

UNIVAC

DATA PROCESSING DIVISION

9200/9300

S Y S T E M S

**CARD REPORT
PROGRAM
GENERATOR**

REFERENCE MANUAL

This manual is published by the Univac Division of Sperry Rand Corporation in loose leaf format as 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 programming effort. For the current status of the programming, 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 and/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 respective Systems.

CONTENTS

CONTENTS	1 to 6
1. INTRODUCTION	1-1 to 1-7
1.1. PURPOSE OF THIS MANUAL	1-1
1.2. AUDIENCE FOR WHOM THIS MANUAL IS INTENDED	1-1
1.3. SUMMARY OF MANUAL CONTENTS	1-1
1.4. DESCRIPTION OF THE RPG	1-2
1.5. RPG OPERATION	1-2
1.6. COMMON FIELDS FOR RPG SPECIFICATION FORMS	1-5
1.6.1. Page Number (Columns 1-2)	1-5
1.6.2. Line Number (Columns 3-5)	1-5
1.6.3. Form Type (Column 6)	1-6
1.6.4. Comments (Column 7)	1-6
1.6.5. Program Identification (Columns 75-80)	1-6
2. FILE DESCRIPTION FORM	2-1 to 2-6
2.1. FORM ENTRIES	2-1
2.1.1. File Name (Columns 7-14)	2-1
2.1.2. File Type (Column 15)	2-2
2.1.3. File Designation (Column 16)	2-3
2.1.4. End of File (Column 17)	2-3
2.1.5. Sequence (Column 18)	2-4
2.1.6. Columns 19-39	2-4
2.1.7. Device (Columns 40-46)	2-4
2.1.8. Columns 47-53	2-6
2.1.9. Comments (Columns 66-74)	2-6
3. INPUT FORMAT SPECIFICATIONS FORM	3-1 to 3-24
3.1. RECORD IDENTIFICATION ENTRIES	3-2
3.1.1. File Name (Columns 7-14)	3-2
3.1.2. Sequence (Columns 15-16)	3-2
3.1.3. Number (Column 17)	3-3
3.1.4. Option (Column 18)	3-3
3.1.5. Record Indicator (Columns 19-20)	3-6
3.1.6. Record Identification Codes (Columns 21-41)	3-7
3.1.6.1. Position (Columns 21-24)	3-8
3.1.6.2. Not (Column 25)	3-8
3.1.6.3. C/Z/D (Character, Zone, Digit) (Column 26)	3-8
3.1.6.4. Char (Character) (Column 27)	3-8
3.1.7. AND Relationship	3-9
3.1.8. OR Relationship	3-9
3.1.9. Stacker Select (Column 42)	3-11

3.2. FIELD DESCRIPTION ENTRIES	3-12
3.2.1. Data Format (Column 43)	3-12
3.2.2. Field Location (Columns 44-51)	3-12
3.2.3. Decimal Position (Column 52)	3-13
3.2.4. Field Name (Columns 53-58)	3-13
3.2.5. Control Level (Columns 59-60)	3-14
3.2.5.1. Split Control Fields	3-14
3.2.6. Matching Fields (Columns 61-62)	3-17
3.2.6.1. Matching Records Indicator – Multifile Input	3-18
3.2.6.2. Sequence Checking	3-20
3.2.7. Field Record Relation (Columns 63-64)	3-20
3.2.8. Field Indicators (Columns 65-70)	3-21
3.2.9. Sterling Sign Position (Columns 71-74)	3-23
4. CALCULATION SPECIFICATIONS FORM	4-1 to 4-26
4.1. CALCULATION SPECIFICATIONS FORM ENTRIES	4-2
4.1.1. Control Level (Columns 7-8)	4-2
4.1.2. Indicators (Columns 9-17)	4-4
4.1.3. Factor 1 (Columns 18-27) and Factor 2 (Columns 33-42)	4-7
4.1.4. Operation (Columns 28-32)	4-8
4.1.4.1. Add (ADD)	4-8
4.1.4.2. Zero and Add (Z-ADD)	4-10
4.1.4.3. Subtract (SUB)	4-10
4.1.4.4. Zero and Subtract (Z-SUB)	4-10
4.1.4.5. Multiply (MULT)	4-10
4.1.4.6. Divide (DIV)	4-11
4.1.4.7. Move Remainder (MVR)	4-11
4.1.4.8. Compare (COMP)	4-12
4.1.4.9. Move (MOVE)	4-12
4.1.4.10. Move Leftmost Characters (MOVEL)	4-13
4.1.4.11. Move Zone	4-14
4.1.4.11.1. Move Low to Low (MLLZO)	4-14
4.1.4.11.2. Move Low to High (MLHZO)	4-14
4.1.4.11.3. Move High to Low (MHLZO)	4-14
4.1.4.11.4. Move High to High (MHHZO)	4-14
4.1.4.12. Table Lookup (LOKUP)	4-15
4.1.4.13. Branch (GOTO)	4-15
4.1.4.14. Exit to Subroutine (EXIT)	4-16
4.1.4.15. Result Label (RLABL)	4-18
4.1.4.15.1. Coding of Subroutines	4-19
4.1.4.15.2. Subroutine Restrictions	4-20
4.1.4.16. Tag (TAG)	4-22
4.1.4.17. Set Indicators Off (SETOF)	4-22
4.1.4.18. Set Indicators On (SETON)	4-22
4.1.5. Result Field (Columns 43-48)	4-22
4.1.6. Field Length (Columns 49-51)	4-22
4.1.7. Decimal Position (Column 52)	4-23
4.1.8. Half Adjust (Column 53)	4-24
4.1.9. Resulting Indicators (Columns 54-59)	4-24
4.1.10. Comments (Columns 60-74)	4-25

5. OUTPUT FORMAT SPECIFICATIONS FORM	5-1 to 5-19
5.1. FILE IDENTIFICATION AND CONTROL FORM ENTRIES	5-3
5.1.1. File Name (Columns 7-14)	5-3
5.1.2. Type (Column 15)	5-3
5.1.3. Stacker Select/Print Option (Column 16)	5-5
5.1.4. Space (Columns 17-18)	5-5
5.1.5. Skip (Columns 19-22)	5-6
5.1.6. Output Indicators (Columns 23-31)	5-8
5.2. FIELD DESCRIPTIONS	5-10
5.2.1. Output Indicators (Columns 23-31)	5-11
5.2.2. Field Name (Columns 32-37)	5-11
5.2.3. Zero Suppress (Column 38)	5-14
5.2.4. Blank After (Column 39)	5-14
5.2.5. End Position In Output Record (Columns 40-43)	5-14
5.2.6. Data Format (Column 44)	5-15
5.2.7. Constant or Edit Word (Columns 45-70)	5-15
5.2.7.1. Constant	5-15
5.2.7.2. Edit Word	5-17
5.2.8. Sterling Sign Position (Columns 71-74)	5-18
6. FILE EXTENSION SPECIFICATIONS FORM	6-1 to 6-9
6.1. FILE EXTENSION SPECIFICATIONS FORM ENTRIES	6-3
6.1.1. Table Name (Columns 27-32)	6-3
6.1.2. Number of Table Entries Per Record (Columns 33-35)	6-4
6.1.3. Number of Table Entries Per Table (Columns 36-39)	6-4
6.1.4. Length of Table Entry (Columns 40-42)	6-4
6.1.5. Numeric (Column 44)	6-4
6.1.6. Sequence (Column 45)	6-4
6.1.7. Table Name (Columns 46-51)	6-5
6.1.8. Length of Table Entry (Columns 52-54)	6-5
6.1.9. Numeric (Column 56)	6-5
6.1.10. Sequence (Column 57)	6-5
6.2. LOKUP PROCEDURE	6-6
6.2.1. Single Table	6-6
6.2.2. Two Tables	6-6
6.2.3. Resulting Indicators	6-7
APPENDIX A. SAMPLE PROGRAMS	A-1 to A-24
A.1. EXAMPLE NO. 1 - SALES ANALYSIS LISTING	A-1
A.1.1. Definition	A-1
A.1.2. Procedure	A-2
A.1.3. File Description Specifications	A-2
A.1.4. Input Specifications	A-2
A.1.5. Calculation Specifications	A-5
A.1.6. Output Specifications	A-6

A.2. EXAMPLE NO. 2 – DAILY GROSS PAY CALCULATION	A-6
A.2.1. Procedure	A-7
A.2.2. File Description Specifications	A-8
A.2.3. Input Format Specifications	A-8
A.2.4. Calculation Specifications	A-10
A.2.5. Output Format Specifications	A-12
A.3. EXAMPLE NO. 3 – INVENTORY APPLICATION	A-14
A.3.1. Procedure	A-15
A.3.2. Description of the Stock Status Report Text	A-19
A.3.3. File Specifications	A-19
A.3.4. Input Specifications	A-20
A.3.5. Calculation Specifications	A-20
A.3.6. Output Specifications	A-21
APPENDIX B. SUMMARY OF PROGRAM INDICATORS	B-1
APPENDIX C. CODE REPRESENTATION AND TRANSLATION	C-1 to C-5
APPENDIX D. STERLING NOTATION	D-1 to D-4
D.1. FORMAT 1	D-1
D.2. FORMAT 2	D-2
D.2.1. Shillings	D-2
D.2.2. Pence	D-3
D.2.3. Format 2 Types	D-3
D.2.4. Signs	D-4
D.2.5. Leading Zeros	D-4
D.3. PENCE FORMAT	D-4
FIGURES	
1-1. RPG Object Program Cycle	1-3
1-2. Example of Common Field Entries	1-7
2-1. File Description Form Headings	2-1
2-2. Examples of Various File Description Entries	2-2
2-3. Example of Device Field Entries	2-5
3-1. Input Format Specifications Form	3-1
3-2. Card File Control Group	3-2
3-3. Example of FILE NAME, SEQUENCE, NUMBER, and OPTION Entries	3-4
3-4. Control Group Record Types	3-5
3-5. Example of RECORD INDICATOR and RECORD IDENTIFICATION CODES Entries	3-7
3-6. Examples of AND, OR, and NOT Relationships	3-11
3-7. Example of a Split Control Field	3-15
3-8. Example of Multiple Split Control Field Specifications	3-16
3-9. Example of Matching Fields	3-17
3-10. Example of Matching a Primary File against a Secondary File	3-19
3-11. Example of Sequence Checking with Matching Fields Specification	3-20
3-12. Example of FIELD RECORD RELATION Entries	3-22
3-13. Example FIELD IDENTIFICATION Entries	3-24

4-1. Calculation Specification Form	4-1
4-2. Example of Control Level Entries	4-3
4-3. Example of Indicators Entries	4-6
4-4. Example of Factor 1 Entries	4-9
4-5. Operation Codes	4-9
4-6. Example of MOVEL Operation	4-13
4-7. Examples of GOTO Operations	4-15
4-8. Format of EXIT Operation	4-17
4-9. Format of RLABL Operation, Field Not Defined Elsewhere	4-19
4-10. Format of RLABL Operation, Field Defined Elsewhere	4-19
4-11. Result Label Statement Indicator Definitions	4-20
4-12. Sample Coding RPG and Associated Subroutine	4-21
4-13. Example of Result Field Entries	4-23
4-14. Example of Completed Calculation Specification Form	4-26
5-1. Output Format Specifications Form	5-1
5-2. Sequence of File Entries	5-2
5-3. Examples of File Name Entries	5-3
5-4. Example of Type H/D/T Entries	5-4
5-5. Example of Entries for Stacker Select/Print Option	5-6
5-6. Examples of Space and Skip Entries	5-7
5-7. Examples of Output Indicator Entries	5-10
5-8. Example of Field Description Entries	5-12 to 5-13
6-1. File Extension Specification Form	6-1
6-2. Example of Table Input Record Formats	6-3
6-3. Examples of LOKUP Operation Formats	6-6
6-4. Table Lookup Applications Example	6-9
A-1. Sales Analysis Data Flow	A-1
A-2. Billing Line Item Detail Card Format	A-2
A-3. Weekly Sales Analysis Listing	A-3
A-4. Sales Analysis File Description	A-4
A-5. Sales Analysis Input Specifications	A-4
A-6. Sales Analysis Calculations Specifications	A-4
A-7. Sales Analysis Listing Output Specifications	A-5
A-8. Daily Gross Pay Data Flow	A-6
A-9. Input and Output Card Formats	A-7
A-10. Daily Gross Pay Report Format	A-9
A-11. Daily Gross Pay File Description Specifications	A-10
A-12. Daily Gross Pay Input Format Specifications	A-10
A-13. Daily Gross Pay Calculation Specifications	A-11
A-14. Daily Gross Pay Output Specifications (Sheet 1 of 2)	A-12
A-14. Daily Gross Pay Output Specifications (Sheet 2 of 2)	A-13
A-15. Stock Status Report Data Flow	A-14
A-16. Input Card Formats (Sheet 1)	A-15
A-16. Input Card Formats (Sheet 2)	A-16
A-17. Output Card Formats	A-17
A-18. Daily Stock Status Report Format	A-18
A-19. Inventory File Description	A-19
A-20. Inventory Input Specifications	A-20
A-21. Inventory Calculation Specifications	A-21

A-22. Inventory Output Specifications (Sheet 1 of 3)	A-22
A-22. Inventory Output Specifications (Sheet 2 of 3)	A-23
A-22. Inventory Output Specifications (Sheet 3 of 3)	A-24
D-1. Maximum Sizes of Sterling Notation Formats	D-1
D-2. Sterling Amounts Expressed in Available Formats of Notation	D-2
D-3. Summary of Sterling Notation Formats	D-3

TABLES

Table C-1. 9000 Series Internal Code	C-2 to C-5
--------------------------------------	------------

1. INTRODUCTION

1.1. PURPOSE OF THIS MANUAL

This manual is a reference guide to the use of the UNIVAC 9200/9300 Report Program Generator (RPG) specification forms. These forms are the tools for applying the capabilities of the RPG program on the UNIVAC 9200/9300 Systems. Each specification sheet is described, and the detailed entries are explained and illustrated to provide the programmer with direct insight into the practical requirements of solving problems with the RPG.

1.2. AUDIENCE FOR WHOM THIS MANUAL IS INTENDED

The manual is designed to be used by personnel with some knowledge of unit record equipment principles and functions. Prior knowledge of computer programming, stored programming concepts, and similar information is not assumed.

Personnel will be able to use the RPG to solve problems of at least the same complexity as their experience with unit record equipment would permit.

1.3. SUMMARY OF MANUAL CONTENTS

Each RPG Specification form is treated separately in the following sections of the manual after a brief explanation of the entries that are common to all forms. The remainder of the manual is organized in the following manner:

Section	Title
1	COMMON FIELDS FOR RPG SPECIFICATION FORMS
2	FILE DESCRIPTION FORM
3	INPUT FORMAT SPECIFICATIONS FORM
4	CALCULATION SPECIFICATIONS FORM
5	OUTPUT FORMAT SPECIFICATIONS FORM
6	FILE EXTENSION SPECIFICATIONS FORM

Within each section the individual elements are described by means of a standard format whose headings follow:

- **Entry**
Is an entry 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 desired by the programmer.
- **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 six major sections, the following appendices are included:

- Appendix A RPG Programming Example
- Appendix B Summary of Program Indicators
- Appendix C Code Representation and Translation
- Appendix D Sterling Notation

1.4. DESCRIPTION OF THE RPG

The Report Program Generator is a program based on a series of RPG specification forms that define the operating parameters of the system for a given processing function. The RPG program as described on the forms is then keypunched into cards, and the entire program is stored in the computer processing unit where it controls the internal operation of the computer system to produce a report or any other specified output.

The programmer may also have the object program punched into cards during the generating run. The punched object program can be used for future runs to eliminate unnecessary regeneration of object code.

1.5. RPG OPERATION

Every record processed by the Report Program Generator goes through an identical 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 1-1 shows one complete cycle of an RPG object program. The steps in the following description are numbered to correspond with the operation numbers appearing in Figure 1-1. A program cycle begins with step 1 and continues through step 16. One complete cycle corresponds to the processing of one input record.

- | Step | Operation |
|------|--|
| ① | An input card 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. Whenever a break is found, the program sets the associated indicator and all lower control level indicators. |
| ④ | The program performs all calculations appropriate to the control level breaks that have occurred. These calculations prepare for the printing and punching of total information to be produced at every control break. |
| ⑤ | The total output records are produced on either punched cards or a printed report. |
| ⑥ | The program tests the last record indicator. If this indicator is on, the program ends. |
| ⑦ | If form overflow has occurred during the total output time of this program cycle, then the OF (overflow) indicator is set. |
| 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 information is printed. The programmer controls this printing by means of the OF, 1P, and other appropriate indicators. Headings are produced only at the top of a page. |
| | <p>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.</p> |
| ⑪ | The output called for by the input record and by the detail time calculations is produced. |
| ⑫ | If the OF indicator has been set, it is reset. This step, in conjunction with steps 7 and 13, guarantees that after form overflow has occurred the following operations will take place: <ol style="list-style-type: none">Any totals appropriate to the page are printed at the bottom of the page.Any heading information is printed at the top of the next page. |

- c. The OF indicator will remain on for one complete detail time (steps 9, 10, and 11).
 - d. If form overflow occurs during detail time output, the OF indicator is on during one complete total time (steps 4 and 5).
- ⑬ If form overflow has occurred during the detail output time of this program cycle, the OF indicator is set.
 - ⑭ All control level indicators are reset in preparation for the next program operation cycle.
 - ⑮ 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.
 - ⑯ If end of file is detected, the last record indicator and all control level indicators are set because end of file signifies a control break on all levels. Total time (steps 4 and 5) is then performed to produce the final total output, after which step 6 terminates the program.

1.6. COMMON FIELDS FOR RPG SPECIFICATION FORMS

The five RPG specification forms have certain common fields which have consistent entries within an RPG program. These fields and their respective entries are described in this section and are not repeated in those sections that deal with each specification form.

1.6.1. Page Number (Columns 1-2)

- Entry
Required.
- Purpose
Establishes proper numerical sequence of specification pages and, in conjunction with a line number entry, establishes an ordered sequence for specification cards to be read by the RPG program.
- Codes
01-99.
- Rules
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.

1.6.2. Line Number (Columns 3-5)

- Entry
Required (partially preprinted).
- Purpose
Establishes sequence discipline on the order in which specification cards are read and processed by the RPG program.

- Codes

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

- Rules

Inserts must be numbered in ascending order when more than one insert is entered between two previously written lines.

1.6.3. Form Type (Column 6)

- Entry

Required.

- Purpose

Identifies the type of form and therefore the type of specification being entered; the code is preprinted in column 6 of each specification form.

- Codes

F – File Description.

I – Input Format Specifications.

C – Calculation Specifications.

O – Output Format Specifications.

E – File Extension Specifications.

1.6.4. Comments (Column 7)

- Entry

Optional.

- Purpose

Permits general descriptions to be entered before, or interspersed among, specification lines. These entries are ignored by the RPG program but will be printed when a listing is made of the specification lines.

- Code

* – specifies comments.

- Rules

Comments may be entered from column 8–80, but it is recommended that columns 75–80 be reserved for program identification.

1.6.5. Program Identification (Columns 75–80)

- Entry

Optional.

- Purpose

Identifies all specification cards associated with a specific program.

- Rules

One through six alphanumeric characters may be used.

2. FILE DESCRIPTION FORM

The File Description form (Figure 2-1) is used to assign a unique name to the files that will come from or go to each input or output device. The form is also used to provide certain basic information about input files that will be used by the program when the records are read.

