file defined in Columns 7-14 of the File Description Specifications form is entered here. This is the file from which the data record is obtained.
Tag Files	The name of the tag file is entered here.	The name of the file being accessed on the basis of the tag file is entered here.
Table Files	If a table is being defined (Columns 27-57), the name of the file containing the table data is entered here.	If the table is to be printed, punched, or written after updating, the name of the table output file is entered here.

Table 6–1. Summary of From File Name/To File Name

6.2.4. To File Name (Columns 19-26)

The entries for this field are the same as those described in from file name (see Table 6-1).

6.2.5. Table Name (Columns 27-32)

- Entry

  Required for table files.

- Purpose

  A table file may include a table of arguments and a table of functions. This specification contains the name of either table.

- Code

  TABnnn - nnn is any combination of alphanumeric characters.

- Rules

  (1) The entry must be left-justified.

  (2) Embedded blanks are not permitted.

6.2.6. Number of Table Entries Per Record (Columns 33-35)

- Entry

  Required for tables.

- Purpose

  Specifies the maximum number of table entries (arguments or functions) contained in each input record.

- Codes

  1-999

- Rules

  (1) The entry must be right-justified; leading zeros may be omitted.

6.2.7. Number of Table Entries Per Table (Columns 36-39)

- Entry

  Required for table files.

- Purpose

  Contains the total number of entries (arguments or functions) for the specified table.

- Codes

  1-9999

- Rules

  (1) The number must be right-justified; leading zeros may be omitted.

6.2.8. Length of Table Entry (Columns 40-42)

- Entry

  Required for table files.

- Purpose

  Specifies the length of each table entry.

- Codes

  1-256 for alphanumeric entries.
  1- 15 for numeric entries.

- Rule

  (1) The number must be right-justified; leading zeros may be omitted.

6.2.9. Packed (Column 43)

- Entry

  Required if the data in the table is in packed decimal format.

- Codes

  P – packed format.
  Blank – unpacked format or alphanumeric data.

6.2.10. Numeric (Column 44)

- Entry

  Conditional.

- Purpose

  Indicates the number of decimal positions for numeric table entries.

- Codes

  0 – No decimal positions.
  1-9 – Number of decimal positions.

- Rules

  (1) For numeric fields, a decimal position must be indicated.
  (2) For alphanumerics, this field must be blank.

6.2.11. Sequence (Column 45)

- Entry

  Conditional.

- Purpose

  Indicates the sequence of the data in the corresponding table.

- Codes

  A – Ascending order.
  D – Descending order.
  Blank – Not in sequence.

6.2.12. Table Name (Columns 46-51)

- Entry

  Optional; used for alternating formats only.

- Purpose

  When the alternating argument and function tables are used, enter the second name here.

- Codes

  TABnnn - nnn is any combination of alphanumeric characters.

- Rules

  (1) The entry must be left-justified.

  (2) Embedded blanks are not permitted.

6.2.13. Length of Table Entry (Columns 52-54)

- Entry

  Optional; used for alternating formats.

- Purpose

  Specifies the length of each table entry.

- Codes

  1-256 for alphanumeric entries.
  1- 15 for numeric entries.

- Rules

  (1) The number must be right-justified; leading zeros may be omitted.

6.2.14. Packed (Column 55)

- Entry

  Required if the data in the alternating table file is in packed decimal format.

- Codes

  P - packed format.
  Blank - unpacked format or alphanumeric data.

6.2.15. Numeric (Column 56)

- Entry

  Conditional; used for alternating formats.

- Purpose

  Indicates the number of decimal positions for numeric table entries.

- Codes

  0 – No decimal positions.
  1-9 – Number of decimal positions.

- Rules

  (1) For numeric fields, a decimal position must be indicated.

  (2) For alphanumerics, this field must be left blank.

6.2.16. Sequence (Column 57)

- Entry

  Conditional; used for alternating formats.

- Purpose

  Indicates the sequence of the data in the corresponding table.

- Codes

  A – Ascending order.
  D – Descending order.
  Blank – Not in sequence.

6.2.17. Comments (Columns 58-74)

Can be used for comments or left blank.

7. LINE COUNTER SPECIFICATIONS FORM

7.1. GENERAL DESCRIPTION

The Line Counter Specifications form (Figure 7-1) allows a report which is to be printed at a later time to be stored on an intermediate tape or disc file. It also provides an internal control when an online printer is used.

A carriage control character is added as the first byte of each record that is stored on an intermediate device. Space before and skip before controls are put out as an independent carriage control character in an individual record that does not print. Space after and skip after controls are put out as modifier bits in conjunction with the write command for the line. Thus, a line that has only space before or skip before control produces two records on the intermediate device, and a line that has skip after or space after control produces only one record. To retrieve these records for printing an auxiliary program written in RPG or another programming language must be used (see Appendix E). The carriage control characters used are described in *UNIVAC 9400 System Data Management System Programmers Reference, UP-7629* (current version). The record format is always blocked variable (VARBLK).

REPORT PROGRAM GENERATOR
LINE COUNTER SPECIFICATIONS

PROGRAM _____ PROGRAMMER _____ DATE _____ PAGE ___ OF ___ PAGES

PG. NO.	LINE NO.	FORM TYPE	FILENAME	1 LINE NO.	1 CH. NO.	2 LINE NO.	2 CH. NO.	3 LINE NO.	3 CH. NO.	4 LINE NO.	4 CH. NO.	5 LINE NO.	5 CH. NO.	6 LINE NO.	6 CH. NO.	7 LINE NO.	7 CH. NO.	8 LINE NO.	8 CH. NO.	9 LINE NO.	9 CH. NO.	10 LINE NO.	10 CH. NO.	11 LINE NO.	11 CH. NO.	12 LINE NO.	12 CH. NO.	PROGRAM IDENT.
1	3 5	6	7 14	15 17		20 22		25 27		30 32		35 37		40 42		45 47		50 52		55 57		60 62		65 67		70 72		75 80

Figure 7-1. Line Counter Specifications Form

7.2. FORM ENTRIES

The following descriptions explain the entries on the Line Counter Specifications form and demonstrate their functions in relation to the RPG program.

7.2.1. File Name (Columns 7-14)

- Entry

  Required; in the UNIVAC 9200/9300 mode of compilation, file name may be specified in Columns 50-57 instead of Columns 7-14.

- Purpose

  Identifies the report file from which lines are printed at a later time.

- Rules

  (1) The entry should be identical to the file name in the File Description form. A maximum of eight characters may be used but the first seven will identify the file.

  (2) The file name must be left-justified.

  (3) On the File Description Specifications form, Column 19 must contain a V and Column 39 must contain an L. The L indicates that a Line Counter Specifications form is required for the output file.

- Example

  Figure 7-2a is a completed Line Counter Specifications form and Figure 7-2b is the printer spacing chart from which it is derived.

7.2.2. Line Number (Columns 15-17)

- Entry

  Required.

- Purpose

  Identifies the line number on which a channel punch is entered when deferred printing takes place.

- Code

  001-999

- Rules

  (1) The entry must be right-justified.

  (2) Leading zeros may be omitted.

REPORT PROGRAM GENERATOR

LINE COUNTER SPECIFICATIONS

PROGRAM _____ PROGRAMMER _____ DATE _____ PAGE ___ OF ___ PAGES

FORM TYPE

PG. NO. 1	LINE NO. 3 5	6	FILENAME 7 14	1 LINE NO. 15 17	1 CH. NO.	2 LINE NO. 20 22	2 CH. NO.	3 LINE NO. 25 27	3 CH. NO.	4 LINE NO. 30 32	4 CH. NO.	5 LINE NO. 35 37	5 CH. NO.	6 LINE NO. 40 42	6 CH. NO.	7 LINE NO. 45 47	7 CH. NO.	8 LINE NO. 50 52	8 CH. NO.	9 LINE NO. 55 57	9 CH. NO.	10 LINE NO. 60 62	10 CH. NO.	11 LINE NO. 65 67	11 CH. NO.	12 LINE NO. 70 72	12 CH. NO.	PROGRAM IDENT. 75 80
	01	L	ACCREC	003	14	011	07	035	09																			
	02	L	* THE REPORT HEADING LINE IS TO BE LOCATED IN LINE 3 -																									
	03	L	* THEREFORE LINE 3 IS INDICATED IN COLUMNS 15-17 IN ASSOCIATION																									
	04	L	* WITH THE HOME PAPER CHANNEL, CHANNEL 14, IN COLUMNS 18-19.																									
	05	L	* THE OVERFLOW LINE (CHANNEL 9 PUNCH) IS LINE 35.																									
	06	L	* SINCE THE REPORT OUTPUT REQUIRES A SKIP TO CHANNEL 7,																									
	07	L	* WHICH IS TO BE LOCATED AT LINE 11, THE ENTRIES IN COLUMNS																									
	08	L	* 20-24 ARE REQUIRED.																									

Figure 7–2a. Example of Completed Line Counter Specifications Form

RPG PRINTER FORMAT CHART

FORM NUMBER _____ APPLICATION _____

FORM PARTS _____ RUN NAME _____ RUN NUMBER _____

TYPE OF PRINTOUT _____ RECORD NAME _____ RECORD NUMBER _____

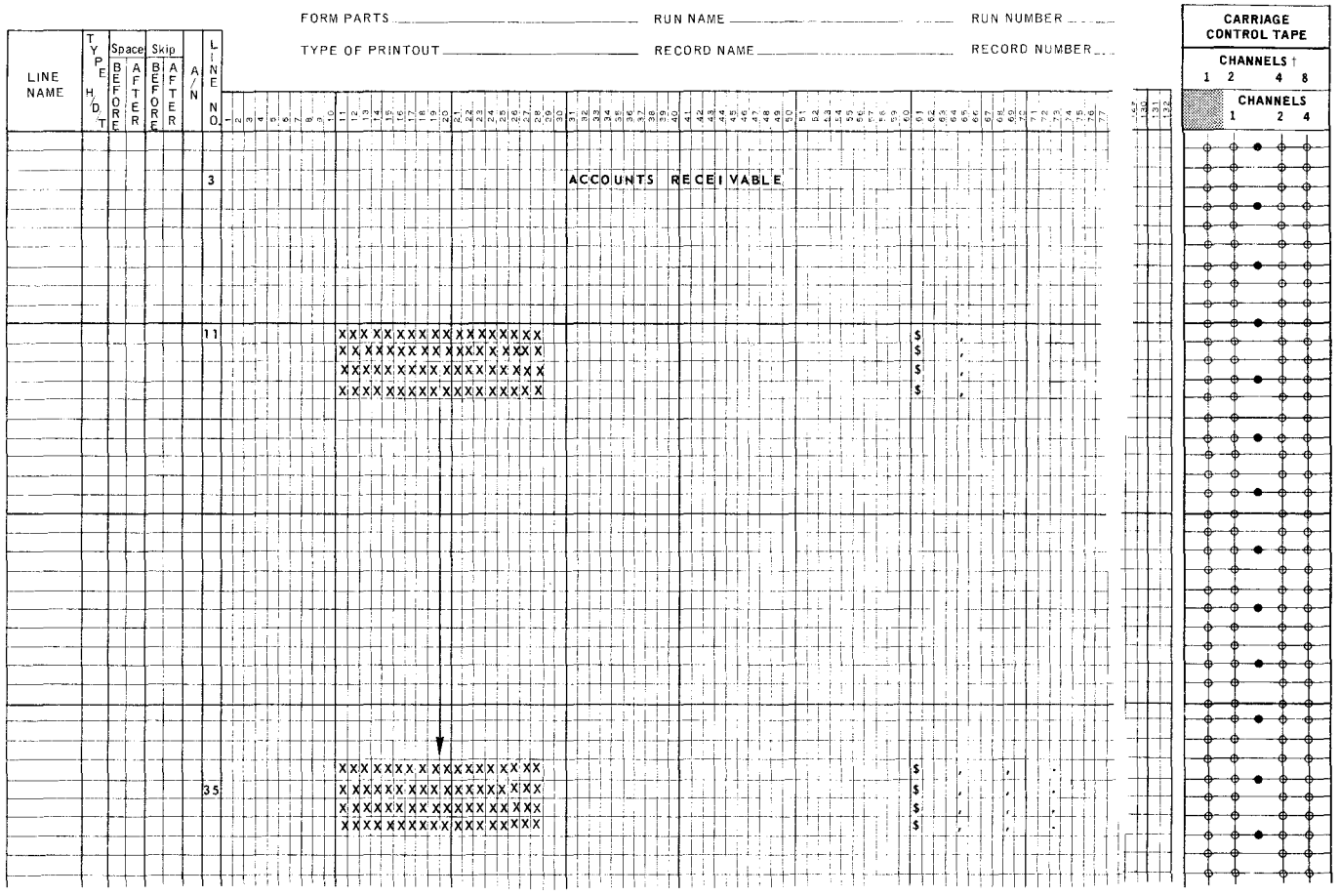


Figure 7–2b. Printer Spacing Chart

7.2.3. Channel Number (Columns 18-19)

- Entry

  Required.

- Purpose

  Identifies the channel number associated with the line number given in the preceding three columns of the Line Counter Specifications form.

- Codes

  04-15 – UNIVAC 9400 channel numbers
  01-07, 15 – UNIVAC 9200/9300 mode programs
  01-12, 15 – IBM 360 Model 20 mode programs
  (See 5.2.6, Rule 3.)

- Rules

  (1) The entry must be right-justified.

  (2) Leading zeros may be omitted.

  (3) When the Line Counter Specifications form is used, at least one of the following entries are required:

  - In UNIVAC 9400 mode of compilation, entries for channel 9 and channel 14 (or 15).
  - In UNIVAC 9200/9300 mode of compilation, entries for channel 1 and channel 7 (or 15).
  - In IBM 360/20 mode of compilation, entries for channel 12 and channel 1 (or 15).

  (4) Form overflow is detected two lines after the specified overflow line is reached. (For the purposes of this rule, each line is defined as a separate output record directed to the printer, tape, or disc file. Note that each "line" requested in the Output-Format Specifications may generate as many as four separate output records. A line with or without a Space After or Skip After request generates a single output record. Additional space and skip requests for the same line generate one additional record each.)

  If a UNIVAC 9200/9300 or IBM 360 Model 20 program is being compiled which includes a skip to the overflow channel, the program may have to be revised to operate properly on the UNIVAC 9400.

NOTE: The five columns (line number and channel number) are repeated across the Line Counter Specifications form allowing for a total of 12 carriage loop punches. (In the UNIVAC 9200/9300 mode, these entries cannot extend beyond Column 49, allowing for a total of seven carriage loop punches.)

8. TABLE HANDLING

8.1. GENERAL DESCRIPTION

The File Extension Specifications form is used to describe data tables (see Figure 6-1).

A table is a collection of data stored in a form suitable for ready reference, usually in sequential machine locations. A table may consist of two parts:

- Argument – in looking up a quantity in a table, the data that identifies the location of the desired value (or function) is called the argument. The argument may also be defined as the variable which, when a certain value is added to its location, determines the location for the value of the function. The argument is more commonly known as the search value used to locate the desired value necessary for calculation.

- Function – a value corresponding to an argument, stated or implied. It is obtained from a table of values in memory.

RPG enables the programmer to use tables in the object program. They are loaded into storage by the RPG object program before any processing.

Tables used in the LOKUP operation may consist of either arguments or functions. As illustrated in Figures 8-1a and 8-1b, table input records may consist of the following:

(1) Arguments (identification)

(2) Function (desired value)

(3) Alternating arguments and functions

(4) Alternating functions and arguments

} alternating input formats

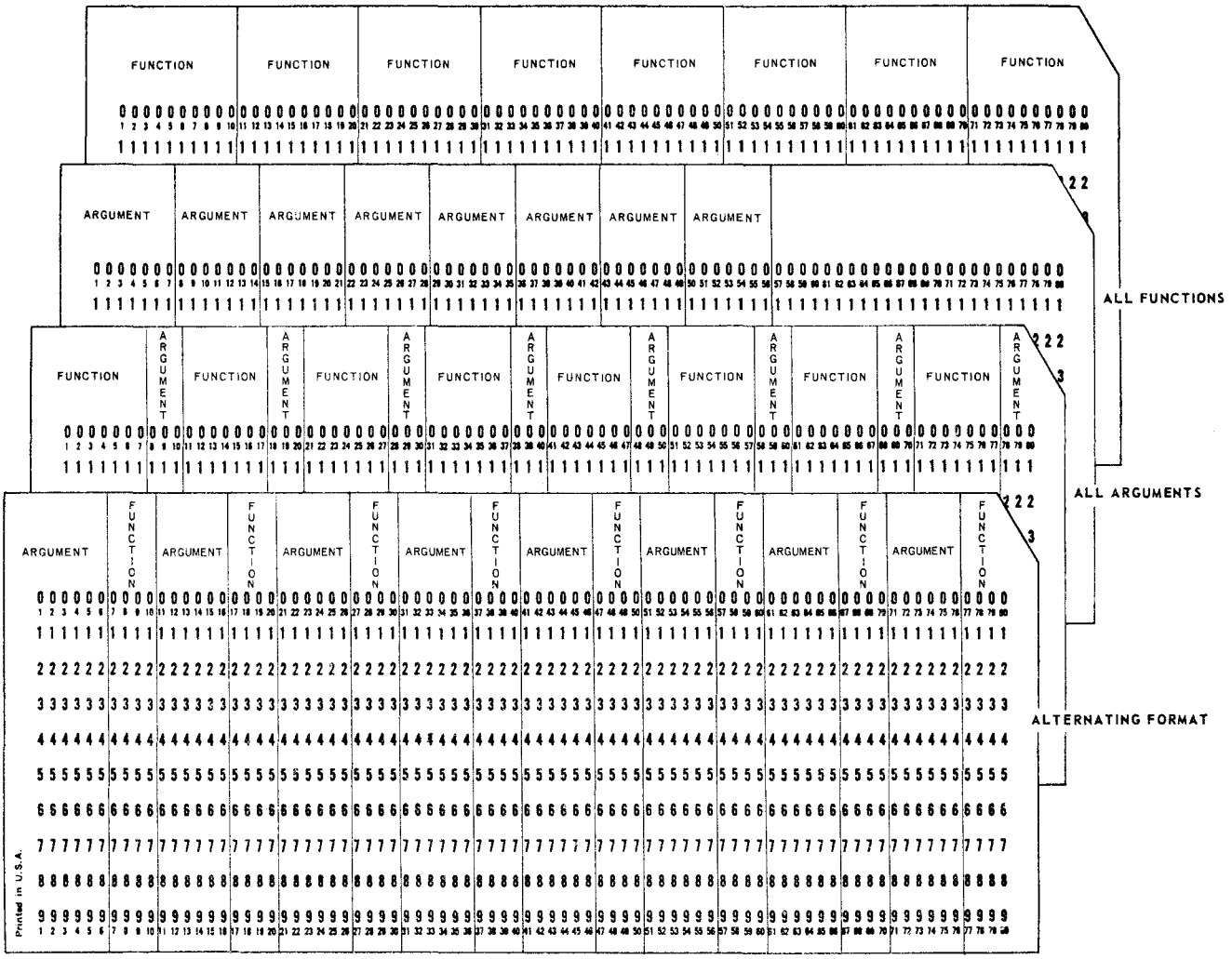
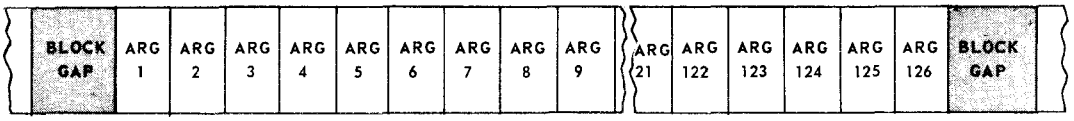
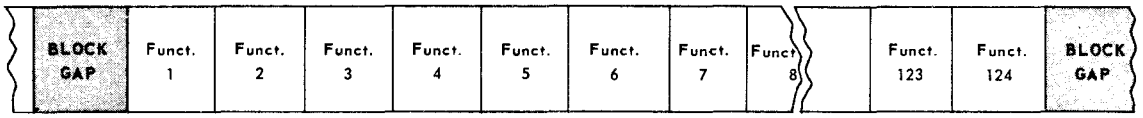


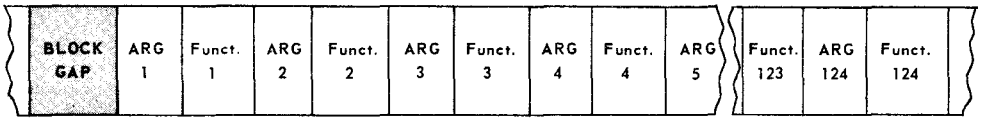
Figure 8–1a. Example of Table Input Record Formats–Cards



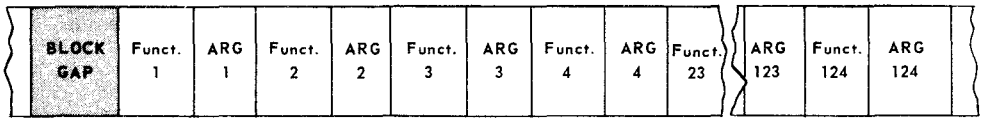
DISC/TAPE TABLE BLOCK – ALL ARGUMENTS



DISC/TAPE TABLE BLOCK – ALL FUNCTIONS



DISC/TAPE TABLE BLOCK – ALTERNATING ARGUMENTS & FUNCTIONS



DISC/TAPE TABLE BLOCK – ALTERNATING FUNCTIONS & ARGUMENTS

Figure 8–1b. Example of Table Input Record Formats–Disc/Tapes

The rules for preparing a table input file are:

(1) Each table is assigned a unique name which can have a maximum of six characters. The first three characters must be TAB, and the remaining three may be either alphanumeric or numeric.

(2) Each unit of table data is a table entry; that is, each argument is a table entry and each function is a table entry.

(3) All tables may be loaded from the same file in the same order that they are specified on the File Extension Specifications form.

(4) Tables may be in ascending or descending sequence or in no sequence. If the tables are not in sequence, only an equal search can be performed.

(5) The records of a table must be in a sequentially organized file.

(6) The first table entry must start in position 1 of the input record and each table entry must be adjacent to the preceding entry with no blank positions in between.

(7) Every record of an input table except the last must contain the same number of entries.

(8) All arguments of a table must be the same size, and all functions of a table must be the same size. Entries may not overflow from one record to another.

(9) For alternating formats, each input record must begin with an entry of the same type; each record must always begin with an argument or each record must always begin with a function.

(10) When alternating tables are used, the table entries must be together; function 5 must be in the same record as argument 5.

(11) Entries may be alphanumeric or numeric. Alphanumeric entries must not exceed 256 characters and the maximum size of numeric entries is 15 digits. Numeric entries may be packed or unpacked.

(12) Columns 46-57 are used on the File Extension Specifications form only when an alternating input format is specified.

8.2. LOKUP PROCEDURE

The LOKUP operation causes the RPG to search a table contained in storage, to indicate a find, and to secure from the table specified data needed in subsequent calculations. The LOKUP operation may specify a single table operation or a two-table operation.

8.2.1. Single Table Operation

Based on conditions specified in resulting indicators, the RPG LOKUP operation searches the table specified in factor 2 for the value specified in factor 1. After the entry from the argument table has been found, the corresponding function from the function table is located and placed in the special holding area of the function table. The resulting indicators may specify the following conditions:

High

Low

Equal

High or Equal

Low or Equal

The following results can be obtained from a single table search.

(1) High (HI) Indicator Specified

- If a value is located in the table which is immediately higher than the search value (factor 1), this located value is moved to the table's special holding area and the HI indicator is set.

- If the highest value in the table is lower than the search value, the HI indicator is reset.

(2) Low (LO) Indicator Specified

- If a value is located in the table which is immediately lower than the search value, this located value is moved to the table's special holding area and the LO indicator is set.

- If the lowest value in the table is higher than the search value, the LO indicator is reset.

(3) Equal (EQ) Indicator Specified

- If the value in factor 1 *is located* in the table, the EQ indicator is set (turned on) and the value from the table is moved to the table's special holding area.

- If the value in factor 1 *is not located* in the table, the EQ indicator is reset (turned off).

(4) High (HI) and Equal (EQ) Indicators Specified

NOTE: Do not specify the same indicator number for both HI and EQ.

- If HI is set, a value immediately higher than the search value is in the table's special holding area.

- If EQ is set, the equal value from the table is in the table's special holding area.

- If both HI and EQ are reset, the highest value in the table is lower than the search value.

(5) Low (LO) and Equal (EQ) Indicators Specified

NOTE: Do not specify the same indicator number for both LO and EQ.

- If LO indicator is set, a value immediately lower than the search value is in the table's special holding area.

- If EQ indicator is set, the equal value from the table is in the table's special holding area.

- If both LO and EQ are reset, the table's lowest value is higher than the search value.

High or low must not be specified unless the table is in either ascending or descending sequence.

Subsequent operations may reference the stored value in the special holding area by referencing the table name. Figure 8–2 illustrates the entries necessary for single table LOKUP.

REPORT PROGRAM GENERATOR
CALCULATION SPECIFICATIONS

PROGRAM _____ PROGRAMMER _____ DATE _____ PAGE ___ OF ___ PAGES

PG. NO.	LINE NO.	FORM TYPE	CONTROL LEVEL	INDICATORS AND	INDICATORS AND	FACTOR 1	OPERATION	FACTOR 2	RESULT FIELD	FIELD LENGTH	DECIMAL POSITIONS	HALF ADJUST	RESULTING INDICATORS COMPARE HI 1-2	RESULTING INDICATORS COMPARE LO 1-2	RESULTING INDICATORS COMPARE EQ 1-2	COMMENTS	PROGRAM IDENTIFICATION
	01	C				PARTNO	LOKUP	TABPNO							21		
	02	C*	INDICATOR 21 IS SPECIFIED FOR AN EQUAL CONDITION														

Figure 8–2. Single Table LOKUP Format

8.2.2. Two-Table Operation

The RPG LOKUP searches the Argument Table specified in factor 2 for the search value specified in factor 1 as modified by the condition specified in resulting indicators. On a find, the corresponding value is secured from the Function Table specified in result field. The conditions may be indicated as follows:

High

Low

Equal

High or Equal

Low or Equal

The following results can be obtained from a two-table search:

(1) High (HI) Indicator Specified

- If a value is located in the Argument Table which is immediately higher than the search value, the corresponding value from the Function Table is moved to the Function Table's special holding area.

- If the highest value in the Argument Table is lower than the search value, the HI indicator is reset.

(2) Low (LO) Indicator Specified

- If a value is located in the Argument Table which is immediately lower than the search value, the corresponding value from the Function Table is moved to the Function Table's special holding area.

- If the lowest value in the Argument Table is higher than the search value, the LO indicator is reset.

(3) Equal (EQ) Indicator Specified

- If the value in factor 1 *is located* in the Argument Table (factor 2), then the EQ indicator is set (turned on) and the corresponding value from the Function Table (result field) is moved to the Function Table's special holding area.

– If the value in factor 1 *is not located* in the Argument Table, the EQ indicator is reset (turned off).