UNIVAC <small>UNIVERSITY MICROFILMS INTERNATIONAL</small>		REPORT PROGRAM GENERATOR FILE DESCRIPTION		UNIVAC 9000 SERIES	
PROGRAM _____			PROGRAMMER _____		DATE _____ PAGE ____ OF ____ PAGES
PG. NO.	LINE NO.	FILE NAME	DEVICE	COMMENTS	PROGRAM IDENTIFICATION
1	3	5 6 7	39 40	46 47	65 66 74 75 80
		0 1 F			

Figure 2-1. File Description Form Headings

2.1. FORM ENTRIES

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

2.1.1. File Name (Columns 7-14)

- **Entry**
Required.

- **Purpose**

Assigns a name to each input and output file. Referencing this file name in the Input and Output Format Specifications makes available to the RPG program all of the information entered in the File Description form relating to this file.

- **Rules**

1. Only alphabetic or numeric characters may be used in a file name. These characters may be intermixed, but no special characters may be used.
2. The first character of the name must be alphabetic and must appear in column 7.
3. One through eight characters may be used for a file name.
4. No spaces are allowed within the name.
5. Each file name must be unique.

■ Examples

Figure 2-2, a partially completed File Description form, illustrates several file name entries.

UNIVAC
DIVISION OF SPERRY RAND CORPORATION

REPORT PROGRAM GENERATOR
FILE DESCRIPTION

FORM TYPE		FILE DESIGNATION FILE TYPE													END OF FILE SEQUENCE	
PG. NO.	LINE NO.	FILE NAME													DEVICE	
1	3	5	6	7	14	15	16	17	18	19	39	40	46	47	5	
	0, 1	F	C	A	R	D	I	N	I	P	E	A				
	0, 2	F	R	E	A	D	P	N	C	H	C	S	A			
	0, 3	F	P	R	I	N	T	O	U	T	O					

Figure 2-2. Examples of Various File Description Entries

2.1.2. File Type (Column 15)

■ Entry

Required.

■ Purpose

Specifies whether this is an input file, output file, or a card file which is both read and punched.

■ Codes

I – Input. An input file consists of cards read from a specified input device during object program execution.

O – Output. An output file can be either a report produced on the printer or cards punched on an output device.

C – Combined. A combined file consists of cards that are read by the program and subsequently punched as output of the system. The punching may be into the same cards that have been read or into blank cards that have been inserted into the combined file. In the latter case, the blank cards must be identified as a record type, and it is these cards that should be used for punching purposes.

■ Rules

Blank or prepunched cards may be specified as an output file. 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 should be made on the Output Specification form.

■ Examples

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

2.1.3. File Designation (Column 16)

■ Entry

Conditional; must be used with file types I and C.

■ Purpose

Indicates whether the file is a primary or secondary file; must coincide with the predetermined sequence of card reads.

■ Codes

P – Primary. Identifies a primary file.

S – Secondary. Identifies a secondary file.

■ Rules

1. This column must be left blank for all output files.
2. If only one input file is specified, a P must be entered in this column.
3. If two input files are specified, either file may be designated as P and the other as S.
4. If three input files are specified, one file must be designated as P and the others as S. This must conform with the matching fields entries assigned in the Input Format Specifications.
5. The sequence of secondary files is determined by the order in which they appear on the File Description form.

■ Examples

If three files (A, B, C) are used and file A is identified as the primary file, then file A is matched against files B and C. Files B and C are not matched against each other. Refer to Figure 2-2 for an illustration of file designation entries.

2.1.4. End of File (Column 17)

■ Entry

Optional.

■ Purpose

Indicates which files are to be checked to determine when the last record has been read in order to turn on the last record (LR) indicator. Should only be used when all end-of-file conditions need not be present to complete processing.

■ Codes

E – Indicates end of file to be checked.

■ Rules

1. If this entry is blank for all input files of a program, then the LR indicator is turned on only after all input files have been processed.
2. An E entry for a specified file indicates that all records for this input must be read and processed before the LR indicator is turned on.

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

- Examples

Refer to Figure 2-2.

2.1.5. Sequence (Column 18)

- Entry

Conditional; must be used when matching fields are designated on the Input Format Specifications form.

- Purpose

Defines the sequence of the card files so that the matching operations in the program are performed correctly.

- Codes

- A – Ascending sequence
- D – Descending sequence

- Rules

1. An entry must be made in the sequence field when matching fields are specified in the Input Format Specifications; otherwise, it must be left blank.
2. Sequence entries apply only to input files.

- Examples

Refer to Figure 2-2.

2.1.6. Columns 19-39

Must be left blank.

2.1.7. Device (Columns 40-46)

- Entry

Required.

- Purpose

Identifies the input/output unit associated with the file name.

■ Codes

ENTRY	INPUT/OUTPUT DEVICE
CCPRI	1001 Card Controller, Primary Feed
CCSEC	1001 Card Controller, Secondary Feed
READER	Card Reader
PUNCH	Column Punch
CRP	Column Read/Punch
ROWPNCH	Row Punch
PRINT63	Printer – 63 Characters
PRINT48	Printer – 48 Characters

■ Rules

1. Each file name must be assigned one of the input/output devices as noted above.
2. The code entered in the device field must be left justified.

■ Examples

Figure 2-3, a completed File Description form, illustrates the entries for the device field.

UNIVAC
DIVISION OF SPERRY RAND CORPORATION

REPORT PROGRAM GENERATOR
FILE DESCRIPTION

PG. NO.	LINE NO.	FILE NAME	FILE DESIGNATION				END OF FILE SEQUENCE	DEVICE	
			FILE TYPE						
1	3	5 6 7	14	15	16	17	18 19	39 40	46 47
	0, 1	F C A R D I N							
	0, 2	F R E A D P N C H C S					A		
	0, 3	F P R I N T O U T O							
	0, 4	F							

The A in column 18 for the input and combined file specifies that the files will be checked for ascending sequence on the field(s) specified on the Input Format Specification under matching fields.

Figure 2-3. Example of Device Field Entries

2.1.8. Columns 47–53

Must be left blank.

2.1.9. Comments (Columns 66–74)

■ Entry

Optional.

■ Purpose

For programmer use in entering any remarks that will be useful in program checking and testing or that are desired for documentation purposes. The comments entry will be listed as a separate field during the generation of the object program.

3. INPUT FORMAT SPECIFICATIONS FORM

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

The form is titled "UNIVAC REPORT PROGRAM GENERATOR INPUT FORMAT SPECIFICATIONS" and "UNIVAC 9000 SERIES". It contains a grid with the following columns: PC NO. (1), LINE NO. (2), FILE NAME (3-6), SEQ. (7), RECORD IDENTIFICATION CODES (8-42), STACKER SELECT (43), DATA FORMAT (44), DECIMAL POSITIONS (45-53), MATCHING FIELDS CONTROL LEVEL (54-63), FIELD LOCATION (64-70), FIELD NAME (71), FIELD RELATION INDICATORS (72-75), STERLING SIGN POSITION (76), and PROGRAM IDENTIFICATION (77-80). A sample row shows: PC NO. 0, LINE NO. 1, FILE NAME, SEQ. 1, RECORD IDENTIFICATION CODES 1, 2, 3, STACKER SELECT, DATA FORMAT, DECIMAL POSITIONS, MATCHING FIELDS CONTROL LEVEL, FIELD LOCATION (START, END), FIELD NAME, FIELD RELATION INDICATORS, STERLING SIGN POSITION, and PROGRAM IDENTIFICATION.

Figure 3-1. Input Format Specifications Form

- identify input files;
- identify input records within an input file;
- define input record sequence; and
- describe data field format and locations within the input record.

The entries for this form are divided into the following two categories:

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.

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.

3.1. RECORD IDENTIFICATION ENTRIES

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

3.1.1. File Name (Columns 7–14)

- Entry

Required.

- Purpose

Assign 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 a File Description form.
2. The file name characteristics must conform with the rules outlined under heading 2.1.1.

3.1.2. Sequence (Columns 15–16)

- Entry

Required.

- Purpose

This field is referenced when the input records must follow a specified sequence within a control group of the input card file. Figure 3–2 illustrates a card file control group. An out-of-sequence condition causes the object program to stop processing.

ENTRIES IN SEQUENCE FIELD ARE:

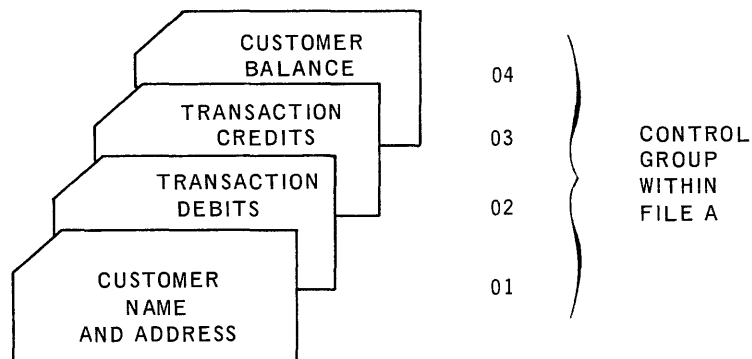


Figure 3–2. Card File Control Group

■ 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 except ND 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 output file.
4. The first entry for each file requiring sequencing must be 01, and succeeding entries must be in consecutive ascending order (02, 03, and so on).
5. A record designating a numeric sequence must have an entry in its associated number field.

■ Examples

Figure 3-3, a partially completed Input Format Specifications form, illustrates various entries that may be included in the sequence field.

3.1.3. 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 is allowable within a control group. Figure 3-4 provides an example of a number of record types within a control group.

■ Codes

- 1 — indicates that only one of that record type can appear within the control group.
- N — indicates that one or more of that record type may appear within the control group.

Blank — when 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. No entry is to be made in this field when the sequence field contains alphabetic data.

■ Examples

Figure 3-3 illustrates entries in this field.

3.1.4. Option (Column 18)**■ Entry**

Optional.

	NUMBER FIELD ENTRIES	CONTROL GROUP RECORD TYPE
CUSTOMER BALANCE	1	4
TRANSACTION CREDITS	N	3
TRANSACTION DEBITS	N	2
CUSTOMER NAME AND ADDRESS	1	1

Figure 3-4. Control Group Record Types

■ Purpose

Defines whether or not a specific record type must be found within each control group.

■ Rules

1. This column must be left blank if the record type must appear in each control group.
2. The letter O is entered when the presence of the record type within the control group is not mandatory.

■ Examples

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

A billing file contains three different record types: name and address, ship to, and details. The ship to card is not required if the address is the same as the one in the name and address card. Figure 3-3 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 three record types are specified on subsequent lines in the sequence that they are to be read.

There should be only one name and address card at the beginning of each control group. This specification is indicated on the form by entering a 01 in the sequence field and a 1 in the number field on the line used to identify the name and address record. No entry (blank) in the option field indicates that this card type must be present to process the group.

A ship to card may be present for a control group, but it must follow the name and address card and is limited to one card per control group. The 02 in the sequence field indicates the order of sequence within the control group, the 1 in the number field specifies the number of the record type allowable, and the letter O in the option field indicates that the record may or may not be present within the control group.

The detail card is in the third record type within the control group, determined by the 03 entry in the sequence field. The N in the number field specifies that one or more of this record type will appear in each control group. The blank option entry indicates that this record type must be present for a control group.

3.1.5. Record Indicator (Columns 19–20)

■ Entry

Required.

■ 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 (specified in the record identification codes – columns 21–41).
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 will process certain specifications of the Calculation and Output 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 will permit 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 type card 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.

The following three types of indicators share the number designations 01 through 99:

- a. Record Indicator (columns 19–20)
- b. Field Indicators (columns 65–70)
- c. Resulting Indicators (columns 54–59 of the Calculation Specifications form)

■ 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.

■ Examples

Figure 3–5 illustrates record indicator entries.

NOTE: A characteristic of the RPG is that only one record indicator is on at one time; when a particular record indicator is turned on, all other record indicators are turned off.

UNIVAC
DIVISION OF ELECTRIC BELL CORPORATION

**REPORT PROGRAM GENERATOR
INPUT FORMAT SPECIFICATIONS**

**UNIVAC
9000
SERIES**

PROGRAM _____ PROGRAMMER _____ DATE _____ PAGE _____ OF _____ PAGES

PG. NO.	LINE NO.	FILE NAME	SEQ.	RECORD INDICATOR	RECORD IDENTIFICATION CODES									FIELD LOCATION		MATCHING FIELDS CONTROL LEVEL	FIELD NAME	FIELD RECORD RELATION					PROGRAM IDENTIFICATION				
					1	2	3	START	END	-	-	OR	5														
					POSITION	NOT	C	Z	CHAR	POSITION	NOT	C	Z	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 44	47 48	51 52 53	58 59	61	63	65	67	69	71	75	80
	0, 1	B, I, L, L, F, I, L, E	0, 1	1, 0	8, 0	Z																					
	0, 2																										
	0, 3																										
	0, 4																										
	0, 5		0, 2	1, 1	8, 0	CE																					
	0, 6																										
	0, 7																										
	0, 8																										
	0, 9		0, 3	1, 2	8, 0	D 5																					
	1, 0																										
	1, 1																										

Figure 3-5. Example of RECORD INDICATOR and RECORD IDENTIFICATION CODES Entries

3.1.6. 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 have only sequence and record identifier 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 identifier by not specifying record identification codes for this last record identifier.

■ Purpose

This section of the Input Format Specifications form identifies the different record types referenced by the record indicator. After identifying this record one time, 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 headed 1, 2, and 3. More than three identifying codes may be specified on successive lines by using AND relationships, as described under heading 3.1.7. 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

3.1.6.1. Position (Columns 21–24)

- Entry
Required.
- Purpose
Indicates the card column that contains the identifying code for a specific card type.
- Rules
 1. The entry must be numeric.
 2. The entry must be right justified.

3.1.6.2. Not (Column 25)

- Entry
Conditional; see codes below.
- 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.

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

- Entry
Required.
- Purpose
Specifies the portion of the card column, indicated in the position entry, that must be examined for identification purposes.
- Codes
C – the entire card column.
Z – the zone portion.
D – the digit portion.

3.1.6.4. Char (Character) (Column 27)

- Entry
Required.
- Purpose
Identifies the character that must be looked for in the input record.

■ Examples

The information written on the input specification form is punched into cards. Effectively, at record identification time in the object program operating cycle, a portion of column 27 on the input specification card is compared for identity of punches and no punches 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. If C is specified, all 12 rows of the column are compared; if Z is specified, rows 12 and 11 are compared; if D is specified, rows 0 through 9 are compared. Thus, what normally appears in the specified record column, what is normally entered for CHAR, and what conditions result in a find are shown in the following table. For a complete discussion of possible variations, see Appendix C.

C/Z/D CODE	CONTENTS OF THE RECORD COLUMN	SPECIFICATION FOR CHAR	CONDITION FOR TESTING EQUAL
C	blank A through Z 0 through 9	blank A through Z 0 through 9	Identity on all twelve rows
Z	& - blank	& - blank	Identity on rows 12 and 11
D	0 through 9	0 through 9	Identity on rows 0 through 9

3.1.7. AND Relationship

■ Entry

Conditional.

■ Purpose

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

■ Rules

1. The additional record identification lines must contain the letters AND in columns 14-16.
2. Columns 17-20 must be left blank.
3. As many AND lines as necessary may be used to identify a record.

3.1.8. OR Relationship

■ 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 more understandable 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 which describe the fields comprising the record.
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). Each field that is not located in the same columns in both record types must be specified twice. The appropriate record indicator is indicated in the field record relation field.
3. When two records have some common fields and some different fields they can be related in an OR relationship. If this is done, each field unique to a record must be so identified in the field record relation entry. It is useful to use an OR relationship in such an instance only if the records contain more common fields than different fields.

■ Rules

1. The additional record identification lines must contain the letters OR in columns 14–15. 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.

■ Examples

1. Figure 3–5 illustrates how the records of the previous billing problem could be identified. Record indicator 10 is assigned to the name and address card. This indicator will be turned on if the zone portion (indicated by Z in column 26) in column 80 (indicated by the 80 in columns 23–24) contains an 11 punch or zone. Indicator number 11 will be turned on if the card has an E punched in column 80. Indicator number 12 will be turned on if the numeric portion of column 80 is equal to 5.
2. Figure 3–6 illustrates how the AND, OR, and NOT relationships may be used. Record indicator 14 is turned on if either a 1 is punched in column 5 or an 11 punch (X zone 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 all of the following four conditions are satisfied by the record:
 - a. No 12 punch (Y zone punch) in column 1.
 - b. The character D is in column 2.
 - c. The character E is in column 3.
 - d. The character D is in column 4.



REPORT PROGRAM GENERATOR
INPUT FORMAT SPECIFICATIONS

PG. NO.	LINE NO.	FORM TYPE	FILE NAME	SEQ.	RECORD IDENTIFICATION CODES												DATA FORMAT	DECIMAL POSITION		MATCHING CONTROL	
					1				2				3					FIELD LOCATION			
					POSITION	NOT	C	Z	POSITION	NOT	C	Z	POSITION	NOT	C	Z		START	END		
1	3	5 6 7	INVENTORY	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
	0 1	I			1 4		5		C	1											
	0 2	I		OR			1		Z	-											
	0 3	I																			
	0 4	I																			
	0 5	I			1 0		8 0		N	D	4										
	0 6	I																			
	0 7	I			1 1		1		N	Z	&		2		C	D		3		C	E
	0 8	I		AND			4		C	D											

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

3.1.9. Stacker Select (Column 42)

- Entry
 - Optional.
- Purpose
 - Permits cards read on the online Column Read/Punch or the 1001 Card Controller to be selected into stackers.
- Rules
 1. Online Column Read/Punch
 - a. A 2 entry will cause stacker 2 to be selected.
 - b. No entry (blank in column 42) will cause stacker 1 to be selected.
 2. 1001 Card Controller – The following entries cause the following stackers to be selected:
 - a. Primary Feed

Entry	Stacker
1	P1
2	P2
3	P3
4	PC

b. Secondary Feed

Entry	Stacker
1	S1
2	S2
3	S3
4	SC

c. If no entry is made for the primary feed, stacker P1 is selected. If no entry is made for the secondary feed, S1 is selected.

3. Stacker selection is not available for the online Column Card Reader.

3.2. FIELD DESCRIPTION ENTRIES

The following descriptions explain the entries for the field description portion of the Input Format Specifications form.

3.2.1. Data Format (Column 43)

- Entry
Conditional.
- Purpose
Indicates to the system whether or not the input data is in packed decimal format.
- Codes
P – indicates a packed input format.
Blank – indicates an unpacked input format (normal entry).
- Rules
Control and/or matching fields may not be specified as packed format.

3.2.2. 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) – These columns specify the location of the first position in a field.
 2. End (Columns 48–51) – These columns specify the location of the last position in a field.

■ 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.

3.2.3. Decimal Position (Column 52)

■ Entry

Conditional; required only for numeric fields.

■ Purpose

Defines for the RPG program the number of decimal positions contained in the field specified in the associated Field Location entry. Also causes the zone punches within the field specified to be removed, except in the units (sign) position.

■ Rules

1. An entry must be made if the field specified is signed (overpunched in the units position).
2. The decimal position must be indicated if any of the following are to be performed on the data field:
 - a. calculation
 - b. zero suppression
 - c. edit
3. A zero must be entered for all numeric fields that contain no decimal positions.
4. An entry may take any value from zero through nine.

3.2.4. Field Name (Columns 53–58)

■ Entry

Required.

■ Purpose

Assigns a name to a defined field so that the programmer does not have to remember the field positions of input records. After a field has been given a name (specified in field name, columns 53–58) all references to this field must be made by using its field name.

■ Rules

1. The name may contain one to six alphanumeric characters.
2. The first character of the name must be alphabetic and be left justified.
3. Special characters and blanks may not be used.
4. 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.

3.2.5. Control Level (Columns 59–60)

■ Entry

Conditional; must contain an entry when the field is used for control purposes.

■ Purpose

Used to determine whether or not a control break has occurred after a record has been read into the system.

When a control break on a given level occurs, the value of the field causing the the break is retained in a level storage 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.

■ Rules

1. Nine control levels may be specified, L1, L2, L3, ..., L9. The number in the entry indicates the level of control; the higher the number, the higher the level.
2. Control fields may be either numeric or alphabetic; when numeric, over-punches are not taken into consideration in comparisons to determine a control break.
3. Control fields may not be in packed format.
4. A control break in the middle of a record type sequence (as specified in the sequence field, columns 15–16 of this form) is considered a break in sequence.
5. If the first record type in a sequence cannot appear more than once in a sequence and all other record types are optional, then a level break must occur at the end of each sequence.

3.2.5.1. Split Control Fields

■ Entry

Optional.

■ Purpose

A control field which consists of many fields in an input record is 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.

■ Rules

1. The fields of a split control field do not have to occupy contiguous column positions.
2. The programmer must observe the following rules when using field record relation indicators with split control field specifications:
 - a. Group control fields by field record relation indicators. All fields with no field record relation indicators must be listed first.
 - b. Noncontrol fields may be listed anywhere within the field descriptions for that file. However, noncontrol fields conditioned by field record relation indicators should be placed within the group of control fields of the same field record relation.
 - c. Control levels within a group must be in ascending sequence.
 - d. A split control field must be entirely numeric or alphabetic.
 - e. The entire field length for each control level must be the same in each group. If, for example, control level (L1) is composed of two fields with an overall length of 20 columns, then the same control level (L1) used with a field record relation indicator and composed of only one field must be only 20 columns in length.
 - f. If a record with split control fields is specified with and without field record relation indicators, then only the fields conditioned by a field record relation indicator are used whenever that record indicator is turned on. The control fields without a field record relation indicator are ignored.
 - g. Split control fields of any one level are placed in memory in the same order as they appear on the input specifications.

NOTE: These rules also must be applied in the use of nonsplit control fields.

■ Examples

1. Figure 3-7 is an example of a split control field. The three fields DEPT, OFFICE, and SECT are not in contiguous positions in the input record but are specified with the same control level (L2). Thus, the three fields are handled as one control field.

UNIVAC		REPORT PROGRAM GENERATOR										UNIVAC																								
9000		INPUT FORMAT SPECIFICATIONS										9000																								
SERIES												SERIES																								
PG. NO.	LINE NO.	FILE NAME	SEQ.	RECORD IDENTIFICATION CODES									FIELD LOCATION		MATCHING FIELDS CONTROL LEVEL		FIELD RECORD RELATION		PROGRAM IDENTIFICATION																	
				1			2			3			START	END																						
				POSITION	NOT	CHAR	POSITION	NOT	CHAR	POSITION	NOT	CHAR																								
				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	75	80
0.1	I	UNIFILE AA	1	1	7	9																														
0.2	I																					1	3													
0.3	I																					4	6													
0.4	I																					1	0	1	2											
0.5	I																					2	3	2	5											
0.6	I																																			
0.7	I																					5	2	5	5											
0.8	I																					5	6	5	9											

Figure 3-7. Example of 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 indicators. Figure 3-8 is an example of this feature.

UNIVAC
DIVISION OF ELECTRIC BELL CORPORATION

REPORT PROGRAM GENERATOR
INPUT FORMAT SPECIFICATIONS

UNIVAC
9000
SERIES

PROGRAM _____ PROGRAMMER _____ DATE _____ PAGE _____ OF _____ PAGES

PG. NO.	LINE NO.	FILE NAME	SEQ.	OPTION NUMBER	RECORD INDICATOR	RECORD IDENTIFICATION CODES						DATA FORMAT		DECIMAL POSITIONS			MATCHING FIELDS CONTROL LEVEL			FIELD RECORD RELATION										
						1		2		3		START	END	FIELD NAME	FIELD INDICATORS	STERLING SIGN POSITION	PROGRAM IDENTIFICATION													
						POSITION	NO. OF CHARACTERS	POSITION	NO. OF CHARACTERS	POSITION	NO. OF CHARACTERS																			
1	3	PAYFILE AA	1415	17	21	7,9	C 2					42	43	44	47	48	51	52	53	58	59	61	63	65	67	69	71	74	75	80
	0,1				2,1																									
	0,2				2,2																									
	0,3				2,3																									
	0,4				2,4																									
	0,5													1		5														
	0,6													6		1,1														
	0,7													1,2		1,4														
	0,8													1,9		2,3														
	0,9													2,4		2,6														
	1,0													2,7		2,9														
	1,1													2,4		3,2								2,3						
	1,2													3,3		3,4								2,3						
	1,3													4,1		4,6								2,3						
	1,4													1,5		2,3								2,4						
	1,5													5,0		5,4									2,4					
	1,6													7,6		7,8														

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

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 whenever that record indicator is turned on. The matching fields without a field record relation indicator are ignored. These fields should be specified first on the Input Specifications form.

3.2.6.1. Matching Records Indicator – Multifile Input

■ Purpose

The function of matching records of one file, called primary, against the records of another file, called secondary, has as 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 cards of a matched control group will be processed before the first secondary card. 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 Calculations and Output Format Specifications forms. It can be used to select unmatched primary and secondary cards from their respective files. If no match is found on the matching fields, the object program processes whichever record is next in 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 lacking these matching fields will contain blanks in the matching fields specification on the input form. The blank field will indicate 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. Three input files may be used with the matching fields specification. This multifile input is accomplished through the use of three input devices: 1001 Card Controller, Card Reader, and Column Read/Punch. The primary file is listed first on the Input Format Specifications form followed by the two secondary files. 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 either secondary file matches a record from the primary file.
3. If matching fields are established for the primary file, the same number of matching fields must be specified for the secondary file.

■ Examples

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 from the file as noted in the example.

The input specifications for this inventory application are 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.

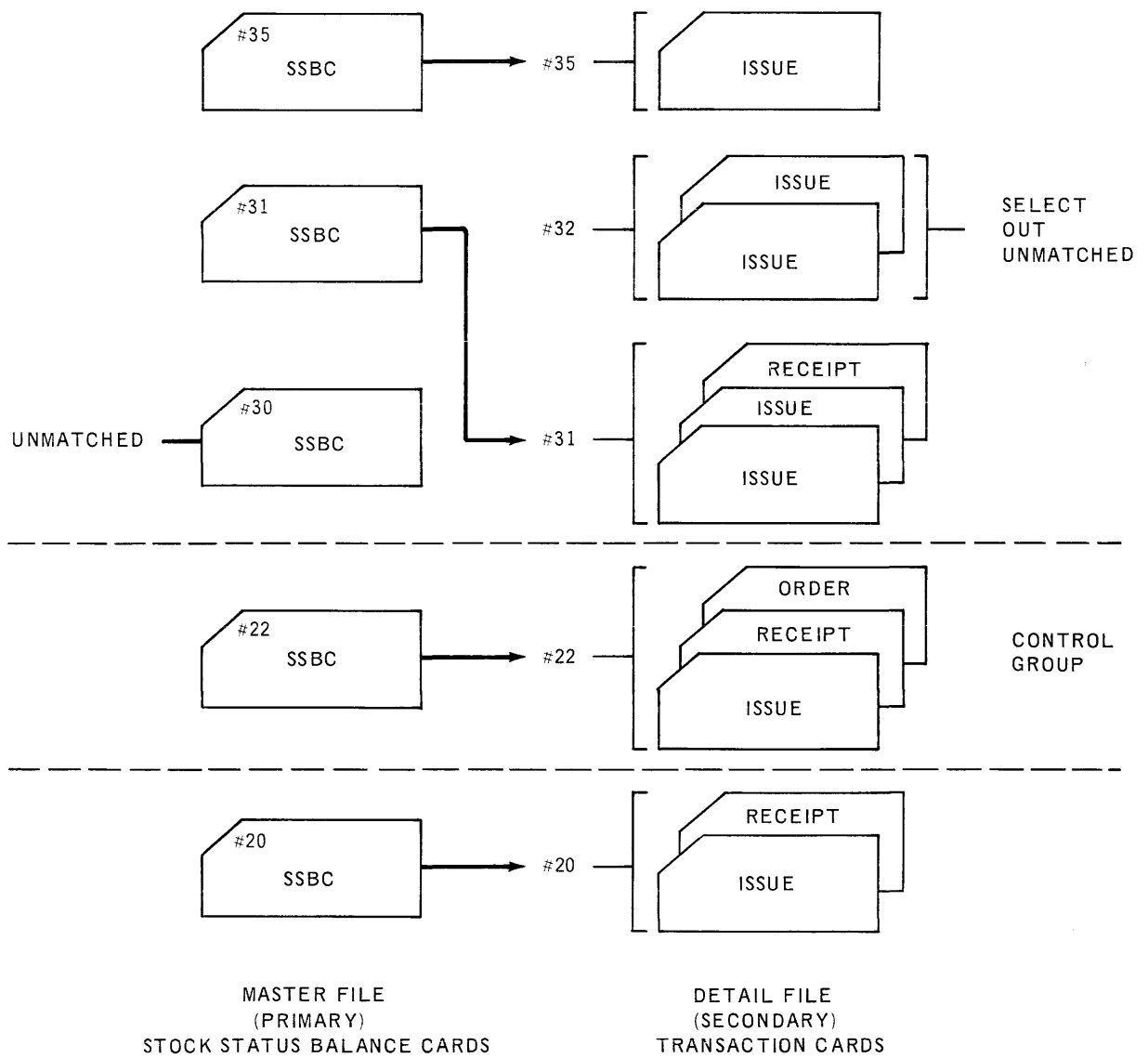


Figure 3-10. Example of Matching a Primary File Against a Secondary File

3.2.6.2. Sequence Checking

■ Purpose

The matching fields indicator may be used to indicate sequence checking when one input file is specified. The specifications of M1 through M3 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), processing will stop immediately.

■ Examples

Figure 3-11 is an example of sequence checking three input fields. M3 specifies RUN, the high order field; PREFIX is the intermediate field indicated by M2; and PARTNO, M1, is the low order field to be checked for sequence. In this example, a compare operation is performed on all three fields at the same time.

REPORT PROGRAM GENERATOR
INPUT FORMAT SPECIFICATIONS



PROGRAMMER _____ DATE _____ PAGE ____ OF ____ PAGES

C H A R 27	ORD IDENTIFICATION CODES						42	43	DATA FORMAT			DECIMAL POSITION			MATCHING FIELDS CONTROL LEVEL			FIELD RECORD RELATION			75	80				
	2		3		FIELD LOCATION				FIELD NAME		FIELD INDICATORS			PROGRAM IDENTIFICATION												
	POSITION	NOT	POSITION	NOT	START	END						+	-		OR											
28	31	32	33	34	35	38	39	40	41	44	47	48	51	52	53	58	59	61	63	65	67	69	71	75	80	
										2		6				DATE										
										7		2	7			DESCR										
										2	8		3	5		PARTNO		M1								
										3	6		3	8		PREFIX		M2								
										3	9		4	1		RUN		M3								
										4	2		4	6		ONHAND										
										4	7		5	1		ORDER										
										5	2		5	6		MIN										
										5	7		6	1		MAX										

Figure 3-11. Example of Sequence Checking with Matching Fields Specification

3.2.7. Field Record Relation (Columns 63-64)

■ Entry

Conditional; used in conjunction with the entries for records in an OR relationship. Also used with split control fields.

■ 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 under headings 3.2.5., Split Control Fields, and 3.1.8., OR Relationship.

■ Rules

1. To increase object code efficiency, fields should be grouped by field record relation indicators.
2. When two records have identical fields but the location of the fields in the records are not identical, only one of the two sets of field specifications need be modified by a field record relation indicator. In such a case those field specifications not including the field record relation indicator must be listed first.

■ 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 2–42. 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 in columns 43–49. Figure 3–12 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 but 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 receipt card (indicated by record indicator 29). Therefore, 29 is written in the field record relation column on the line with QTY. Similarly, 30 in the field record relation column relates ISSUE to the issue card.

3.2.8. Field Indicators (Columns 65–70)**■ Entry**

Optional.

■ Purpose

An entry in this field causes a field indicator to be turned on depending upon the condition of a field. The field is tested for positive, negative, blank, or zero status. The indicator, when turned on, can be used to control calculations, output specifications, or to stop processing of the object program.

Indicators used to test fields for plus, minus, or zero/blank conditions are set (turned on) if certain predefined conditions occur when a record is read. 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 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 the indicator is tested and the results of the test are used to control further processing of the object program.

- b. Halt indicators – These indicators are designated H1 and H2. They are used to halt the processing of the object program when error conditions (as determined by the programmer) have been detected. Halt indicators can also be used to control calculation and output specifications. If H1 or H2 is turned on during processing of a record, then the object program is stopped when the processing of that record is completed.

■ Rules

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

■ Examples

Figure 3-13 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.

The job cards should be in sequence by employee number (EMPNO). This is specified by writing M1 under matching fields on the entry line for EMPNO. By specifying L1 under control level on the same line, the indicator L1 is turned on when a change in employee number occurs; L1 only indicates that a change has occurred in this control field (EMPNO). An M1 is entered in matching fields to indicate that a sequence check on employee number is desired. Field indicator 34 will be turned on if the OTHRS field is zero. Indicator 34 can be used to eliminate overtime processing where there is no entry in the OTHRS field.

3.2.9. 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 and familiarize himself with 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 indicated right justified in columns 71-74.

1. *Control indicators (columns 7–17)* define the record types on which and the conditions under which calculations are to be performed. If calculations are to be performed based on a control break, this specification determines the exact control level at which the calculation will be executed.
2. *Columns 18–53* of the Calculation Specifications form determine the type of calculation that is to be performed – add, subtract, multiply, divide, compare, and so forth. 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 will be stored.
3. *Columns 54–59* of the form permit the results of calculations to be tested, thereby controlling subsequent calculations or output specifications.

4.1. CALCULATION SPECIFICATIONS FORM ENTRIES

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

4.1.1. 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 card is read into the system. If a control break has occurred, total calculations are performed before the card causing the control break is processed. Whenever 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 throughout the program. 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 card 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

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

■ Examples

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

UNIVAC
DIVISION OF SPERRY RAND CORPORATION

PG. NO.	LINE NO.	FORM TYPE		CONTROL LEVEL			INDIC
		5	6	7	9	11	
	0 1		C				1
	0 2		C				
	0 3		C	L	0		2
	0 4		C				
	0 5		C	L	1		3
	0 6		C				
	0 7		C	L	2		4
	0 8		C				
	0 9		C	L	R		5
	1 0		C				
	1 1		C				
	1 2		C				
	1 3		C				

Figure 4-2. Example of Control Level Entries

1. Because the control level columns are blank (spaces) 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 detail time for the last record in the input file. When the LR indicator is set, so are level indicators L1-L9.

4.1.2. Indicators (Columns 9–17)

■ Entry

Optional.

■ Purpose

Defines the conditions under which the calculation is to be performed.

■ Codes

Indicator specifications are grouped into the following categories:

1. If the control level and indicators specifications are blank, the calculation indicated on that specification line will be executed for each detail record.
2. A record indicator code (01–99) indicates the record type on which the specified calculation will be made. The computation 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 card which caused the control break. This permits a calculation to occur only once, on the first card, after the specified control break.
6. The halt indicators (H1 and H2), when specified on a calculation line in conjunction with an N prefix, will cause that calculation to be suppressed and the program will be halted at the end of the current record if the specified halt indicator is on. If the halt indicator is off, the calculation is executed. The halt indicator may have been turned on because of some error condition detected in the input data or because of a prior calculation.
7. The form overflow (OF) indicator permits a calculation specification line to be executed only if form overflow has occurred. If the OF indicator has not been turned on, the specification line containing the OF indicator is not performed.
8. The matching record (MR) indicator code indicates that the calculation is performed only if there is a matching record in a secondary input file.

■ Rules

1. A maximum of three indicators 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 will be performed.

2. If columns 7 through 17 are blank (spaces), the associated calculation is performed at every detail time.
3. 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.
4. 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 will be 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. For example, in the following

UNIVAC
DIVISION OF SPERRY RAND CORPORATION

PG. NO.	LINE NO.	FORM TYPE	CONTROL LEVEL	INDICATORS							
				A N D	A N D	A N D	A N D				
1	3	5	6	7	9	11	12	14	15	17	18
	0 1	C			1	0					
	0 2	C				*					
	0 3	C				*		1	1		
	0 4	C				*					
		C									

none of lines 1-4 will be executed if indicator 10 is off; lines 1 and 2 will be executed if indicator 10 is on; and lines 3 and 4 will be executed only if indicators 10 and 11 are on.

- a. Indicators are tested only when they are explicitly specified. For example, in the following

UNIVAC
DIVISION OF SPERRY RAND CORPORATION

PG. NO.	LINE NO.	FORM TYPE	CONTROL LEVEL	INDICATORS							
				A N D	A N D	A N D	A N D				
1	3	5	6	7	9	11	12	14	15	17	18
	0 1	C			1	0					
	0 2	C				*					
	0 3	C				*					
	0 4	C				*					

indicator 10 is tested only preparatory to executing line 1. The RPG generator assumes that if line 1 is executed, then lines 2, 3, and 4 should be executed also, and produces object code that does so, with no further testing of indicators. If indicator 10 were specified as a resulting indicator on line 2, it is thus possible that the above might not give the results the programmer desires.

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

■ Examples

Figure 4-3 illustrates the conditions that must be present for the associated calculations to be performed. 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.
5. When the overflow indicator is set.

UNIVAC
DIVISION OF SPERRY RAND CORPORATION

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

Figure 4-3. Example of Indicators Entries

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.

4.1.3. Factor 1 (Columns 18–27) and Factor 2 (Columns 33–42)

■ Entry

Required.

■ Purpose

Identifies the predefined field name or "literal" that is to be processed.

■ Literal Definition

A literal is actual data rather than the assigned name which represents the address or location of the data. The various types of literals include the following:

1. *Numeric Literals* – A numeric literal may contain only digits. Alphabets and special characters are not permitted. One algebraic sign, plus (+) or minus (–), and one decimal point 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, if used, may appear anywhere in the literal.
 - d. Numeric literals must not be enclosed in apostrophes.
 - e. Maximum length is ten digits.
2. *Alphanumeric Literals* – An alphanumeric literal is any group of characters or spaces that is enclosed in apostrophes ('). Alphanumeric literals may never be 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 a string of one through six hexadecimal digits preceded by an X` and followed by an apostrophe. Each hexadecimal digit represents a half byte of information (four bits). The hexadecimal digits and their corresponding 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

Examples of hexadecimal literals and their values are as follows:

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

■ Rules

1. The factor specification must be a field name or a literal.
2. The field name in factor must be left justified, and a maximum of six characters is permitted.
3. If the factor entry is a field name, it must be entered on the Input Format Specifications form or defined as a result field (columns 43–48). If the factor entry is defined as a result field, it has an initial value of zero.