(4) High (HI) and Equal (EQ) Indicators Specified

NOTE: Do not specify the same indicator number for both HI and EQ.

– If HI indicator is set, a value from the Function Table will be in the Function Table's special holding area. This value corresponds to a value in the Argument Table which is immediately higher than the search value.

– If EQ indicator is set, a value, from the Function Table is in the Function Table's special holding area. This value corresponds to a value in the Argument Table which is equal to the search value.

– If both HI and EQ are reset, the highest value in the table is lower than the search value.

(5) Low (LO) and Equal (EQ) Indicator Specified

NOTE: Do not specify the same indicator number for both LO and EQ.

– If LO indicator is set, a value from the Function Table is in the Function Table's special holding area. This value corresponds to a value in the Argument Table which is immediately lower than the search value.

– If EQ indicator is set, a value from the Function Table is in the Function Table's special holding area. This value corresponds to a value in the Argument Table which is equal to the search value.

– If both LO and EQ are reset, the table's lowest value in the table is higher than the search value.

High or low must not be specified unless the table is in either ascending or descending sequence.

Subsequent operations may reference the stored value by using the name of the Function Table. Also, the high, low, or equal resulting indicators that were specified in the LOKUP operation are turned on. These indicators can now be used to control calculation and output operations. Figure 8-3 is an example of a two-table LOKUP format.

REPORT PROGRAM GENERATOR
CALCULATION SPECIFICATIONS

PROGRAM _____ PROGRAMMER _____ DATE _____ PAGE ___ OF ___ PAGES

PG. NO.	LINE NO.	FORM TYPE	CONTROL LEVEL	INDICATORS AND	INDICATORS AND	FACTOR 1	OPERATION	FACTOR 2	RESULT FIELD	FIELD LENGTH	DECIMAL POSITIONS	HALF ADJUST	RESULTING INDICATORS COMPARES HI 1-2	RESULTING INDICATORS COMPARES LO 1-2	RESULTING INDICATORS COMPARES EQ 1-2	COMMENTS	PROGRAM IDENTIFICATION
	01	C				PARTNO	LOKUP	TABNO	TABQTY				16		17		
	02	C*		INDICATOR 16 IS SPECIFIED FOR A GREATER THAN CONDITION AND													
	03	C*		INDICATOR 17 IS SPECIFIED FOR AN EQUAL CONDITION													

Figure 8–3. Two-Table LOKUP Format

8.2.3. Resulting Indicators

The high, low, or equal resulting indicators specified in the LOKUP operation determine the type of search to be performed. Therefore, at least one resulting indicator must be specified as a prerequisite for this operation. Any available indicator number may be used. If an equal indicator (Columns 58-59) is specified, the RPG attempts to locate a table entry equal to the search value or argument. If a low indicator (Columns 56-57) is specified, it causes the RPG to search for that table entry which is nearest to, but smaller than, the search value. A specified high indicator (Columns 54-55) causes the RPG to search for a table entry that is nearest to, but greater than, the search value. If two indicators are specified, then the RPG searches for the table entry which is high or equal, or low or equal to the search value. If the search value cannot be found in the Argument Table, then the specified resulting indicators are reset (turned off).

The compare operations used for the LOKUP operation are logical for alphanumeric arguments and algebraic for numeric arguments. The search values must be identical in format to the table entries against which they are compared. If the search value is numeric, then the table entries must be numeric in format. Decimal alignment is not performed if numeric search arguments and table entries have different decimal lengths.

NOTE: The File Extension Specifications form must specify the sequence of the Argument Table if the high or low indicators are used.

8.2.4. General Rules

The following points should be considered when the table lookup operation is used in a program:

(1) Indicators and control level entries (Columns 7-17) are optional.
(2) The argument or search value is always specified in factor 1 and must be the same length as the entries in the Argument Table.
(3) The search value may be a field name or a literal and may contain either numeric or alphanumeric data.
(4) For single tables, the table name is entered in factor 2, and the result field is left blank.
(5) When two tables are used, the Argument Table is specified in factor 2 (Columns 33-38) and the Function Table in the result field (Columns 43-48).
(6) Columns 49-53 must be blank.
(7) If a table name is used in factor 1 or factor 2 of any operation except LOKUP, the data used for this table name is the data that was located or used by the last performed LOKUP operation referring to this table. If no LOKUP has yet been performed in the program, the first entry in the table is used.
(8) If a table name is specified as the search value (factor 1) of the LOKUP operation, then the data used for factor 1 is the data that was located or used on the last performed LOKUP operation referring to this table. If no LOKUP has yet been performed in the program, the first entry in the table is used.
(9) If a table name is specified in the result field of an RLABL statement, the data located or used by the last performed LOKUP operation referring to this table may be used in subroutines contained in the program.
(10) Table names may be referred to in a subroutine through an RLABL statement.

8.2.5. Examples

Figure 8-4 is an example of the following table lookup application.

The table input records are in the alternating argument and function format. TABPNO, the Argument Table, contains the part numbers of 150 critical stock items. TABQTY, the Function Table, contains the corresponding quantity of each part number contained in the Argument Table. The TABPNO entries are six numeric positions in length and are in ascending sequence. Each TABQTY function entry is four numeric positions long.

The cost of inventory is determined by extending the cost of each part number (found in the data input record) by its quantity as found in the Function Table.

The field PARTNO, contained in Columns 1-6 of the data input record, is used as a search value. This field contains the stock number of an inventory item which is to be costed. The cost field in Columns 7-11 indicates the actual cost of the part whose number PARTNO is indicated in the same card.

TABPNO	TABQTY	TABPNO	TABQTY	TABPNO	TABQTY	TABPNO	TABQTY	TABPNO	TABQTY	TABPNO	TABQTY	TABPNO	TABQTY	TABPNO	TABQTY
1–6	7–10	11–16	17–20	21–26	27–30	31–36	37–40	41–46	47–50	51–56	57–60	61–66	67–70	71–76	77–80

REPORT PROGRAM GENERATOR
FILE DESCRIPTION

Line No. (3–5)	Form Type (6)	File Name (7–14)	File Type (15)	File Designation (16)	Device (40–46)
01	F	TABLE	I	T	E READER
02	F	PARTCARD	I	P	READER

REPORT PROGRAM GENERATOR
FILE EXTENSION SPECIFICATIONS

Line No.	Form Type	From File Name (11–18)	To File Name (19–26)	Table Name (27–32)	Number of Table Entries per Record (33–35)	Number of Table Entries per Table (36–39)	Length of Table Entry (40–42)	Packed (43)	Numeric (44)	Sequence (45)	Table Name (46–51)	Length of Table Entry (52–54)	Packed (55)	Numeric (56)	Sequence (57)
01	E	TABLE		TABPNP	8	150	6		0	A	TABQTY	4		0	

PART NO.	COST	DESCRIPTION		CARD TYPE
000000	00000	00000000000000000000	000000000000000000000000000000000000000000000000	0
1 2 3 4 5 6	7 8 9 10 11	12–31	32–79	80
111111	11111	11111111111111111111	111111111111111111111111111111111111111111111111	1
222222	22222	22222222222222222222	222222222222222222222222222222222222222222222222	■

REPORT PROGRAM GENERATOR
CALCULATION SPECIFICATIONS

Line No.	Form Type	Comment (7)	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 (53)	Resulting Indicators EQ (58–59)
01	C			PARTNO	LOKUP	TABPNO	TABQTY				01
02	C	*		SEARCH ARGUMENT TABLE (TABPNO) TO LOCATE A PART NUMBER EQUAL							
03	C	*		TO THE SEARCH VALUE (PARTNO) AS SPECIFIED BY INDICATOR 01 IN							
04	C	*		COLUMN 58-59. IF THE CORRESPONDING FUNCTION IS FOUND, INDICATOR							
05	C	*		01 IS TURNED ON.							
06	C		01	COST	MULT	TABQTY	TOTAL	8	2	H	
07	C	*		THE COST IS MULTIPLIED BY THE QUANTITY AND THE RESULT IS PLACED							
08	C	*		HALF ADJUSTED IN THE FIELD NAMED TOTAL.							
09	C		N01		GOTO	ORDER					
10	C	*		IF INDICATOR 01 IS NOT TURNED ON LINE 06 IS NOT PERFORMED.							
11	C	*		INSTEAD GOTO ORDER WILL TRANSFER TO A ROUTINE WHICH WILL							
12	C	*		INDICATE THAT THE PART NUMBER IS NOT IN THE TABLE.							

Figure 8–4. Table Lookup Applications Example

8.3. EXTENDING TABLES

The LOKUP operation has the facility to update tables and to add new entries to tables.

8.3.1. Updating Tables

Figure 8-5 illustrates updating the Argument Table and the Function Table.

REPORT PROGRAM GENERATOR
CALCULATION SPECIFICATIONS

PROGRAM _____ PROGRAMMER _____ DATE _____ PAGE ___ OF ___ PAGES

PG. NO.	LINE NO.	FORM TYPE	CONTROL LEVEL	INDICATORS AND	INDICATORS AND	FACTOR 1	OPERATION	FACTOR 2	RESULT FIELD	FIELD LENGTH	DECIMAL POSITIONS	HALF ADJUST	RESULTING INDICATORS COMPARE HI 1-2	RESULTING INDICATORS COMPARE LO 1-2	RESULTING INDICATORS COMPARE EQ 1-2	COMMENTS	PROGRAM IDENTIFICATION
	01	C				+10	LOKUP	TABARG	TABFUN						25		
	02	C		25			MOVE	+365	TABFUN								
	03	C		*			MOVE	+15	TABARG								
	04	C															

Figure 8–5. Table Updating Example

(1) On line 1, the literal "+10" in factor 1 is the search argument. The table lookup (LOKUP) operation searches the Argument Table (TABARG) in factor 2 for an entry equal to the search value (+10). A search for equality is specified by indicator 25 entered in Columns 58 and 59 of the resulting indicator section of the Calculation Specifications form.

If the search is successful, indicator 25 is turned on, the Argument Table entry location is noted, and the contents of the located argument are moved to the Argument Table's special holding area; also, the Function Table's corresponding entry location is noted and the contents of the located function are moved to the Function Table's special holding area. If the search is not successful, indicator 25 is reset.

(2) On line 2, if indicator 25 is set (on), the new function value (factor 2) "+365" is moved to the Function Table's entry and the special holding area; this transfer replaces the entry located by the previous LOKUP operation. If indicator 25 is reset (off), then line 2 is skipped.

(3) On line 3, if indicator 25 is set (on - indicated by asterisk in Column 11), the new argument value (factor 2) "+15" is moved to the Argument Table's entry and the special holding area; this transfer replaces the entry located by the previous LOKUP operation. If indicator 24 is reset (off), then line 3 is also skipped.

8.3.2. Adding New Entries

Another example of table updating is that of adding new entries. The entry is new only in the sense that the data is new to the table. Space in the table must have been provided when the table was designed and loaded. For example, a table may be designed and loaded with zero entries; a zero entry would specify that the particular location is empty and would provide space for inserting table entries. Once loaded, a location is no longer available for receiving data, so the LOKUP operation proceeds to the next zero entry.

Figure 8-6 illustrates the addition of an entry to a single table in which empty entries are indicated by zeros.

REPORT PROGRAM GENERATOR
CALCULATION SPECIFICATIONS

PROGRAM _____ PROGRAMMER _____ DATE _____ PAGE ____ OF ____ PAGES

PG. NO.	LINE NO.	FORM TYPE	CONTROL LEVEL	INDICATORS: AND	INDICATORS: AND	FACTOR 1	OPERATION	FACTOR 2	RESULT FIELD	FIELD LENGTH	DECIMAL POSITIONS	HALF ADJUST	RESULTING INDICATORS: COMPARE HI 1 2	RESULTING INDICATORS: COMPARE LO 1 2	RESULTING INDICATORS: COMPARE EQ 1 2	COMMENTS	PROGRAM IDENTIFICATION
	01	C		10		000	LOKUP	TABADD							20		
	02	C		*	20		MOVE	NEW	TABADD								

Figure 8–6. Adding a New Entry to a Single Table

The following explains the entry addition operation.

(1) On line 1, the LOKUP operation searches for the next empty location in the table named TABADD. Indicator 20 is turned on only if an empty location is found; if the table is full, indicator 20 will be reset (turned off).

(2) On line 2, if indicator 20 is set (on), the contents of NEW are moved to the indicated empty location and to the special holding area.

Figure 8-7 illustrates the addition of entries to two related tables (Argument and Function). In this example, it is assumed that the need to make entries into the tables is based on the fact that the new data has not been previously stored.

The problem facts are:

- NEWLAB is the name of the new argument entry.
- NEWCON is the name of the new function entry.
- TABNAM is the name of the Argument Table.
- TABADD is the name of the Function Table.
- Indicator 28 specifies that NEWLAB and NEWCON are available for storage.
- Indicator 30 specifies that the new entry (NEWLAB) is a duplicate of one already in storage.
- Indicator 32 specifies that there is space available for the new entries.
- Indicator N32 specifies that the table is full and has no room for the new entries.

REPORT PROGRAM GENERATOR
CALCULATION SPECIFICATIONS

PROGRAM _____ PROGRAMMER _____ DATE _____ PAGE ___ OF ___ PAGES

PG. NO.	LINE NO.	FORM TYPE	CONTROL LEVEL	INDICATORS AND	INDICATORS AND	FACTOR 1	OPERATION	FACTOR 2	RESULT FIELD	FIELD LENGTH	DECIMAL POSITIONS	HALF ADJUST	RESULTING INDICATORS +	RESULTING INDICATORS −	RESULTING INDICATORS 0	COMPARE HI 1>2	COMPARE LO 1<2	COMPARE EQ 1=2	COMMENTS	PROGRAM IDENTIFICATION
	01	C		28		NEWLAB	LOKUP	TABNAM							30					
	02	C		30			GOTO	ERROR												
	03	C		28		000	LOKUP	TABNAM							32					
	04	C		N32			GO TO	FULL												
	05	C		32	28		MOVE	NEW LAB	TABNAM											
	06	C		*			MOVE	NEWCON	TABADD											
	07	C		*			GOTO	CONTIN												
	08	C		30		ERROR	TAG													
	09	C		*															ERROR	
	10	C		*															SUBROUTINE	
	11	C		*																
	12	C		*																
	13	C		*			GOTO	CONTIN												
	14	C		N32		FULL	TAG													
	15	C		*																
	16	C		*															FULL TABLE	
	17	C		*															SUBROUTINE	
	18	C		*																
	19	C		*																
	20	C		*																
	21	C				CONTIN	TAG													
	22	C																		
	23	C																	CONTINUE	
	24	C																	PROGRAM	
	25	C																		

Figure 8–7. Updating Two Related Tables

The following explains Figure 8-7 in detail:

Line 1 - Checks for a duplicate entry NEWLAB; if found, indicator 30 is set; if not found, indicator 30 is reset (off).

Line 2 - If indicator 30 is set, a duplicate entry has been found, and the program jumps to the ERROR subroutine. If indicator 30 is reset, line 2 is skipped.

Line 3 - A search is made for the next empty entry (000); if not found, indicator 32 is turned off (reset). If found, indicator 32 is turned on.

Line 4 - If indicator 32 is reset, the table is full, and the program jumps to the FULL subroutine on line 14. If indicator 32 is set, line 4 is skipped.

Lines 5 and 6 - If indicator 32 is set, space is available in the tables. Move the contents of NEWLAB to the empty space in the Argument Table TABNAM and the contents of NEWCON to the empty space in the Function Table TABADD.

Line 7 - Jump to the subroutine CONTIN on line 21 to continue program.

Lines 8 through 12 - The ERROR subroutine; these lines are executed only if indicator 30 is on.

Line 13 - Jump to subroutine CONTIN on line 21 to continue program.

Lines 14 through 20 - The FULL subroutine; these lines are executed only if indicator 32 is reset (off).

Line 21 - Continuation of program.

APPENDIX A. SAMPLE PROGRAM

A1. INTRODUCTION

The following three problem examples demonstrate the application of RPG techniques and procedures. The typical operations chosen to illustrate the use of the RPG are as follows:

(1) Sales Analysis Listing

(2) Daily Gross Pay Calculation

(3) Stock Status Report (Disc)

A2. EXAMPLE NUMBER 1 – SALES ANALYSIS LISTING

A Sales Analysis Listing is to be prepared using billing line item detail cards. The cards are in sequence by territory, salesman, and product number. Figure A-1 illustrates the required data flow, and Figure A-2 shows the order card format.

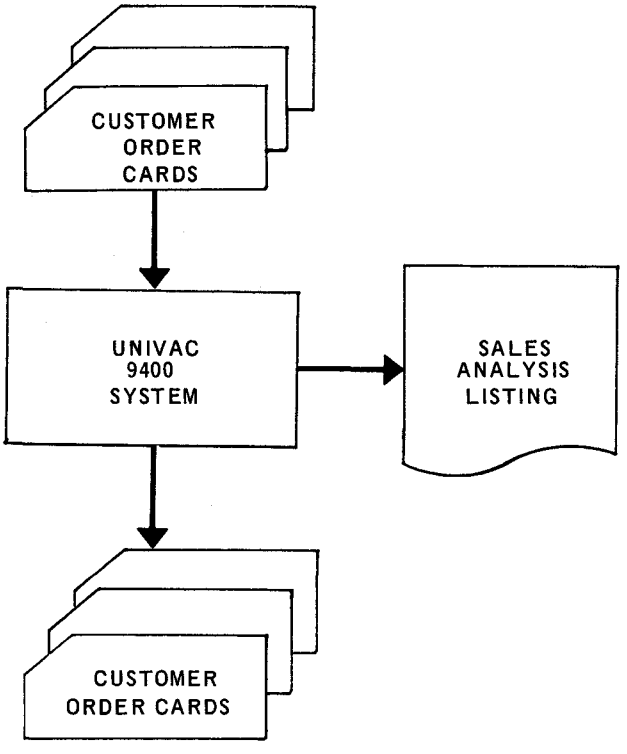


Figure A–1. Sales Analysis Data Flow

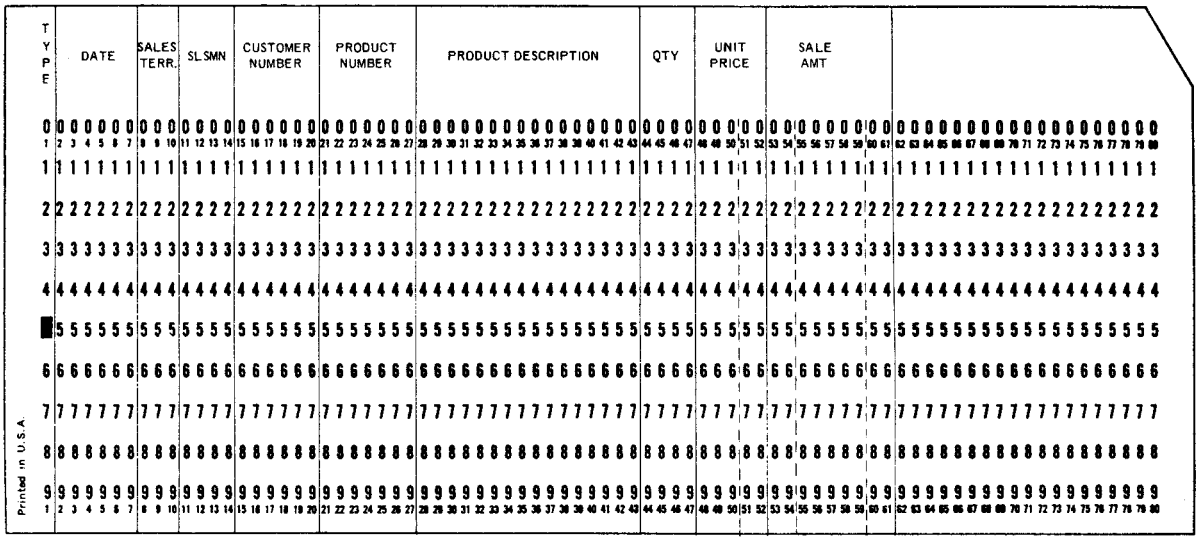


Figure A–2. Billing Line Item Detail Card Format

A2.1. Procedure

(1) Print field headings at the top of every page. One line is to be printed for each customer order card.

(2) Quantity and sale amount totals are to be printed when there is a control break in product (minor), salesman (intermediate), or territory (major).

(3) Overall totals are also accumulated and printed.

(4) All fields are to be zero suppressed and edited as indicated on the printer format chart (Figure A-3). Figures A-4 through A-7 illustrate how the RPG forms would be completed for this problem.

A2.2. File Description Specifications – Figure A-4

Line 1 specifies the input file read into the system; line 2 indicates the printed report.

A2.3. Input Specifications – Figure A-5

Lines 1-7 of the input specifications identify and describe billing line item detail cards. Only those fields needed in actual processing and output are specified.

(1) Line 1 defines the input file and relates it, by file name, to the File Description Specifications.

(2) Lines 2-7 describe the fields within the input file.

RPG PRINTER FORMAT CHART

FORM NUMBER _____ APPLICATION _____

FORM PARTS _____ RUN NAME _____ RUN NUMBER _____

TYPE OF PRINTOUT _____ RECORD NAME _____ RECORD NUMBER _____

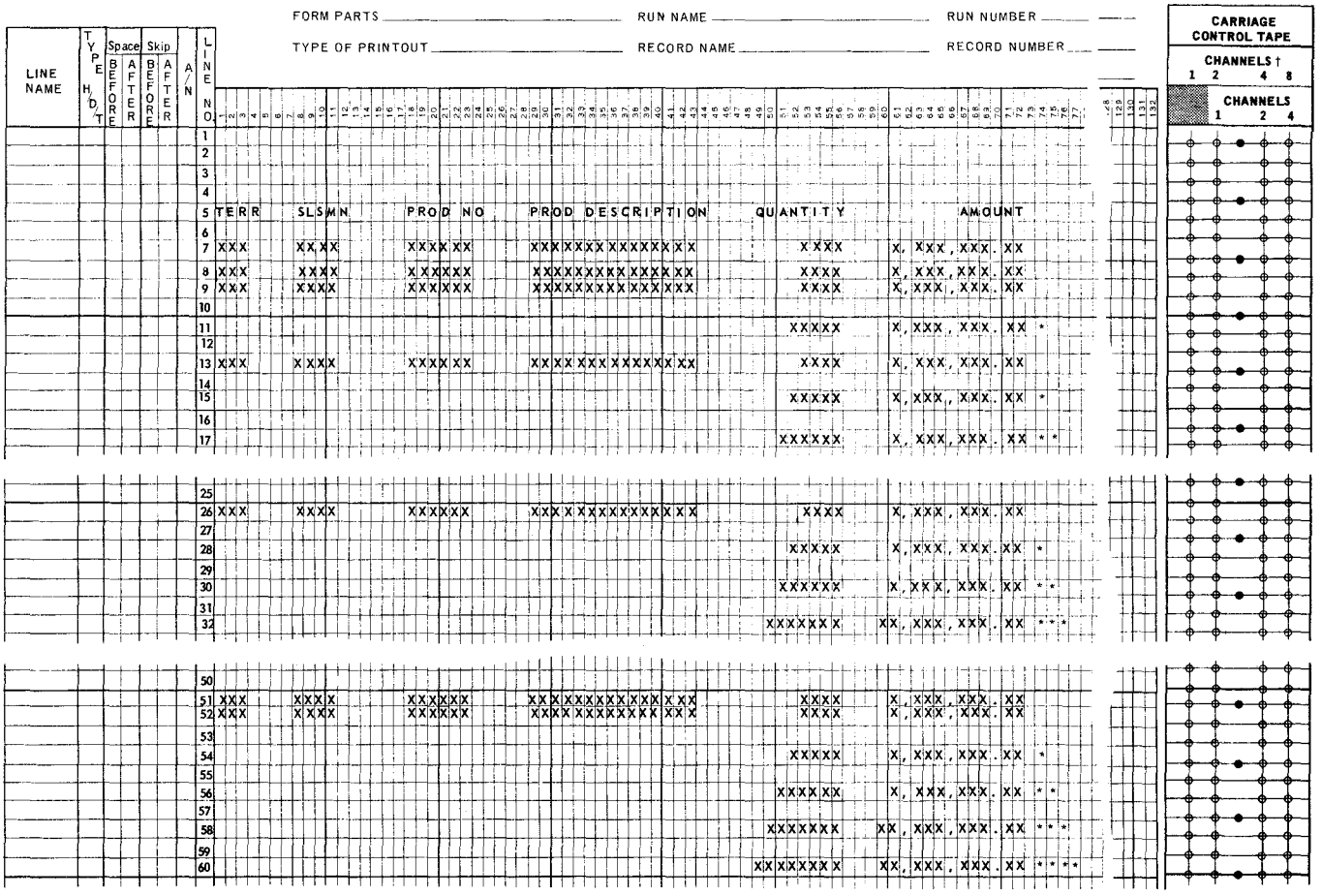


Figure A–3. Weekly Sales Analysis Listing

REPORT PROGRAM GENERATOR

FILE DESCRIPTION

PROGRAM _____ PROGRAMMER _____ DATE _____ PAGE ___ OF ___ PAGES

PG. NO.	LINE NO.	FORM TYPE	FILE NAME	FILE DESIGNATION	FILE TYPE	END OF FILE SEQUENCE	FORMAT	BLOCK LENGTH	RECORD LENGTH	RA TYPE	MODE	RA FIELD LENGTH	FILE ORGANIZATION	OVERFLOW INDICATOR	KEY FIELD START LOCATION	EXTENSION	DEVICE	LABELS	LOGICAL UNIT NUMBER	FILE ADDITION	LOGICAL UNIT NUMBER††	NAME OF LABEL EXIT†	CYLINDER OVERLOW	REWIND	PROGRAM IDENTIFICATION
1	3 5	6	7 14	15	16	17	18	19 20 23	24 27	28	29 30	31	32	33	34 35 38	39	40 46	47 52	53	54 59	60 65	66	67	68 69	70 71 74	75 80
	0 1	F	INPUT	I	P		F	80									READER									
	0 2	F	PRINT	O			F	132						OF			PRINTER									

Figure A–4. Sales Analysis File Description

REPORT PROGRAM GENERATOR
INPUT FORMAT SPECIFICATIONS

PROGRAM _____ PROGRAMMER _____ DATE _____ PAGE ___ OF ___ PAGES

PG. NO.	LINE NO.	FORM TYPE	FILE NAME	SEQ.	OPTION NUMBER	RECORD INDICATOR	RECORD IDENTIFICATION CODES 1: POSITION	1: NOT	1: C/Z/D	1: CHAR	2: POSITION	2: NOT	2: C/Z/D	2: CHAR	3: POSITION	3: NOT	3: C/Z/D	3: CHAR	STACKER SELECT	DATA FORMAT	DECIMAL POSITIONS	FIELD LOCATION: START	FIELD LOCATION: END	MATCHING FIELDS CONTROL LEVEL: FIELD NAME		FIELD RECORD RELATION	FIELD INDICATORS: +	FIELD INDICATORS: -	FIELD INDICATORS: ø OR b	STERLING SIGN POSITION	PROGRAM IDENTIFICATION
	01	I	INPUT	AA		01	1		C	5																						
	02	I																				8	10	0 TERR	L3						
	03	I																				11	14	0 SLSMN	L2						
	04	I																				21	27	0 PROD	L1						
	05	I																				28	43	DESC							
	06	I																				44	47	0 QTY							
	07	I																				53	61	2 AMT							

Figure A–5. Sales Analysis Input Specifications

A2.4. Calculation Specifications – Figure A-6

(1) Lines 1 and 2 specify addition calculations for the detail billing line items, detail cards in which product quantity (PQTY), and product amount totals (PAMT) are accumulated.