■ Examples

Figure 4–4 illustrates several types of factor entries.

4.1.4. Operation (Columns 28–32)

■ Entry

Required.

■ Purpose

Defines the processing step or calculation that must be applied to the specified data. An operation is defined by entering its operation code in the operation field. The operation codes shown in Figure 4–5 may be specified.

4.1.4.1. 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 by 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.

4.1.4.2. 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.

4.1.4.3. 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.

4.1.4.4. 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 the Z-SUB operations, a hyphen is required between the letter Z and the word ADD or SUB.

4.1.4.5. 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 to be 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 will be retained in the product. Any excess digits at the low-order end of the result will be dropped. If, at this point, the result is larger than 15 positions, all digits in excess of 15 will be 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 will be truncated (no error condition is indicated by the RPG in this case):

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

where

L_1 = length in positions of factor 1 (multiplicand)

D_1 = number of decimal positions in factor 1

L_2 = length in positions of factor 2 (multiplier)

D_2 = number of decimal positions in factor 2

D_r = number of decimal positions in the result (product)

4.1.4.6. 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 (MVR) operation is specified on the following line. Factor 2 (divisor) must not contain a value of 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. $L_2 + D_1 - D_2 - D_r \leq 15$ and
2. $L_1 - D_1 + D_2 + D_r \leq 15$ if no half-adjustment, or
3. $L_1 - D_1 + D_2 + D_r \leq 14$ if half-adjustment

where

- L_1 = length in positions of factor 1 (dividend)
- D_1 = number of decimal positions in factor 1
- L_2 = length in positions of factor 2 (divisor)
- 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 will be 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 padded 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, an error will result.

4.1.4.7. 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. 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:

$$\text{Remainder} = \text{Dividend} - (\text{Divisor} \times \text{Quotient})$$

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 not transferred.
3. A numeric field may be moved to an alphanumeric field and vice versa. This may be used for unpacking and packing data.
4. Resulting indicators cannot be specified on a MOVE operation.

4.1.4.10. Move Leftmost Characters (MOVE)

The MOVE 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 MOVE operation are as follows:

1. The MOVE operation does not use the factor 1, half-adjust, and resulting indicators specifications.
2. Both alphanumeric and numeric data may be moved into either type field. If alphanumeric data is moved into a numeric field, then all zones are stripped off except the rightmost zone position. Decimal alignment is not performed.
3. Field length and decimal positions specifications (if result field is numeric) are required in this operation if they have not as yet been defined in the program for the result field.

Results of the MOVE operation for the examples listed below are illustrated in Figure 4-6.

REPORT PROGRAM GENERATOR
CALCULATION SPECIFICATIONS

OPERATION	FACTOR 2	RESULT FIELD	FIELD LENGTH	DECIMAL POSITION	HALF ADJUS	RES	INDI	FACTOR 2	RESULT FIELD (before move)	RESULT FIELD (after move)
						HI	CO			
① MOVE	2,3,4,5,6	AMOUNT	8	3				2,3,4,5,6	8,8,8,8,8,8	2,3,4,5,6,8,8
② MOVE	2,3,4,5,6,7,8	RESULT	5	2				2,3,4,5,6,7,8	3,3,3,3,3	2,3,4,5,6
③ MOVE	RESULT	FINAL	7					2,3,4,5,6	A,A,A,A,A,A,A	2,3,4,5,6,A,A,A
④ MOVE	FLDA	FLDX	5					2,3,4,5,6,7,8	A,A,A,A,A	2,3,4,5,6
⑤ MOVE	'HERE'	FLDY	7					HERE	A,A,A,A,A,A,A	HERE,A,A,A

Figure 4-6. Example of MOVE Operation

1. If the result field is longer than factor 2, the excess (righthand) positions of the result field remain untouched. (See case 1, Figure 4-6.)
2. If the result field is shorter than factor 2, the excess (righthand) positions of factor 2 are not moved. (See case 2.)
3. If factor 2 is shorter than the (numeric) result field, then the sign of factor 2 is not moved. (See case 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 case 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 case 3.)
6. If an alphanumeric result field is shorter than the numeric factor 2 field, then the sign is not moved. (See case 4.)

4.1.4.11. 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. It is usually in the rightmost position of the field.

4.1.4.11.1. 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.

4.1.4.11.2. 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.

4.1.4.11.3. 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.

4.1.4.11.4. 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.

NOTE: Factor 1 cannot be specified in any of the move zone operations.

4.1.4.12. Table Lookup (LOKUP)

The LOKUP operation causes the RPG to search a table contained in memory and secure from it specific data needed in following calculations. The LOKUP operation is used in conjunction with the File Extension Specifications form and is described, together with this form, in complete detail in Section 6 of this manual.

4.1.4.13. Branch (GOTO)

The GOTO statement causes the RPG to transfer control to some other operation which is not next in sequence. This operation may be used as either a conditional or 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 tag line, illustrated in Figure 4-7, must be in the following format:



REPORT PROGRAM GENERATOR
CALCULATION SPECIFICATIONS



PC. NO.	LINE NO.	FORM TYPE	CONTROL LEVEL	INDICATORS			FACTOR 1	OPERATION	FACTOR 2	RESULT FIELD	FIELD LENGTH	DECIMAL POSITIONS	HALF ADJUST	RESULTING INDICATORS			COMMENTS	PROGRAM IDENTIFICATION												
				A	N	D								HI	LO	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	56	58	60	74	75	80	
	0.1	C						GOTO	ISSUE																				UNCONDITIONAL	
	0.2	C					ORDER	TAG																						
	0.3	C																												
	0.4	C																												
	0.5	C						GOTO	ONHAND																					UNCONDITIONAL
	0.6	C					ISSUE	TAG																						
	0.7	C																												
	0.8	C																												
	0.9	C		1.7		1.4		GOTO	ORDER																					CONDITIONAL
	1.0	C					ONHAND	TAG																						
	1.1	C																												
	1.2	C																												
	1.3	C																												
	1.4	CL1					USAGE	TAG																						
	1.5	C																												
	1.6	C																												
	1.7	CL1		1.2	N	1.0		GOTO	ADDUP																					CONDITIONAL

Figure 4-7. Examples of GOTO Operations

1. Factor 1 - the name of the routine, as specified by the GOTO statement.
2. Operation field (columns 28-30) - must contain the constant TAG

■ Rules

General

1. The GOTO operation must *not* be used to transfer control from detail to total or from total to detail calculations.
2. A control level entry (L0-L9) must be specified when a GOTO statement is used in total calculations.

3. 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:

4. 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.
5. The name in factor 2 must be unique and may consist of one through six alphanumeric characters.
6. The first character of the name must be alphabetic and left justified; special characters and blanks may not be used.

Conditional GOTO statements:

7. 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.
8. The name specified in factor 2 is the address to which control is transferred if the conditions indicated in columns 9–17 are satisfied.

■ Examples

Figure 4–7 illustrates various entries for the GOTO operation:

1. The GOTO statement in line 01 transfers control unconditionally to the tag line (ISSUE) in line 06.
2. 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.
3. 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.

4.1.4.14. Exit to Subroutine (EXIT)

The EXIT operation provides the ability to perform subroutines written in the 9200/9300 Assembler language. These subroutines usually perform operations either not available or not easily accomplished in the RPG. Subroutines of this type may consist of routines to compute such things as trigonometric functions or logarithms.

The EXIT operation causes a transfer of control from the RPG program to a subroutine (written in Assembler language). The subroutine is performed, and control is transferred back to the main RPG program.

NOTE: The programmer should possess a thorough understanding of the 9200/9300 Assembler language to use the EXIT operation properly.

■ Rules

1. The mnemonic EXIT must be placed in columns 28–31 of the operation field and the name of the subroutine to which control is to be transferred entered in factor 2 (columns 33–36).
2. The name of the subroutine is limited to four characters. The first character must be alphabetic and left justified; the other characters may be alphabetic or numeric. Special characters or blanks may not be used.
3. This operation may be used as a conditional operation by specifying a control level or indicators (columns 7–17).
4. 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.
 - a. 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.
 - b. 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.
 - c. 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.
 - d. If the EXIT operation is the last total calculation, control is transferred to the indicated subroutine prior to printing or punching totals.

■ Examples

Figure 4–8 shows an example of the EXIT operation usage and format.

UNIVAC
DIVISION OF SPERRY RAND CORPORATION

REPORT PROGRAM GENERATOR
CALCULATION SPECIFICATIONS

PG. NO.	LINE NO.	FORM TYPE	CONTROL LEVEL	INDICATORS		FACTOR 1	OPERATION	FACTOR 2	RESULT FIELD	FIELD LENGTH	DECIMAL POSITION	HALF ADJUST	RES												
				A N D	A N D								INDI												
1	3	5	6	7	9	11	12	14	15	17	18	27	28	32	33	42	43	48	49	51	52	53	54	+	
	0 1	C	L	1	1	6	N	1	0																
	0 2	C																							
	0 3	C			OPTIONAL																				
	0 4	C																							
	0 5	C																							

Figure 4–8. Format of EXIT Operation

4.1.4.15. Result Label (RLABL)

The RLABL statement is used to provide 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. The use of the RLABL statement is described below.

Field Definition – If a field is used in *both* a subroutine and the RPG program, then it must be defined in the RPG program.

■ Rules

1. The format of the RLABL statement consists of the mnemonic RLABL placed in the operation field (columns 28–32), and the name of the field (used in both the subroutine and the RPG) is indicated in the result field (columns 43–46).
2. The field name is limited to four characters; the first character must be alphabetic and be left justified; the other characters may be alphabetic or numeric.
3. Field length (columns 49–51) and decimal positions (column 52) must be specified if the 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 positions entries may be left blank.
4. Columns 7–27, 33–42, and 53–59 must be blank.
5. A field name used in a tag statement must not be used in a subroutine. Therefore, a RLABL statement must not contain a name used in a tag line.
6. All RLABL entries for an RPG program must be placed at the end of the Calculation Specifications form.

■ Examples

Figures 4–9 and 4–10 illustrate RLABL formats.

Indicator Definition: If the condition of RPG result indicators must be known in a subroutine, then those indicators must be defined in an RLABL statement.

■ Rules

1. The entry in columns 43–46 of the result field consists of the letters IN followed by the indicator number or letters.
2. 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.

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.

REPORT PROGRAM GENERATOR
CALCULATION SPECIFICATIONS

OPERATION	FACTOR 2	RESULT FIELD	FIELD LENGTH	DECIMAL POSITION	HALF ADJUST	RESULT INDICA	
						HI 1-2	LO 1-2
27 28	32 33	42 43	48 49 51 52	53	54	56	
R L A B L		F L D A	0 1 0 2				

Figure 4-9. Format of RLABL Operation, Field Not Defined Elsewhere

REPORT PROGRAM GENERATOR
CALCULATION SPECIFICATIONS

OPERATION	FACTOR 2	RESULT FIELD	FIELD LENGTH	DECIMAL POSITION	HALF ADJUST	RESULT INDICA	
						HI 1-2	LO 1-2
27 28	32 33	42 43	48 49 51 52	53	54	56	
R L A B L		F L D B					

Figure 4-10. Format of RLABL Operation, Field Defined Elsewhere

■ Examples

1. If the condition of the matching record indicator is required in the subroutine, then the label INMR is entered in the result field of the RLABL line. The RPG is notified by the RLABL statement that the condition of MR is to be tested in the subroutine.
2. Figure 4-11 illustrates the entry for an indicator definition in an RLABL statement.

4.1.4.15.1. Coding of Subroutines

The EXIT operation may only be linked to subroutines that are coded in Assembler language. A subroutine may have only one entry point, and this must be its first byte. A subroutine that is to be incorporated into an RPG program must not contain input/output operations for devices specified on the File Description form. The programmer should observe the following points concerning the use of registers:

1. Register 14 is used by the RPG to store the return address from the subroutine. The return address is the address of the operation following the EXIT statement and to which control is transferred upon completion of the subroutine.

REPORT PROGRAM GENERATOR
CALCULATION SPECIFICATIONS

OPERATION		FACTOR 2		RESULT FIELD		FIELD LENGTH		DECIMAL POSITION		HALF ADJUST		RESULTING INDICATORS		
												+	-	0
												COMPARE		
												HI	LO	EQ
												>2	<2	1=2
27	28	32	33	42	43	48	49	51	52	53	54	56	58	60
R	L	A	B	L		I	N	M	R					
R	L	A	B	L		I	N	O	I					
R	L	A	B	L		I	N	I	6					

Figure 4-11. Result Label Statement Indicator Definitions

- The subroutine must preserve and restore the contents of the registers it uses.

Subroutines to be incorporated into an RPG program must be assembled separately by the 9200/9300 Assembler. The subroutines must be relocatable and all Assembler linking conventions must be observed. The object program deck of a subroutine is loaded at RPG program generation time.

A field length, defined in an RPG specification, is implied with every use of the field name in the RPG program. This is not the case when the same field name is used in a subroutine. The length of the field must be specified in every statement referring to this field since the length as specified in the RPG program was not available to the subroutine when assembled.

Figure 4-12 illustrates the coding necessary for an RPG program and an associated subroutine that uses SUBA and SUBC fields as defined in the RPG program.

4.1.4.15.2. Subroutine Restrictions

The programmer must observe the following restrictions when using subroutines written in Assembler language.

- Subroutines must not contain input/output operations for devices specified by the RPG program on the File Description form.
- Control cannot be transferred from one subroutine to another.
- Calculation fields defined in a subroutine must not be used in another subroutine.
- Numeric data transferred between the RPG program and a subroutine must be in packed format.

REPORT PROGRAM GENERATOR
CALCULATION SPECIFICATIONS

FACTOR 1	OPERATION	FACTOR 2	RESULT FIELD	FIELD LENGTH	DECIMAL POSITION	HALF ADJUST	RESULTING INDICATORS			COMMENTS					
							+	-	0						
							COMPARE								
							HI 1>2	LO 1<2	EQ 1=2						
27	28	32	33	42	43	48	49	51	52	53	54	56	58	60	
	⋮														
	EXIT	CALC													
	⋮														
	R,LABL		SUBA	8	2										
	⋮														
	R,LABL		SUBC	1	03										

PROGRAM _____ PROGRAMMER _____

1	LABEL	8	OPERATION	14	OPERAND	6
	CALC		START			
			EXITRN		SUBA	
			⋮			
			EXITRN		SUBC	
			⋮			
			INSTRUCTIONS			
			B,C		1,5,0,(,14)	← EXIT
			D,C		Y,(SUBA)	
			⋮			
			D,C		Y,(SUBC)	

Figure 4-12. Sample Coding RPG And Associated Subroutine

4.1.4.16. Tag (TAG)

The TAG entry is referenced and defined under the GOTO operation.

4.1.4.17. 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, 56–57, 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 may also be made a conditional statement through the use of control levels and indicators.

4.1.4.18. 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.

4.1.5. Result Field (Columns 43–48)

- Entry

As required by the operation.

- Purpose

Defines the area in storage where the result of a calculation is to be stored. The units position of the area always contains the algebraic sign when an arithmetic operation is performed.

- Rules

1. The name of the result field may be alphanumeric, but the first character must be alphabetic. Special characters or spaces are not allowed.
2. The field name may contain from one through six characters and must be left justified.
3. 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.

- Examples

Figure 4–13 describes some typical field name entries in the result field.

4.1.6. Field Length (Columns 49–51)

- Entry

Optional.

- Purpose

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

REPORT PROGRAM GENERATOR
CALCULATION SPECIFICATIONS



OPERATION	FACTOR 2	RESULT FIELD	FIELD LENGTH	DECIMAL POSITION	HALF ADJUST	RESULTING INDICATORS			COMMENTS	PROGRAM IDENTIFICATION							
						-	-	0									
						COMPARE											
HI >2	LO <2	EQ =2															
27	28	32	33	42	43	48	49	51	52	53	54	56	58	60	74	75	80

Figure 4-13. Example of Result Field Entries

■ Rules

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

- decimal fields - 15 digits
- alphanumeric field - 256 characters

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

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

4.1.7. Decimal Position (Column 52)

■ Entry

Optional.

■ Purpose

Defines 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. Every decimal result field must have a decimal point specified for it. When the result field of an arithmetic calculation has no decimal 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 be left blank.

4.1.8. 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.

- Codes

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

- Rules

1. This specification must be blank for all alphanumeric fields.

4.1.9. Resulting Indicators (Columns 54–59)

- Entry

Optional.

- Purpose

The resulting indicators specification causes the result of the specified operation to be tested after it is completed. Based on the results of this test, an indicator may be turned on and then be used to control subsequent calculation 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:
 - a. Factor 1 > Factor 2---Set HI indicator (columns 54–55)
 - b. Factor 1 < Factor 2---Set LO indicator (columns 56–57)
 - c. 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 adjusted; it is set if its condition is met; it is reset if the condition is not met.

■ Codes

Indicator codes 01–99 may be specified. Halt indicators H1 and H2 may also be specified. If the operation is SETON or SETOF, control level indication L1–L9 may also be specified.

■ 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 indicators 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 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.

4.1.10. 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.

■ Examples

Figure 4–14 illustrates the following example and entries on 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.

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 will be 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.)

REPORT PROGRAM GENERATOR
CALCULATION SPECIFICATIONS

OPERATORS	FACTOR 1	OPERATION	FACTOR 2	RESULT FIELD	FIELD LENGTH	DECIMAL POSITION	HALF ADJUST	RESULTING INDICATORS													
								+	-	0											
								COMPARE													
AND								HI	LO	EQ											
14	15	17	18	23	27	28	32	33	42	43	48	49	51	52	53	54	56	58	60		
	FIELD A	ADD	FIELD B	FIELD 1	8	2	H											6	5	①	
N 6	5	FIELD C	ADD	FIELD 1	FIELD 1		H													②	
	FIELD D	SUB	FIELD E	FIELD 3	6	0												9	8	9	③
		Z - SUB	FIELD F	FIELD G																	④
	FIELD H	MULT	3,6,2,5	FIELD 4	1	2	3														⑤
	FIELD J	DIV	FIELD K	FIELD 5	1	5	0														⑥
	FIELD L	COMP	4,8,0,0,0,0															7	5		⑦
		MOVE	FIELD M	FIELD 6	1	5	0														⑧
		MOVE	FIELD N	FIELD 7	8	0															⑨

Figure 4-14. Example of a Completed Calculation Specification Form

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. FIELD F is stored with reversed sign in FIELD G, as defined on the Input Format Specifications form.
5. FIELD H is multiplied by the numeric literal 3625, and the result is stored in FIELD 4, containing three decimal places.
6. FIELD J is divided by FIELD K and the result is stored in FIELD 5. Note required entry of zero decimal places.
7. FIELD L is compared to the literal 480000. If FIELD L is algebraically greater than or equal to 480000, indicator 75 is set.
8. A numeric field is moved from FIELD M to FIELD 6.
9. An alphanumeric field is moved from FIELD N to FIELD 7. (No entry in the decimal positions specification indicates an alphanumeric field.)

5. OUTPUT FORMAT SPECIFICATIONS FORM

The Output Format Specifications form (Figure 5-1) defines the types of output which are to be produced on the printer or punch units and also specifies the location of the specific output fields. 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 and punch card files that are to be produced. These specifications govern the spacing on printed reports and stacker selection of cards processed through the punch. This section also controls the conditions under which printed and punched 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 and punched. Each file identification line entered on the output format specification sheet 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.