(2) Lines 3 and 4 are executed only on a level 1 (minor) break, that is, when there is a change in product number. The two entries cause the salesman quantity (SQTY) and salesman amount (SAMT) totals to be accumulated.

(3) Lines 5 and 6 specify that on a level 2 (intermediate) break (change in salesman number), territory quantity (TQTY) and territory amount totals (TAMT) are accumulated.

(4) Lines 7 and 8 specify that on a level 3 (major) break (change in territory number), final quantity (FQTY) and final amount totals (FAMT) are accumulated.

NOTE: When a level break occurs, all calculations for lower level entries are performed prior to the calculation for the higher level.

REPORT PROGRAM GENERATOR
CALCULATION SPECIFICATIONS

PROGRAM _____ PROGRAMMER _____ DATE _____ PAGE ___ OF ___ PAGES

PG. NO.	LINE NO.	FORM TYPE	CONTROL LEVEL	INDICATORS: AND	INDICATORS: AND	FACTOR 1	OPERATION	FACTOR 2	RESULT FIELD	DECIMAL POSITIONS: FIELD LENGTH	HALF ADJUST	RESULTING INDICATORS: COMPARE HI 1>2	RESULTING INDICATORS: COMPARE LO 1<2	RESULTING INDICATORS: COMPARE EQ 1=2	COMMENTS	PROGRAM IDENTIFICATION
	01	C		01		QTY	ADD	PQTY	PQTY	5 0						
	02	C		*		AMT	ADD	PAMT	PAMT	9 2						
	03	C	L1			PQTY	ADD	SQTY	SQTY	6 0						
	04	C	L1			PAMT	ADD	SAMT	SAMT	9 2						
	05	C	L2			SQTY	ADD	TQTY	TQTY	7 0						
	06	C	L2			SAMT	ADD	TAMT	TAMT	1 0 2						
	07	C	L3			TQTY	ADD	FQTY	FQTY	8 0						
	08	C	L3			TAMT	ADD	FAMT	FAMT	1 0 2						

Figure A–6. Sales Analysis Calculation Specifications

A2.5. Output Specifications – Figure A-7

(1) Lines 1-8 provide for printing headings at the top of each page.

(2) Detail line editing and printing is specified on lines 9-15.

(3) Lines 16-18 cause the printing of the level 1 (L1) totals (product number totals), including the constant asterisk (*).

(4) Lines 19-21 provide for editing and printing of level 2 (L2) totals (salesman totals).

(5) Lines 22-24 provide for editing and printing of level 3 (L3) totals (territory totals).

(6) The final total line (LR) is specified on lines 25-27.

(7) All fields, except the description, are zero suppressed. The amount field for the detail and total lines has comma and decimal point editing. On total lines, the total level is identified by one through four asterisks printed to the right of the amount total. All total fields are cleared to zeros after the specific total level has been printed (B in Column 39).

REPORT PROGRAM GENERATOR
OUTPUT FORMAT SPECIFICATIONS

PROGRAM _____ PROGRAMMER _____ DATE _____ PAGE ___ OF ___ PAGES

PG. NO.	LINE NO.	FORM TYPE	FILE NAME	TYPE H/D/T	STACKER SELECT	SPACE AFTER	SPACE BEFORE	SKIP BEFORE	SKIP AFTER	OUTPUT INDICATORS AND	OUTPUT INDICATORS AND	FIELD NAME	ZERO SUPPRESS	BLANK AFTER	END POSITION IN OUTPUT RECORD	CONSTANT OR EDIT WORD	STERLING SIGN POSITION	PROGRAM IDENTIFICATION
1	3 5	6	7 14	15	16	17 18		19 20	21 22	23 25	26 28	29 31	32 37	38	39	40 43	44 45 70	71 74	75 80
	01	O	PRINT	H		2		1	4	0F									
	02	O		OR						1P									
	03	O														4	'TERR'		
	04	O														12	'SLSMN'		
	05	O														24	'PROD NO'		
	06	O														44	'PROD DESCRIPTION'		
	07	O														56	'QUANTITY'		
	08	O														72	'AMOUNT'		
	09	O	PRINT	D		0	1			01									
	10	O											TERR	Z		3			
	11	O											SLSMN	Z		11			
	12	O											PROD	Z		23			
	13	O											DESC			43			
	14	O											QTY	Z		56			
	15	O											AMT			72	' 0. '		
	16	O	PRINT	T		1	2			L1									
	17	O											PQTY	Z	B	56			
	18	O											PAMT		B	74	' 0. &* '		
	19	O	PRINT	T		0	2			L2									
	20	O											SQTY	Z	B	56			
	21	O											SAMT		B	74	' 0. &** '		
	22	O	PRINT	T		0	2			L3									
	23	O											TQTY	Z	B	56			
	24	O											TAMT		B	76	' 0. &*** '		
	25	O	PRINT	T		0	2			LR									
		O											FQTY	Z	B	56			
		O											FAMT		B	77	' 0. &**** '		

Figure A–7. Sales Analysis Listing Output Specifications

A3. EXAMPLE NUMBER 2 – DAILY GROSS PAY CALCULATION

A daily job payroll report and employee payroll summary card are to be prepared from daily job cards. The cards are in sequence by employee number. For each job card the regular, shift, overtime, and total earnings are calculated. These extensions are printed with the detail information from the job card. When a control break in employee number occurs, total hours and earnings are printed, and an extended payroll summary card is punched. Final totals will be printed at the end of the report. Figure A-8 shows the required data flow, and Figure A-9 shows the input and output card formats. The completed forms are shown in Figures A-10 through A-14.

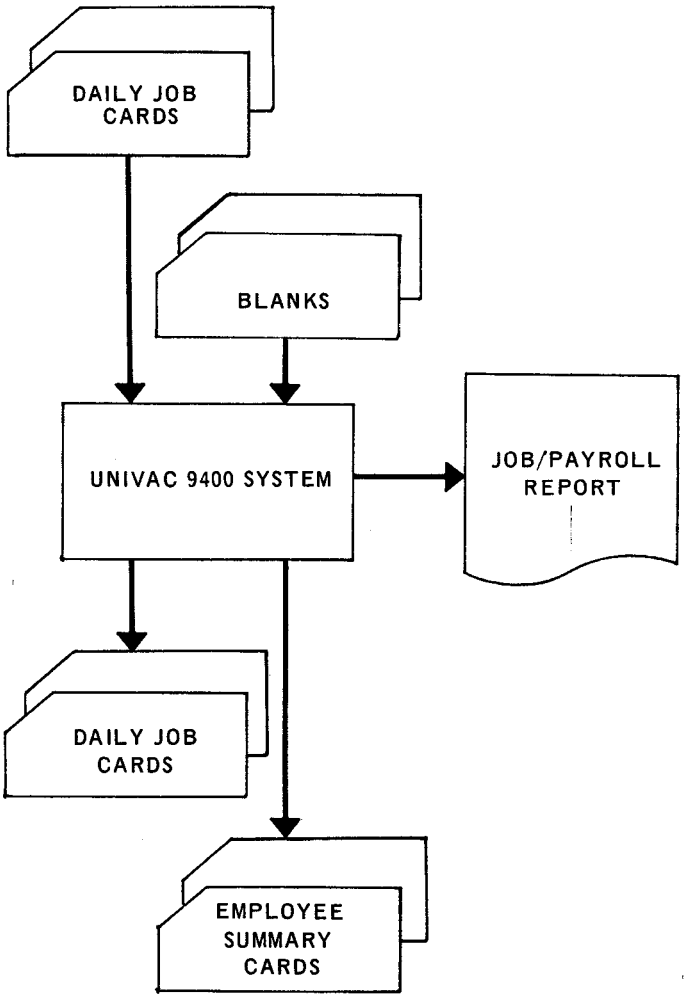


Figure A–8. Daily Gross Pay Data Flow

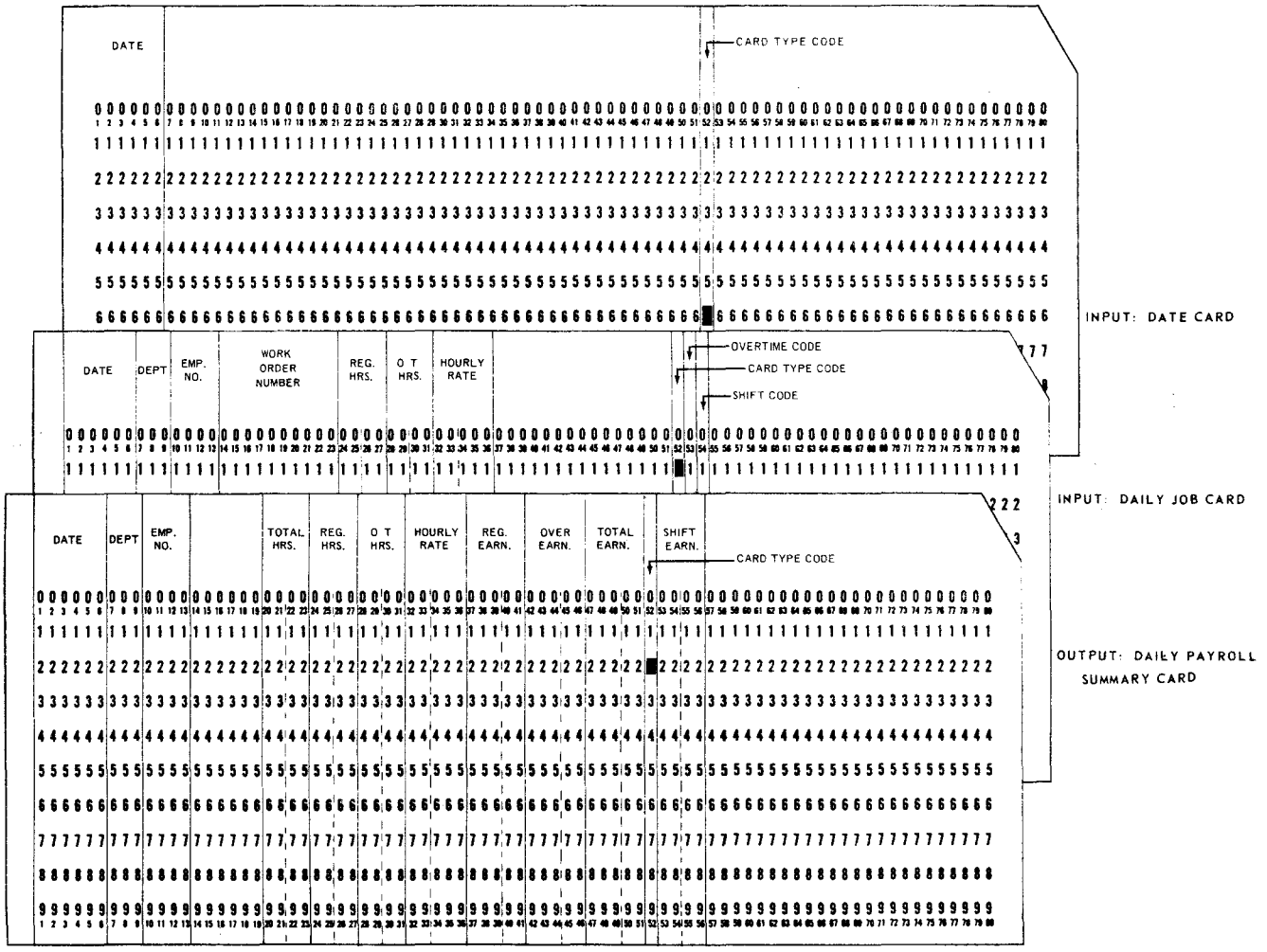


Figure A–9. Input and Output Card Formats

A3.1. Procedure

(1) Sequence check employee number.

(2) Calculate regular earnings and half adjust (round to dollars and cents). Regular earnings = regular hours x rate.

(3) Calculate overtime earnings and half adjust (round to dollars and cents). Check overtime code in Column 53 for 1 or 2.

If Column 53 is 1, then
overtime earnings = 1.5 x overtime hours x rate.

If Column 53 is 2, then
overtime earnings = 2.0 x overtime hours x rate.

(4) Calculate gross pay.

Gross pay = overtime earnings + regular earnings + shift bonus.

(5) Calculate shift bonus. Check shift code in Column 54 for a 1.

If Column 54 is 1, then
shift bonus = gross pay x 5%.

If Column 54 is blank, this calculation is omitted.

RPG PRINTER FORMAT CHART

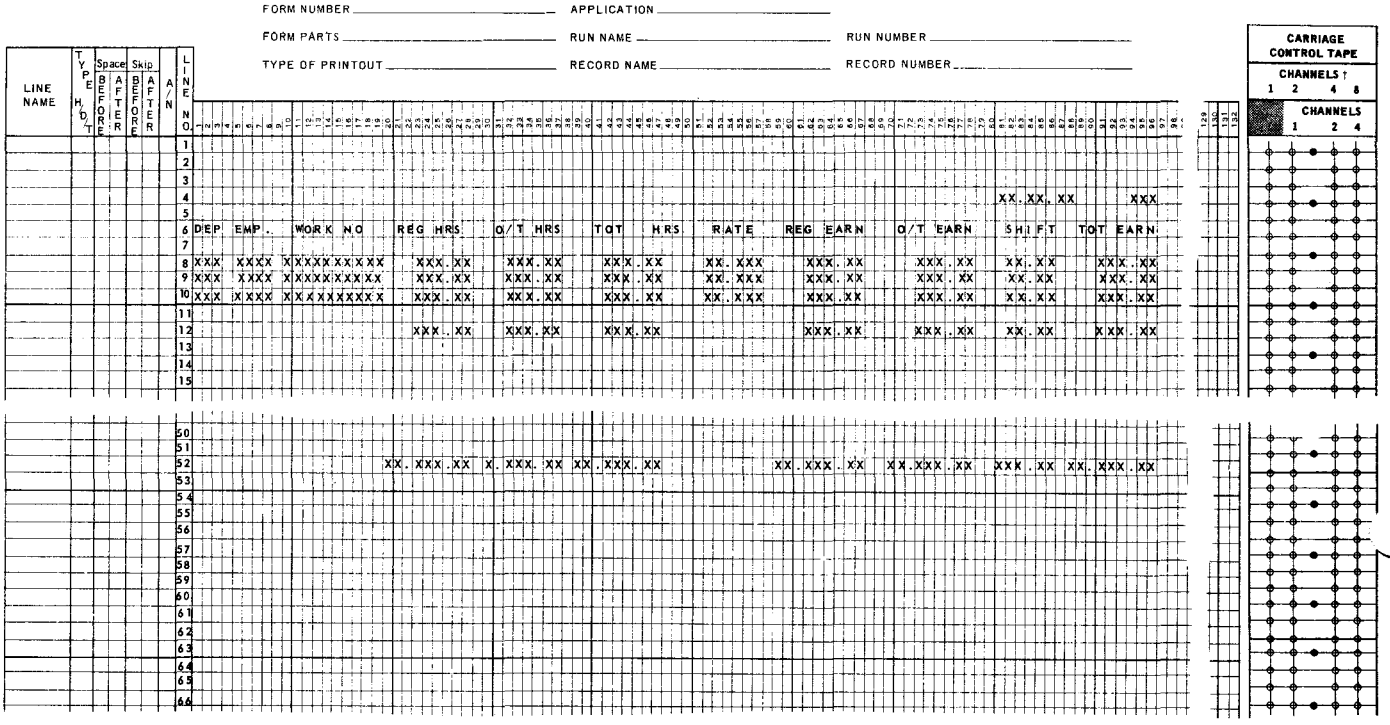


Figure A–10. Daily Gross Pay Report Format

(6) Calculate total hours.

Total hours = regular hours + overtime hours.

(7) Accumulate the following employee totals for summary card output and payroll listing:

Regular hours
Overtime hours
Total hours
Regular earnings
Overtime earnings
Shift bonus
Total earnings

(8) Accumulate final totals for all fields specified in step 7.

(9) The first card of the input file is a date card. This date is printed at the top of every page on the same line as the page number. The second line of printing on every page is the column heading.

A3.2. File Description Specifications – Figure A-11

Only three lines are needed on the File Description form. Line 1 specifies the input file: the job cards in the reader. Lines 2 and 3 specify the two outputs: the printer and punch.

REPORT PROGRAM GENERATOR
FILE DESCRIPTION

PROGRAM _____ PROGRAMMER _____ DATE _____ PAGE ___ OF ___ PAGES

PG. NO.	LINE NO.	FORM TYPE	FILE NAME	FILE DESIGNATION / FILE TYPE	END OF FILE SEQUENCE	FORMAT	BLOCK LENGTH	RECORD LENGTH	MODE	RA TYPE	RA FIELD LENGTH	FILE ORGANIZATION / EXTENSION	OVERFLOW INDICATOR	KEY FIELD START LOCATION	E	DEVICE	LABELS / LOGICAL UNIT NUMBER	FILE ADDITION / LOGICAL UNIT NUMBER†† / NAME OF LABEL EXIT†	CYLINDER OVERLOW	REWIND	PROGRAM IDENTIFICATION
	01	F	JOBCRDS	I	P	A	F	80	80							READER					
	02	F	PRNTOUT	O			F	132	132				OV			PRINTER					
	03	F	SUMMARY	O			F	80	80							PUNCH					

Figure A–11. Daily Gross Pay File Description Specifications

A3.3. Input Format Specifications – Figure A-12

The input form identifies the job card. Record indicator 01 is turned on if Column 52 contains a 1 punch. All the fields requiring processing are indicated under field name. Employee number (EMP) is the level 1 control (L1) and is to be sequence checked (M1). When the card is read, the overtime hours field (OHD) is checked for zeros. If the field is all zeros, indicator 10 is turned on. This eliminates the necessity of performing the detail calculations for overtime. If the shift code (Column 54) is blank, indicator 15 is turned on. If indicator 15 is not on, shift bonus is calculated and added to gross pay.

A date card, which is identified by a 6 punch in Column 52, turns on record indicator 06. The date is punched in Columns 1-6 in the form MMDDYY.

REPORT PROGRAM GENERATOR
INPUT FORMAT SPECIFICATIONS

PROGRAM _____ PROGRAMMER _____ DATE _____ PAGE ___ OF ___ PAGES

PG. NO.	LINE NO.	FORM TYPE	FILE NAME	OPTION NUMBER SEQ.	RECORD INDICATOR	RECORD IDENTIFICATION CODES 1 POSITION	1 NOT	1 C/Z/D	1 CHAR	2 POSITION	2 NOT	2 C/Z/D	2 CHAR	3 POSITION	3 NOT	3 C/Z/D	3 CHAR	STACKER SELECT	DATA FORMAT	FIELD LOCATION START	FIELD LOCATION END	DECIMAL POSITIONS	MATCHING FIELDS CONTROL LEVEL	FIELD NAME	FIELD RECORD RELATION	FIELD INDICATORS +	FIELD INDICATORS -	FIELD INDICATORS 8 OR blank	STERLING SIGN POSITION	PROGRAM IDENTIFICATION
1	3	5	7	14 15	17 18 19	21	24 25	26	27	28	31 32	33	34	35	38 39	40	41	42	43	44	47 48	51	52	53	58 59	61	63	65	67	69	71	74 75	80
	0 1	1	JOBCRDS	AA	0 1	5 2		C	1																							
	0 2	1																		7	9	0		DEPT							
	0 3	1																		1 0	1 3	0		EMP	L 1	M 1					
	0 4	1																		1 4	2 3	0		WONO							
	0 5	1																		2 4	2 7	2		RHD							
	0 6	1																		2 8	3 1	2		OHD					1 0		
	0 7	1																		3 2	3 6	3		RATE							
	0 8	1																		5 3	5 3	0		CODE							
	0 9	1																		5 4	5 4			SHIFT					1 5		
	1 0	1		BB	0 6	5 2		C	6																						
	1 1	1																		1	6	0		DATE							

Figure A–12. Daily Gross Pay Input Format Specifications

A3.4. Calculation Specifications – Figure A-13

Every calculation line is checked to determine whether indicator 01 is on so that calculations are only performed on detail cards and not on the date card.

Line 1 calculates regular earnings. Lines 2 and 3 check for overtime codes. If indicator 10 is not on (N10), the overtime code is compared to 1. If equal, the indicator 11 is turned on. If indicator H1 is turned on, the processor is halted at the end of the current detail card cycle. This indicates that the overtime code is missing from Column 53.

If indicator 10 is not on (N10), line 3 compares the overtime code to 2. If they are equal, indicator 12 is turned on. If the code is greater than 2 (an error), H1 is turned on and the program is brought to a halt.

Lines 4-6 calculate overtime earnings. Lines 7-17 are the detail accumulations which occur for every job card. Totals by employee are accumulated. On lines 8 and 9 shift bonus is calculated and added to gross pay if indicator 15 is not on (N15). On line 16 shift earnings are added to shift total only if indicator 15 is not on (N15). Note that on lines 11-17 (also lines 18-25), the factor 2 field initially has a value of zero. Its contents are replaced by the accumulated sum as totals are developed.

Lines 18-24 are executed only when a control break occurs on employee number (when indicator L1 is on). These lines accumulate final totals.

REPORT PROGRAM GENERATOR
CALCULATION SPECIFICATIONS

PROGRAM _____ PROGRAMMER _____ DATE _____ PAGE ___ OF ___ PAGES

PG. NO.	LINE NO.	FORM TYPE	CONTROL LEVEL	INDICATORS AND	INDICATORS AND	FACTOR 1	OPERATION	FACTOR 2	RESULT FIELD	FIELD LENGTH	DECIMAL POSITIONS	HALF ADJUST	RESULTING INDICATORS +	RESULTING INDICATORS −	RESULTING INDICATORS 0	COMPARE HI 1>2	COMPARE LO 1<2	COMPARE EQ 1=2	COMMENTS	PROGRAM IDENTIFICATION
1	3 5	6	7	9 11	12 14	15 17	18 27	28 32	33 42	43 48	49 51	52	53	54	55	56	57	58	60 74	75 80
	01	C		01			RHD	MULT	RATE	RED	5	2	H							
	02	C		01	N10		CODE	COMP	1								H1	11		
	03	C		01	N10		CODE	COMP	2				H	1				12		
	04	C		01	N10		OHD	MULT	RATE	OED	5	2	H							
	05	C		01	11		OED	MULT	1..5	OED			H							
	06	C		01	12		OED	ADD	OED	OED									NOTE:	
	07	C		01			RED	ADD	OED	TED	5	2							USE FOLLOWING LEGEND AS GUIDE TO DECODE FIELD NAMES	
	08	C		01	N15		TED	MULT	.05	SHED	4	2	H							
	09	C		*			SHED	ADD	TED	TED									D = DETAIL	
	10	C		01			RHD	ADD	OHD	THD	5	2							E = EARNINGS	
	11	C		*			RHD	ADD	RHT	RHT	4	2							FT = FINAL TOTAL	
	12	C		01	N10		OHD	ADD	OHT	OHT	4	2							H = HOURS	
	13	C		01			THD	ADD	THT	THT	4	2							O = OVERTIME	
	14	C		*			RED	ADD	RET	RET	5	2							R = REGULAR	
	15	C		01	N10		OED	ADD	OET	OET	5	2							SH = SHIFT	
	16	C		01	N15		SHED	ADD	SHET	SHET	4	2							T = TOTAL	
	17	C		01			TED	ADD	TET	TET	5	2								
	18	C	L1				RHT	ADD	RHFT	RHFT	7	2								
	19	C		*			OHT	ADD	OHFT	OHFT	6	2								
	20	C		*			THT	ADD	THFT	THFT	7	2								
	21	C		*			RET	ADD	REFT	REFT	7	2								
	22	C		*			OET	ADD	OEFT	OEFT	7	2								
	23	C		*			SHET	ADD	SHEFT	SHEFT	5	2								
	24	C		*			TET	ADD	TEFT	TEFT	7	2								

Figure A–13. Daily Gross Pay Calculation Specifications

A3.5. Output Format Specifications – Figure A-14

The output form specifies two output files named PRINTOUT and SUMMARY (punching). Printing occurs at header, detail, and total time. Punching occurs only at total time.

For each different line of print it is necessary to specify the file name, the type of line (header, detail, or total), and the indicator and the fields to be printed on that line.

Lines 1-4 identify the first header line. The fields DATE and PAGE are printed as headers when overflow occurs or when indicator 06 is on. Page numbering starts with page 1. Indicator 06 is on for the first card only (date card). The form advances to Channel 7 (top of next page) before printing the first header line and spaces two lines after printing. Lines 5-10 designate the second header line (column headers). This line is generated by the RPG.

Lines 11-22 specify the detail line. The hour and earning fields are edited according to the edit words in Columns 45-70. The apostrophes indicate the beginning and the end of the edit word. The zero in each edit word specifies the end of zero suppression.

REPORT PROGRAM GENERATOR
OUTPUT FORMAT SPECIFICATIONS

PROGRAM _____ PROGRAMMER _____ DATE _____ PAGE ___ OF ___ PAGES

PG. NO.	LINE NO.	FORM TYPE	FILE NAME	TYPE H/D/T	STACKER SELECT	SPACE AFTER	SPACE BEFORE	SKIP BEFORE	SKIP AFTER	OUTPUT INDICATORS AND	OUTPUT INDICATORS AND	FIELD NAME	ZERO SUPPRESS	BLANK AFTER	END POSITION IN OUTPUT RECORD	DATA FORMAT CONSTANT OR EDIT WORD	STERLING SIGN POSITION	PROGRAM IDENTIFICATION
	01	O	PRNTOUT	H		2	1	4		OV								
	02	O	OR							06								
	03	O										DATE			88	' 0 / / '		
	04	O										PAGE	Z		96			
	05	O	PRNTOUT	H		2				*								
	06	O													17	'DEP EMP # WORK NO'		
	07	O													37	'REG HRS O/T HRS'		
	08	O													56	'TOT HRS RATE'		
	09	O													78	'REG EARN O/T EARN'		
	10	O													96	'SHIFT TOT EARN'		
	11	O	PRNTOUT	D		1				01								
	12	O										DEPT	Z		3			
	13	O										EMP	Z		8			
	14	O										WONO	Z		19			
	15	O										RHD			28	' 0. '		
	16	O										OHD			37	' 0. '		
	17	O										THD			47	' 0. '		
	18	O										RATE			57	' 0. '		
	19	O										RED			67	' 0. '		
	20	O										OED		B	78	' 0. '		
	21	O										SHED		B	86	' 0. '		
	22	O										TED			96	' 0. '		
	23	O	SUMMARY	T						L1								
	24	O										DATE			6			
	25	O										DEPT			9			

Figure A–14. Daily Gross Pay Output Specifications (Part 1 of 2)

Decimal points indicate where decimals are to be inserted. The B in Column 39 indicates that the fields OED and SHED are to be cleared to zeros after printing. Lines 23-25 (Figure A-14, Part 1) and lines 1-10 (Figure A-14, Part 2) specify punching at total time when the L1 level indicator is on.

Line 9 of part 2 causes a 2, generated by the RPG, to be punched in Column 52 of the summary card.

Lines 11-18 of part 2 specify the total line which is printed when the L1 indicator is on. All fields are set to zeros after printing (B in Column 39) so that new totals may be accumulated for the next input card. All specified fields are edited according to the edit words in Columns 45-70. All fields are zero suppressed with decimals inserted. The paper advances two spaces after printing. Lines 19-26 of part 2 specify the final totals which are printed when the LR (last record) indicator is on. All fields are edited according to the specified edit words. The processor is brought to a halt after the final totals are printed.

REPORT PROGRAM GENERATOR
OUTPUT FORMAT SPECIFICATIONS

PROGRAM _____ PROGRAMMER _____ DATE _____ PAGE ___ OF ___ PAGES

PG. NO.	LINE NO.	FORM TYPE	FILE NAME	TYPE H/D/T	STACKER SELECT	SPACE	SKIP	OUTPUT INDICATORS	FIELD NAME	ZERO SUPPRESS	BLANK AFTER	END POSITION IN OUTPUT RECORD	DATA FORMAT	CONSTANT OR EDIT WORD	STERLING SIGN POSITION	PROGRAM IDENTIFICATION
	01	O							EMP			13				
	02	O							THT			23				
	03	O							RHT			27				
	04	O							OHT			31				
	05	O							RATE			36				
	06	O							RET			41				
	07	O							OET			46				
	08	O							TET			51				
	09	O										52		'2'		
	10	O							SHET			56				
	11	O	PRNTOUT	T	1	2		*								
	12	O							RHT		B	28		'0.'		
	13	O							OHT		B	37		'0.'		
	14	O							THT		B	47		'0.'		
	15	O							RET		B	67		'0.'		
	16	O							OET		B	78		'0.'		
	17	O							SHET		B	86		'0.'		
	18	O							TET		B	96		'0.'		
	19	O	PRNTOUT	T	0	0		LR								
	20	O							RHFT			28		'0.'		
	21	O							OHFT			37		'0.'		
	22	O							THFT			47		'0.'		
	23	O							REFT			67		'0.'		
	24	O							OEFT			78		'0.'		
	25	O							SHEFT			86		'0.'		
		O							TEFT			96		'0.'		

Figure A–14. Daily Gross Pay Output Specifications (Part 2 of 2)

A4. EXAMPLE NUMBER 3 – INVENTORY APPLICATION USING DISC STOCK STATUS BALANCE RECORD

A daily stock status listing and summary record updating are to be prepared using the following records or cards in sequence by part number:

(1) The stock status balance record on disc which contains bin quantity, on-order quantity, available quantity, usage last year, usage this year, and so on.

(2) On-order cards which increase the on-order quantity and the available quantity.

(3) Receipt cards, which increase the bin quantity and decrease the on-order quantity.

(4) Issue cards which decrease bin quantity and available quantity, and increase the usage this year quantity.

A detail line is printed for each input record. An updated stock status summary record is calculated and, on a change in part number, is printed and written out on disc. If the total available quantity is less than or equal to the reorder point, an order card must also be calculated and punched. Figure A-15 shows the pattern of data flow, and Figures A-16a, 16b, and 16c show the input record formats. Figures A-17a, 17b, and 17c show the output record formats. Figures A-18 through A-22 illustrate the completed forms and report.

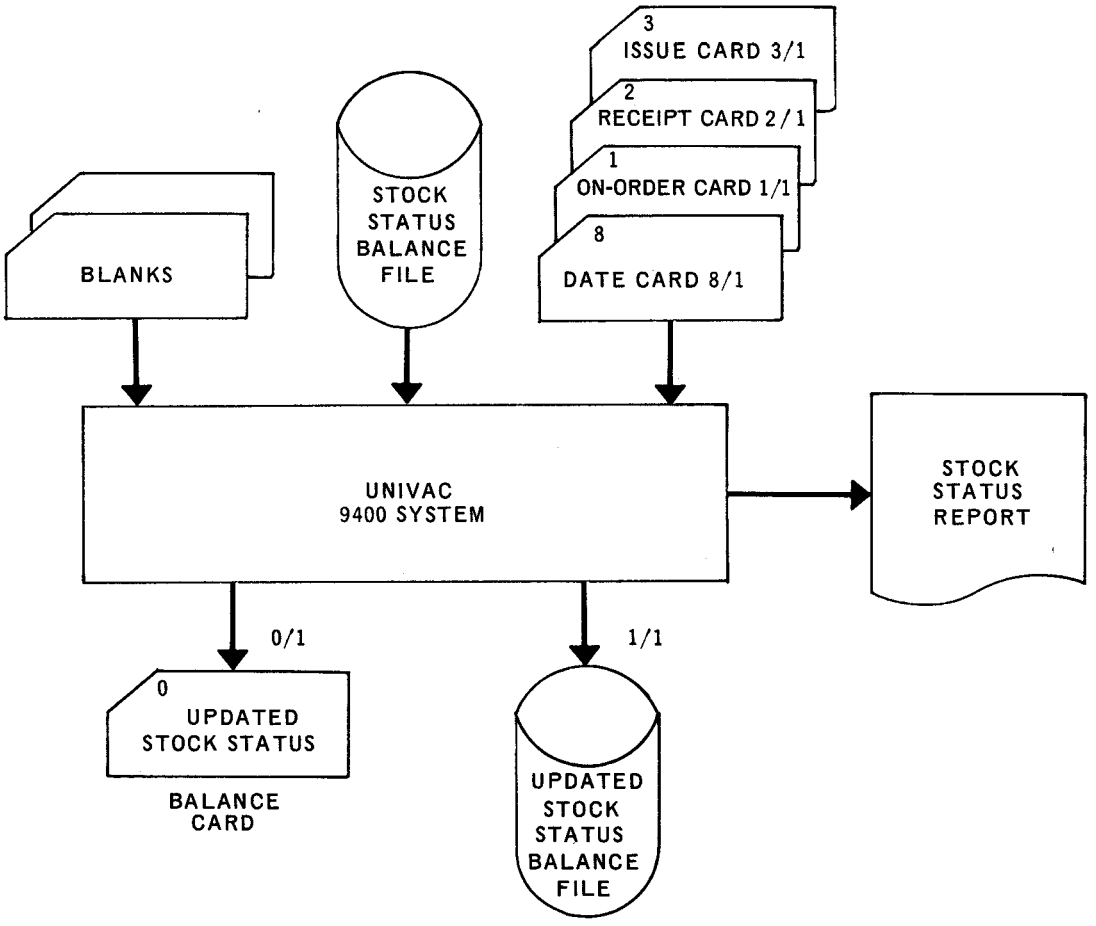


Figure A–15. Stock Status Report Data Flow

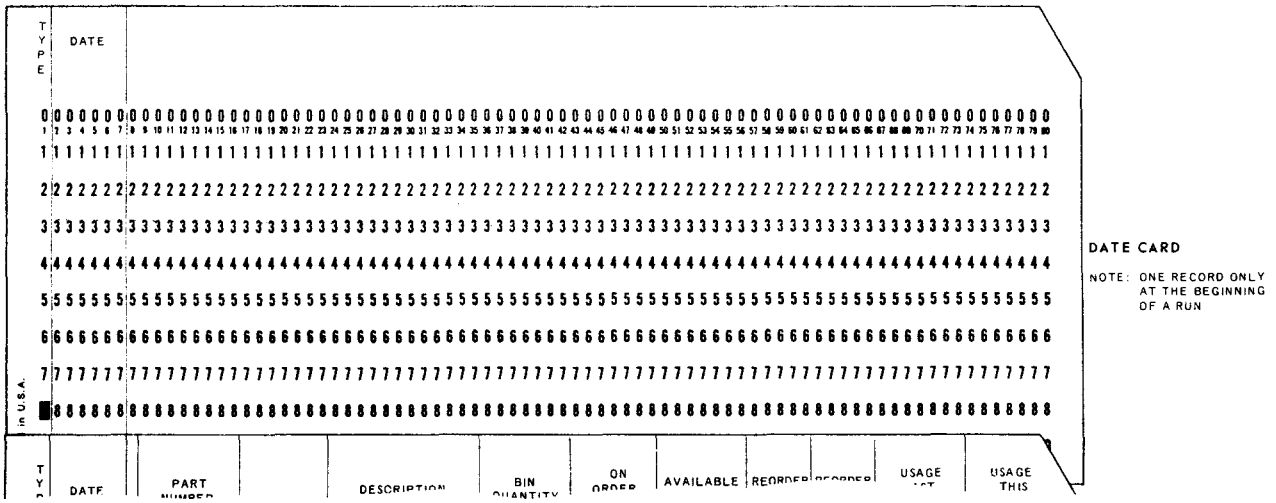


Figure A–16a. Date Card Input

	DIVISION	LAST ACTIVITY DATE	PART NUMBER	ASSEMBLY NUMBER	DESCRIPTION	BIN QUANTITY	ON–ORDER QUANTITY	AVAILABLE QUANTITY	REORDER POINT	REORDER QTY.	USAGE LAST YEAR	USAGE THIS YEAR	
	12	3 8	9 16	17 23	24 35	36 42	43 49	50 56	57 61	62 66	67 73	74 80	

Figure A–16b. Stock Status Disc Record Input

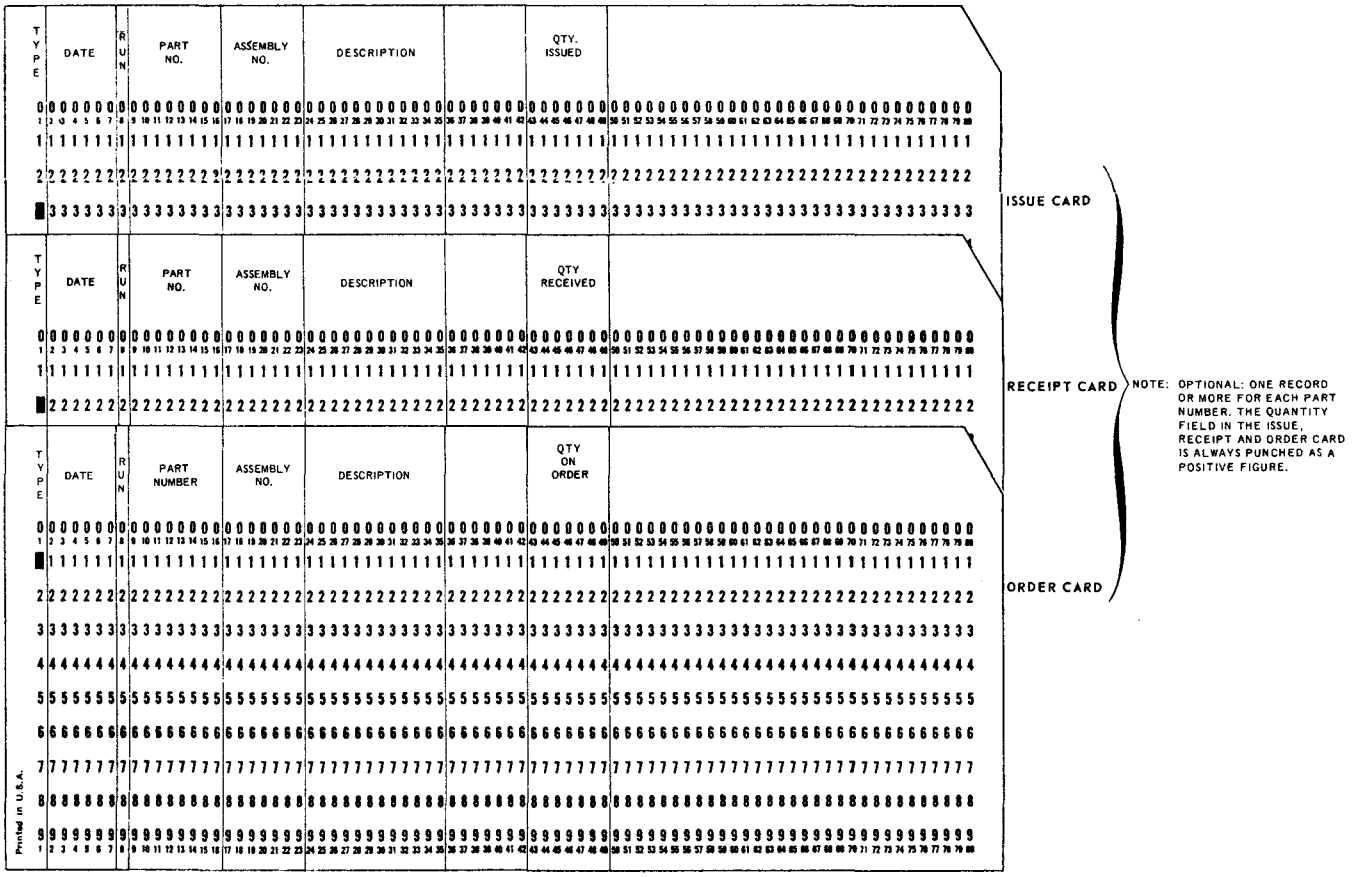


Figure A–16c. Input Card Formats Types 1, 2, and 3

DIVISION	CURRENT OR LAST ACTIVITY DATE	PART NUMBER	ASSEMBLY NUMBER	DESCRIPTION	NEW BIN QUANTITY	NEW ON–ORDER QUANTITY	NEW AVAILABLE QUANTITY	REORDER POINT	REORDER QTY.	USAGE LAST YEAR	NEW USAGE THIS YEAR
12	3 8	9 16	17 23	24 35	36 42	43 49	50 56	57 61	62 66	67 73	74 80

Figure A–17a. Stock Status Disc Record Output

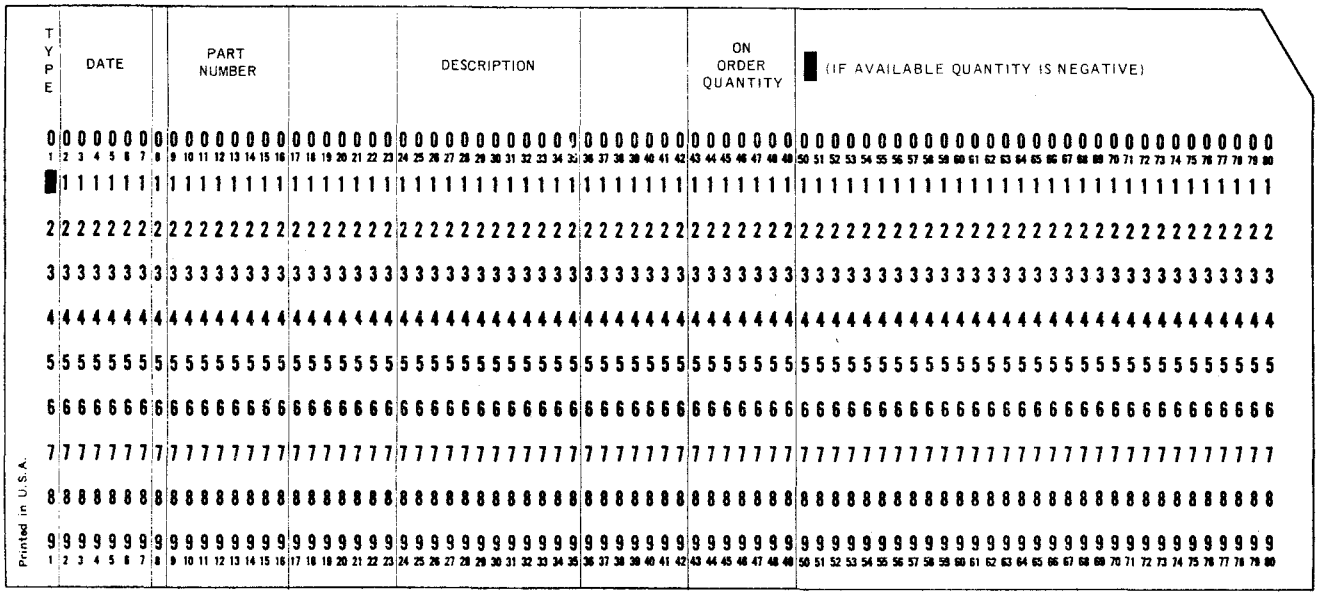


Figure A–17b. Order Card Output

If negative available quantity is indicated on the stock status balance record printout, then an X-punch (11 zone) is punched in Column 50 of the order card.

An order card is present only if replenishment is necessary.

RPG PRINTER FORMAT CHART

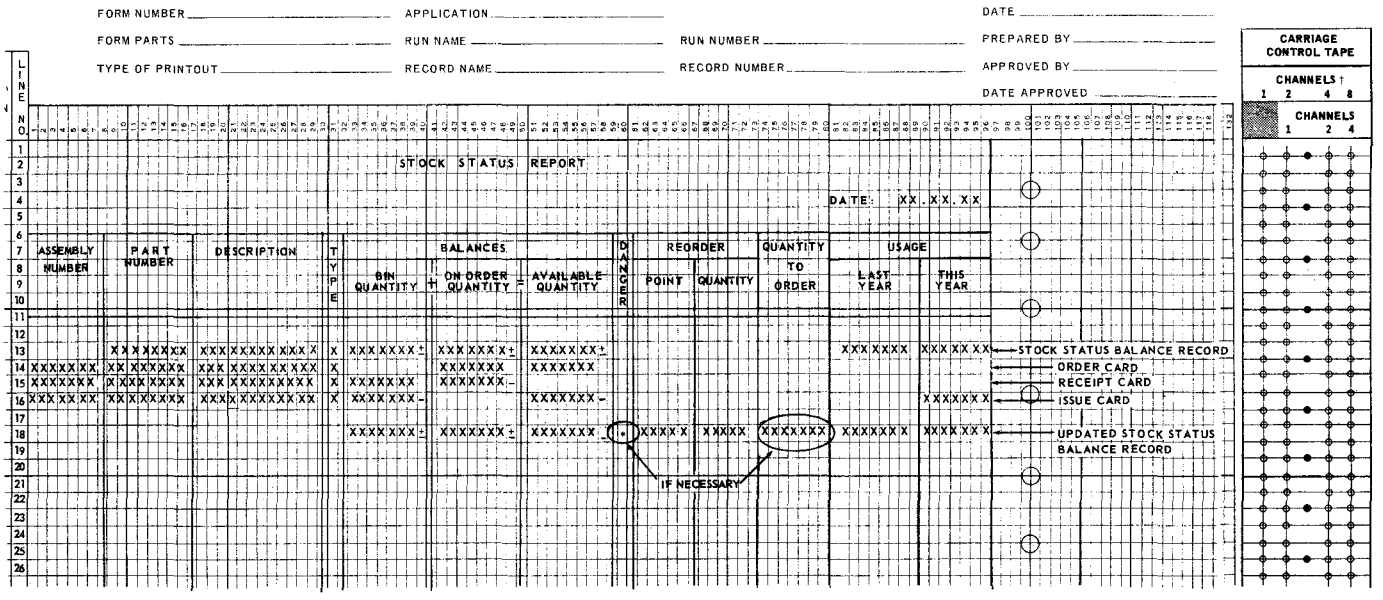


Figure A–17c. Daily Stock Status Report Format

A4.1. Procedure

(1) The input files are sequence checked by part number.

(2) The stock status balance record from disc is stored for updating and change.

(3) The on-order quantity (Columns 43-49) from the on-order card is added to the summary record totals of on-order and available quantities.

(4) Quantity received (Columns 43-49) from the receipt card is added to the summary record total of bin quantity and subtracted from the summary record total of on-order quantity.

(5) Quantity issued (Columns 43-49) from the issue card is treated as follows:

(a) Subtracted from the summary record total of bin quantity;

(b) Subtracted from the summary record total of available quantity;

(c) Added to the summary record total of usage this year.

(6) When a change in part number occurs, the summary record total of available quantity is compared to the reorder point field. If the available quantity is less than or equal to the reorder point, the following calculation is performed and an order card is produced: Quantity to be Ordered = Reorder Point + Reorder Quantity – Available Quantity.

If an order card is produced, quantity to be ordered must also be printed in print positions 74-80. If the accumulated available quantity is negative, an asterisk (*) is placed in print position 60 of the stock status report, and an X-punch (11 zone) is punched in Column 50 of the on-order card.

A4.2. Description of the Stock Status Report Text

The layout of this report is shown in Figure A-17c.

(1) The report header line consists of the current date from the date card.

(2) The stock status balance input record is to be printed as shown on line 13 of Figure A-17c.

(3) Designating information from the on-order card, the receipt card, and the issue card are printed as shown on lines 14, 15, and 16, respectively. The on-order card quantity field is printed as a positive quantity in print positions 42-48 and 51-57 of line 14. The receipt card quantity field is printed as a positive quantity in print positions 33-39 and as a negative quantity in print positions 42-48 (line 15). Issue card quantity field is printed as a negative or a positive quantity in print positions 33-39 and 51-57, and as a positive quantity in print positions 90-96 (line 16).

(4) The updated stock status balance record is printed as shown on line 18 of Figure A-17c.

A4.3. File Specifications – Figure A-18

The File Description form specifies the file name of the two input files and the three output files. The device associated with each file is specified. The entry A under sequence indicates that the input files are in ascending order. This specification is related to the M1 entries on the Input Format Specifications form. Each file except the printer uses two I/O areas to improve processing efficiency.

REPORT PROGRAM GENERATOR
FILE DESCRIPTION

PROGRAM _____ PROGRAMMER _____ DATE _____ PAGE ___ OF ___ PAGES

PG. NO.	LINE NO.	FILE NAME	FORM TYPE	FILE DESIGNATION / FILE TYPE	END OF FILE SEQUENCE	FORMAT	BLOCK LENGTH	RECORD LENGTH	MODE	RA FIELD LENGTH	RA TYPE	OVERFLOW INDICATOR	FILE ORGANIZATION EXTENSION: KEY FIELD START LOCATION	E	DEVICE	LABELS	LOGICAL UNIT NUMBER	FILE ADDITION: LOGICAL UNIT NUMBER / NAME OF LABEL EXIT	CYLINDER OVERLOW	REWIND	PROGRAM IDENTIFICATION
	01	FDISCIN	I	P	A	F	800	80				1			DISC		S				
	02	FINPUT	I	S	A	F	80	80				1			READER						
	03	FPRINT	O			F	132	132				OF			PRINTER						
	04	FPUNCH	O			F	80	80				1			PUNCH						
	05	FDISCOUT	O			F	800	80				1			DISC		S				

Figure A–18. Inventory File Description

A4.4. Input Specifications – Figures A-19 and A-20

(1) Lines 1 and 2 specify the date card.

(2) The OR relationship is used on lines 3-5 to describe the on-order, receipt, and issue cards because all three records have the same fields in the same positions. Since part number is also specified on these card types, the L1 and M1 entries are made on line number 7. A level 1 control break (L1) occurs upon a change in part number. Part number is also sequence checked (M1).

REPORT PROGRAM GENERATOR
INPUT FORMAT SPECIFICATIONS

PROGRAM _____ PROGRAMMER _____ DATE _____ PAGE ___ OF ___ PAGES

PG. NO.	LINE NO.	FILE NAME	FORM TYPE	OPTION NUMBER: SEQ.	RECORD INDICATOR	RECORD IDENTIFICATION CODES 1: POSITION	1: NOT	1: C/Z/D	1: CHAR	2: POSITION	2: NOT	2: C/Z/D	2: CHAR	3: POSITION	3: NOT	3: C/Z/D	3: CHAR	STACKER SELECT	DATA FORMAT	FIELD LOCATION: START	FIELD LOCATION: END	DECIMAL POSITIONS	FIELD NAME	MATCHING FIELDS CONTROL LEVEL	FIELD RECORD RELATION	FIELD INDICATORS: +	FIELD INDICATORS: –	FIELD INDICATORS: 8 OR 12	STERLING SIGN POSITION	PROGRAM IDENTIFICATION
	01	INPUT	I	AA	08	1			C8																					
	02		I																	2	7	0	DATE							
	03		I	BB	01	1			C1																					
	04		I	OR	02	1			C2																					
	05		I	OR	03	1			C3																					
	06		I																	1	1	0	TYPE							
	07		I																	9	16	0	PART	L1	M1					
	08		I																	17	23	0	NUM							
	09		I																	24	35		DESC							
	10		I																	43	49	0	QTY							

Figure A–19. Inventory Card Input Specifications

REPORT PROGRAM GENERATOR
INPUT FORMAT SPECIFICATIONS

PROGRAM _____ PROGRAMMER _____ DATE _____ PAGE ___ OF ___ PAGES

PG. NO.	LINE NO.	FORM TYPE	FILE NAME	SEQ.	OPTION NUMBER	RECORD INDICATOR	RECORD IDENTIFICATION CODES 1: POSITION	1: NOT	1: C/Z/D	1: CHAR	2: POSITION	2: NOT	2: C/Z/D	2: CHAR	3: POSITION	3: NOT	3: C/Z/D	3: CHAR	STACKER SELECT	DATA FORMAT	DECIMAL POSITIONS	FIELD LOCATION: START	FIELD LOCATION: END	MATCHING FIELDS CONTROL LEVEL: FIELD NAME			FIELD RECORD RELATION: FIELD INDICATORS: +	-	B OR b	STERLING SIGN POSITION	PROGRAM IDENTIFICATION
1	3 5	6	7 14	15	17	18 19	21 24	25	26	27	28 31	32	33	34	35 38	39	40	41	42	43	44 47	48 51	52	53 58	59	61	63	65	67	69	71 74	75 80
	0 1	1	DISCIN	CC		10																										
	0 2	1																			1	8 0	R E C O R D									
	0 3	1																				2 0	D I V									
	0 4	1																			9	1 6 0	P A R T		M1							
	0 5	1																			1 7	2 3 0	A S S E M									
	0 6	1																			2 4	3 5	D E S C									
	0 7	1																			3 6	4 2 0	S B I N									
	0 8	1																			4 3	4 9 0	S O R D E R									
	0 9	1																			5 0	5 6 0	S A V A I L					1 7				
	1 0	1																			5 7	6 1 0	S P O I N T									
	1 1	1																			6 2	6 6 0	S Q T Y									
	1 2	1																			6 7	7 3 0	S L A S T									
	1 3	1																			7 4	8 0	S T H I S									

Figure A–20. Inventory Tape Record Input Specifications

Lines 1-13 are the specifications for the input stock status balance disc record. Part number is sequence checked (M1). If the available quantity field (SAVAIL) is negative, field indicator 17 is turned on. The matching record indicator (MR) is turned on when a disc record matches the input card records.