UNIVAC <small>DIVISION OF ELECTRIC BELL CORPORATION</small>		REPORT PROGRAM GENERATOR OUTPUT FORMAT SPECIFICATIONS										UNIVAC 9000 SERIES				
PROGRAM _____		PROGRAMMER _____										DATE _____		PAGE _____ OF _____ PAGES		
PG. NO.	LINE NO.	FILE NAME	TYPE H/D/T	SPACE	SKIP	OUTPUT INDICATORS		FIELD NAME	ZERO SUPPRESS	BLANK AFTER	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
0, 1	O															
0, 2	O															
0, 3	O															
0, 4	O	FILE IDENTIFICATION AND CONTROL														
0, 5	O															
0, 6	O															
0, 7	O															
0, 8	O															
0, 9	O															

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 overflow headings first, followed by the specifications for the detail lines, and finally the required specifications for all control levels. In fact, files must be listed on the output form in this order. Moreover, in any one grouping, files should be listed in the order 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.

If the object program is to produce both printed and punched output, it is advantageous from an object program running time point of view to specify the punching before the printing.

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



PG. NO.	LINE NO.	FILE NAME	TYPE		STACKER SELECT / PRINT OPTION								ZERO SUPPRESS			BLANK AFTER		DATA FORMAT		STERLING SIGN POSITION	PROGRAM IDENTIFICATION											
			N/D/T		SPACE	SKIP	OUTPUT INDICATORS				FIELD NAME	END POSITION IN OUTPUT RECORD	CONSTANT OR EDIT WORD																			
7	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
0	1	0																														
	2	0																														
	3	0																														
0	4	0																														
	5	0																														
	6	0																														
	7	0																														
0	8	0																														
	9	0																														
	10	0																														
	11	0																														
	12	0																														
	13	0																														
	14	0																														
	15	0																														
	16	0																														
	17	0																														
	18	0																														
	19	0																														
	20	0																														
	21	0																														
	22	0																														
	23	0																														

Figure 5-2. Sequence of File Entries

5.1. FILE IDENTIFICATION AND CONTROL FORM ENTRIES

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

5.1.1. File Name (Columns 7-14)

- **Entry**
Required.
- **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 and punching operation as specified on the File Description form. Consequently, one file name would be assigned to the printing operation, and a different file name to the punching operation.
- **Rules**
 1. A file name must be left justified and must consist of alphanumeric characters.
 2. Blanks and special characters are not permitted.
 3. The corresponding file name must be entered whenever a new output operation is specified.

- **Examples**
Figure 5-3 illustrates several file name entries.



REPORT PROGRAM GENERATOR
OUTPUT FORMAT SPECIFICATIONS

PG. NO.	LINE NO.	FILE NAME	TYPE H/D/T	STACKER SELECT			OUTPUT INDICATORS	FIELD NAME	ZERO SUPPRESS	BLANK AFTER	END POSITION IN OUTPUT RECORD	DATA FORMAT	CONSTANT												
				SPACE	SKIP	SKIP																			
				B E F O R E	A F T E R	A F T E R	A N D	A N D																	
				17	18	19	20	21	22	23	25	26	28	29	31	32	37	38	39	40	43	44	45		
	0 1	OPRINTRPT																							
	0 2	O																							
	0 3	OPRINTRPT																							
	0 4	O																							
	0 5	OPUNHCRD																							

Figure 5-3. Examples of File Name Entries

5.1.2. Type (Column 15)

- **Entry**
Required.
- **Purpose**
Classifies the file entry by type: heading, detail, or total specification.

■ Codes

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

D – Detail. Generally the information specified by the detail specification uses the input record as its source, or it is calculated within the program at detail time.

T – Total. Before detail input fields are operated on, a test for a control break occurs. 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. This output specification defines total time operations which are initiated when a control break occurs.

■ Rules

- The following sequence is required for file name entries.
 - heading entries
 - detail entries
 - total entries
- 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 Format Specifications form relates a file name directly to either the printer or the punch.

■ Examples

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



REPORT PROGRAM GENERATOR
OUTPUT FORMAT SPECIFICATIONS

PG. NO.	LINE NO.	FILE NAME	TYPE H/D/T	STACKER SELECT		OUTPUT INDICATORS						FIELD NAME	ZERO SUPPRESS	BLANK AFTER	END POSITION IN OUTPUT RECORD	DATA FORMAT	CONSTAN												
				SPACE	SKIP	A	N	D	A	N	D																		
1	3	5	7	14	15	16	17	18	19	20	21	22	23	25	26	28	29	31	32	37	38	39	40	43	44	45			
	0, 1	O P, R, N, T, R, P, T	H																										HEADING LINE
	0, 2	O																											
	0, 3	O P, R, N, T, R, P, T	D																										DETAIL LINE
	0, 4	O																											
	0, 5	O P, U, N, H, C, R, D	D																										DETAIL CARD
	0, 6	O																											
	0, 7	O P, R, N, T, R, P, T	T																										TOTAL LINE
	0, 8	O																											
	0, 9	O P, U, N, H, C, R, D	T																										TOTAL CARD
	1, 0	O																											
	1, 1	O P, R, N, T, R, P, T	T																										TOTAL LINE

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

5.1.3. Stacker Select/Print Option (Column 16)

■ Entry

Optional.

■ Purpose

This entry is used as either a punch or print specification.

1. *Punch Specification* – The stacker select specification permits cards on the punch to be selected into stacker 2 if the punch is equipped with the stacker select option.
2. *Print Specification* – The print option specification permits the programmer to specify which print rate, 250 lpm alphanumeric or 500 lpm numeric, is to be used with a print order.

A 48-character bar printer is available as an option with the 9200 processor. It allows a 250 lpm alphanumeric print rate and a 500 lpm numeric rate. If the 500 lpm numeric rate is used, then only the following 16 characters can be printed:

0123456789.\$* - / ,

■ Codes

Stacker Selection:

Blank – cards to Stacker 1
2 – cards to Stacker 2

Print Option:

Blank – 250 lpm with 48-character set or 63-character set
N – 500 lpm with 16-character set

■ Rules

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.

■ Examples

Figure 5-5 illustrates entries for this option.

5.1.4. Space (Columns 17-18)

■ Entry

Conditional; required only for output files to be printed.

■ Purpose

Controls the spacing of printed reports.

UNIVAC
DIVISION OF SPERRY RAND CORPORATION

PG. NO.	FORM TYPE		FILE NAME	TYPE H/D/T			STACKER SELECT			
	LINE NO.						SPACE	A	B	
1	3	5 6 7		14	15	16	17	18	19	
	0, 1,		OP,R,N,T,R,P,T,			H				PRINT ALPHANUMERIC OPTION
	0, 2,		O							PRINT NUMERIC OPTION
	0, 3,		OP,R,N,T,R,P,T,			D	N			DETAIL CARD TO STACKER ONE
	0, 4,		O							PRINT NUMERIC OPTION
	0, 5,		OP,U,N,H,C,R,D,			D				TOTAL STACK TO STACKER TWO
	0, 6,		O							PRINT NUMERIC OPTION
	0, 7,		OP,R,N,T,R,P,T,			T	N			PRINT NUMERIC OPTION
	0, 8,		O							PRINT NUMERIC OPTION
	0, 9,		OP,U,N,H,C,R,D,			T	2			PRINT NUMERIC OPTION
	1, 0,		O							PRINT NUMERIC OPTION
	1, 1,		OP,R,N,T,R,P,T,			T	N			PRINT NUMERIC OPTION

Figure 5-5. Example of Entries for Stacker Select/Print Option

■ Codes

Space Before (Column 17).

- 0 – no space
- 1 – single space
- 2 – double space

Space After (Column 18).

- 0 – no space
- 1 – single space
- 2 – double space

■ Rules

1. Column 18 must contain an entry for every record that is to be printed
2. If column 17 is left blank, the RPG considers a zero to have been entered.
3. See rules for skip entries.

■ Examples

Figure 5-6 presents samples of valid spacing entries.

5.1.5. Skip (Columns 19-22)

■ Entry

Conditional

■ Purpose

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

■ Codes

Skip before (Columns 19–20) – The entries 01 through 07 in the specification form cause the form to advance to the specified channel before printing occurs.

Skip after (Columns 21–22) – The entries 01 through 07 in this area of the specification form cause the form to advance to the specified channel after printing occurs.

NOTE: If paper is to be skipped at a certain point, then it cannot be spaced at that point, nor can paper be skipped if spacing is specified. This means that both spacing and skipping cannot be specified before printing. The same applies to spacing and skipping after printing.

From an execution time point of view, it is advantageous to advance paper after printing rather than before.

■ Rules

If, in an OR relationship, Space and Skip specifications are entered in the first line, then none need be entered in the subsequent lines, provided the specifications in the first line are to be applicable for all subsequent lines. Otherwise, all the rules given for spacing and skipping specifications must be followed.

■ Examples

Figure 5–6 presents samples of skip entries.

UNIVAC
DIVISION OF SPERRY RAND CORPORATION

REPORT PROGRAM GENERATOR
OUTPUT FORMAT SPECIFICATIONS

PG. NO.	LINE NO.	FORM TYPE	FILE NAME	TYPE H/D/T	STACKER SELECT							ZERO SUPPRESS		BLANK AFTER		DATA FORMAT		CONSTANT													
					SPACE			SKIP				OUTPUT INDICATORS		FIELD NAME	END POSITION IN OUTPUT RECORD	44	45														
					B E F O R E	A F T E R	H I S T O R Y	B E F O R E	A F T E R	A N D	A N D	31	32						37	38	39	40	43	44	45						
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				
	0 1		O P R N T R P T	H					0	7	0	2																			
	0 2		O																												
	0 3		O P R N T R P T	D				1																							
	0 4		O																												
	0 5		O P R N T R P T	T	2	2																									
	0 6		O																												
	0 7		O P R N T R P T	T	2				0	7																					
	0 8		O																												

Figure 5–6. Examples of Space and Skip Entries

5.1.6. Output Indicators (Columns 23–31)

■ Entry

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

■ Purpose

To determine 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 had 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 produced 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 (generally 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 1P (first page) indicator (turned on when the RPG object program is initiated and permanently turned off at the end of detail time of the first object program processing cycle) 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 specifies that the output operation is to be performed only if there is a matching record in a secondary input file.
8. The OF indicator specifies output to be produced only when form overflow has occurred. The OF indication is generally used to produce headings on the printer. A channel-1 punch in the printer carriage control tape designates a control point associated with the end of a page. The detection of this punch in the carriage control tape is referred to as form overflow. When a channel-1 punch is detected on the carriage control tape, the fact is noted, and at the end of this output time (total or detail) the OF (overflow) indicator is turned on. If this occurred during detail time, the OF indicator remains on for one

complete cycle; it is turned off following the next detail time. If it occurred during total time, the OF 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 OF indicator, automatic skipping to channel 7 (home paper position) is provided whenever the overflow condition is detected.

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

- OF – Overflow indicator
- 1P – First page
- L0 – Level zero indicator
- L1 – L9 – Control level indicators
- LR – Last record
- 01 – 99 – Record indicators, field indicators, resulting indicators
- MR – Matching record indicator

■ Rules

1. A maximum of three indicators may be entered on a line. At least one must be specified; the specification may not be left blank. 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 next specification line, and the additional indicators are entered in the output indicator area of this line. Columns 17–22 must be blank.
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 the OR indicator entered on this 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.
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.

5.2.1. Output Indicators (Columns 23-31)

■ Entry

Optional.

■ Purpose

Specifies the point during output processing that a field will be printed or punched. The same types of indicators previously specified for file identification may also be specified in defining individual fields.

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 type. Time and space are saved in utilizing this technique. 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. For example, in the following

PG. NO.	LINE NO.	FORM TYPE	FILE NAME	TYPE H/D/T	STACKER SELECT /PRINT OPTION																
					SPACE		SKIP		OUTPUT INDICATORS												
					B E F O R E	A F T E R	B E F O R E	A F T E R	A N D	A N D	A N D	A N D									
1	3	5	6	7	14	15	16	17	18	19	20	21	22	23	25	26	28	29	31	32	
	0 1	0														1	0				
	0 2	0														*					
	0 3	0														*		1	1		
	0 4	0														*					

none of lines 1-4 will be executed if indicator 10 is off; lines 1 and 2 will be executed if indicator 10 is on; and lines 3 and 4 will be executed only if indicators 10 and 11 are on. The two instances in which this asterisk convention cannot be used are as follows:

- a. The first line of a group of field description lines.
- b. A total time line if the preceding line is a detail line.

5.2.2. Field Name (Columns 32-37)

■ Entry

Required unless the output is to be a constant.

■ Purpose

Defines the specific field that is to be printed or punched.

■ Rules

1. Only names previously defined on 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. Field name is blank when a constant is specified in columns 45–70.

■ Examples

See Figure 5–8.



REPORT PROGRAM GENERATOR
OUTPUT FORMAT SPECIFICATIONS

PG. NO.	LINE NO.	FORM TYPE	FILE NAME	TYPE H/D/T	STACKER SELECT		OUTPUT INDICATORS		FIELD NAME	ZERO SUPPRESS	BLANK AFTER	END POSITION IN OUTPUT RECORD	DATA FORMAT	CONSTANT OR EDIT WORD																
					SPACE	SKIP	AND	AND																						
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						
0	1	O	PRNTRPT	H			0	7	0	2	O	F																		
0	2	O		OR																										
0	3	O																					1	8	'FIELD ONE HEADER'					
0	4	O																						2	7	'FLD,2'				
0	5	O																						3	6	'FLD,3'				
0	6	O																						4	9	'FIELD,4'				
0	7	O																							6	4	'FIELD,5'			
0	8	O																							8	2	'FIELD, SIX'			
0	9	O																							9	8	'FIELD, 7'			
0	10	O																							1	1	6	'FIELD, 8'		
1	1	O	PRNTRPT	D			1					1	0																	
1	2	O																								2	0	FIELD,1		
1	3	O																									2	8	FIELD,2	
1	4	O																									3	8	FIELD,3	
1	5	O																									5	2	FIELD,4 Z	
1	6	O																									6	8	FIELD,5 Z	
1	7	O																									8	5	' , \$0. & CR'	
1	8	O																									1	0	2	' , \$0. & -'
1	9	O																									1	2	0	' , \$0. & CR'
2	0	O	PUNHCRD	D								1	0	N	2	1	N	2	2											
2	1	O																												
2	2	O																												
2	3	O																												
2	4	O																												
2	5	O																												

Figure 5–8. Example of Field Description Entries

REPORT PROGRAM GENERATOR
OUTPUT FORMAT SPECIFICATIONS

PG. NO.	LINE NO.	FORM TYPE	FILE NAME	TYPE H/D/T	STACKER SELECT		OUTPUT INDICATORS					FIELD NAME	ZERO SUPPRESS	BLANK AFTER	END POSITION IN OUTPUT RECORD	DATA FORMAT	CONSTANT OR EDIT WORD																
					SPACE	SKIP	AFTER	BEFORE	AFTER	BEFORE	AND							AND															
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						
0	1	O	PRNTRPT	T		2	2								L1																		
0	2	O													FIELDP									2	5	'			\$0.		&CR**'		
0	3	O													FIELDN									4	5	'			\$0.		&CR**'		
0	4	O													FIELDF	B								7	0	'			\$0.		&CR**'		
0	5	O													FIELDG	B								9	0	'			\$0.		&CR**'		
0	6	O													FIELDH	B								1	1	0	'			\$0.		&CR**'	
0	7	O	PUNHCRD	T	2										L1																		
0	8	O																						3	0	'			1,2,				
0	9	O													FIELDP	B								6	0								
1	0	O													FIELDN	B								8	0								
1	1	O	PRNTRPT	T		2			0	7					LR																		
1	2	O													FIELDX									6	5	'			\$0.		&CR**'		
1	3	O													FIELDY									8	5	'			\$0.		&CR**'		
1	4	O													FIELDZ									1	0	5	'			\$0.		&CR**'	
1	5	O																															
1	6	O																															
1	7	O																															

Figure 5-8. Example of Field Description Entries (cont.)

NOTE: Page numbering may be accomplished automatically 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 end position in output record.

Rules

1. PAGE is a restricted name. It may be used in field name only to indicate page numbering.
2. Pages are numbered from 1-99999. The RPG object program allocates a five-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.
3. 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 would contain the desired starting page number less one because the page number is always increased by one before printing occurs.
4. Any output indicators specified in a PAGE line are checked before printing. If all indicators are on, the page counter is reset to one, instead of being incremented by one.

5.2.3. 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.

- Codes

The letter Z is entered in column 38 to specify 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 will print as an alphabetic letter or special character. These same fields, when specified for punching, generally should not specify zero suppression so that the sign position will be punched into the card.
2. The zero suppress column must be blank if an edit word is specified for a field.

- Examples

See Figure 5–8.

5.2.4. Blank After (Column 39)

- Entry

Optional.

- Purpose

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

- Codes

The letter B is entered in column 39 to specify 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; furthermore, it eliminates duplicates.

- Examples

See Figure 5–8.

5.2.5. End Position In Output Record (Columns 40–43)

- Entry

Required.

- Purpose

Defines the print position or card column of the rightmost character of the output field.

- Rules

Entries are right justified; leading zeros may be omitted.

- Examples

See Figure 5-8.

5.2.6. Data Format (Column 44)

- Entry

Optional; used only for packed format punched output.

- Purpose

Specifies that a field is to be punched in packed or unpacked format.

- Codes

P – packed format

Blank – unpacked format

- Rules

Individual fields may be punched in either packed or unpacked format as required.

- Examples

See Figure 5-8.

5.2.7. Constant or Edit Word (Columns 45-70)

- Entry

Optional.

- Purpose

Provides for the creation of constants that may be printed or punched, and editing of numeric fields on printed reports.

- Rules

Constants and edit words must be left justified.

5.2.7.1. 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 up to 22 hexadecimal digits preceded by X` followed by an apostrophe sign ('). Each hexadecimal digit represents a half byte of information. The hexadecimal digits and their values are as follows:

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

*Each hexadecimal digit represents a half byte of information.

Some examples of hexadecimal constants and their values are as follows:

Hexadecimal Representation	Binary Value
X`D`	00001101
X`101`	00000001 00000001
X`3FFF`	00111111 11111111

5.2.7.2. Edit Word

■ Purpose

Permits the programmer to punctuate a print field with commas, a decimal point, and preceding dollar sign; to suppress leading zeros; and to express signed conditions of the field. The edit word consists of two parts: body and status.

1. Body – That portion which begins with the leftmost character of the edit word and continues to the right to the character controlling the transfer of the units position of the data field.

The body of an edit word is made up of two kinds of characters, digit select characters and other characters. Digit select characters indicate positions in the body into which digits of the field are to be inserted. (Digits from the field replace digit select characters in the body from left to right.) Characters other than digit select characters retain their position in the body. (An exception is the ampersand, which is replaced by a space. This is also true of ampersands appearing in the status.)

All spaces are digit select characters.

As each digit is moved from the field to the body, it is inspected to determine if its value is zero. If it is, it is suppressed; that is, it is replaced by the fill character (defined below). This process of suppression continues until whichever of the two following conditions occurs first:

- a. A nonzero digit is detected in the field.
- b. The significance start character (defined below) in the body is replaced by the fill character.

At this point the suppression process ceases, and the remaining digits of the field are moved to the body without any further screening. All characters other than digit select characters that lie in the body to the left of the point where suppression stops are also suppressed.

Moving from the left to right across the body, the first zero or asterisk encountered 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.

If a dollar sign appears in the body immediately to the left of the significance start character, a dollar sign is floated to the position immediately preceding the first nonsuppressed character in the output field. The dollar sign as it originally appears in the body is also a digit select character. If the significance start character is an asterisk, then a dollar sign cannot be floated.

2. Status – That portion of the word to the right of the body that continues through the credit symbol (CR) or minus sign (-), if used. The status portion of the edit word is undisturbed if the data field is negative. If the data field is positive, the status is replaced by spaces.

Asterisks may appear to the right of the status in an edit word. They are undisturbed by the edit operation and are usually used to indicate a level of total.

■ Examples

Figure 5-8 presents examples of constant and edit word entries in various situations. The following text describes the examples shown:

1. This header line shows examples of constant specifications. The constant FIELD ONE HEADER is printed in the output record ending at print position 18. Similar print instruction information is included in all the other column titles. The field name specification must be blank for constants. Column 44 is blank to indicate unpacked format.
2. This detail print line has eight specified fields. The first three are alphanumeric and are printed right justified as indicated by the column specified in the end position field. The next two are numeric and are zero suppressed. The last three fields are numeric and have edit words specified for zero suppression, dollar sign, commas, decimal point, and credit inserts, if necessary.
3. This detail punched card shows the punching of four fields. Note that the constant 11 will be punched in columns 29 and 30 in each card of this file as a card code. Note that FIELD8 will be punched in columns 73 through 80 but not suppressed as on the printed line.
4. This total print line shows five edited fields. The last three fields will be reset to zeros because of the B in the blank after field.
5. This total punched card shows a constant 12 in columns 29 and 30 for a card code. The B in the blank after field will reset the two fields to zero after punching.
6. This total print line shows the printing of three final totals to complete the report. The fields are edited.

5.2.8. Sterling Sign Position (Columns 71-74)

■ Entry

Conditional; used only when sterling currency is being processed.

■ Purpose

The user should read and familiarize himself with Appendix D before using this specification.

■ Codes

S – is entered in column 74 for printed output. For punched output the letter S is entered in column 74 if the sterling output field is signed as specified under the heading entitled “Format 2” in Appendix D. If the sterling output field does not conform to the sign specifications (indicated in Appendix D), then the position of the sign must be indicated right justified in columns 71–74.

■ Rules

1. If a field is defined as sterling on input but is not defined 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. Zero suppression will be, in effect, in the tens position of the shillings and pence fields. The sign is associated with the least significant pence position (integral or fractional).
3. The rules governing the use of edit control words are identical to those for decimal fields.

6. FILE EXTENSION SPECIFICATIONS FORM

The File Extension Specifications form (Figure 6-1) is used by the RPG program to identify table input files which are subsequently referenced in the object program through the LOKUP operation.

■ Definitions

Table – a collection of data stored in memory in a form suitable for ready reference, usually in sequential machine locations.

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, the location for the value of the function is determined. The argument is more commonly known as the search value used to locate the desired value necessary for calculation.

UNIVAC
DIVISION OF SPERRY RAND CORPORATION

REPORT PROGRAM GENERATOR
FILE EXTENSION SPECIFICATIONS

UNIVAC
9000
SERIES

PROGRAM _____ PROGRAMMER _____ DATE _____ PAGE _____ OF _____ PAGES

PG. NO.	LINE NO.	FORM TYPE	TABLE NAME		NUMBER OF TABLE ENTRIES PER RECORD		NUMBER OF TABLE ENTRIES PER TABLE		LENGTH OF TABLE ENTRY		NUMERIC SEQUENCE		TABLE NAME		LENGTH OF TABLE ENTRY		NUMERIC SEQUENCE		COMMENTS	PROGRAM IDENTIFICATION			
			26	27	32	33	35	36	39	40	42	43	44	45	46	51	52	54		55	56	57	58
1	3	5	6	7																			
	0, 1	E																					
	0, 2	E																					
	0, 3	E																					
	0, 4	E																					
	0, 5	E																					
	0, 6	E																					
	0, 7	E																					
	0, 8	E																					
	0, 9	E																					
	1, 0	E																					
	1, 1	E																					
	1, 2	E																					
	1, 3	E																					
	1, 4	E																					
	1, 5	E																					
	1, 6	E																					
	1, 7	E																					
	1, 8	E																					
	1, 9	E																					
	2, 0	E																					
	2, 1	E																					
	2, 2	E																					
	2, 3	E																					
	2, 4	E																					
	2, 5	E																					

Figure 6-1. File Extension Specifications Form

Function – A value corresponding to an argument, stated or implied. It is obtained from a table of values in memory.

Tables used in the LOKUP operation may consist of either arguments or functions. They may be ordered in either ascending or descending sequence or they may have no sequence at all. The tables used in this operation are loaded into memory storage at program generation time.

As illustrated in Figure 6–2, 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

The rules for preparing a table input file are as follows:

1. Each table must be assigned a unique name which can be a maximum of six characters long. The first three characters must be TAB, and the remaining three are optional alphabetic or numeric.
2. The first table entry must start in column 1 of the input card record.
3. Every record (card) of a table input deck except the last must contain the same number of entries.
4. Each table entry must follow the preceding entry with no blank columns between entries.
5. All arguments of a table must be identical in size. Also, all functions of a table must be the same size. Finally, entries may not spill over from one record to another.
6. For alternating formats, each input record card must begin with the same type of table entry.
7. Entries may be alphanumeric or numeric.
8. Alphanumeric entries must not exceed 80 characters.
9. The maximum size of numeric entries is 15 digits.

The general rules for File Extension Specifications entries are as follows:

1. When a single (nonalternating) format is required, columns 27–42, 44, and 45 are used.
2. When an alternating input format is specified, columns 27–42, 44–54, 56, and 57 are used.

FUNCTION								FUNCTION								FUNCTION								FUNCTION								FUNCTION								FUNCTION								FUNCTION															
0000000000								0000000000								0000000000								0000000000								0000000000								0000000000								0000000000								0000000000							
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20								21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40								41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60								61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80								81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 00																															
1111111111								1111111111								1111111111								1111111111								1111111111								1111111111								1111111111								1111111111							
ARGUMENT								ARGUMENT								ARGUMENT								ARGUMENT								ARGUMENT								ARGUMENT								ARGUMENT															
00000000								00000000								00000000								00000000								00000000								00000000								00000000								00000000							
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20								21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40								41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60								61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80								81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 00																															
11111111								11111111								11111111								11111111								11111111								11111111								11111111								11111111							
FUNCTION		ARGUMENT	FUNCTION		ARGUMENT	FUNCTION		ARGUMENT	FUNCTION		ARGUMENT	FUNCTION		ARGUMENT	FUNCTION		ARGUMENT	FUNCTION		ARGUMENT	FUNCTION		ARGUMENT	FUNCTION		ARGUMENT																																					
00000000		00000000	00000000		00000000	00000000		00000000	00000000		00000000	00000000		00000000	00000000		00000000	00000000		00000000	00000000		00000000	00000000		00000000																																					
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20		21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40	41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60		61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80	81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 00		00000000	00000000		00000000	00000000		00000000	00000000		00000000	00000000		00000000	00000000		00000000	00000000		00000000																																					
11111111		11111111	11111111		11111111	11111111		11111111	11111111		11111111	11111111		11111111	11111111		11111111	11111111		11111111	11111111		11111111	11111111		11111111																																					
ARGUMENT		FUNCTION	ARGUMENT		FUNCTION	ARGUMENT		FUNCTION	ARGUMENT		FUNCTION	ARGUMENT		FUNCTION	ARGUMENT		FUNCTION	ARGUMENT		FUNCTION	ARGUMENT		FUNCTION	ARGUMENT		FUNCTION																																					
00000000		00000000	00000000		00000000	00000000		00000000	00000000		00000000	00000000		00000000	00000000		00000000	00000000		00000000	00000000		00000000	00000000		00000000																																					
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20		21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40	41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60		61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80	81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 00		00000000	00000000		00000000	00000000		00000000	00000000		00000000	00000000		00000000	00000000		00000000	00000000		00000000																																					
11111111		11111111	11111111		11111111	11111111		11111111	11111111		11111111	11111111		11111111	11111111		11111111	11111111		11111111	11111111		11111111	11111111		11111111																																					
22222222		22222222	22222222		22222222	22222222		22222222	22222222		22222222	22222222		22222222	22222222		22222222	22222222		22222222	22222222		22222222	22222222		22222222																																					
33333333		33333333	33333333		33333333	33333333		33333333	33333333		33333333	33333333		33333333	33333333		33333333	33333333		33333333	33333333		33333333	33333333		33333333																																					
44444444		44444444	44444444		44444444	44444444		44444444	44444444		44444444	44444444		44444444	44444444		44444444	44444444		44444444	44444444		44444444	44444444		44444444																																					
55555555		55555555	55555555		55555555	55555555		55555555	55555555		55555555	55555555		55555555	55555555		55555555	55555555		55555555	55555555		55555555	55555555		55555555																																					
66666666		66666666	66666666		66666666	66666666		66666666	66666666		66666666	66666666		66666666	66666666		66666666	66666666		66666666	66666666		66666666	66666666		66666666																																					
77777777		77777777	77777777		77777777	77777777		77777777	77777777		77777777	77777777		77777777	77777777		77777777	77777777		77777777	77777777		77777777	77777777		77777777																																					
88888888		88888888	88888888		88888888	88888888		88888888	88888888		88888888	88888888		88888888	88888888		88888888	88888888		88888888	88888888		88888888	88888888		88888888																																					
99999999		99999999	99999999		99999999	99999999		99999999	99999999		99999999	99999999		99999999	99999999		99999999	99999999		99999999	99999999		99999999	99999999		99999999																																					
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20		21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40	41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60		61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80	81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 00		00000000	00000000		00000000	00000000		00000000	00000000		00000000	00000000		00000000	00000000		00000000	00000000		00000000																																					

Figure 6-2. Example of Table Input Record Formats

6.1. FILE EXTENSION SPECIFICATIONS FORM ENTRIES

The following definitions explain the entries for the File Extension Specifications form and their relation to the RPG program.

6.1.1. Table Name (Columns 27-32)

- Entry
Required.
- Purpose
The name of the argument or function table that appears first on the table input card record is indicated by this specification.

6.1.2. Number of Table Entries Per Record (Columns 33–35)

- Entry
Required.
- Purpose
Defines the number of entries of the specified table contained in one card record.
- Rules
The number must be right justified.

6.1.3. Number of Table Entries Per Table (Columns 36–39)

- Entry
Required.
- Purpose
Contains the total number of entries for the specified table.
- Rules
The number must be right justified.

6.1.4. Length of Table Entry (Columns 40–42)

- Entry
Required.
- Purpose
Defines the number of columns of each entry of the table specified.
- Rules
The number must be right justified.

6.1.5. Numeric (Column 44)

- Entry
Conditional.
- Purpose
Indicates the number of decimal positions for numeric table entries.
- Rules
 1. For numeric fields, a decimal position (0–9) must be indicated.
 2. For alphanumerics, this field must be blank.

6.1.6. Sequence (Column 45)

- Entry
Conditional.
- Purpose
Indicates the sequence of the corresponding table.

- Codes
 - A – Ascending order
 - D – Descending order
 - Blank – Not in sequence.

6.1.7. Table Name (Columns 46–51)

- Entry
 - Optional; used for alternating formats.
- Purpose
 - The name of the argument or function table that appears last on the table input record is indicated by this entry.

6.1.8. Length of Table Entry (Columns 52–54)

- Entry
 - Optional; used for alternating formats.
- Purpose
 - Defines the number of columns or length of each entry of the table specified.
- Rules
 - The number must be right justified.

6.1.9. Numeric (Column 56)

- Entry
 - Conditional.
- Purpose
 - Indicates the number of decimal positions for numeric table entries.
- Rules
 - 1. For numeric fields, a decimal position (0–9) must be indicated.
 - 2. For alphanumerics, this field must be left blank.

6.1.10. Sequence (Column 57)

- Entry
 - Conditional.
- Purpose
 - Indicates the sequence of the corresponding table.
- Codes
 - A – Ascending order
 - D – Descending order
 - Blank – Not in sequence.

6.2. LOKUP PROCEDURE

The LOKUP operation causes the RPG to search a table contained in memory and secure from it specific data needed in following calculations. The LOKUP operation may be used with two tables (alternating input format) or with a single table.

6.2.1. Single Table

The RPG searches the table specified in factor 2 for that value indicated in factor 1 which fits the conditions specified in the resulting indicators. If the value is located, then the specified resulting indicators are turned on. Figure 6-3 illustrates the entries required for the single-table LOKUP operation.

UNIVAC
DIVISION OF ELECTRIC BOND CORPORATION

REPORT PROGRAM GENERATOR
CALCULATION SPECIFICATIONS

PG. NO.	LINE NO.	FORM TYPE	CONTROL LEVEL	INDICATORS		FACTOR 1	OPERATION	FACTOR 2	RESULT FIELD	FIELD LENGTH	DECIMAL POSITION	HALF ADJUST	RESULTING INDICATORS												
				A	N								H	L	E										
				11	12	14	15	17	18	27	28	32	33	42	43	48	49	51	52	53	54	56	58	60	
	0, 1	C																							
	0, 2	C																							
	0, 3	C																							
	0, 4	C		1, 6	N, 1, 0	P, A, R, T, N, O	L, O, K, U, P	T, A, B, P, N, O	T, A, B, Q, T, Y												1, 7			1, 6	
	0, 5	C																							
	0, 6	C																							
	10, 9	C																							
	11, 0	C				P, A, R, T, N, O	L, O, K, U, P	T, A, B, P, N, O																	
	11, 1	C																							

search value

argument table

function table

TWO TABLE ENTRY

SINGLE TABLE ENTRY

Figure 6-3. Examples of LOKUP Operation Formats

6.2.2. Two Tables

The programmer specifies in factor 1 the field name or literal (alphanumeric or numeric) which the RPG uses as the search value (argument). The RPG searches the argument table specified in factor 2 for the value which fits the conditions indicated in the high, low, or equal resulting indicators (columns 54-59). If the search value is found in the argument table, then the corresponding value in the function table (specified in the result field) is made available for use in calculations. Subsequent operations may reference this value by using the name of the function table. Also, the resulting indicators high, low, or equal that were specified in the LOKUP operation are turned on. These indicators can now be used to control calculation and output operations.

6.2.3. 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 in the range 01-99 may be used. If an equal indicator (columns 58-59) is specified, the RPG will attempt to locate a table entry equal to the search value or argument. An indicator in low (columns 56-57) causes the RPG to search for that table entry which is nearest to but smaller than the search value. A specified high indicator 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.

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.

■ Rules

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 equal in length to 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.

Examples of the above rules are shown in Figure 6-3.

NOTE: The following points should be considered when the table lookup operation is used in a program:

1. The argument table name is always specified in factor 2 of a LOKUP operation.
2. If a table name is used in factor 1 or factor 2 of any operation except LOKUP, the data used for this table name will be the data that was located or used by the last performed LOKUP operation referring to this table.

3. If a table name is specified as the search value (factor 1) of a LOKUP operation, then the data used for factor 1 will be the data that was located or used on the last performed LOKUP operation referring to this table.
4. LOKUP and RLABL are the only operations that may specify a table name in the result field.
5. 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.
6. Table names referred to in a subroutine through an RLABL statement must be in a special format. The table name must not exceed four characters in length. The first three characters must be TAB, and the remaining character may be alphabetic or numeric (special character or blank not permitted).

■ Examples

Figure 6-4 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 TABNO entries are eight numeric positions in length and are in ascending sequence. Each TABQTY function entry is four numeric positions long.

The cost of inventory will be 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, will be used as the 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.

The RPG searches the argument table (TABPNO) until it locates a part number equal to the search value (PARTNO) as specified by the indicator in columns 58-59. Then, the corresponding function (or quantity) from the table TABQTY is delivered to the result field for subsequent calculations, and indicator 01 is turned on. If indicator 01 is on, the COST field will be multiplied by the TABQTY (or quantity) field and the result placed half adjusted in the field named TOTAL.

If indicator 01 is not turned on, the multiply operation will not be performed because this specific part number has not been located in the table. The operation (GOTO) in line 03 of the calculation form will be performed if indicator 01 is not on. The GOTO operation will cause a transfer to a routine which will indicate that this stock number is not in the table and should be reviewed.

APPENDIX A. SAMPLE PROGRAMS

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

A.1. EXAMPLE NO. 1 – SALES ANALYSIS LISTING

A.1.1. Definition

A Sales Analysis Listing will 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.

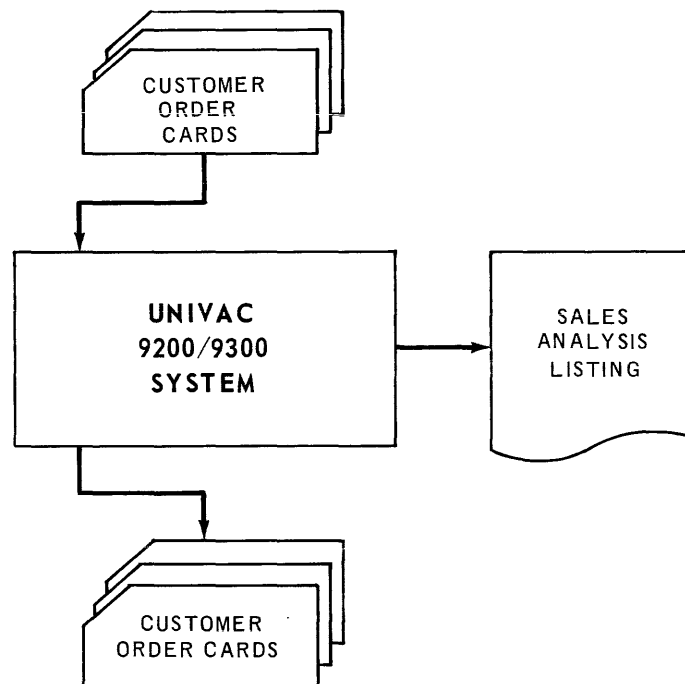


Figure A-1. Sales Analysis Data Flow

TYPE	DATE	SALES TERR.	SLSMN	CUSTOMER NUMBER	PRODUCT NUMBER	PRODUCT DESCRIPTION	QTY	UNIT PRICE	SALE AMT	
0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7	7	7	7
8	8	8	8	8	8	8	8	8	8	8
9	9	9	9	9	9	9	9	9	9	9

BILLING LINE ITEM DETAIL CARD, DIGIT 5 IN COLUMN 1

Printed in U.S.A.

INPUT

Figure A-2. Billing Line Item Detail Card Format

A.1.2. 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 will 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 will 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.

A.1.3. File Description Specifications

Line 1 specifies the input file read into the system; line 2 indicates the printed report.

A.1.4. Input Specifications

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 (Figure A-4).
2. Lines 2-7 describe the fields within the input file.
3. The control levels (columns 59-60) specified in lines 02-04 indicate that TERR is the major control; SLSMN the intermediate; PROD the minor control.