A4.5. Calculation Specifications – Figure A-21

(1) Lines 1 and 2 specify the calculations for each of the on-order cards.

(2) Lines 3 and 4 specify the calculations for each of the receipt cards.

(3) Lines 5-7 specify the calculations for each of the issue cards.

(4) Lines 8-11 specify the calculations executed at total time.

NOTES: (1) Matching record indicator (MR) indicates that lines 1–7 are processed only when there is a match on part number between the disc record and the card records.

(2) Resulting indicator 17 represents a negative available quantity and is used to control output.

Resulting indicator 18 represents a negative on-order quantity; it is used to clear to zero the on-order field at total time.

Resulting indicator 16 is turned on when the available quantity is less than or equal to the reorder point. When this indicator is on, lines 10 and 11 are executed.

REPORT PROGRAM GENERATOR
CALCULATION SPECIFICATIONS

PROGRAM _____ PROGRAMMER _____ DATE _____ PAGE ___ OF ___ PAGES

PG. NO.	LINE NO.	FORM TYPE	CONTROL LEVEL	INDICATORS AND	INDICATORS AND	FACTOR 1	OPERATION	FACTOR 2	RESULT FIELD	FIELD LENGTH	DECIMAL POSITIONS	HALF ADJUST	RESULTING INDICATORS COMPARE HI 1>2	RESULTING INDICATORS COMPARE LO 1<2	RESULTING INDICATORS COMPARE EQ 1=2	COMMENTS	PROGRAM IDENTIFICATION
	01	C		01	MR	QTY	ADD	SORDER	SORDER								
	02	C		*		QTY	ADD	SAVAIL	SAVAIL					17			
	03	C		02	MR	QTY	ADD	SBIN	SBIN								
	04	C		*		SORDER	SUB	QTY	SORDER					18			
	05	C		03	MR	SBIN	SUB	QTY	SBIN								
	06	C		*		SAVAIL	SUB	QTY	SAVAIL					17			
	07	C		*		QTY	ADD	STHIS	STHIS								
	08	C	L1	18			Z-ADD	+00000000	SORDER								
	09	C	L1			SAVAIL	COMP	SPOINT						16	16		
	10	C		*	16	SPOINT	SUB	SAVAIL	ORQTY	7	0						
	11	C		*		SQTY	ADD	ORQTY	ORQTY								

Figure A–21. Inventory Calculation Specifications

A4.6. Output Specifications – Figure A-22

(1) Lines 1–3 (Figure A-22, Part 1) provide for the printing of the data at the top of each page.

(2) Lines 4–12 (Figure A-22, Part 1) provide for the printing of the input stock status balance record.

(3) Lines 13–26 (Figure A-22, Part 1) provide for the printing of the on-order, receipt, and issue cards.

(4) Lines 1–7 (Figure A-22, Part 2) provide for the punching of an order card at total time and only if indicator 16 is on.

NOTE: An X punch (11 zone) is placed in Column 50 of the card if available quantity is negative.

(5) Lines 8–22 (Figure A-22, Part 2) provide for writing the updated stock status balance record out on disc.

(6) Lines 21–25 (Figure A-22, Part 2) and lines 1–5 (Figure A-22, Part 3) provide for the printing of the updated stock status totals. An asterisk (*) is printed in print position 60 if the available quantity is negative. Quantity to be ordered is printed in print positions 74-80 if an order card has been punched.

REPORT PROGRAM GENERATOR
OUTPUT FORMAT SPECIFICATIONS

PROGRAM _____ PROGRAMMER _____ DATE _____ PAGE ___ OF ___ PAGES

PG. NO.	LINE NO.	FORM TYPE	FILE NAME	TYPE H/D/T	STACKER SELECT	SPACE BEFORE	SPACE AFTER	SKIP BEFORE	SKIP AFTER	OUTPUT INDICATORS AND	OUTPUT INDICATORS AND	FIELD NAME	ZERO SUPPRESS	BLANK AFTER	END POSITION IN OUTPUT RECORD	DATA FORMAT	CONSTANT OR EDIT WORD	STERLING SIGN POSITION	PROGRAM IDENTIFICATION
	01	O	PRINT	H				1	04	01									
	02	O		OR						08									
	03	O										DATE			94		'0 '		
	04	O	PRINT	D		01				10									
	05	O										PART	Z		16				
	06	O										DESC			29				
	07	O										TYPE			31				
	08	O										SBIN			40		'0 -'		
	09	O										SORDER			49		'0 -'		
	10	O										SAVAIL			58		'0 -'		
	11	O										SLAST	Z		88				
	12	O										STHIS	Z		96				
	13	O	PRINT	D		01				01									
	14	O		OR						02									
	15	O		OR						03									
	16	O										NUM	Z		7				
	17	O										PART	Z		16				
	18	O										DESC			29				
	19	O										TYPE			31				
	20	O								N01		QTY	Z		39				
	21	O								N03		QTY	Z		48				
	22	O								N02		QTY	Z		51				
	23	O								03		QTY	Z		96				
	24	O								03					40		'-'		
	25	O								02					49		'-'		
		O								03					58		'-'		

Figure A–22. Inventory Output Specifications (Part 1 of 3)

REPORT PROGRAM GENERATOR
OUTPUT FORMAT SPECIFICATIONS

PROGRAM _____ PROGRAMMER _____ DATE _____ PAGE ____ OF ____ PAGES

PG. NO.	LINE NO.	FORM TYPE	FILE NAME	TYPE H/D/T	STACKER SELECT	SPACE BEFORE	SPACE AFTER	SKIP BEFORE	SKIP AFTER	OUTPUT INDICATORS AND	OUTPUT INDICATORS AND	FIELD NAME	ZERO SUPPRESS	BLANK AFTER	END POSITION IN OUTPUT RECORD	CONSTANT OR EDIT WORD	STERLING SIGN POSITION	PROGRAM IDENTIFICATION
	01	O	PUNCH	T						L1	16							
	02	O													1	'1'		
	03	O										DATE			7			
	04	O										PART			16			
	05	O										DESC			35			
	06	O										ORQTY			49			
	07	O								17					50	'-'		
	08	O	DISCOUT	T						NMR	10							
	09	O										RECORD			80			
	10	O	DISCOUT	T						L1								
	11	O										DIV			2			
	12	O										DATE			8			
	13	O										PART			16			
	14	O										DESC			35			
	15	O										SBIN			42			
	16	O										SORDER			49			
	17	O										SAVAIL			56			
	18	O										SPOINT			61			
	19	O										SQTY			66			
	20	O										SLAST			73			
	21	O										STHIS			80			
	22	O										ASSEM			23			

Figure A–22. Inventory Output Specifications (Part 2 of 3)

REPORT PROGRAM GENERATOR
OUTPUT FORMAT SPECIFICATIONS

PROGRAM _____ PROGRAMMER _____ DATE _____ PAGE ____ OF ____ PAGES

PG. NO.	LINE NO.	FORM TYPE	FILE NAME	TYPE H/D/T	STACKER SELECT	SPACE BEFORE	SPACE AFTER	SKIP BEFORE	SKIP AFTER	OUTPUT INDICATORS AND	OUTPUT INDICATORS AND	FIELD NAME	ZERO SUPPRESS	BLANK AFTER	END POSITION IN OUTPUT RECORD	CONSTANT OR EDIT WORD	STERLING SIGN POSITION	PROGRAM IDENTIFICATION
	01	O	PRINT	T		1	2			L1								
	02	O										SBIN			40	'0-'		
	03	O										SORDER			49	'0-'		
	04	O										SAVAIL			58	'0-'		
	05	O								17					60	'*'		
	06	O										SPOINT	Z		66			
	07	O										SQTY	Z		72			
	08	O								16		ORQTY	Z		80			
	09	O										SLAST	Z		88			
	10	O										STHIS	Z		96			

Figure A–22. Inventory Output Specifications (Part 3 of 3)

APPENDIX B. SUMMARY OF MODAL PROCESSING FEATURES

B1. INTRODUCTION

The UNIVAC 9400 Report Program Generator is capable of compiling, for execution on the 9400 only, source programs written in the RPG language for the 9400 (as described in this document), the UNIVAC 9200/9300 (as described in *UNIVAC 9200/9300 Systems Tape and Disc Report Program Generator, UP-7620,* current version), and IBM System 360 Model 20 (as described in *IBM System 360 Model 20 Disc and Tape Programming Systems Report Program Generator, IBM document number C24-9001-3).*

Since the three languages are slightly different, the user of the UNIVAC 9400 RPG who wishes to compile a program originally written for the UNIVAC 9300 or IBM 360 Model 20 is required to include an appropriate entry in Column 7 of the RPG control card (see Appendix E):

'3' for compilation of UNIVAC 9200/9300 programs;

'2' for compilation of IBM 360 Model 20 programs.

Any other entry in Column 7 (including blank) causes the compiler to assume that its input is a UNIVAC 9400 source program. Hence, there are considered to be three modes of compilation.

The compiler always assumes that the UNIVAC 9200/9300 or IBM 360 Model 20 source program is written in the most advanced version of RPG available for the given system; that is, disc RPG rather than tape or card RPG. Therefore, any fields of RPG source statements which are permitted to contain comments in the card or tape version, but are reserved for significant entries in the disc version, may have to be changed to blanks for successful compilation.

The following major distinctions are observed by UNIVAC 9400 RPG in the three modes of compilation.

B1.1. File Format, Block Length, and Record Length

(1) UNIVAC 9200/9300 and IBM 360/20 Mode - file format, block length, and record length may be omitted for card and printer files. The following assumptions are made by the compiler:

Card files - format F, block and record length of 80;

Printer files - format F, block and record length of 132.

(2) UNIVAC 9400 mode - file format, block length, and record length should be specified.

B1.2. Card Zone and Digit Testing

(1) UNIVAC 9200/9300 mode - card columns to be tested are translated to UNIVAC 9200/9300 compressed code. Zone testing is performed on bits 6 and 7 of the input and master characters, while digit testing is performed on bits 0 through 5.

(2) IBM 360/20 mode - EBCDIC zones and digits are compared, except for &, -, and blank, which are considered to have the same zones as A, J, and 1, respectively.

(3) UNIVAC 9400 mode - zones and digits, as those words are used in punched-card processing systems, are compared without reference to internal representation. The possible zones are:
(a) 12-zone (characters A-I, &, plus zero)

(b) 11-zone (characters J-R, –, minus zero)

(c) 0-zone (characters S-Z)

(d) no-zone (characters 0-9, blank)

All other characters are treated as having the same zone and the same digit.

B1.3. Printer Loop Channel Numbers

Table B-1 shows all the valid printer channel numbers in the three modes of compilation. The compiler treats UNIVAC 9200/9300 channel numbers and IBM 360/20 channel numbers internally as the corresponding UNIVAC 9400 channel number. The printer loop for the execution of the compiled object program must be prepared in accordance with this table.

	UNIVAC 9400	UNIVAC 9200/9300	IBM 360 MODEL 20
Overflow Channel	9	1	12
Home Paper Channel	14 15	7 15	1 15
Other Channels	4 - 6	4 - 6	4 - 6
	7	-	7
	8	-	8 - 9
	10 - 11	-	10 - 11
	12 - 13	2 - 3	2 - 3

NOTES: (1) Channels 8 and 9 of IBM 360 Model 20 programs are both translated as Channel 8. IBM 360 Model 20 programs that make use of both channels must be modified.

(2) Channel 15 may be specified in any mode, and causes printer spacing at the rate of 8 lines per inch instead of 6 lines per inch.

Table B–1. Valid Printer Channel Numbers

B1.4. Tape Rewind

Table B-2 shows the valid codes and the corresponding rewind actions.

9400 Mode	9200/9300 Mode	IBM 360/20 Mode	Action At Start of Processing	Action At End of Processing
R	–	Blank	Rewind, No Lock	Rewind, No Lock
U or Blank	Blank	U	Rewind, No Lock	Rewind, Lock
N	–	–	No Rewind	No Rewind
K	–	N	Rewind, No Lock	No Rewind
L	–	–	No Rewind	Rewind, No Lock
M	N	–	No Rewind	Rewind, Lock

Table B–2. Valid Codes and Tape Rewind Actions

B1.5. Sign Validity Checking

The validity of the signs of certain fields is checked and corrected as required. The checking is performed for the MOVE, MOVEL, MLLZO and MHLZO operations and for numeric fields in the input data.

B1.5.1. MOVE and MOVEL

When an alphanumeric field is moved into a numeric field, checking and correcting is performed as follows:

(1) For the UNIVAC 9400 mode, signs 6_{16} and B_{16} are changed to D_{16}, D_{16} is unchanged, and all other signs are changed to C_{16}.

(2) For the UNIVAC 9200/9300 mode programs, sign checking is not performed.

(3) For the IBM 360 Model 20 programs, signs C_{16} through F_{16} are unchanged, sign 6_{16} is changed to D_{16} only if the accompanying digit is 0_{16}, and all other signs are changed to C_{16}.

B1.5.2. Move Low to Low (MLLZO) and Move High to Low (MHLZO)

When factor 2 is alphanumeric and the result field is numeric, sign checking is performed as follows:

(1) For UNIVAC 9400 mode programs, signs 6_{16} and B_{16} are changed to D_{16}, D_{16} is unchanged, and all other signs are changed to C_{16}.

(2) For UNIVAC 9200/9300 mode programs, signs 6_{16} and 9_{16} are changed to D_{16}, A_{16} through F_{16} are unchanged, and all other signs are changed to C_{16}.

(3) For IBM 360 Model 20 mode programs, signs C_{16} through F_{16} are unchanged, sign 6_{16} is changed to D_{16} only if the accompanying digit is 0_{16}, and all other signs are changed to C_{16}.

B1.5.3. Input

The signs of numeric fields in the input data including table data are checked for validity and corrected if necessary as follows:

(1) For the UNIVAC 9400 mode, 6_{16} is changed to D_{16}, D_{16} remains unchanged, and all other signs are changed to C_{16}.

(2) For the UNIVAC 9200/9300 mode, no checking is performed.

(3) For IBM 360 Model 20 mode programs, signs C_{16} through F_{16} are unchanged, sign 6_{16} is changed to D_{16} only if the accompanying digit is 0_{16}, and all other signs are changed to C_{16}.

B1.6. Fixed Dollar Sign Editing

Fixed dollar sign editing is not explicitly provided for UNIVAC 9200/9300 mode programs. See 5.3.7.2.

B1.7. Record Sequence of Chaining Files

(1) UNIVAC 9200/9300 mode - record sequence (Columns 7-8 of the File Extension Specifications) may be blank for chaining files and is otherwise ignored except for a syntactic validity check.

(2) UNIVAC 9400 and IBM 360 Model 20 modes - record sequence is required for chaining file specifications.

APPENDIX C. SUMMARY OF PROGRAM INDICATORS

INDICATOR	LOCATION	WHERE USED	TURNED ON **	TURNED OFF **	REMARKS
Resulting Indicator 00	Internal	Output Indicators	At beginning of processing before any input records are read.		
Field Indicators 01-99 Plus Minus	Input Specifications Form	Indicators – (Calculation Form) Output Indicators	 By Blank or Zero in specified field. By Plus in specified field. By Minus in specified field.	The next time the associated field is read.	More than one Field Indicator can be on at one time.*
Halt Indicators H0-H9	Internal	Field Indicators – (Input Form) Indicators – (Calculation Form) Resulting Indicators– (Calculation Form) Output Indicators	If condition tested for occurs.		
Last Record LR	Internal	Control Level (Calculation Form) Output Indicators	At end of processing of last record read.	At the beginning of processing.	
Level Zero L0	Internal	Control Level	At beginning of processing before any input records are read.		
Control Level Indicators L1-L9	Internal	Control Level Indicators – (Calculation Form) Output Indicators	By appropriate control level break.	At end of following detail cycle.	Remains on during detail calculation and output cycle.
Matching Record MR	Internal	Indicators – (Calculation Form) Output Indicators	After total time, if there is a matching record in a secondary file.	After all total calculation and output operations are completed.	Remains on during the complete processing of all records containing matching fields.
Overflow Indicators OA, OB, OC, OD, OE, OF, OG, and OV	Internal	Indicators – (Calculation Form) Output Indicators	After detail or total time if overflow line detected during report output.	After detail time.	Remains on for detail time if turned on after total. Remains on for one complete cycle (total and next detail) if turned on after detail.
Record Indicators 01-99	Input Specifications Form	Indicators – (Calculation Form) Output Indicators	When specified record has been read and is ready for processing.	When another Record Indicator is turned on.	More than one Record Indicator can be on at one time only if chaining files are present.*
Resulting Indicators 01-99, H0-H9 Plus Minus Zero Compare Operation High Low Equal	Calculation Specifications Form	Indicators – (Calculation Form) Output Indicators: Field Record Relation	 By Plus balance in field By Minus balance in field. By Zero balance in field. If Factor 1>Factor 2 If Factor 1<Factor 2 If Factor 1=Factor 2	The next time a calculation is performed for which the program specifies the indicator as a Resulting Indicator.	More than one Resulting Indicator can be on at one time.*
First Page 1P	Internal	Output Indicators	At beginning of processing before any input records are read.	At the end of detail time.	

* Field Indicators, Record Indicators and Resulting Indicators are numbered in the range 01-99 and are mutually exclusive (there are only 99 indicators in total).

**All indicators can be turned on or off by using SETON or SETOF operation.

APPENDIX D. STERLING NOTATION

D1. INTRODUCTION

Sterling notation is the expression of monetary fields in terms of pounds, shillings, and pence. One pound is equal to 20 shillings; one shilling equals 12 pence. Entries for sterling notation are made on the control card and on the Input and Output Specifications form. Monetary fields using sterling notation must conform to the specific rules indicated in this section. RPG uses three basic formats for the representation of sterling amounts.

D1.1. Format 1

This format is used if the output is to be printed. Format 1 processes sterling amounts in the standard notation of pounds, shillings, and pence, including decimal fractions of a penny.

Format 1 allows for a maximum of 10 positions for pounds, 2 positions for shillings, 2 for pence, and 3 for decimals. Table D-1 illustrates maximum size for each format.

FORMAT	POUNDS (£)	s	d	DECIMALS (f)	USE
1	££££££££££	ss	dd	fff	PRINTING (OUTPUT)
2C, 2D	££££££££££	s	d	fff	READING/PUNCHING (INPUT/OUTPUT)
2A, 2B	££££££££££	ss	d	fff	READING/PUNCHING (INPUT/OUTPUT)
Pence	ddddddddddddd			fff	RPG

Legend:
£ pound digit
s shilling digit
d penny digit
f penny fractional digit

Table D–1. Maximum Sizes of Sterling Notation Formats

D1.2. Format 2

Format 2 is used for both input and output. Input cards are read and output cards are punched in this format.

Format 2 allows for a maximum of 10 positions for pounds and 3 positions for decimals. Shillings and pence may be represented in either the British Standards Institution (BSI) or Hollerith code.

Table D-2 lists examples of sterling amounts written in each of the formats.

	UNSIGNED				NEGATIVE				POSITIVE			
	£	19	11	.15	−£425	15	10	.14	+£188	0	7	
Format	Pounds	Shillings	Pence	Decimal	Pounds	Shillings	Pence	Decimal	Pounds	Shillings	Pence	Decimal
1	0	19	11	15	425	15	10	1M	188	00	0G	
2A	0	19	+	15	425	15	−	1M	18H	00	7	
2B	0	19	−	15	425	15	+	1M	18H	00	7	
2C	0	I	−	15	425	E	+	1M	18H	0	7	
2D	0	I	+	15	425	E	−	1M	18H	0	7	
PENCE	239			15	102190			1M	4512G			

Table D–2. Sterling Amounts Expressed in Available Formats of Notation

D1.2.1. Shillings

BSI Code

A single column is used to represent the shillings field. Amounts of 0 through 9 are represented by the punches 0 through 9, ten shillings by a 12 punch in the column, and eleven through nineteen shillings by the A through I punches, respectively.

Hollerith Code

Two columns are used to represent the shillings field. Decimal notation is used with the first column representing the tens position (that is, it contains either 0 or 1) and with the second column for the units (digits 0 through 9).

D1.2.2. Pence

BSI Code

The pence field is represented by a single column. Amounts of 0 through 9 are represented by the punches 0 through 9, ten pence by a 12 punch, and eleven pence by an 11 punch.

Hollerith Code

The Hollerith code is similar to the pence field of the BSI code in that both use a single column to represent pence. Amounts of 0 through 9 are represented by the punches 0 through 9. However, ten pence is represented by an 11 punch and eleven pence by a 12 punch. Format 2 may be used in any of the following combinations of BSI and Hollerith codes.

D1.2.3. Format 2 Types

TYPE OF FORMAT	SHILLINGS	PENCE
Format 2A	Hollerith Code	Hollerith Code
Format 2B	Hollerith Code	BSI Code
Format 2C	BSI Code	BSI Code
Format 2D	BSI Code	Hollerith Code

Table D-1 illustrates the maximum size of Format 2. Refer to Table D-2 for an example of sterling amounts written in Format 2. Table D-3 is a summary of the formats for sterling notation.

	PRINTING FORMAT	CARD READING/PUNCHING FORMAT	
FIELDS	FORMAT 1	FORMATS 2C, 2D	FORMATS 2A, 2B
POUNDS	1–10 Positions	1–10 Positions	1–10 Positions
SHILLINGS	2 Positions (contents not to exceed 19)	1 Position BSI Code	2 Positions (contents not to exceed 19) Hollerith Code
PENCE	2 Positions (contents not to exceed 11)	1 Position BSI or Hollerith Code	1 Position BSI or Hollerith Code
DECIMALS	0–3 Positions	0–3 Positions	0–3 Positions

Table D–3. Summary of Sterling Notation Formats

D1.2.4. Signs

The sign is found in the least significant digit of the decimal portion of a sterling field. However, if no decimal fractions of a penny exist in the field, the zone punch identifying the sign is placed in the units position of the pounds field.

The sign codes are as follows:

–(minus) = 11 punch
+(plus) = 12 punch or blank

If a sterling input field is not signed according to the above specifications, then the position of the sign can be specified in Columns 71-74 of the Input Specifications form.

NOTE: An 11 punch always requires a digit underpunched in the same column. A 12 punch in the appropriate position is always used to signify positive output amounts.

D1.2.5. Leading Zeros

Leading zeros can be represented by blanks in sterling input fields.

D1.3. Pence Format

The pence format is a "pence only" notation of a sterling amount (all pounds and shilling fields are converted to pence).

Pence format allows for a maximum of 16 positions - 13 positions for pence and 3 decimal positions.

Table D-1 illustrates the maximum size of the pence format. Refer to Table D-2 for an example of sterling amounts in pence format.

NOTE: RPG processing of sterling amounts requires that certain information must be entered on the Input and Output Specifications forms. No entries are required on the Calculation Specifications form, but the user should remember that all calculations are done by the RPG in pence format. Therefore, this fact must be considered when defining the length of result fields or when using factors 1 and 2.

D1.4. Specifying Sterling Notation

The presence of sterling fields is indicated to the RPG by additional entries in the RPG control card (Columns 17-20).

Sterling input information can be represented in two formats:

- Hollerith
- British Standards Institution (BSI)

The RPG converts the input fields into pence format fields. If the output is to be printed, the fields are converted with shillings and pence printed in two positions each. If the output is to be punched, the output is converted to either Hollerith or BSI format.

D1.4.1. RPG Control Card

All sterling input to any one program must be in one format and all sterling output punched by any given program is in one format, not necessarily that of input. To select the required sterling routines, the RPG needs information regarding the input and output formats. This information is entered in Columns 17-20 of the RPG control card and is as follows:

COLUMNS

17 Shilling input format
Codes:
1 = Hollerith
2 = BSI
Blank = Input has no sterling currency fields, or fields are in pence format only.

18 Pence input format
Codes:
1 = Hollerith
2 = BSI
Blank = Input has no sterling currency fields, or fields are in pence format only.

19 Shilling output format
Codes:
1 = Hollerith
2 = BSI
Blank or 0 = No output sterling currency fields, or fields are in printer format.

20 Pence output format
Codes:
1 = Hollerith
2 = BSI
Blank or 0 = No output sterling currency fields, or fields are in printer format.

NOTE: All fields represented as sterling fields are converted (if required) according to the specifications in the RPG control card.

D1.4.2. Input Specifications

The sign of sterling fields may be in standard or nonstandard position. If the sterling field has decimal positions, the standard position of the sign is in the rightmost decimal position of the field. If the sterling field has no decimal positions, the standard position of the sign is in the units position of the pounds field. The following input rules must be considered:

(1) One of the digits 0, 1, 2, or 3 must be entered in Column 52 of the Input Format Specifications form to indicate the required number of decimal positions.

(2) Column 43 (Data Format) of the Input Format Specifications form must be blank, since sterling fields may not be specified in packed format.

(3) On input, the pounds field must consist of at least one and no more than nine positions.

(4) If the sign is in standard position, enter an S in Column 74. If not, enter the actual position in Columns 71-74.

D1.4.3. Output Specifications

Column 44 (Data Format) of the Output Format Specifications form must be blank, since sterling fields may not be specified in packed format.

D1.4.3.1. Tape, Disc, and Punched Output

The field may be specified as any combination of Hollerith or BSI shilling and pence formats.

Output which is not printed may not be edited or zero suppressed.

D1.4.3.2. Printed Output

For printed output, the sign appears in the least significant digit of the sterling field. Shillings and pence are printed in two positions each. It is necessary to add two additional positions for printed output when the shilling input is in the BSI format. One position must be added when the shilling input is in the Hollerith or pence-only format. Also, an extra position is required to avoid the loss of the high order digit as described under lengths of pence format fields (see D1.4.5).

Hence, three extra output print positions are required when the shilling input is specified in BSI format. Two extra output print positions are required when the shilling input is specified in Hollerith or pence-only format.

If neither zero suppression nor an edit word is specified, leading zeros in the tens digit of the shillings and pence fields are suppressed.

Editing is permitted only on printed output files. The rules governing the use of edit words are the same as those for decimal fields. The features available are:

- Zero suppression in the pounds field.
- Zero suppression in the shillings field, if both pound and shilling values are zero.
- Zero suppression in the pence fields, if pound, shilling, and pence values are zero.
- Zero suppression preceding signs, and suppression of separation marks between pounds and shillings, shillings and pence, and pence and decimals.

D1.4.3.3. Example of Proper Edit Masks

If the RPG control card contains 1 in Column 17 and sterling field STERA has a length of 6 (defined on either an input specification or a calculation specification), a correct edit word for the field is the following:

'ΔΔΔΔ&ΔΔ&0ΔCR'

Since the shilling input is in Hollerith format, the sum of the digit select characters and the significant start character must be two greater than the length of the field.

If the RPG control card contains a 2 in Column 17, a correct edit word for the field is:

'ΔΔΔΔΔ&ΔΔ&OΔCR'

Since the shilling input is in BSI format, the sum of the digit select characters and the significant start character must be three greater than the length of the field.

D1.4.4. Calculation Specifications

While no additional entries are required in this form, the user should keep in mind that all calculations are done in pence format. This must be considered when defining the length of result fields or when using factors 1 and 2.

D1.4.5. Lengths of Pence Format Fields

If a pence format result field is to be reconverted into a sterling output field, the highest amount it is permitted to contain is 2,399,999,999,999,999. This converts to a field containing ten pounds positions, which is the maximum allowed.

In order to avoid the possible loss of the high order digit, fields that are read in as Hollerith or BSI sterling always contain one more position when written as Hollerith or BSI sterling. For example, the five position sterling input field 9919+ (Hollerith format) converts to the five-position pence field 23999. When the RPG writes this field, it must allow for a five-position pence field containing up to 99999 pence (which converts to 416133, Hollerith format), the field on output is six positions long.

APPENDIX E. OPERATING CONSIDERATIONS

E1. INTRODUCTION

This appendix provides operational and related information which enables the programmer to compile and execute his RPG program. Information is presented for the following:

- RPG source program deck arrangement
- Job control statements
- RPG control card format
- Halt indicator processing
- Variable-length records
- Printing line counter tape/disc files
- Table entry at execution time
- RPG console messages

For additional information related to the descriptions which follow, see *UNIVAC 9400 System Job Control Programmers Reference, UP-7793* (current version).

E2. RPG SOURCE PROGRAM DECK ARRANGEMENT

The programmer prepares the deck and arranges it as shown in Figure E–2.

(1) Job control statements

(2) RPG control card

(3) File description specifications

(4) File extension specifications

(5) Line counter specifications

(6) Input format specifications

(7) Calculation specifications

(8) Output format specifications

(9) Terminating job control statement (/* in columns 1–2).

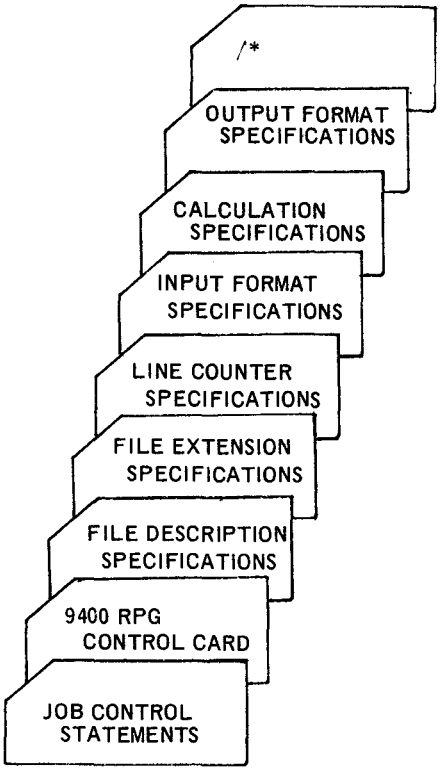


Figure E–1. Source Program Deck Arrangement

E3. JOB CONTROL STATEMENTS

Job control statements are provided in a sequence known as the control stream. Control stream preparation for tape and disc is described in the following paragraphs.

- Tape RPG

LABEL (1)	OPERATION (10)	OPERAND (16)
// JOB	RPGRUN	
// DVC	3	// LFD PRNTR
// DVC	4	// LFD OBJFIL
// DVC	6	// LFD SCR1
// DVC	7	// LFD SCR2
// DVC	8	*OPTIONAL; SEE NOTE.
// LFD	filename	
// EXEC	RPG	
// PARAM		*SEE TABLE E-1.
/$		
.		}
.		SOURCE PROGRAM STATEMENTS
.		}
/*		
/&		

NOTES:

(1) Source programs in a source library on tape can be compiled. This is possible only on systems where five or more tapes are available. The filename assigned to the library tape must appear in both the LFD statement and the PARAM statement. In addition, the name used to identify the program in the source library must be specified in the PARAM statement (the PARAM statements for RPG are given in Table E–1). Source libraries on the system reserve resident volume library (SYSRES) tape are not supported.

(2) At execution time, filenames specified on LFD statements should be identical to the filenames in the source program's File Description specifications. Exception: for 9400 mode and IBM 360 Model 20 mode programs only, the last character of eight-character filenames must be omitted from the LFD statements. Note that all files must be allocated by means of DVC and LFD statements, including card reader files.

- Disc RPG

LABEL	OPERATION	OPERAND	COMME
1	10	16	
//JOB	RPGRUN		
//DVC	3	//LFD PRNTR	
//DVC	20	//VOL 000001	
//DVC	21	//VOL 000002	//LFD SYSPOOL
//EXEC	DRPG,LOADSLIB,,REL		
//PARAM		*SEE TABLE E-1	
/$			
.			
.	SOURCE	PROGRAM STATEMENTS	
.			
/*			
/&			

The logical DVC assignments could vary depending on the structure of the Supervisor being used and the physical device types and the location on a given system. The PARAM statements for RPG and their specifications are given in Table E–1.

CONTROL STATEMENT	OPERAND	DESCRIPTION	TOS	DOS
// PARAM	CNL = {N / E}	N – cancels compiler job if an error occurs that prevents output of an object module.	X	X
		E – cancels compiler job if object module has serious errors or no object module is output.		
// PARAM	IN = program-name/filename	Indicates program name and file name of specified source program module in source code library.	X	X
		NOTE: Source code library on tape is available for use with TOS and DOS; source code library on disc is available only with DOS.		
// PARAM	LST = {K / M / N / S}	K – inhibits error flags for sequence errors.	X	X
		S – inhibits memory map listing	X	X
		N – inhibits all listable output (this option takes precedence over the other LST options).	X	X
		M – inhibits source listing and diagnostics.	X	X
// PARAM	OUT = {N / T}	N – inhibits writing of primary output.	X	
		T – produces an object file (OBJFIL) output tape (available with DOS only).		X
// PARAM	VER = mm/nn	mm specifies level number; nn specifies update number. Both mm and nn are expressed as decimal numbers.	X	X

NOTES:

- When specifying the PARAM control statement, at least one space must appear between the last slash and the operation field, and also between the operation and operand fields. For additional information concerning the PARAM control statement, see UNIVAC 9400 System Job Control Programmers Reference, UP-7793 (current version).
- Information contained within braces represents necessary entries of which one must be chosen.
- Under DOS, output will be produced in the module complex library (MCL). If OUT=T is specified, the compiler will then produce, in addition, an output to an OBJFIL tape. (MCL is the disc equivalent of OBJFIL.) For additional information concerning MCL and OBJFIL, see UNIVAC 9400 System Disc Librarian Programmers Reference, UP-7745 (current version) and UNIVAC 9400 System Tape Librarian Programmers Reference, UP-7667 (current version), respectively.
- X = currently available.

Table E–1. PARAM Job Control Statements for RPG

- Tape/Disc RPG

  If a combination tape/disc supervisor is constructed, the tape RPG control statements may be used while accessing RPG modules from the disc resident supervisor. Use of tape RPG permits the use of a disc and three tapes for compiling RPG programs. If a fourth tape is available, access of source code on tape is permitted.

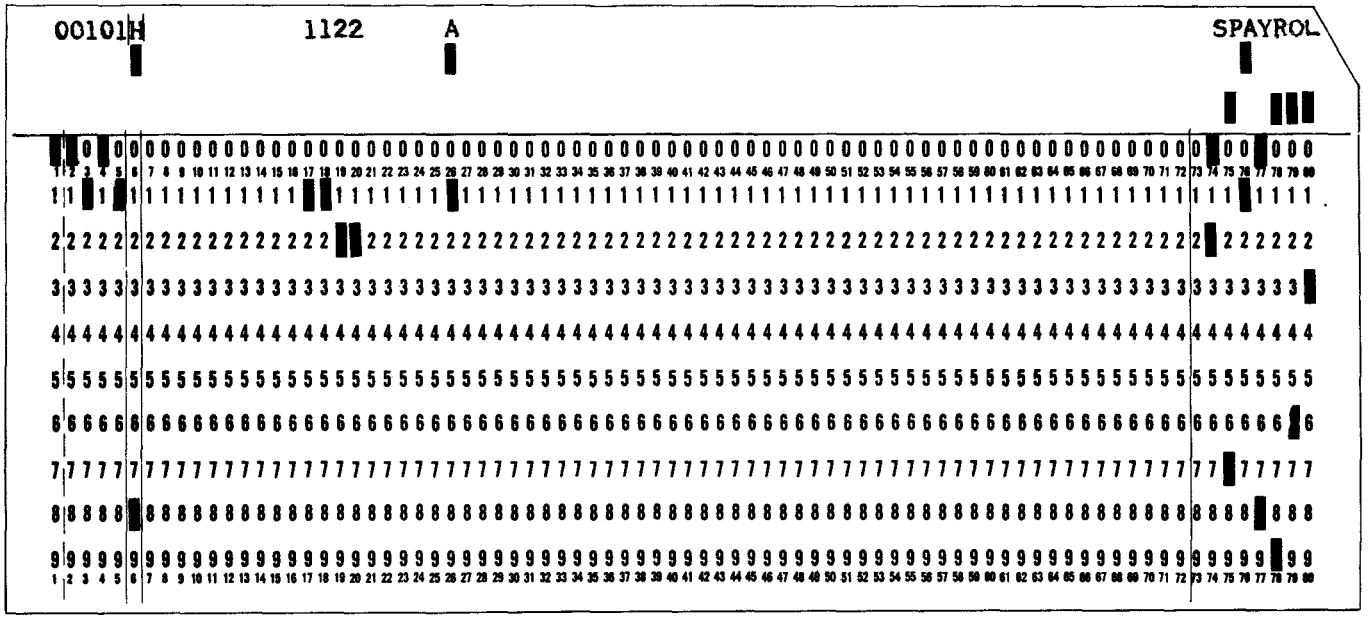
E4. RPG CONTROL CARD FORMAT

The RPG control card provides the compiler with essential information not included in the RPG specification statements. Each source deck must be preceded by only one control card. The contents and format of the control card are as follows:

COLUMNS	CONTENTS	DESCRIPTION
1–5	Any EBCDIC characters including blanks	Insignificant (page and line number can appear here).
6	H	Identifies this as a control (header) card.
7	3	Compilation takes place in UNIVAC 9300 mode (see Appendix B).
	2	Compilation takes place in IBM 360/20 mode (see Appendix B).
	Any other EBCDIC character including blanks	Compilation takes place in UNIVAC 9400 mode (see appendix B).
8	D or other nonblank character	Generates halt indicator processing code
9–16	Any EBCDIC characters including blanks	Insignificant.
17	1	Sterling shillings field on input is in Hollerith format.
	2	Sterling shillings field on input is in BSI format.
	Blank	Input has no sterling currency fields.
18	1	Sterling pence field on input is in Hollerith format.
	2	Sterling pence field on input is in BSI format.
	Blank	Input has no sterling currency fields.
19	1	Sterling shillings field on output is in Hollerith format.
	2	Sterling shillings field on output is in BSI format.
	0 or Blank	No output sterling currency fields, or fields in printer format.
20	1	Sterling pence field on output is in Hollerith format.
	2	Sterling pence field on output is in BSI format.
	0 or Blank	No output sterling currency fields, or fields in printer format.

COLUMNS	CONTENTS	DESCRIPTION
21	I	Inverted print option: numeric literals and edit words use the European conventions of punctuation (comma for decimal point and vice versa).
	Blank	No inverted print option.
22–25	Any EBCDIC characters including blanks	Insignificant.
26	A	An external subroutine is used to translate the sequence of a matching field to the internal collating sequence of the UNIVAC 9400. The name of the external subroutine must be ALTSEQ.
	Blank or 4	Standard collating sequence is assumed for matching fields. (A 4 provides compatibility with the UNIVAC 9200/9300 RPG control card.)
27–73	Any EBCDIC characters including blanks	Insignificant.
74	S	The RPG object program is to be executed as a subroutine only.
	Blank	The RPG object program is to be executed as a main program only.
75–80	XXXXX	Program identification. If column 75 is blank, the program name is assumed to be RPGOBJ. The program name is included in the object module output.

An example of a control card is shown in Figure E–2.



COLUMNS	CONTENTS	DESCRIPTION
1–5	00101	Page and line number
6	H	Header card
7	Blank	UNIVAC 9400 mode
8–16	Blank	--
17	1	Hollerith sterling input
18	1	Hollerith pence input
19	2	BSI sterling output
20	2	BSI pence output
21	Blank	No inverted print option
22–25	Blank	--
26	A	Alternate sequence
27–73	Blank	--
74	S	Subroutine object program
75–80		Program name

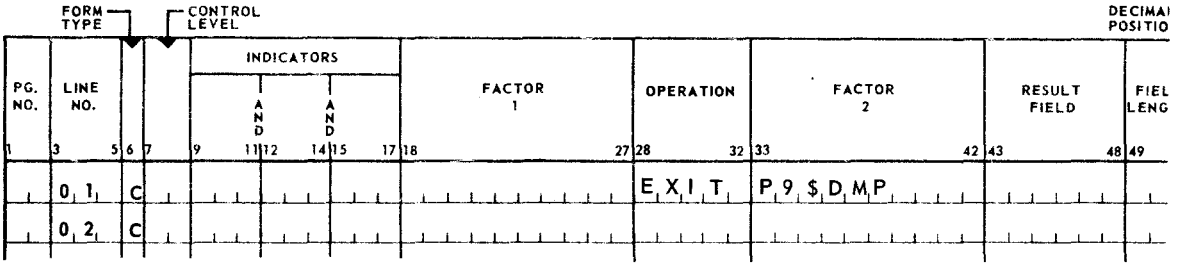
Figure E–2. Example of RPG Control Card

E5. HALT INDICATOR PROCESSING

Normally, when any halt indicator (H0 through H9) is set, the RPG program is cancelled. At the user's option, this processing can be altered to do any of the following:

- End job step without cancelling (EOJ)
- Turn off the halt indicator and resume processing (but this does not remove the cause of the halt; and if H0 was set, further processing will probably be erratic).
- Print a halt analysis describing the reason for the halt, the current settings of all indicators, and the current contents of all data fields, then cancel the program.
- Print a system dump, then cancel the program.
- Print a console message permitting the operator to select the desired processing option.

Halt processing occurs after header or detail output time in the RPG processing cycle. Therefore, at compile time, a D or other nonblank character should be placed in column 8 of the RPG header (H) card. If, however, it is necessary to do this processing during calculation time, column 8 of the H card should be blank, and the calculations should include the following operation:



All other operating instructions still apply, except that control is returned to the operation following the EXIT.

Note that column 8 of the H card is ignored if the RPG object module is a subroutine (see Appendix G).

E5.1. SET COMREG Statement

The SET COMREG statement is used to control processing of an RPG program when a halt indicator is set. At execution time, the SET COMREG statement is placed in the job control stream (see Figure E–1) before the EXEC statement. The first ten bytes of the communication region in the job preamble are associated with the ten halt indicators, H0 through H9. Each of these communication bytes should be set to indicate the processing option to be followed when the corresponding halt indicator is set.

The format of the SET COMREG statement for RPG is:

// SET COMREG,C'character-string'

POSITIONAL PARAMETER 1

COMREG – indicates that positional parameter 2 is to be stored in the communications region of the job preamble.

POSITIONAL PARAMETER 2

C'character-string' – may be one or more of the following characters:

C or blank	– normal processing (cancel job)
E	– end job step (EOJ)
I	– turn off indicator and resume processing
D	– print halt analysis and cancel job
S	– print system dump and cancel job
M	– print console message

Additional Information:

When the D option is specified, a DVC and LFD statement for a print file with the name PRNTR must be present in the job control deck.

If the M option is specified, the following message is printed on the console when the halt indicator is processed:

PG05 HALT IND SET REPLY C,D,E,I Hn_1 Hn_2 ...

Also, where Hn_1, Hn_2, ... are the halt indicators currently set, the reply to this message is:

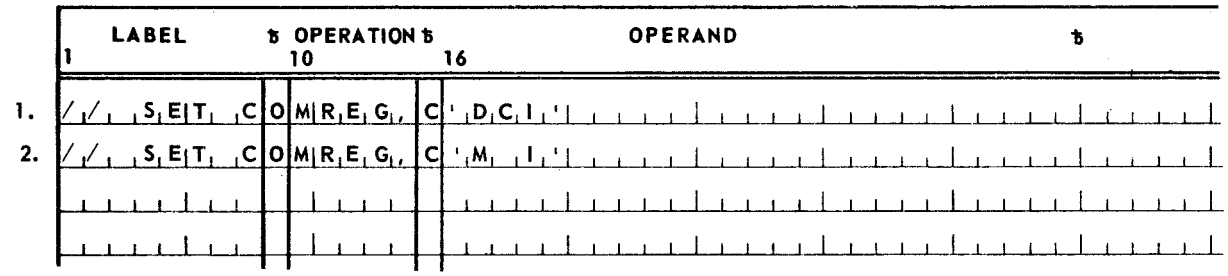
nnR character-stringEOM

where nn is the job number; C,D,E, and I are the options described above for character-string; and EOM is the end-of-message character. The D option is usable only if a print file (PRNTR) has been defined in the job control stream. If any option other than M is specified, the following message is printed on the console when the halt indicator is processed:

PG05 HALT IND Hn_1 Hn_2 ...

Regardless of the option character specified, table output (if any), is done and all files are closed before the program is terminated. For the user to suppress table output and file closing, he must specify the M character, then cancel or dump the job using the standard CANCEL or DUMP command when the console message appears.

Examples:



1. This statement would print a halt analysis and resume printing if H0 were set, cancel the job if H1 were set, and ignore H2.

2. This statement would print a console message (PG05) if H0 were set, cancel the job if H1 were set, and ignore H2.

E6. VARIABLE-LENGTH RECORDS

When specifying block length and record length, the block length must be equal to or greater than the largest record length specified. The record length entry must be the largest possible record that can be contained in any block. RPG constructs a block length and record length larger than the lengths specified by generating a four-byte block length field before each block constructed and a four-byte record length before each record within a block.

If the variable-length records are input to another RPG program, these length fields can be ignored by the user. The block length and record length will be the same as those in the output program.

If the variable-length records are input to a program other than RPG, the block length and record length fields must be taken into consideration. The calculation to determine the largest block generated by RPG is as follows:

$$\frac{(4 \times \text{block length})}{\text{record length}} + \text{block length} + 24$$

E7. PRINTING LINE COUNTER TAPE/DISC FILES

The utility program UTRPG is available for printing tape and disc report files stored by use of the line counter feature of RPG (see Section 7). A maximum of eight tape and/or disc files (one report to each file) may be printed in one run. The files are printed in their entirety, one file at a time. All output is directed to a single printer.

- Input requirements

  Tape files must have standard labels. Filenames on LFD statements for tape files are TREPT1 through TREPT8; for disc files, filenames on LFD statements for disc files are DREPT1 through DREPT8.

- Output requirements

  An LFD statement with the name PRNTR is required for a print file.

■ Sample control stream

LABEL	OPERATION	OPERAND	COMMENTS
// JOB	PRPG		
// DVC	3	// LFD PRNTR	
// DVC	6	// VOL 000001	// LBL (label-info) // LFD TREPT1
// DVC	7	// VOL 000002	// LBL (label-info) // LFD TREPT2
// DVC	21	// VOL 000003	// LBL (label-info) // LFD DREPT1
// DVC	21	// VOL 000003	// LBL (label-info) // LFD DREPT2
// DVC	21	// VOL 000003	// LBL (label-info) // LFD DREPT3
// EXEC	UTRPG		
/&			

With this control stream, five reports are printed: first the tape file TREPT1 on device 6, then a second tape file TREPT2 on device 7, then three disc files, DREPT1, DREPT2, and DREPT3, on the same disc pack 000003. The various files do not have to be defined in any particular order. Regardless of the specified order, all tape files are printed before all disc files, and both tape and disc files are printed in ascending numerical order.

E8. TABLE ENTRY AT EXECUTION TIME

Table and data cards entered through the card reader must be placed in a specific order because their images are double buffered. When table and data files are opened (initialized), they are opened in the order in which the File Description cards are entered; as each file is opened, the first card of that file is read into main storage. Thus,double buffering is used during actual execution to provide a standby image.

The following two examples show the specific order in which table and data cards are to be entered. In the first example, each table is entered as a separate file. In the second, all three tables are entered as one file. Each example enters one data file. Note that cards are placed directly in the reader for DOS and following the /$ statement for TOS.

■ Example 1:

PROGRAM _____ PROGRAMMER _____

PG. NO.	LINE NO.	FORM TYPE	FILE NAME	FILE DESIGNATION / FILE TYPE	END OF FILE / SEQUENCE	FORMAT	BLOCK LENGTH	RECORD LENGTH	MODE	RA TYPE / RA FIELD LENGTH	OVERFLOW INDICATOR	FILE ORGANIZATION / EXTENSION / KEY FIELD START LOCATION	E	DEVICE	LOGICAL UNIT NUMBER
	01	F	TBLX	I	T								E	READER	
	02	F	TBLY	I	T								E	READER	
	03	F	TBLZ	I	T								E	READER	
	04	F	AA	I	PEA	F								READER	

FORM TYPE — CHAINING FIELD NUMBER

PG. NO. 1	LINE NO. 3 5	FORM TYPE 6	RECORD SEQUENCE† 7 8	CHAINING FIELD NUMBER 9 10	FROM FILE NAME 11 18	TO FILE NAME 19 26	TABLE NAME 27 32	NUMBER OF TABLE ENTRIES PER RECORD 33 35	NUMBER OF TABLE ENTRIES PER TABLE 36 39	LENGTH OF TABLE ENTRY 40 42	PACKED 43	NUMERIC SEQUENCE 44 45	TABLE NAME 46
	01	E			TBLX		XX	3	10	2		0A	
	02	E			TBLY		YY	2		9		A	
	03	E			TBLZ		ZZ	1	15	7		A	

Three separate table files named TBLX, TBLY, and TBLZ and data file AA are described on the file description form. TBLX, TBLY, and TBLZ are defined on the File Extension Specifications form as Table XX, Table YY, and Table ZZ, respectively. Arrangement of these table and data cards for entry into the reader at execution time is:

```
    First card of Table XX
    First card of Table YY
    First card of Table ZZ          } Initialize files
    First card of data file AA
    Remaining cards of Table XX
/*  (card indicates end of file)
    Remaining cards of Table YY
/*
    Remaining cards of Table ZZ
/*
    Remaining cards of data file AA
/*
```

■ Example 2:

PROGRAM _____ PROGRAMMER _____

FORM TYPE — FILE DESIGNATION FILE TYPE — END OF FILE SEQUENCE — MODE — RA TYPE — FILE ORGANIZATION EXTENSION — LABE

PG. NO. 1	LINE NO. 3 5	FORM TYPE 6	FILE NAME 7 14	FILE DESIGNATION FILE TYPE 15	MODE 16	END OF FILE SEQUENCE 17	FORMAT 18	BLOCK LENGTH 19 23	RECORD LENGTH 24 27	RA TYPE 28	RA FIELD LENGTH 29 30	FILE ORGANIZATION EXTENSION 31	OVERFLOW INDICATOR 32 33	KEY FIELD START LOCATION 34 38	E 39	DEVICE 40 46	LOGICAL UNIT NUMBER 47
	01	F	TBLS	I	T										E	READER	
	02	F	AA	I	P	E	A	F								READER	

FORM TYPE — CHAINING FIELD NUMBER

PG. NO. 1	LINE NO. 3 5	FORM TYPE 6	RECORD SEQUENCE† 7 8	CHAINING FIELD NUMBER 9 10	FROM FILE NAME 11 18	TO FILE NAME 19 26	TABLE NAME 27 32	NUMBER OF TABLE ENTRIES PER RECORD 33 35	NUMBER OF TABLE ENTRIES PER TABLE 36 39	LENGTH OF TABLE ENTRY 40 42	PACKED 43	NUMERIC SEQUENCE 44 45	TABLE NAME 46
	01	E			TBLS		XX	3	10	2		0A	
	02	E			TBLS		YY	2	5	9		A	
	03	E			TBLS		ZZ	1	15	7		A	

Table file TBLS and data file AA are described on the File Description form. TBLS contains three tables: Table XX, Table YY, and Table ZZ. Each table is defined on the File Extension Specifications form. Arrangement of these table and data cards is as follows:

```
First card of Table XX      }
First card of data file AA  }  Initialize files
Remaining cards of Table XX
All cards of Table YY
All cards of Table ZZ
/*
Remaining cards of data file AA
/*
```

E9. RPG CONSOLE MESSAGES

The following messages may be printed at the console during the running of the RPG program:

PG01 FATAL IO ERROR

Error in work or output file. RPG compiler aborted.

PG02 PRNTR FILE NOT FOUND

No LFD statement in the control stream defining the PRNTR file (see E3). RPG compiler aborted.

PG03 COMPILER FILES CANNOT BE OPENED

Error detected while opening work files or byte count error in source library. RPG aborted.

PG04 CANNOT LOCATE SOURCE PROGRAM

Source library not found on specified tape or disc file. RPG compiler aborted.

PG05 HALT IND SET

Specified halt indicator(s) set in user program; resulting action depends on halt indicator processing action taken by programmer (see E4).

APPENDIX F. RPG PROGRAM DIAGNOSTICS LISTING

F1. ERROR MESSAGES

The following RPG error messages are listed in order by code number.

NOTE 000 END OF FILE HAS BEEN ENCOUNTERED IN INPUT STREAM PRIOR TO AN OUTPUT SPECIFICATION. EXECUTION IS DELETED.

NOTE 001 STERLING INPUT SPECIFICATION (COLUMNS 17-18) IS INVALID. HOLLERITH IS ASSUMED.

NOTE 002 STERLING OUTPUT SPECIFICATION (COLUMNS 19-20) IS INVALID. HOLLERITH IS ASSUMED.

NOTE 003 INVERTED PRINT ENTRY (COLUMN 21) IS INVALID. ENTRY OF I IS ASSUMED.

NOTE 004 RPG CONTROL CARD IS MISSING. COMPILATION IS BYPASSED.

NOTE 005 MAIN OR SUBROUTINE CHARACTER (COLUMN 74) IS INVALID. MAIN IS ASSUMED.

NOTE 006 FILE TYPE (COLUMN 15) IS INVALID. SPECIFICATION IS NOT PROCESSED.

NOTE 007 INVALID ENTRY IN COLUMNS 28, 31, OR 32. SPECIFICATION IS NOT PROCESSED.

NOTE 008 RECORD ADDRESS FIELD (COLUMNS 29-30) HAS INVALID LENGTH, IS MISSING, OR IS NOT RIGHT-JUSTIFIED. ENTRY OF 08 IS AS - SUMED FOR RAF FILE. ENTRY OF 10 IS ASSUMED FOR TAG FILE.

NOTE 009 MORE THAN ONE RECORD ADDRESS FILE IS PRESENT. SUCCEEDING ONES ARE NOT PROCESSED.

NOTE 010 EXTENSION CODE (COLUMN 39) IS INVALID. ENTRY OF BLANK IS ASSUMED.

NOTE 011 INPUT FILE DESIGNATION (COLUMN 16) IS INVALID OR MISSING. ENTRY OF R IS ASSUMED FOR TAG FILE. OTHERWISE S IS ASSUMED.