UNIVAC
DIVISION OF SPERRY RAND CORPORATION

REPORT PROGRAM GENERATOR
FILE DESCRIPTION

PG. NO.	LINE NO.	FILE NAME	FILE DESIGNATION FILE TYPE	END OF FILE SEQUENCE	DEVICE
1	0,1	INPUT	IP		READER
	0,2	PRINT	O		PRINT 63

Figure A-4. Sales Analysis File Description

UNIVAC
DIVISION OF SPERRY RAND CORPORATION

REPORT PROGRAM GENERATOR
INPUT FORMAT SPECIFICATIONS

UNIVAC
9000
SERIES

PG. NO.	LINE NO.	FILE NAME	SEQ.	RECORD INDICATOR	RECORD IDENTIFICATION CODES			FIELD LOCATION		MATCHING FIELDS CONTROL LEVEL	FIELD RECORD RELATION			STERLING SIGN POSITION	PROGRAM IDENTIFICATION
					POSITION	POSITION	POSITION	START	END		FIELD NAME	+	-		
1	0,1	INPUT	AA	0,1	1	C	5								
	0,2							8	1,0	TERR	L3				
	0,3							1,1	1,4	SLSMN	L2				
	0,4							2,1	2,7	PROD	L1				
	0,5							2,8	4,3	DESC					
	0,6							4,4	4,7	QTY					
	0,7							5,3	6,1	AMT					
	0,8														

Figure A-5. Sales Analysis Input Specifications

UNIVAC
DIVISION OF SPERRY RAND CORPORATION

REPORT PROGRAM GENERATOR
CALCULATION SPECIFICATIONS

PG. NO.	LINE NO.	CONTROL LEVEL	INDICATORS		FACTOR 1	OPERATION	FACTOR 2	RESULT FIELD	FIELD LENGTH	DECIMAL POSITION	HALF ADJU
			AND	AND							
	0,1	C	0,1		QTY	ADD	PQTY	PQTY	5	0	
	0,2	C	*		AMT	ADD	PAMT	PAMT	9	2	
	0,3	CL1			PQTY	ADD	SQTY	SQTY	6	0	
	0,4	CL1			PAMT	ADD	SAMT	SAMT	9	2	
	0,5	CL2			SQTY	ADD	TQTY	TQTY	7	0	
	0,6	CL2			SAMT	ADD	TAMT	TAMT	10	2	
	0,7	CL3			TQTY	ADD	FQTY	FQTY	8	0	
	0,8	CL3			TAMT	ADD	FAMT	FAMT	10	2	

Figure A-6. Sales Analysis Calculation Specifications

A.1.6. Output Specifications

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 from 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).

A.2. EXAMPLE NO. 2 – DAILY GROSS PAY CALCULATION

A daily job payroll report and employee payroll summary card will 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 will be calculated. These extensions will be printed with the detail information from the job card. When a control break in employee number occurs, total hours and earnings will be printed, and an extended payroll summary card will be 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 format. The completed forms are shown in Figures A–10 through A–14.

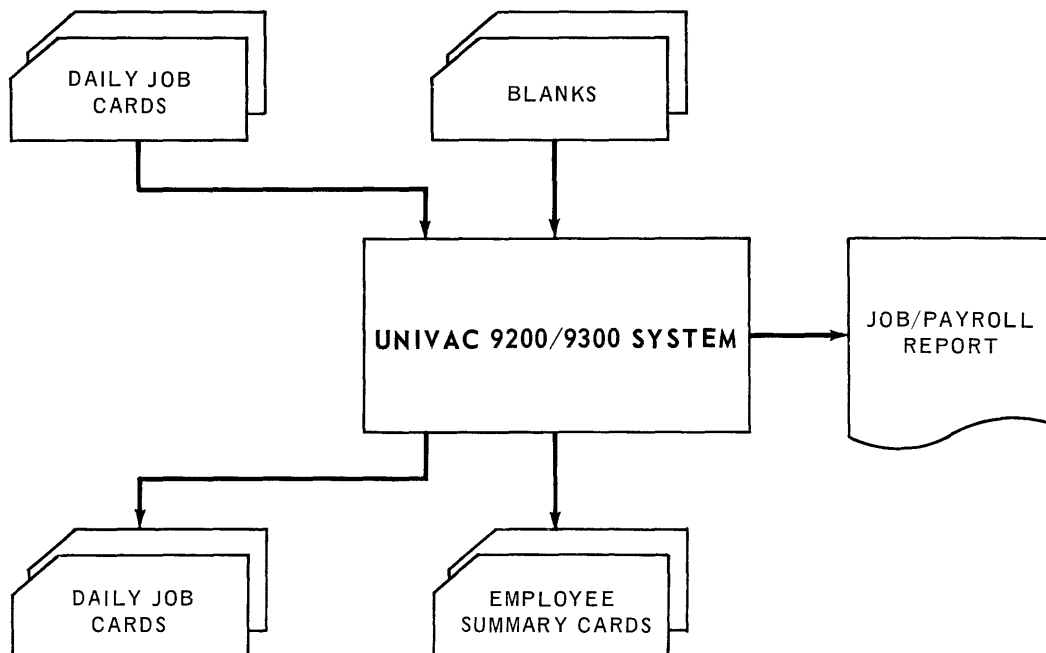


Figure A-8. Daily Gross Pay Data Flow

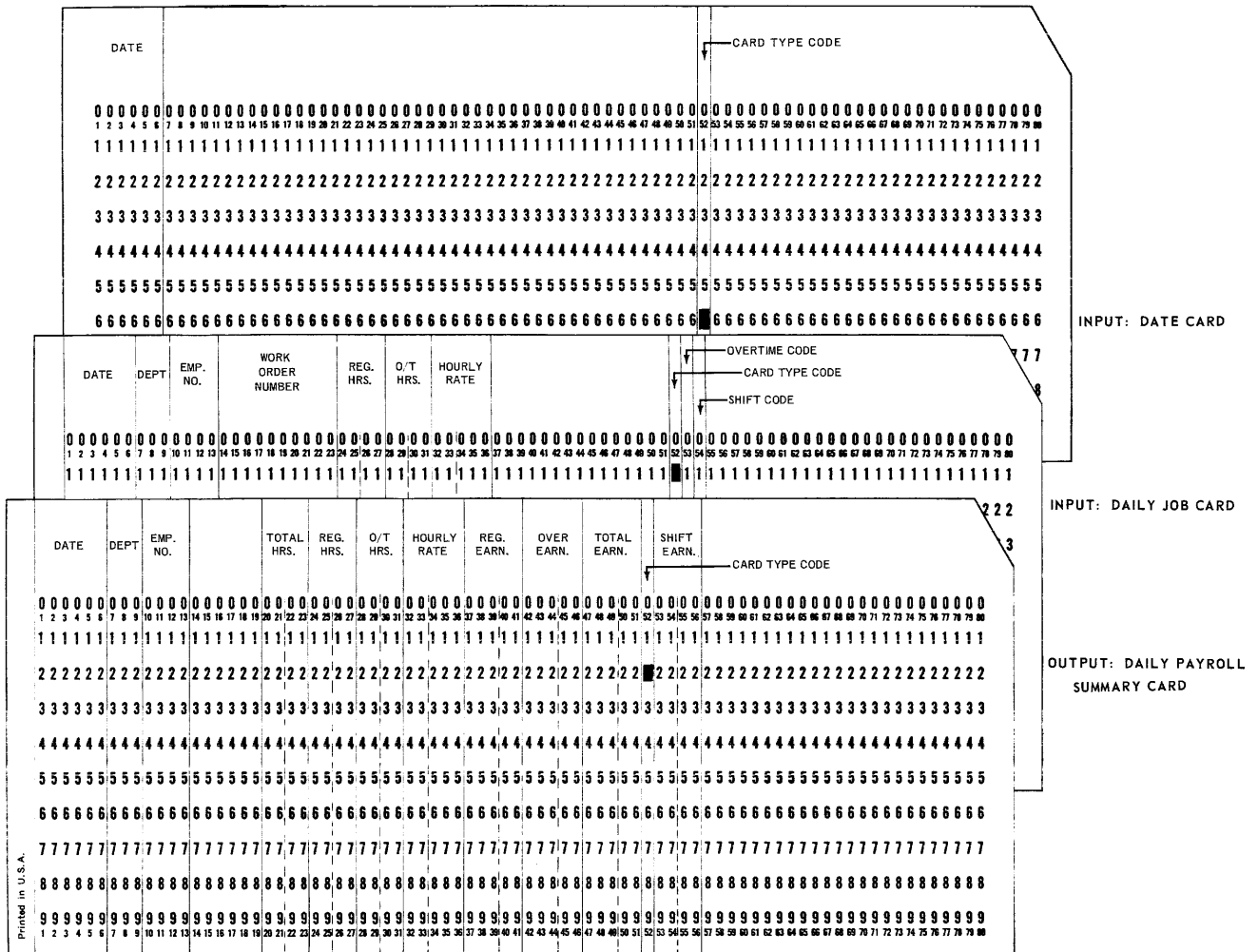


Figure A-9. Input and Output Card Formats

A.2.1. 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

$$\text{Overtime earnings} = 1.5 \times \text{overtime hours} \times \text{rate}$$

If column 53 is 2, then

$$\text{Overtime earnings} = 2.0 \times \text{overtime hours} \times \text{rate}$$

4. Calculate gross pay.

$$\text{Gross pay} = \text{overtime earnings} + \text{regular earnings} + \text{shift bonus.}$$

5. Calculate shift bonus. Check shift code in column 54 for a 1.
If a 1 is present, then
$$\text{Shift bonus} = \text{gross pay} \times 5\%$$
If shift code is blank, this calculation is omitted.
6. Calculate total hours
$$\text{Total hours} = \text{regular hours} + \text{overtime hours}$$
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 will be printed at the top of every page on the same line as the page number. The second line of printing on every page will be column headings.

A.2.2. File Description Specifications

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.

A.2.3. Input Format Specifications

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 format MMDDYY. The programmer will notice that the date card on the input form appears after the job card. The reason for this sequence is that most frequently used specifications should be stated first to save object program running time.

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 will initially have a value of zero. Its contents will be 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.

UNIVAC

REPORT PROGRAM GENERATOR
CALCULATION SPECIFICATIONS

UNIVAC
9200
SYSTEM

PG. NO.	LINE NO.	FORM TYPE	CONTROL LEVEL	INDICATORS			FACTOR 1	OPERATION	FACTOR 2	RESULT FIELD	FIELD LENGTH	RESULTING INDICATORS			COMMENTS	PROGRAM IDENTIFICATION											
				A	N	D						H	L	E			Q										
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		0,1				R,H,D	M,U,L,T	R,A,T,E	R,E,D	5,2	H															
0,2	C		0,1	N,1,0			C,O,D,E	C,O,M,P	1													H,1	1,1				
0,3	C		0,1	N,1,0	N,1,1		C,O,D,E	C,O,M,P	2													H,1	H,1	1,2			
0,4	C		0,1	N,1,0			O,H,D	M,U,L,T	R,A,T,E	O,E,D	5,2	H															
0,5	C		0,1		1,1		O,E,D	M,U,L,T	1, . 5	O,E,D		H															
0,6	C		0,1		1,2		O,E,D	A,D,D	O,E,D	O,E,D																	
0,7	C		0,1				R,E,D	A,D,D	O,E,D	T,E,D	5,2																
0,8	C		0,1	N,1,5			T,E,D	M,U,L,T	. 0,5	S,H,E,D	4,2	H															
0,9	C		*				S,H,E,D	A,D,D	T,E,D	T,E,D																	
1,0	C		0,1				R,H,D	A,D,D	O,H,D	T,H,D	5,2																
1,1	C		*				R,H,D	A,D,D	R,H,T	R,H,T	5,2																
1,2	C		0,1	N,1,0			O,H,D	A,D,D	O,H,T	O,H,T	5,2																
1,3	C		0,1				T,H,D	A,D,D	T,H,T	T,H,T	5,2																
1,4	C		*				R,E,D	A,D,D	R,E,T	R,E,T	5,2																
1,5	C		0,1	N,1,0			O,E,D	A,D,D	O,E,T	O,E,T	5,2																
1,6	C		0,1	N,1,5			S,H,E,D	A,D,D	S,H,E,T	S,H,E,T	4,2																
1,7	C		0,1				T,E,D	A,D,D	T,E,T	T,E,T	5,2																
1,8	C	L,1					R,H,T	A,D,D	R,H,F,T	R,H,F,T	7,2																
1,9	C		*				O,H,T	A,D,D	O,H,F,T	O,H,F,T	6,2																
2,0	C		*				T,H,T	A,D,D	T,H,F,T	T,H,F,T	7,2																
2,1	C		*				R,E,T	A,D,D	R,E,F,T	R,E,F,T	7,2																
2,2	C		*				O,E,T	A,D,D	O,E,F,T	O,E,F,T	7,2																
2,3	C		*				S,H,E,T	A,D,D	S,H,E,F,T	S,H,E,F,T	5,2																
2,4	C		*				T,E,T	A,D,D	T,E,F,T	T,E,F,T	7,2																
2,5	C																										

Figure A-13 Daily Gross Pay Calculation Specifications

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 of page 1 and lines 1–10 of page 2 specify punching at total time when the L1 level indicator is on.

Line 9 of page 2 causes a 2, generated by the RPG, to be punched in column 52 of the summary card.

Lines 11–18 of page 2 specify the total line which will print 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 will be edited according to the edit words in columns 45–70. All fields will be zero suppressed with decimals inserted. The paper will advance two spaces after printing. Lines 19–26 of page 2 specify the final totals which will print when the LR (last record) indicator is on. All fields will be edited according to the specified edit words. The processor is brought to a halt after the final totals are printed.

UNIVAC
DIVISION OF SPERRY RAND CORPORATION

REPORT PROGRAM GENERATOR
OUTPUT FORMAT SPECIFICATIONS

UNIVAC
9000
SERIES

PG. NO.	LINE NO.	FORM TYPE	FILE NAME	TYPE H/D/T	STACKER		OUTPUT INDICATORS					FIELD NAME	END POSITION IN OUTPUT RECORD	CONSTANT OR EDIT WORD	STERLING SIGN POSITION	PROGRAM IDENTIFICATION	
					SPACE	SKIP	A	N	D	A	N						D
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
0	2	0										EMP	13				
	0	2										TH T	23				
	0	3										RH T	27				
	0	4										OH T	31				
	0	5										RA TE	36				
	0	6										RE T	41				
	0	7										OET	46				
	0	8										TE T	51				
	0	9											52	' 2 '			
	1	0										SH ET	56				
	1	1	OPRINT,OUTT			12				*							
	1	2										RH T	28	' 0 . '			
	1	3										OH T	37	' 0 . '			
	1	4										TH T	47	' 0 . '			
	1	5										RE T	67	' 0 . '			
	1	6										OET	78	' 0 . '			
	1	7										SH ET	86	' 0 . '			
	1	8										TE T	96	' 0 . '			
	1	9	OPRINT,OUTT			00				LR							
	2	0										RH FT	28	' 0 . '			
	2	1										OH FT	37	' 0 . '			
	2	2										TH FT	47	' 0 . '			
	2	3										RE FT	67	' 0 . '			
	2	4										OFT	78	' 0 . '			
	2	5										SH FT	86	' 0 . '			
	2	6										TE FT	96	' 0 . '			

Figure A-14. Daily Gross Pay Output Specifications (Sheet 2 of 2)

A.3. EXAMPLE NO. 3 – INVENTORY APPLICATION

Stock Status Report

A daily stock status listing and summary card updating are to be prepared using the following cards in sequence by part number:

1. Stock status balance card containing bin amount, on-order amount, available amount, usage last year, usage this year, and so on.
2. On-order cards which increase the on-order amount and the available amount.
3. Receipt cards which increase the bin amount and usage this year amount and decrease the on-order amount.
4. Issue cards which decrease bin amount and available amount.

A detail line is printed for each input card. An updated stock status summary card is calculated and, on a change in part number, printed and punched. If the total available amount 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-16 and A-17 show the input and output card formats. Figures A-18 through A-22 illustrate the completed forms.

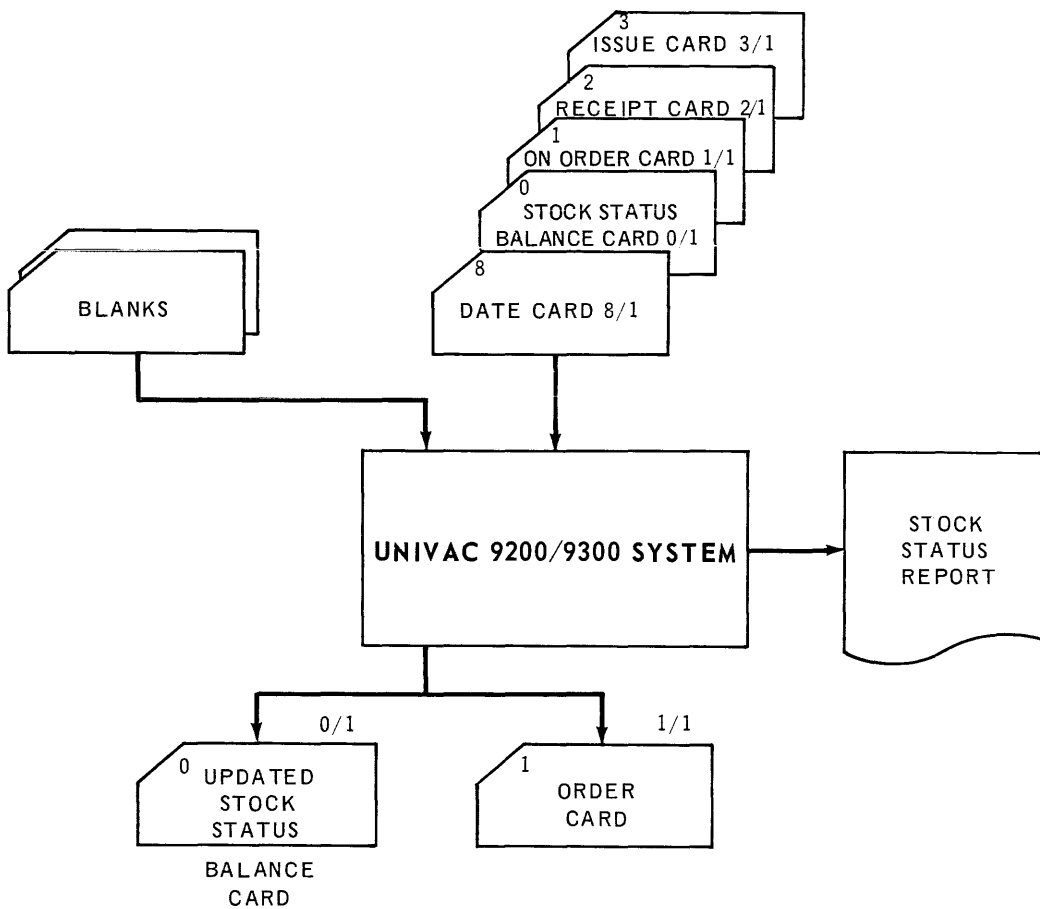


Figure A-15. Stock Status Report Data Flow

TYPE	DATE	RUN	PART NO.	ASSEMBLY NO.	DESCRIPTION	QTY. ISSUED AMT.	QTY. RECEIVED AMT.	ON ORDER AMT.
0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7	7
8	8	8	8	8	8	8	8	8
9	9	9	9	9	9	9	9	9

NOTE: OPTIONAL; ONE RECORD OR MORE FOR EACH PART NUMBER. THE AMOUNT FIELD IN THE ISSUE, RECEIPT AND ORDER CARD IS ALWAYS PUNCHED AS A POSITIVE FIGURE.

Figure A-16. Input Card Formats (Sheet 1 of 2)

A.3.1. Procedure

1. Sequence check by part number.
2. Check that each part number group has only one stock status balance input card.
3. The following fields from the stock status balance card are accumulated for summary card punching and printing:

Card Columns

Bin Amount	36-42
On-Order Amount	43-49
Available Amount	50-56
Usage This Year	74-80

Part number (columns 9-16) and description (columns 24-35) are stored for summary card punching. Reorder point (columns 57-61), reorder quantity (columns 62-66), and usage last year (columns 67-73) are to be stored for printing and summary card punching.