NOTE 012 OVERFLOW INDICATOR (COLUMN 33) IS NOT O. ENTRY OF O IS ASSUMED.

NOTE 013 OVERFLOW INDICATOR (COLUMNS 33-34) IS INVALID. SPECIFICATION IS NOT PROCESSED.

NOTE 014 MORE THAN ONE PRIMARY FILE IS SPECIFIED. FILE IS ASSUMED TO BE A SECONDARY FILE.

NOTE 015 MODE OF PROCESSING (COLUMN 28) IS INVALID. ENTRY OF R IS ASSUMED.

NOTE 016 FIXED FORMAT IS SPECIFIED, BUT BLOCK LENGTH IS NOT A MULTIPLE OF RECORD LENGTH. BLOCK LENGTH IS INCREASED TO NEXT HIGHEST MULTIPLE.

NOTE 017 TYPE OF FILE ORGANIZATION (COLUMN 32) IS NOT BLANK. ENTRY OF BLANK IS ASSUMED.

NOTE 018 END-OF-FILE CODE (COLUMN 17) IS INVALID. ENTRY OF BLANK IS ASSUMED.

NOTE 019 SEQUENCE (COLUMN 18) IS INVALID. ENTRY OF BLANK IS ASSUMED.

NOTE 020 MODE OF PROCESSING (COLUMN 28) IS NOT BLANK. ENTRY OF BLANK IS ASSUMED.

NOTE 021 RECORD ADDRESS TYPE (COLUMN 31) IS NOT BLANK. ENTRY OF BLANK IS ASSUMED.

NOTE 022 EXTENSION CODE (COLUMN 39) IS INVALID. ENTRY OF E IS ASSUMED.

NOTE 023 WARNING – FORMAT (COLUMN 19) IS NOT F OR V. ENTRY OF F IS ASSUMED FOR INDEX SEQUENTIAL. OTHERWISE ENTRY OF V IS ASSUMED.

NOTE 024 BLOCK LENGTH (COLUMNS 20-23) IS MISSING, INVALID, LESS THAN THE RECORD LENGTH, OR NOT RIGHT-JUSTIFIED. BLOCK LENGTH IS ASSUMED EQUAL TO RECORD LENGTH.

NOTE 025 RECORD LENGTH (COLUMNS 24-27) IS MISSING, INVALID, OR NOT RIGHT-JUSTIFIED. RECORD LENGTH OF 0080 IS ASSUMED.

NOTE 026 FILENAME (COLUMNS 7-14) IS MISSING, INVALID, OR NOT LEFT-JUSTIFIED. SPECIFICATION IS NOT PROCESSED.

NOTE 027 OUTPUT FILE DESIGNATION (COLUMN 16) IS NOT BLANK. ENTRY OF BLANK IS ASSUMED.

NOTE 028 PROGRAM EXCEEDS LIMIT OF TEN VALID FILE NAMES. ADDITIONAL FILE DESCRIPTION SPECIFICATIONS ARE NOT PROCESSED.

NOTE 029 KEY FIELD STARTING LOCATION (COLUMNS 35-38) IS INVALID, NOT RIGHT-JUSTIFIED, OR NOT LESS THAN RECORD LENGTH. ENTRY OF 0001 IS ASSUMED.

NOTE 030 DEVICE (COLUMNS 40-46) IS INVALID. SPECIFICATION IS NOT PROCESSED.

NOTE 031 'LABELS' (COLUMN 53) IS INVALID. ENTRY OF S IS ASSUMED FOR TAPE AND DISC FILES. ENTRY OF BLANK IS ASSUMED FOR CARD AND PRINTER FILES.

NOTE 032 NAME OF LABEL EXIT (COLUMNS 54-59) IS MISSING, INVALID, OR NOT LEFT-JUSTIFIED. SPECIFICATION IS NOT PROCESSED.

NOTE 033 NAME OF LABEL EXIT (COLUMNS 54-59) MUST BE BLANK FOR DEVICES OTHER THAN TAPE OR DISC. ENTRY OF BLANKS IS ASSUMED.

NOTE 034 OVERFLOW INDICATOR (COLUMNS 33-34) IS APPLICABLE TO OUTPUT FILES ONLY. ENTRY OF BLANKS IS ASSUMED.

NOTE 035 FILE ADDITION ENTRY FOR INDEX SEQUENTIAL (COLUMN 66) IS INVALID. ENTRY OF A IS ASSUMED.

NOTE 036 KEY LENGTH (COLUMNS 29-30) IS INVALID OR MISSING. SPECIFICATION IS NOT PROCESSED.

NOTE 037 'NUMBER OF TRACKS FOR CYLINDER OVERFLOW' ENTRY FOR INDEX SEQUENTIAL (COLUMN 67) IS INVALID. ENTRY OF ZERO IS ASSUMED.

NOTE 038 INVALID REWIND OPTION FOR A TAPE FILE. ENTRY OF U IS ASSUMED.

NOTE 039 FORM TYPE (COLUMN 6) IS INVALID OR OUT OF SEQUENCE. SPECIFICATION IS NOT PROCESSED.

NOTE 040 FILE DESCRIPTION SPECIFICATIONS ARE MISSING. EXECUTION IS DELETED.

NOTE 041 WARNING – PRIMARY FILE IS NOT SPECIFIED.

NOTE 042 WARNING – FILE EXTENSION SPECIFICATION IS MISSING.

NOTE 043 FILENAME (COLUMNS 7-14) IS MULTI-DEFINED. SPECIFICATION IS NOT PROCESSED.

NOTE 044 WARNING – LENGTH OF RAF FIELD (COLUMNS 29-30) MUST BE 10. ENTRY OF 10 IS ASSUMED.

NOTE 045 DIRECT FILE CANNOT BE SPECIFIED. SPECIFICATION IS NOT PROCESSED.

NOTE 046 KEY FIELD LOCATION IS INVALID. SPECIFICATION IS NOT PROCESSED.

NOTE 047 COLUMNS 66-67 MUST BE BLANK. ENTRY OF BLANKS IS ASSUMED.

NOTE 048 RECORD LENGTH (COLUMNS 24-27) IS INVALID. ENTRY OF 132 IS ASSUMED.

NOTE 049 MORE THAN ONE RECORD ADDRESS FILE IS SPECIFIED ON FILE EXTENSION SPECIFICATION. SPECIFICATION IS NOT PROCESSED.

NOTE 050 'FROM FILENAME' (COLUMNS 11-18) IS NOT SPECIFIED AS ON FILE DESCRIPTION SPECIFICATION. SPECIFICATION IS NOT PROCESSED.

NOTE 051 EXTENSION CODE (COLUMN 39) OF FILE DESCRIPTION SPECIFICATION IS NOT E. SPECIFICATION IS NOT PROCESSED.

NOTE 052 LENGTH OF TABLE ENTRY (COLUMNS 40-42 OR 52-54) EXCEEDS 256 CHARACTERS FOR AN ALPHANUMERIC FIELD. ENTRY OF 256 IS ASSUMED.

NOTE 053 CHAINING FIELD (COLUMNS 9-10) IS MISSING, INVALID, OR NOT RIGHT-JUSTIFIED. SPECIFICATION IS NOT PROCESSED.

NOTE 054 NUMBER OF FIELD NAMES EXCEEDS ALLOCATED MEMORY STORAGE. ADDITIONAL SPECIFICATIONS CONTAINING TABLE NAMES WILL NOT BE PROCESSED.

NOTE 055 'TO FILENAME' (COLUMNS 19-26) IS NOT SPECIFIED AS ON FILE DESCRIPTION SPECIFICATION. SPECIFICATION IS NOT PROCESSED.

NOTE 056 'TO FILENAME' (COLUMNS 19-26) IS NOT SPECIFIED AS A CHAINED FILE ON FILE DESCRIPTION SPECIFICATION. SPECIFICATION IS NOT PROCESSED.

NOTE 057 LENGTH OF TABLE ENTRY (COLUMNS 40-42 OR 52-54) EXCEEDS 15 DIGITS FOR A NUMERIC FIELD. ENTRY OF 15 IS ASSUMED.

NOTE 058 'TO FILENAME' (COLUMNS 19-26) IS NOT SPECIFIED AS A PRIMARY OR SECONDARY FILE ON FILE DESCRIPTION SPECIFICATION. SPECIFICATION IS NOT PROCESSED.

NOTE 059 TABLE SEQUENCE (COLUMNS 45 OR 57) IS INVALID. ENTRY OF BLANK IS ASSUMED.

NOTE 060 TABLE NAME (COLUMNS 27-32 OR 46-51) IS MULTI-DEFINED. SPECIFICATION IS NOT PROCESSED.

NOTE 061 'TO FILENAME' (COLUMNS 19-26) IS NOT SPECIFIED AS ON FILE DESCRIPTION SPECIFICATION. ENTRY OF BLANKS IS ASSUMED.

NOTE 062 'TO FILENAME' (COLUMNS 19-26) IS NOT SPECIFIED AS AN OUTPUT FILE ON FILE DESCRIPTION SPECIFICATION. ENTRY OF BLANKS IS ASSUMED.

NOTE 063 TABLE NAME (COLUMNS 27-32 OR 46-51) IS MISSING, INVALID, OR NOT LEFT-JUSTIFIED. SPECIFICATION IS NOT PROCESSED.

NOTE 064 FIRST THREE CHARACTERS OF TABLE NAME (COLUMNS 27-29 OR 46-48) ARE NOT 'TAB'. ENTRY OF 'TAB' IS ASSUMED.

NOTE 065 NUMBER OF TABLE ENTRIES PER RECORD (COLUMNS 33-35) IS MISSING, INVALID, OR NOT RIGHT-JUSTIFIED. ENTRY OF 008 IS ASSUMED.

NOTE 066 NUMBER OF TABLE ENTRIES PER TABLE (COLUMNS 36-39) IS MISSING, INVALID, OR NOT RIGHT-JUSTIFIED. ENTRY OF 0150 IS ASSUMED.

NOTE 067 'LENGTH OF TABLE' ENTRY (COLUMNS 40-42 OR 52-54) IS MISSING, INVALID, OR NOT RIGHT-JUSTIFIED. ENTRY OF 010 IS ASSUMED.

NOTE 068 'PACKED' (COLUMN 43 OR 55) IS INVALID. ENTRY OF BLANK IS ASSUMED.

NOTE 069 'DECIMAL POSITIONS' (COLUMN 44 OR 56) IS INVALID. ENTRY OF ZERO IS ASSUMED.

NOTE 070 RECORD SEQUENCE OF THE CHAINING FILE (COLUMNS 7-8) IS INVALID. BOTH POSITIONS MUST BE EITHER NUMERIC OR ALPHABETIC. SPECIFICATION IS NOT PROCESSED.

NOTE 071 FILENAME (COLUMNS 7-16) IS NOT SPECIFIED AS ON FILE DESCRIPTION SPECIFICATION. SPECIFICATION IS NOT PROCESSED.

NOTE 072 OVERFLOW OR HOME PAPER CHANNEL IS MISSING OR INVALID. SPECIFICATION IS NOT PROCESSED.

NOTE 073 FILENAME IS NOT SPECIFIED AS AN OUTPUT FILE OR AN OUTPUT FILE REQUIRING A LINE COUNTER SPECIFICATION. SPECIFICATION IS NOT PROCESSED.

NOTE 074 LINE NUMBER OR CHANNEL NUMBER IS INVALID OR MISSING. SPECIFICATION IS NOT PROCESSED.

NOTE 075 CHANNEL NUMBER IS MULTI-DEFINED. SPECIFICATION IS NOT PROCESSED.

NOTE 076 LINE COUNTER FILE NAME (COLUMNS 7-14) IS MISSING, INVALID, OR NOT LEFT-JUSTIFIED. SPECIFICATION IS NOT PROCESSED.

NOTE 077 'FROM FILE NAME' (COLUMNS 11-18) IS MISSING. SPECIFICATION IS NOT PROCESSED.

NOTE 078 'FROM FILE NAME' (COLUMNS 11-18) IS INVALID OR NOT LEFT-JUSTIFIED. SPECIFICATION IS NOT PROCESSED.

NOTE 079 'TO FILE NAME' (COLUMNS 19-26) IS INVALID OR NOT LEFT-JUSTIFIED. SPECIFICATION IS NOT PROCESSED.

NOTE 080 THERE ARE NO VALID INPUT SPECIFICATIONS IN THIS PROGRAM. EXECUTION IS DELETED.

NOTE 081 FIELD INDICATOR IS SPECIFIED BUT IS NOT VALID. INDICATOR IS NOT PROCESSED.

NOTE 082 INDICATOR L0 IS SPECIFIED AS A FIELD INDICATOR, BUT IS NOT ALLOWED. INDICATOR IS IGNORED.

NOTE 083 FIELD-RECORD RELATION (COLUMNS 63-64) IS INVALID. ENTRY OF 00 IS ASSUMED.

NOTE 084 'POSITION' (COLUMNS 21-24, 28-31, OR 35-38), 'START' (COLUMNS 44-47), 'END' (COLUMNS 48-51), OR 'STERLING SIGN POSITION' (COLUMNS 71-74) IS ZERO. ENTRY OF 1 IS ASSUMED.

NOTE 085 IF HOLLERITH SHILLING IS SPECIFIED, STERLING INPUT FIELD MUST HAVE MORE THAN THREE NON-DECIMAL POSITIONS. IF BSI SHILLING IS SPECIFIED, STERLING INPUT FIELD MUST HAVE MORE THAN TWO NON-DECIMAL POSITIONS. VALUE OF FIELD IS ASSUMED TO BE ZERO.

NOTE 086 WARNING – INDICATOR 00 SHOULD BE USED ONLY IN OUTPUT SPECIFICATIONS.

NOTE 087 COLUMNS 7-42 OF FIELD DESCRIPTION SPECIFICATION SHOULD CONTAIN BLANKS. ENTRY OF BLANKS IS ASSUMED.

NOTE 088 FORM TYPE (COLUMN 6) IS NOT I, C, O, OR T, AND COLUMN 7 IS NOT AN ASTERISK. SPECIFICATION IS NOT PROCESSED.

NOTE 089 UNDEFINED FILENAME (COLUMNS 7-14). SPECIFICATION IS NOT PROCESSED.

NOTE 090 FILENAME (COLUMNS 7-14) HAS BEEN DEFINED AS A TABLE, RAF, TAG, OR OUTPUT FILE. SPECIFICATION IS NOT PROCESSED.

NOTE 091 'AND' SPECIFICATION (COLUMNS 14-16) IS OUT OF SEQUENCE – I.E., FIRST INPUT SPECIFICATION OR FOLLOWS FIELD NAME OR FOLLOWS INVALID 'OR', 'AND', OR FILE NAME. SPECIFICATION IS NOT PROCESSED.

NOTE 092 NO RECORD IDENTIFICATION CODES (COLUMNS 21-41) IN THE CARD BEFORE AN 'AND' CARD. SPECIFICATION IS NOT PROCESSED.

NOTE 093 'OR' SPECIFICATION (COLUMNS 14-15) IS OUT OF SEQUENCE – I.E., FIRST INPUT SPECIFICATION OR FOLLOWS FIELD NAME OR FOLLOWS INVALID 'OR', 'AND', OR FILE NAME. SPECIFICATION IS NOT PROCESSED.

NOTE 094 RECORD IDENTIFICATION FOLLOWS AN INVALID FILE TYPE SPECIFICATION. SPECIFICATION IS NOT PROCESSED.

NOTE 095 FILE AND FIELD NAMES ARE BOTH PRESENT ON THE SAME SPECIFICATION. FILENAME IS ASSUMED.

NOTE 096 SEQUENCE (COLUMNS 15-16) IS BLANK. ENTRY OF AA IS ASSUMED.

NOTE 097 ALPHABETIC SEQUENCE FOUND AFTER NUMERIC SEQUENCE. NUMERIC SEQUENCE EQUAL TO PREVIOUS NUMERIC SEQUENCE IS ASSUMED.

NOTE 098 NUMERIC SEQUENCE NOT CONSECUTIVE IN FILE BEGINNING WITH 01. EXECUTION IS DELETED.

NOTE 099 NUMBER (COLUMN 17) IS NOT N OR 1 FOR A NUMERIC SEQUENCE. ENTRY OF N IS ASSUMED.

NOTE 100 OPTION (COLUMN 18) IS NOT 0 OR BLANK. ENTRY OF 0 IS ASSUMED.

NOTE 101 RESULTING INDICATOR (COLUMNS 19-20) IS BLANK OR INVALID. INDICATOR OF 99 IS ASSUMED.

NOTE 102 WARNING – INVALID OR UNSUPPORTED STACKER SELECT ENTRY (COLUMN 42). ENTRY OF BLANK IS ASSUMED.

NOTE 103 'NOT' (COLUMNS 25, 32, OR 39) IS NOT N OR BLANK. ENTRY OF N IS ASSUMED.

NOTE 104 'C/Z/D' (COLUMNS 26, 33, OR 40) IS NOT C, Z, OR D. ENTRY OF C IS ASSUMED.

NOTE 105 RECORD IDENTIFICATION IS OUT OF SEQUENCE – I.E., FIRST INPUT SPECIFICATION OR FOLLOWING AN INVALID 'OR', 'AND', OR FILE NAME. SPECIFICATION IS NOT PROCESSED.

NOTE 106 'START' (COLUMNS 44-47) OR 'END' (COLUMNS 48-51) IS BLANK. ENTRY OF 1 IS ASSUMED.

NOTE 107 'START' (COLUMNS 44-47) IS GREATER THAN 'END' (COLUMNS 48-51). 'END' IS ASSUMED TO HAVE A VALUE EQUAL TO 'START'.

NOTE 108 DECIMAL POSITION (COLUMN 52) IS NOT NUMERIC. ENTRY OF ZERO IS ASSUMED.

NOTE 109 UNPACKED NUMERIC FIELD IS MORE THAN 15 BYTES LONG. LENGTH OF 15 IS ASSUMED.

NOTE 110 STERLING FIELD IS INDICATED WITH MORE THAN THREE DECIMAL POSITIONS. THE DECIMAL PORTION OF THE FIELD IS TRUNCATED TO THREE POSITIONS. THE 'END' POSITION OF THE FIELD IS ALTERED TO ALLOW THIS TRUNCATION.

NOTE 111 'PACKED' (COLUMN 43) IS NEITHER P NOR BLANK. ENTRY OF P IS ASSUMED.

NOTE 112 ALPHANUMERIC FIELD LENGTH IS MORE THAN 256 BYTES LONG. LENGTH OF 256 IS ASSUMED.

NOTE 113 NUMERIC RESULTING INDICATOR (COLUMNS 65-66 OR 67-68) IS SPECIFIED FOR AN ALPHANUMERIC FIELD. INDICATOR IS SET OFF.

NOTE 114 IMPROPER ENTRY FOR STERLING SIGN POSITION (COLUMNS 71-74). 'END' FIELD LOCATION ASSUMED.

NOTE 115 STERLING INPUT SPECIFIED (COLUMNS 71-74), BUT NOT INDICATED ON PROCESSOR CONTROL CARD. ENTRY OF BLANKS IS ASSUMED.

NOTE 116 CALCULATED LENGTH OF STERLING FIELD IS GREATER THAN 15 BYTES. LENGTH OF 15 IS ASSUMED.

NOTE 117 'AND' TYPE OR 'OR' TYPE SPECIFICATION DOES NOT HAVE ALL BLANKS IN COLUMNS 43-74. ENTRY OF BLANKS IS ASSUMED.

NOTE 118 'AND' TYPE INPUT SPECIFICATION DOES NOT LEAVE BLANKS IN COLUMNS 17-20 AND 42. ENTRY OF BLANK IS ASSUMED.

NOTE 119 FILENAME HAS BEEN REFERENCED PREVIOUSLY IN INPUT SPECIFICATIONS. SPECIFICATION IS NOT PROCESSED.

NOTE 120 CONTROL LEVEL IS SPECIFIED BUT COLUMN 59 IS NOT L. ENTRY OF L IS ASSUMED.

NOTE 121 CONTROL LEVEL IS SPECIFIED BUT COLUMN 60 IS NOT 1-9. ENTRY OF 1 IS ASSUMED.

NOTE 122 MATCHING OR CHAINING FIELD IS SPECIFIED BUT COLUMN 62 IS NOT 1-9. ENTRY OF 1 IS ASSUMED.

NOTE 123 MATCHING OR CHAINING FIELD IS SPECIFIED BUT COLUMN 61 IS NOT M OR C. ENTRY OF M IS ASSUMED.

NOTE 124 'PACKED' (COLUMN 43) IS NOT BLANK, BUT AN ALPHANUMERIC OR STERLING IS SPECIFIED. ENTRY OF BLANK IS ASSUMED.

NOTE 125 SEQUENCE CHARACTERS (COLUMNS 15-16) ARE NOT BOTH ALPHABETIC OR BOTH NUMERIC. ENTRY OF AA IS ASSUMED.

NOTE 126 'POSITION' (COLUMNS 21-24, 28-31, OR 35-38) IS INVALID OR NOT RIGHT-JUSTIFIED. ENTRY OF 0001 IS ASSUMED.

NOTE 127 FILE NAME (COLUMNS 7-14) IS INVALID OR NOT LEFT-JUSTIFIED. SPECIFICATION IS NOT PROCESSED.

NOTE 128 'START' (COLUMNS 44-47) OR 'END' (COLUMNS 48-51) IS INVALID OR NOT RIGHT-JUSTIFIED. ENTRY OF 0001 IS ASSUMED.

NOTE 129 PACKED NUMERIC FIELD IS MORE THAN 8 BYTES LONG. LENGTH OF 8 IS ASSUMED.

NOTE 130 FIELD NAME (COLUMNS 53-58) IS INVALID OR NOT LEFT-JUSTIFIED. SPECIFICATION IS NOT PROCESSED.

NOTE 131 STERLING SIGN POSITION (COLUMNS 71-74) IS SPECIFIED FOR ALPHANUMERIC FIELD. STERLING SIGN POSITION ENTRY OF BLANKS IS ASSUMED.

NOTE 132 WARNING – FILENAME (COLUMNS 7-14) IS DEFINED BUT NEVER USED.

NOTE 133 UNDEFINED TABLE SPECIFIED IN LOKUP OPERATION. SPECIFICATION IS NOT PROCESSED.

NOTE 134 DECIMAL POSITION IS INVALID. ENTRY OF ZERO IS ASSUMED FOR NUMERIC FIELD. ENTRY OF BLANK IS ASSUMED FOR ALPHANUMERIC FIELD.

NOTE 135 FIELD LENGTH IS IMPROPERLY SPECIFIED OR IS NOT SPECIFIED. ENTRY OF ZERO IS ASSUMED FOR INVALID CHARACTER. WHEN REQUIRED LENGTH IS NOT SPECIFIED, ENTRY OF 4 IS ASSUMED.

NOTE 136 OPERATION CODE (COLUMNS 28-32) IS INVALID OR MISSING. SPECIFICATION IS NOT PROCESSED.

NOTE 137 REQUIRED ENTRY IN FACTOR 1 (COLUMNS 18-27) IS MISSING OR INVALID. SPECIFICATION IS NOT PROCESSED.

NOTE 138 REQUIRED ENTRY IN FACTOR 2 (COLUMNS 33-42) IS MISSING OR INVALID. SPECIFICATION IS NOT PROCESSED.

NOTE 139 REQUIRED ENTRY IN RESULT FIELD (COLUMNS 43-48) IS MISSING OR INVALID. SPECIFICATION IS NOT PROCESSED.

NOTE 140 FORM TYPE (COLUMN 6) IS INVALID. SPECIFICATION IS NOT PROCESSED.

NOTE 141 'NOT' (COLUMNS 9, 12, OR 15) IS NOT N OR BLANK. ENTRY OF N IS ASSUMED.

NOTE 142 CONTROL LEVEL IS IMPROPERLY SPECIFIED. ENTRY OF L0 IS ASSUMED.

NOTE 143 RESULTING INDICATOR IS INVALID. INDICATOR IS NOT PROCESSED.

NOTE 144 'HALF ADJUST' ENTRY (COLUMN 53) IS INVALID. ENTRY OF H IS ASSUMED.

NOTE 145 FIELD NAME IS IMPROPERLY USED. SPECIFICATION IS NOT PROCESSED.

NOTE 146 INDICATOR (COLUMNS 10-11, 13-14, OR 16-17) IS INVALID. ENTRY OF 00 IS ASSUMED.

NOTE 147 REQUIRED RESULTING INDICATOR (COLUMNS 54-55, 56-57, OR 58-59) IS NOT SPECIFIED. SPECIFICATION IS NOT PROCESSED.

NOTE 148 'MVR' DOES NOT FOLLOW 'DIV', OR FOLLOWS A 'DIV' WITH HALF ADJUST SPECIFIED. SPECIFICATION IS NOT PROCESSED.

NOTE 149 OPERATION CODE IS NOT SUPPORTED BY THIS VERSION OF RPG.

NOTE 150 DETAIL CALCULATION SPECIFICATION FOLLOWS A TOTAL CALCULATION SPECIFICATION. DETAIL SPECIFICATION IS NOT PROCESSED.

NOTE 151 RESULT FIELD LENGTH (COLUMNS 49-51) IS GREATER THAN ALLOWED. A LENGTH OF 256 IS ASSUMED FOR AN ALPHANUMERIC FIELD. A LENGTH OF 15 IS ASSUMED FOR A NUMERIC FIELD.

NOTE 152 OPERATION CODE OF TESTZ IS INVALID FOR THE 9300 MODE OF COMPUTATION. SPECIFICATION IS NOT PROCESSED.

NOTE 153 STERLING SPECIFIED IN COLUMNS 71-74, BUT NOT SPECIFIED ON PROCESSOR CONTROL CARD. ENTRY OF BLANKS IS ASSUMED.

NOTE 154 INAPPROPRIATE OUTPUT FIELD. SPECIFICATION IS NOT PROCESSED.

NOTE 155 FILENAME (COLUMNS 7-14) IS MISSING, OR RECORD TYPE (COLUMN 15) IS IN WRONG ORDER. SPECIFICATION IS NOT PROCESSED.

NOTE 156 CORRESPONDING FILENAME (COLUMNS 7-14) CANNOT BE DETERMINED. SPECIFICATION IS NOT PROCESSED.

NOTE 157 'STACKER SELECT' (COLUMN 16) IS INVALID. ENTRY OF BLANK IS ASSUMED.

NOTE 158 'SPACE BEFORE' (COLUMN 17) IS INVALID. ENTRY OF 1 IS ASSUMED.

NOTE 159 'SPACE AFTER' (COLUMN 18) IS INVALID. ENTRY OF 1 IS ASSUMED.

NOTE 160 'SKIP BEFORE' (COLUMNS 19-20) IS INVALID. ENTRY OF 01 IS ASSUMED.

NOTE 161 'SKIP AFTER' (COLUMNS 21-22) IS INVALID. ENTRY OF 01 IS ASSUMED.

NOTE 162 RECORD TYPE (COLUMN 15) IS NOT H, D, OR T. SPECIFICATION IS NOT PROCESSED.

NOTE 163 COLUMNS 17-22 MUST BE BLANK FOR 'AND' TYPE SPECIFICATIONS. ENTRY OF BLANK IS ASSUMED.

NOTE 164 COLUMNS 7-13 MUST BE BLANK FOR 'AND' OR 'OR' TYPE SPECIFICATIONS. ENTRY OF BLANK IS ASSUMED.

NOTE 165 CORRESPONDING RECORD SPECIFICATION IS MISSING OR INVALID. SPECIFICATION IS NOT PROCESSED.

NOTE 166 'ZERO SUPPRESS' (COLUMN 38) IS INVALID. ENTRY OF BLANK IS ASSUMED.

NOTE 167 'PACKED FIELD' (COLUMN 44) IS INVALID. ENTRY OF BLANK IS ASSUMED.

NOTE 168 FIELD NAME (COLUMNS 32-37) IS NOT LEFT-JUSTIFIED. SPECIFICATION IS NOT PROCESSED.

NOTE 169 'END POSITION' (COLUMNS 40-43) IS INVALID OR MISSING. SPECIFICATION IS NOT PROCESSED.

NOTE 170 LEADING OR CLOSING APOSTROPHE (') IN EDIT WORD IS NOT CORRECT. ENTRY OF BLANKS IN COLUMNS 45-70 IS ASSUMED.

NOTE 171 'BLANK AFTER' (COLUMN 39) IS INVALID. ENTRY OF BLANK IS ASSUMED.

NOTE 172 PUNCH AND PRINT FUNCTIONS ARE SPECIFIED FOR THE SAME FILE. ENTRY OF BLANKS IS ASSUMED FOR COLUMNS 17-22.

NOTE 173 ZERO SUPPRESSION (COLUMN 38) MAY NOT BE SPECIFIED FOR CONSTANTS OR EDIT WORDS. ENTRY OF BLANK IN COLUMN 38 IS ASSUMED.

NOTE 174 FIELD NAME (COLUMNS 32-37) IS UNDEFINED. SPECIFICATION IS NOT PROCESSED.

NOTE 175 WARNING–'BLANK AFTER' (COLUMN 39) IS SPECIFIED FOR CONSTANT. ALL IDENTICAL CONSTANTS WILL BE BLANKED.

NOTE 176 CONSTANT (COLUMNS 45-70) IS NOT LEFT-JUSTIFIED. SPECIFICATION IS NOT PROCESSED.

NOTE 177 EDIT WORD (COLUMNS 45-70) IS NOT LEFT-JUSTIFIED. ENTRY OF BLANKS IN COLUMNS 45-70 IS ASSUMED.

NOTE 178 'PACKED FIELD' (COLUMN 44) MAY NOT BE SPECIFIED WITH CONSTANT, EDIT WORD OR STERLING ENTRY. ENTRY OF BLANK IN COLUMN 44 IS ASSUMED.

NOTE 179 FILENAME (COLUMNS 7-14) IS NOT LEFT-JUSTIFIED. SPECIFICATION IS NOT PROCESSED.

NOTE 180 EDIT WORD (COLUMNS 45-70) CONTAINS NO DIGIT POSITIONS OR MORE THAN FIFTEEN (SIXTEEN FOR STERLING). ENTRY OF BLANKS IN COLUMNS 45-70 IS ASSUMED.

NOTE 181 LEADING OR CLOSING APOSTROPHE (') IN CONSTANT IS NOT CORRECT. SPECIFICATION IS NOT PROCESSED.

NOTE 182 'AND' OR 'OR' FOLLOWING A FIELD NAME SPECIFICATION OR AS FIRST OUTPUT SPECIFICATION IS INVALID. SPECIFICATION IS NOT PROCESSED.

NOTE 183 STERLING ENTRY (COLUMNS 71-74) MAY NOT BE SPECIFIED WITH CONSTANT OR PAGE (N). ENTRY OF BLANK IN COLUMNS 71-74 IS ASSUMED.

NOTE 184 STERLING ENTRY (COLUMNS 71-74) IS INVALID. ENTRY OF BLANKS IS ASSUMED.

NOTE 185 OUTPUT INDICATOR (COLUMNS 24-25, 27-28, OR 30-31) IS INVALID OR UNDEFINED. ENTRY OF L0 IS ASSUMED.

NOTE 186 OUTPUT INDICATORS SHOULD START IN COLUMNS 23-25, THEN 26-28, AND FINALLY 29-31. ENTRY IS SHIFTED LEFT.

NOTE 187 'NOT' (COLUMNS 23, 26, OR 29) IS NOT BLANK OR N. ENTRY OF N IS ASSUMED.

NOTE 188 WARNING – OVERFLOW INDICATOR IS SPECIFIED IN 'AND' TYPE SPECIFICATION. RECORD WILL NOT BE PUT OUT AS OVERFLOW LINE.

NOTE 189 DECIMAL POSITIONS MUST BE ZERO FOR PAGE(N) FIELD. ENTRY OF ZERO IS ASSUMED.

NOTE 190 SPECIFICATION TYPE CANNOT BE DETERMINED. RECORD AND FIELD DEFINITIONS ARE SPECIFIED IN SAME LINE OR BOTH ARE BLANK. SPECIFICATION IS NOT PROCESSED.

NOTE 191 FORM TYPE (COLUMN 6) IS INVALID (NOT O) SPECIFICATION IS NOT PROCESSED.

NOTE 192 NO OUTPUT INDICATOR (COLUMNS 24-25, 27-28, OR 30-31) IS SPECIFIED FOR 'AND' OR 'OR' TYPE SPECIFICATION. SPECIFICATION IS NOT PROCESSED.

NOTE 193 HEXADECIMAL CONSTANT (COLUMNS 45-70) CONTAINS INVALID CHARACTER. SPECIFICATION IS NOT PROCESSED.

NOTE 194 LEADING OR CLOSING APOSTROPHE (') IN HEXADECIMAL CONSTANT IS NOT CORRECT. SPECIFICATION IS NOT PROCESSED.

NOTE 195 WARNING: PAGE1 NOT SUPPORTED AS AN AUTOMATIC PAGE COUNTER IF IN 9200/9300 MODE OF PROCESSING.

NOTE 196 WARNING – PAGE2,...,PAGE7 ARE NOT SUPPORTED AS AUTOMATIC PAGE COUNTERS IN 9200/9300 AND MOD 20 MODE OF PROCESSING.

NOTE 197 DECIMAL POSITIONS MUST BE BLANK FOR DATE FIELD. ENTRY OF BLANK IS ASSUMED.

NOTE 198 FIELDNAME (COLUMNS 32-37) IS INVALID. SPECIFICATION IS NOT PROCESSED.

NOTE 199 FILENAME (COLUMNS 7-14) IS INVALID. SPECIFICATION IS NOT PROCESSED.

NOTE 200 WARNING: TABLE FILE MUST BE SUBMITTED AT EXECUTION.

NOTE 201 RESULTING INDICATOR IS INVALID OR UNDEFINED. ENTRY OF L0 IS ASSUMED.

NOTE 202 WARNING – RESULTING INDICATOR IS UNREFERENCED.

NOTE 203 FIELD NAME IS UNDEFINED. FIELD IS PROCESSED WITH ASSUMED LENGTH OF 004.

NOTE 204 WARNING – FIELD NAME IS MULTI-DEFINED.

NOTE 205 WARNING – FIELD NAME IS UNREFERENCED.

NOTE 206 THE COMBINED LENGTHS OF LITERALS AND FIELD NAMES EXCEEDS ALLOCATED MAIN STORAGE.

NOTE 207 WARNING – MULTI-FILE PROGRAM (WITH PRIMARY AND SECONDARY FILES) IS SPECIFIED WITHOUT MATCHING FIELDS OF THE PRIMARY FILE.

NOTE 208 WARNING – MULTI-FILE PROGRAM (WITH PRIMARY AND SECONDARY FILES) IS SPECIFIED WITHOUT MATCHING FIELDS FOR THE SECONDARY FILE(S).

NOTE 209 THE SUM OF THE LENGTHS OF THE MATCHING FIELDS FOR THE PRIMARY FILE DOES NOT EQUAL THAT OF EACH SECONDARY FILE. EXECUTION IS DELETED.

NOTE 210 THE SUM OF THE LENGTHS OF THE MATCHING FIELDS IS NOT CONSTANT IN EACH RECORD WHICH SPECIFIED MATCHING FIELDS FOR A FILE. EXECUTION IS DELETED.

NOTE 211 WARNING – THE SUM OF THE LENGTHS OF THE CONTROL FIELDS IS NOT CONSTANT IN EACH RECORD WHICH SPECIFIED CONTROL FIELDS.

NOTE 212 AN EXCESSIVE AMOUNT OF OBJECT CODE HAS BEEN GENERATED FOR A SINGLE RECORD GROUP. EXECUTION IS DELETED.

NOTE 213 THIS SPECIFICATION CONTAINS AN ASTERISK CONDITION INDICATION AND FOLLOWS A SPECIFICATION THAT HAS BEEN DELETED. SPECIFICATION IS NOT PROCESSED.

NOTE 214 FIELDS USED IN AN ALPHANUMERIC COMPARE MUST BE EQUAL IN LENGTH OR MUST BE LESS THAN OR EQUAL TO 200 BYTES. SPECIFICATION IS NOT PROCESSED.

NOTE 215 FIELD LENGTHS ARE INVALID FOR THIS OPERATION. SPECIFICATION IS NOT PROCESSED.

NOTE 216 PLUS AND/OR MINUS RESULTING INDICATORS (COLUMNS 54-55 OR 56-57) ARE NOT ALLOWED FOR TESTING ALPHANUMERIC FIELDS. INDICATORS ARE IGNORED.

NOTE 217 FIELD TYPE IS INVALID FOR THIS OPERATION. SPECIFICATION IS NOT PROCESSED.

NOTE 218 ENTRY IN COLUMNS 16-18 VALID FOR INDEXED-SEQUENTIAL ADD ONLY. ENTRY IS IGNORED.

NOTE 219 REQUIRED 'ADD' ENTRY (COLUMNS 16-18) IS MISSING. 'ADD' IS ASSUMED.

NOTE 220 FILE SPECIFIED ON OUTPUT FORMAT SPECIFICATIONS IS UNDEFINED OR NOT AN OUTPUT FILE (U,C, O, OR INDEX SEQUENTIAL INPUT WITH ADDED RECORDS). SPECIFICATION IS NOT PROCESSED.

NOTE 221 WARNING -- FILENAME (COLUMNS 7-14) IS NOT REFERENCED ON OUTPUT SPECIFICATIONS.

NOTE 222 NO VALID OUTPUT SPECIFICATIONS ARE PRESENT. EXECUTION IS DELETED.

NOTE 223 ALL OUTPUT LINES OF A PRINTER FILE MUST INDICATE EITHER SPACING AND/OR SKIPPING. SINGLE SPACING IS ASSUMED FOR ALL OUTPUT LINES OF NAMED FILE WHICH HAVE NO PRINT FUNCTION.

NOTE 224 STACKER SELECT MAY NOT BE SPECIFIED WITH PRINT FILE. STACKER SELECT IS IGNORED AND SINGLE SPACING IS ASSUMED FOR ALL LINES OR NAMED FILE.

NOTE 225 PRINT OR PUNCH FUNCTION MAY NOT BE SPECIFIED FOR AN OUTPUT RECORD OF TAPE OR DISC FILE. STACKER SELECT, SPACING, OR SKIPPING IS IGNORED ON ALL RECORDS OF NAMED FILE.

NOTE 226 PRINT FUNCTION MAY NOT BE SPECIFIED FOR OUTPUT RECORD OF PUNCH FILE. SPACE AND SKIP ENTRIES ARE IGNORED FOR ALL RECORDS OF NAMED FILE.

NOTE 227 NUMBER OF LINES OF OUTPUT EXCEEDS THE CAPACITY OF RPG. MAXIMUM NUMBER IS 1023. EXECUTION IS DELETED.

NOTE 228 IMPROPER USE OF PACKING OR ZERO SUPPRESSION ON ALPHANUMERIC OR PACKED FIELD. ENTRY OF BLANK IS ASSUMED FOR INVALID CODE.

NOTE 229 END POSITION SPECIFIED FOR THE FIELD IS GREATER THAN THE RECORD LENGTH. ALL OR PART OF THE FIELD IS LOST, STARTING WITH THE RIGHTMOST POSITION.

NOTE 230 END POSITION IS LESS THAN THE FIELD LENGTH. FIELD IS NOT PROCESSED.

NOTE 231 FIELD TO BE EDITED IS GREATER THAN THE EDIT WORD. RIGHTMOST DIGITS WILL BE LOST.

NOTE 232 FIELD TO BE EDITED IS NOT NUMERIC. NO EDITING IS PERFORMED.

NOTE 233 STERLING SIGN POSITION IS SPECIFIED AS OTHER THAN NORMAL FOR PRINTED LINE. NORMAL POSITION IS ASSUMED.

NOTE 234 LOCATION FOR STERLING SIGN EXCEEDS RECORD LENGTH. NORMAL POSITION FOR SIGN IS ASSUMED.

NOTE 235 STERLING FIELD IS SPECIFIED WITH DECIMAL LENGTH GREATER THAN THREE. FIELD IS NOT PROCESSED.

NOTE 236 STERLING FIELDS MAY BE EDITED ONLY FOR PRINT FILES. NO EDITING IS PERFORMED.

NOTE 237 STERLING FIELD IS SPECIFIED WITH MORE THAN NINE POUNDS POSITIONS. LEFTMOST DIGITS WILL BE LOST.

NOTE 238 MAXIMUM NUMBER OF EXTERNAL SYMBOLS (EXIT SUBROUTINES, ULABL ENTRIES, AND LABEL EXITS) HAS BEEN EXCEEDED. EXECUTION IS DELETED.

APPENDIX G. USER CODED SUBROUTINES AND USE OF RPG AS A SUBROUTINE

G1. INTRODUCTION

This Appendix deals with the following four topics:

- User-coded EXIT subroutines
- User-coded ALTSEQ subroutines
- User label processing subroutines
- Using RPG object programs as subroutines

In the discussion, familiarity with UNIVAC 9400 System Assembler Language programming is assumed (see *UNIVAC 9400 System Assembler/Central Processor Unit Programmers Reference, UP-7600 (current version)*).

G2. USER-CODED EXIT SUBROUTINES

The EXIT operation is coded in the Calculation Specifications to permit the execution of user-coded subroutines at detail or total time. The subroutines may be written in assembler language, RPG, or (with some restrictions) in FORTRAN or COBOL, and must be compiled separately. Before execution, the RPG object program and all subroutines must be linked together by the Linkage Editor to form a single load module. If more than one subroutine is present in the load module and the program is to be executed on a system with storage of at least 32,768 bytes, the user may specify that the Linkage Editor create a load module with overlay structure, so that the various subroutines are in storage only when required and occupy the same locations, thus reducing overall storage requirements.

When RPG passes control to an assembly language subroutine, certain conventions must be observed in the use of the general registers.

Upon entry to the subroutine:

Register 15 contains the address of the first executable instruction of the subroutine. (The name of this instruction should be the same as or equivalent to the name specified as factor 2 of the EXIT statement. The factor 2 name should appear in the subroutine as the name of a START or CSECT statement or as the operand of an ENTRY statement.)

Register 14 contains the address to which the subroutine must branch when its processing is finished.

Register 13 contains the address of an 18-word area (aligned on a fullword boundary) in which the subroutine may save the contents of any or all registers.

When processing is finished, and the subroutine has returned control to RPG, the contents of registers 2 through 13 must be the same as they were at the time of entry. The contents of other registers need not be the same. The SAVE and RETURN macro instructions may conveniently be used for properly saving and restoring all registers, except 13, and returning to RPG (see Figure G-1, Register Conventions for an EXIT subroutine).

Note that no input or output operations affecting files used by the main RPG program are permitted in the subroutine.

Fields defined in RPG may be used in an assembler language subroutine if they are specified in RLABL statements in RPG and in EXTRN statements in the subroutine. Fields defined in the subroutine may be used in RPG if they are specified in ULABL statements in RPG and in ENTRY statements in the subroutine. Numeric fields in RPG are always stored in packed decimal format.

Any RPG program indicator may be tested, set, or reset by a subroutine. Each indicator occupies one byte in storage; the set (on) value is X'F0' and the reset (off) value is X'00'. An indicator has a name of the form INxx (such as, INH0, IN94, or IN1P), which must be specified in an RPG RLABL statement and a subroutine EXTRN statement.

An EXIT statement may be used to call a COBOL or FORTRAN subroutine, and is subject to the usual restriction on input and output operations. Parameters cannot be passed directly to such a subroutine. To use a parameterized COBOL or FORTRAN subroutine, the user must include in the load module an additional, short assembler language subroutine to receive control from RPG and to call and pass parameters to the higher level language subroutine.

ASSEMBLER CODING FORM

PROGRAM _____ PROGRAMMER _____ DATE _____ PAGE ____ OF ____ PAGES

LABEL	OPERATION	OPERAND	COMMENTS
*			
*, OUTLINE OF A USER SUBROUTINE			
*			
SUBR	CSECT		SUBR IS FACTOR 2 OF AN RPG EXIT
*			OPERATION.
	USING	*,12	BASE REGISTER ASSIGNMENT.
BEGIN	SAVE	(14,12)	SAVE ALL REGISTERS EXCEPT 13
	LR	12,15	LOAD BASE REGISTER
	ST	13,SAVER13	SAVE REGISTER 13.
			PROCESSING
	L	13,SAVER13	RESTORE REGISTER 13
	RETURN	(14,12)	RESTORE OTHER REGISTERS AND
*			RETURN VIA REGISTER 14.
SAVER13	DC	F'0'	
	END	BEGIN	

Figure G–1. Register Conventions for an EXIT Subroutine

Example:

A FORTRAN subroutine named COMP calculates compound interest from four parameters: principal, annual rate of interest, number of interest periods per year, and number of years. Four RPG fields, containing appropriate parameter values, are defined in RLABL statements. The subroutine in Figure G-2 passes those parameters to the FORTRAN subroutine.

Data passed to FORTRAN subroutines must conform to FORTRAN standards (see *UNIVAC 9400 System FORTRAN Supplementary Reference, UP-7693* (current version)). Integer constants must be specified with a length of 9 in RPG. Real constants cannot be processed or printed by RPG. Subroutine results should be passed back to the main program by redefining formal parameters. Thus, in the example given, it is assumed that the result is assigned to the first parameter (PRIN).

All numeric and alphanumeric fields in an RPG program can also be processed in a COBOL subroutine. RPG indicators should be described in the COBOL Linkage Section as PICTURE X. An indicator is set by using MOVE ZERO..., and reset by using MOVE LOW-VALUE... .

G3. USER-CODED ALTSEQ SUBROUTINES

If the user enters an A in Column 26 of the RPG control card (see Appendix E), the RPG object program calls a special user subroutine each time a record with matching fields is read. In such a subroutine the user can:

- change the collating sequence of the matching fields (by translating the data in the matching fields hold area) before sequence checking of records takes place;
- do any other processing of the matching fields hold area; or
- do any desired processing of any input record fields before control or chaining fields are processed and before the data is made available for detail calculations and output.

The following rules apply:

- The subroutine should be written in assembler language.
- The entry point of the subroutine must have the name ALTSEQ.
- The subroutine must be linked with the RPG object program and any other subroutines before execution by the Linkage Editor. Automatic overlay is possible on systems of at least 32,768 bytes (see G2 for discussion of EXIT subroutines).

ASSEMBLER CODING FORM

PROGRAM _____ PROGRAMMER _____ DATE _____ PAGE ____ OF ____ PAGES

LABEL	OPERATION	OPERAND	COMMENTS
*			
*		PASSING PARAMETERS TO A FORTRAN SUBROUTINE	
*			
SUBR	CSECT		SUBR IS FACTOR 2 OF AN RPG
*			EXIT OPERATION.
	USING	*,12	BASE REGISTER ASSIGNMENT
BEGIN	SAVE	(14,12)	SAVE REGISTERS.
	LR	12,15	LOAD BASE REGISTER.
	ST	13,SAVEAREA+4	SAVE REGISTER 13.
	LA	13,SAVEAREA	LOAD ADDRESS OF NEW SAVE AREA
*			FOR CALLING NEXT SUBROUTINE
	CALL	COMPR,(PRIN,AR01,NIPPY,YRS))	CALL ROUTINE AND PASS FOUR
*			PARAMETERS.
	EXTRN	PRIN	ALL 4 LABELS HAVE
	EXTRN	AR01	APPEARED IN RLABL
	EXTRN	NIPPY	STATEMENTS IN THE
	EXTRN	YRS	RPG PROGRAM.
	L	13,SAVEAREA+4	ON RETURN FROM COMPR, RESTORE
*			REGISTER 13.
	RETURN	(14,12)	RESTORE OTHER REGISTERS AND
*			RETURN.
SAVEAREA	DS	18F	
	END	BEGIN	

Figure G–2. EXIT Subroutine to Call a FORTRAN Subroutine

- RLABL and ULABL statements in the RPG program may be used in conjunction with EXTRN and ENTRY statements in the ALTSEQ subroutine in the same manner as in an EXIT subroutine.

- Conventions for saving and restoring registers are identical to those for EXIT subroutines (see G2). Registers 13 and 14 at entry to the ALTSEQ subroutine contain the values specified for EXIT subroutines. In addition:

  - Register 15 contains the address of the entry point of the ALTSEQ subroutine.

  - Register 1, bits 0-7, contains the total length of the matching fields minus one, occurring in this record.

  - Register 1, bits 8-31, contains the address of the high order byte of the matching fields hold area.

  - Register 0 contains the address of the first byte of the input record. (For variable-format records, this is the first byte following the four-byte record length control field).

- Record identification codes in the input record can be changed; however, record identification is made before the ALTSEQ subroutine is called and is not changed by any alteration of the codes.

G4. USER LABEL PROCESSING SUBROUTINES

User subroutines are required for processing user header and trailer labels on tape or disc files or nonstandard labels on tape files. Rules for coding these subroutines may be found in *UNIVAC 9400 System Data Management System Programmers Reference, UP-7629* (current version). The following additional points should be noted:

- The subroutines must be written in assembler language.

- The name of the entry point of each subroutine must correspond exactly to the names given in Columns 54-59 of the File Description Specifications form.

- The subroutines must be linked with the RPG object program and other subroutines before execution by the Linkage Editor. Automatic overlay of label subroutines is not possible.

- RLABL and ULABL statements in the RPG program may be used in conjunction with EXTRN and ENTRY statements in the label subroutine in the same manner as in an EXIT subroutine.

G5. USING RPG OBJECT PROGRAMS AS SUBROUTINES

The object program output of an RPG compilation can be used as a subroutine if an S is coded in Column 74 of the RPG control card (see Appendix E). (A subroutine produced in this way cannot also be used as a main program.) When an RPG subroutine is called, it opens all files and does all processing specified before returning to the calling program.

The standard register usage conventions are observed by the RPG object program:

- At entry, registers 15, 14, and 13 must be initialized as follows:

  Register 15 must contain the address of the entry point of the subroutine. (The name of the entry point is the same as that coded in Columns 75-80 of the RPG control card, or must be RPGOBJ if Columns 75-80 are blank.)

  Register 14 must contain the return address of the calling program.

  Register 13 must contain the address of an (aligned) 18-word register save area.

- The RPG subroutine saves all registers except register 15. Upon return to the calling programs, register 15 contains binary 0 if the subroutine ended normally, binary 4 if a user halt indicator (H1-H9) was set, or binary 8 if the RPG hold indicator (H0) was set.

APPENDIX H. TAG FILES

H1. INTRODUCTION

A tag file is a sequentially organized input file used to obtain random access to records from a second input file (the "data file") on disc. Only one tag file may be specified in a program, and if a tag file is specified, a record address file may not be specified.

A tag file may be on cards, tape, or disc, and must be sequential. The data file must be on disc, and may have sequential or direct organization.

H2. ENTERING TAG FILE INFORMATION ON RPG SPECIFICATION FORMS

The user identifies the tag and data files in his program on three specification forms as follows:

(1) File Description Specifications

Tag File:	I	- file type	- Column 15
	R	- file designation	- Column 16
	Blank	- mode of processing	- Column 28
	10	- length of record address field	- Columns 29-30
	Blank	- record address type	- Column 31
	T	- type of file organization	- Column 32
	E	- extension code	- Column 39

Data File:	I	- file type	- Column 15
	P or S	- file designation	- Column 16
	R	- mode of processing	- Column 28
	Blank	- length of record address field	- Columns 29-30
	I or R	- record address type	- Column 31
	D	- type of file organization	- Column 32
	E	- extension code	- Column 39

(2) File Extension Specifications

Tag File: enter file name in Columns 11-18.
Data File: enter file name in Columns 19-26.

(3) Input Specifications

Tag File: no entry necessary.
Data File: enter file name in Columns 7-14.

NOTE: A data file only may be chained to one or more indexed sequential files. In such a case, the extension code of the data file (File Description, Column 39) must be E, and File Extension Specifications for chaining should be included as usual (see 6.2.2).

H3. TAG FILE FORMATS

Each record of a tag file contains one or more entries of ten bytes each. Each entry contains a pointer to a logical record of the data file. At execution time, when a new record is to be obtained from the data file, the next sequential entry in the tag file is examined, and this entry is used to locate and read the required data record. Thus, for instance, an unsorted disc file can be processed in sorted sequence by means of an associated tag file that contains entries which correspond to the desired order of processing. The first entry should contain a pointer to the first data record to be processed, and so forth.

Tag file entries have the following format:

(1) Each entry contains ten bytes (number 0-9) that are divided into two parts,

bytes 0-7: block address
bytes 8-9: logical record pointer

(2) The block address may have either of two formats: physical disc track address or relative block. Column 31 of the File Description form indicates which format the RPG compiler is to process, an I for physical disc track address, or an R for relative block number.

(a) Physical disc track address (I):

byte 0 = logical unit number (X'00' – X'03')
bytes 1-2 = X'0000'
bytes 3-4 = cylinder number (X'0000' – X'0007')
bytes 5-6 = head number (X'0000' – X'0009')
byte 7 = sequential number of the block on the track (X'01' – X'FF')

(b) Relative block number (R):

bytes 0-3 = X'00000000'
bytes 4-7 = relative block number of the block to be accessed in the file (X'00000001' - X'FFFFFFFF'; the number of the first block in the file is 1)

(3) The logical record pointer is the displacement in bytes (relative to zero) of the desired record within the block identified by bytes 0-7. This pointer always refers to the first byte within the record, and therefore, for variable format files, to the beginning of the four-byte record length field which precedes every logical record.

In the particular case of unblocked files:

bytes 8-9 = X'0000' for fixed format files
bytes 8-9 = X'0004' for variable format files (since a four-byte block length field for the whole block precedes the record length field of the first record of all variable format files).

(4) If bytes 0-7 of any entry are all blanks, this and all remaining entries of the current tag file record are skipped, a new record is read from the tag file, and processing resumes with the first entry of the new record.


UNIVAC
SPERRY RAND