4. The on-order amount (columns 43-49) from the on-order card is added to the summary card totals of on-order and available amounts.
5. Quantity received (columns 43-49) from the receipt card is added to the summary card total of bin amount and subtracted from the summary card total of on-order amount.
6. Quantity issued (columns 43-49) from the issue card is treated as follows:
 - a. Subtracted from the summary card total of bin amount.
 - b. Subtracted from the summary card total of available amount.
 - c. Added to the summary card total of usage this year.
7. When a change in part number occurs, the summary card total of available amount is compared to the reorder point field. If the available amount is less than or equal to the reorder point, the following calculation is performed and an order card produced: Quantity to be Ordered = Reorder Point + Reorder Quantity - Available Amount.

If an order card is produced, quantity to be ordered must also be printed in print positions 74-80. If the accumulated available amount is negative, an asterisk (*) is placed in print position 60 of the stock status report.

TYPE	DATE	PART NUMBER	DESCRIPTION	BIN AMOUNT	ON ORDER AMOUNT	AVAILABLE AMOUNT	REORDER POINT	REORDER QTY.	USAGE LAST YEAR	USAGE THIS YEAR
0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7	7	7	7
8	8	8	8	8	8	8	8	8	8	8
9	9	9	9	9	9	9	9	9	9	9

DATE CARD
NOTE: ONE RECORD ONLY AT THE BEGINNING OF A RUN.

STOCK STATUS BALANCE CARD
NOTE: ONE RECORD FOR EACH PART NUMBER. A NEGATIVE FIELD IS DENOTED BY AN X ZONE PUNCH IN ITS LEAST SIGNIFICANT POSITION.

Figure A-16. Input Card Formats (Sheet 2 of 2)



REPORT PROGRAM GENERATOR
FILE DESCRIPTION

PG. NO.	LINE NO.	FILE NAME	FILE DESIGNATION FILE TYPE	END OF FILE SEQUENCE	DEVICE
1	3	5 6 7	14 15 16 17 18 19		39 40 46 47
	0 1	F I N P U T	I P A		R E A D E R
	0 2	F P R I N T	O		P R I N T 6 3
	0 3	F P U N C H	O		R O W P N C H

Figure A-19. Inventory File Description

A.3.2. Description of the Stock Status Report Text

The layout of this report is shown in Figure A-18.

1. The report header line consists of the current date from the date card.
2. The stock status balance input card is to be printed as shown on line 13 of Figure A-18.
3. Designating information from the on-order card, the receipt card and issue card are printed as shown on lines 14, 15, and 16, respectively. The on-order card amount field is printed as a positive amount in print positions 42-48 and 51-57 of line 14. The receipt card amount field is printed as a positive amount in print positions 33-39 and as a negative amount in print positions 42-48 (line 15). Issue card amount field is printed as a negative amount in print positions 33-39 and 51-57, and as a positive amount in print positions 90-96 (line 16).
4. The updated stock status balance line is printed as shown on line 18 of Figure A-18.

A.3.3. File Specifications

The File Description form specifies the file name of the input file and the two output files. The device associated with each file is specified. The entry A under sequence indicates that the input file is in ascending order. This specification is related to the M1 entries on the Input Format Specifications form.

- NOTE: 1. Resulting indicator 17 represents a negative available amount and is used to control output.
2. Resulting indicator 16 is turned on when the available amount is less than or equal to the reorder point. When this indicator is on, lines 9 and 10 are executed.

UNIVAC
DIVISION OF SPERRY RAND CORPORATION

REPORT PROGRAM GENERATOR
CALCULATION SPECIFICATIONS

UNIVAC
9000
SERIES

PG. NO.	LINE NO.	FORM TYPE	CONTROL LEVEL	INDICATORS		FACTOR 1	OPERATION	FACTOR 2	RESULT FIELD	FIELD LENGTH	DECIMAL POSITION	HALF ADJUST	RESULTING INDICATORS			COMMENTS	PROGRAM IDENTIFICATION											
				A N D	A N D								+	-	0													
1	3	5	6	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		0	1			A.M.T.	A.D.D.	S.O.R.D.E.R.	S.O.R.D.E.R.																	
0	2	C		*				A.M.T.	A.D.D.	S.A.V.A.I.L.	S.A.V.A.I.L.												1	7				
0	3	C		0	2			A.M.T.	A.D.D.	S.B.I.N.	S.B.I.N.																	
0	4	C		*				S.O.R.D.E.R.	S.U.B.	A.M.T.	S.O.R.D.E.R.																	
0	5	C		0	3			S.B.I.N.	S.U.B.	A.M.T.	S.B.I.N.																	
0	6	C		*				S.A.V.A.I.L.	S.U.B.	A.M.T.	S.A.V.A.I.L.												1	7				
0	7	C		*				A.M.T.	A.D.D.	S.T.H.I.S.	S.T.H.I.S.																	
0	8	C	L	1				S.A.V.A.I.L.	C.O.M.P.	S.P.O.I.N.T.													1	6	1	6		
0	9	C	L	1	1	6		S.P.O.I.N.T.	S.U.B.	S.A.V.A.I.L.	O.R.Q.T.Y.											7	0					
1	0	C		*				S.Q.T.Y.	A.D.D.	O.R.Q.T.Y.	O.R.Q.T.Y.																	

Figure A-21. Inventory Calculation Specifications

A.3.6. Output Specifications

- Lines 1-3 of page 1 provide for the printing of the date at the top of each page.
- Lines 4-12 of page 1 provide for the printing of the input stock status balance card.
- Lines 13-25 of page 1 and line 1 of page 2 provide for the printing of the on-order, receipt, and issue cards.
- Lines 2-8 of page 2 provide for the punching of an order card only if indicator 16 is on.

NOTE: An asterisk (*) will be placed in column 50 of the card if the available amount is negative.

- Lines 9-20 of page 2 provide for the punching of the updated stock status balance card.
- Lines 21-25 of page 2 and 1-5 of page 3 provide for the printing of the updated stock status totals. An asterisk (*) is printed in print position 60 if the available amount is negative. Quantity to be ordered is printed in print positions 74-80 if an order card has been punched.

UNIVAC
DIVISION OF SPERRY RAND CORPORATION

**REPORT PROGRAM GENERATOR
OUTPUT FORMAT SPECIFICATIONS**

PG. NO.	LINE NO.	FORM TYPE	FILE NAME	TYPE H/D/T	STACKER SELECT			ZERO SUPPRESS		BLANK AFTER	DATA FORMAT	CONSTANT
					SPACE	SKIP	OUTPUT INDICATORS					
1	3	5 6 7	14	15 16	B A F T E R	B E F O R E	A F T E R	A N D	A N D	37 38 39	40	43 44 45
0,1	0,1	O	P R I N T	H		0,7	0,2	0,0				
	0,2	O		O R				0,8				
	0,3	O									9,4	' - - '
	0,4	O	P R I N T	D	0,1			1,0				
	0,5	O								P A R T	Z	1,6
	0,6	O								D E S C		2,9
	0,7	O								T Y P E		3,1
	0,8	O								S B I N		4,0
	0,9	O								S O R D E R		4,9
	1,0	O								S A V A I L		5,8
	1,1	O								S L A S T	Z	8,8
	1,2	O								S T H I S	Z	9,6
	1,3	O	P R I N T	D	0,1			0,1				
	1,4	O		O R				0,2				
	1,5	O		O R				0,3				
	1,6	O								N U M	Z	7
	1,7	O								P A R T	Z	1,6
	1,8	O								D E S C		2,9
	1,9	O								T Y P E		3,1
	2,0	O						N,0,1		A M T	Z	3,9
	2,1	O						N,0,3		A M T	Z	4,8
	2,2	O						N,0,2		A M T	Z	5,7
	2,3	O						0,3		A M T	Z	9,6
	2,4	O						0,3				4,0
	2,5	O						0,2				4,9

Figure A-22. Inventory Output Specifications (Sheet 1 of 3)

UNIVAC
DIVISION OF SPERRY RAND CORPORATION

**REPORT PROGRAM GENERATOR
OUTPUT FORMAT SPECIFICATIONS**

PG. NO.	LINE NO.	FORM TYPE	FILE NAME	TYPE H/D/T	STACKER SELECT			OUTPUT INDICATORS			FIELD NAME	ZERO SUPPRESS	BLANK AFTER	END POSITION IN OUTPUT RECORD	DATA FORMAT	CONSTANT										
					SPACE	SKIP		A	N	D																
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
0,2	0,1	O												0,3										5,8	' - '	
	0,2	O	PUNCH	T	1									L,1	1,6											
	0,3	O																					1	' 1 '		
	0,4	O																					7			
	0,5	O																					1,6			
	0,6	O																					3,5			
	0,7	O																					4,9			
	0,8	O												1,7									5,0	' * '		
	0,9	O	PUNCH	T	2									L,1												
	1,0	O																					1	' 0 '		
	1,1	O																					7			
	1,2	O																					1,6			
	1,3	O																					3,5			
	1,4	O																					4,2			
	1,5	O																					4,9			
	1,6	O																					5,6			
	1,7	O																					6,1			
	1,8	O																					6,6			
	1,9	O																					7,3			
	2,0	O																					8,0			
	2,1	O	PRINT	T	1 2									*												
	2,2	O																					4,0	' 0 - '		
	2,3	O																					4,9	' 0 - '		
	2,4	O																					5,8	' 0 - '		
	2,5	O												1,7									6,0	' * '		

Figure A-22. Inventory Output Specifications (Sheet 2 of 3)

UNIVAC
DIVISION OF BULLOCK AND LEBRON

**REPORT PROGRAM GENERATOR
OUTPUT FORMAT SPECIFICATIONS**

PG. NO.	LINE NO.	FILE NAME	TYPE H/D/T	STACKER SELECT			ZERO SUPPRESS			BLANK AFTER			DATA FORMAT												
				SPACE	SKIP	OUTPUT INDICATORS	FIELD NAME	END POSITION IN OUTPUT RECORD	CONSTAI																
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
0	3	0, 1, 0																	S P O I N T Z				6	6	
		0, 2, 0																	S Q T Y Z				7	2	
		0, 3, 0										1	6						O R Q T Y Z				8	0	
		0, 4, 0																	S L A S T Z				8	8	
		0, 5, 0																	S T H I S Z				9	6	

Figure A-22. Inventory Output Specifications (Sheet 3 of 3)

APPENDIX B. SUMMARY OF PROGRAM INDICATORS

INDICATOR	LOCATION	WHERE USED	TURNED ON	TURNED OFF	REMARKS
Field Indicators (01-99) Zero and Blank 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 SEE NOTE
H1 and H2	Internal	Field Indicators (Input) Indicators (Calculation) Resulting Indicators (Calculation) Output Indicators	If condition tested for occurs	when the system is started after the system is stopped by the error	
LR (Last Record)	Internal	Control Level (Calculation) Output Indicators	at end of processing of last record read	at the beginning of processing	
L0 (Level Zero)	Internal	Control Level			This indicator is always on
Control Level Indicators (L1-L9)	Internal	Control Level Indicators (Calculation) Output Indicators	by appropriate control level break	at end of following detail cycle	Remains on during detail calculation and output cycle
MR (Matching Record)	Internal	Indicators (Calculation) Output Indicators	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
OF (Overflow)	Internal	Indicators (Calculation) Output Indicators	after detail or total time if channel 1 punch detected during output	after detail time	Remains on for detail time if turned on after total--Remains on for 1 complete cycle (total and next detail) if turned on after detail
Record Indicators (01-99)	Input Specifications Form	Indicators (Calculation) Output Indicators	When specified record has been read and is ready for processing	when another Record Indicator is turned on	Only one Record Indicator can be on at one time SEE NOTE
Resulting Indicators (01-99, H1, H2) Plus Minus Zero Compare Operation High Low Equal	Calculation Specifications Form	Indicators (Calculation) 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 SEE NOTE
1P (First Page)	Internal	Output Indicators	at beginning of processing before any input records are read	before the first detail card is read	

NOTE: 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).

APPENDIX C. CODE REPRESENTATION AND TRANSLATION

At generator time, each source card is read in compressed mode and is then translated by way of a read table. The standard translation is shown in the table at the end of this appendix. If a nonstandard card code is being used, a different translation table can be employed. The only requirement is that the graphic-to-internal-code relation (shown in the table at the end of this appendix) be maintained.

For record identification purposes, the generator takes the character specified for CHAR and translates it by means of a table that is the inverse of the read table. This, in effect, translates the character from its internal code back to its compressed code.

If C is specified in C/Z/D field, the character in its compressed code form is adopted as the code to be tested against. If D is specified, bit positions six and seven of the compressed code are erased, and the result is adopted as the code to be tested against. If Z is specified, bits zero through five are erased, and the result (or its complement) is adopted as the immediate part of a TM instruction.

At object code time each input record is read in compressed mode. Record identification is then done. If C was specified at generator time, the character in the position indicated by POSITION is compared logically against the code set up by the generator. If D was specified, the character in the position indicated is moved to a temporary storage, bit positions six and seven of the character are erased, and the result is compared logically against the code. If Z, the TM instruction generated at generator time is executed with respect to the character in the position indicated.

After a record has been identified, it is then translated to internal code. The table C-1 is a complete listing of the internal code.

TWO MOST SIGNIFICANT BITS OF ZONE - 00

DIGIT	TWO LEAST SIGNIFICANT BITS OF ZONE			
	00	01	10	11
0000	12-0-9-8-1	12-11-9-8-1	11-0-9-8-1	12-11-0-9-8-1
0001	12-9-1	11-9-1	0-9-1	9-1
0010	12-9-2	11-9-2	0-9-2	9-2
0011	12-9-3	11-9-3	0-9-3	9-3
0100	12-9-4	11-9-4	0-9-4	9-4
0101	12-9-5	11-9-5	0-9-5	9-5
0110	12-9-6	11-9-6	0-9-6	9-6
0111	12-9-7	11-9-7	0-9-7	9-7
1000	12-9-8	11-9-8	0-9-8	9-8
1001	12-9-8-1	11-9-8-1	0-9-8-1	9-8-1
1010	12-9-8-2	11-9-8-2	0-9-8-2	9-8-2
1011	12-9-8-3	11-9-8-3	0-9-8-3	9-8-3
1100	12-9-8-4	11-9-8-4	0-9-8-4	9-8-4
1101	12-9-8-5	11-9-8-5	0-9-8-5	9-8-5
1110	12-9-8-6	11-9-8-6	0-9-8-6	9-8-6
1111	12-9-8-7	11-9-8-7	0-9-8-7	9-8-7

Table C-1. 9000 Series Internal Code. (Sheet 1 of 4)

CARD CODE
GRAPHIC, IF ANY

TWO MOST SIGNIFICANT BITS OF ZONE - 01

DIGIT	TWO LEAST SIGNIFICANT BITS OF ZONE			
	00	01	10	11
0000	⊥	12 &	11 -	12-11-0
0001	12-0-9-1	12-11-9-1	0-1	12-11-0-9-1
0010	12-0-9-2	12-11-9-2	11-0-9-2	12-11-0-9-2
0011	12-0-9-3	12-11-9-3	11-0-9-3	12-11-0-9-3
0100	12-0-9-4	12-11-9-4	11-0-9-4	12-11-0-9-4
0101	12-0-9-5	12-11-9-5	11-0-9-5	12-11-0-9-5
0110	12-0-9-6	12-11-9-6	11-0-9-6	12-11-0-9-6
0111	12-0-9-7	12-11-9-7	11-0-9-7	12-11-0-9-7
1000	12-0-9-8	12-11-9-8	11-0-9-8	12-11-0-9-8
1001	12-8-1	11-8-1	0-8-1	8-1
1010	12-8-2 ϕ	11-8-2 !	12-11	8-2
1011	12-8-3 .	11-8-3 \$	0-8-3	8-3 #
1100	12-8-4 <	11-8-4 *	0-8-4 %	8-4 @
1101	12-8-5 (11-8-5)	0-8-5 —	8-5 /
1110	12-8-6 +	11-8-6 ;	0-8-6 >	8-6 =
1111	12-8-7	11-8-7 ┘	0-8-7 ?	8-7 "

Table C-1. 9000 Series Internal Code. (Sheet 2 of 4)

CARD CODE GRAPHIC, IF ANY

TWO MOST SIGNIFICANT BITS OF ZONE - 10

DIGIT	TWO LEAST SIGNIFICANT BITS OF ZONE			
	00	01	10	11
0000	12-0-8-1	12-11-8-1	11-0-8-1	12-11-0-8-1
0001	12-0-1	12-11-1	11-0-1	12-11-0-1
0010	12-0-2	12-11-2	11-0-2	12-11-0-2
0011	12-0-3	12-11-3	11-0-3	12-11-0-3
0100	12-0-4	12-11-4	11-0-4	12-11-0-4
0101	12-0-5	12-11-5	11-0-5	12-11-0-5
0110	12-0-6	12-11-6	11-0-6	12-11-0-6
0111	12-0-7	12-11-7	11-0-7	12-11-0-7
1000	12-0-8	12-11-8	11-0-8	12-11-0-8
1001	12-0-9	12-11-9	11-0-9	12-11-0-9
1010	12-0-8-2	12-11-8-2	11-0-8-2	12-11-0-8-2
1011	12-0-8-3	12-11-8-3	11-0-8-3	12-11-0-8-3
1100	12-0-8-4	12-11-8-4	11-0-8-4	12-11-0-8-4
1101	12-0-8-5	12-11-8-5	11-0-8-5	12-11-0-8-5
1110	12-0-8-6	12-11-8-6	11-0-8-6	12-11-0-8-6
1111	12-0-8-7	12-1-8-7	11-0-8-7	12-11-0-8-7

Table C-1. 9000 Series Internal Code. (Sheet 3 of 4)

CARD CODE
GRAPHIC, IF ANY

TWO MOST SIGNIFICANT BITS OF ZONE - 11

DIGIT	TWO LEAST SIGNIFICANT BITS OF ZONE			
	00	01	10	11
0000	12-0	11-0	0-8-2	0 0
0001	12-1 A	11-1 J	11-0-9-1	1 1
0010	12-2 B	11-2 K	0-2 S	2 2
0011	12-3 C	11-3 L	0-3 T	3 3
0100	12-4 D	11-4 M	0-4 U	4 4
0101	12-5 E	11-5 N	0-5 V	5 5
0110	12-6 F	11-6 O	0-6 W	6 6
0111	12-7 G	11-7 P	0-7 X	7 7
1000	12-8 H	11-8 Q	0-8 Y	8 8
1001	12-9 I	11-9 R	0-9 Z	9 9
1010	12-0-9-8-2	12-11-9-8-2	11-0-9-8-2	12-11-0-9-8-2
1011	12-0-9-8-3	12-11-9-8-3	11-0-9-8-3	12-11-0-9-8-3
1100	12-0-9-8-4	12-11-9-8-4	11-0-9-8-4	12-11-0-9-8-4
1101	12-0-9-8-5	12-11-9-8-5	11-0-9-8-5	12-11-0-9-8-5
1110	12-0-9-8-6	12-11-9-8-6	11-0-9-8-6	12-11-0-9-8-6
1111	12-0-9-8-7	12-11-9-8-7	11-0-9-8-7	12-11-0-9-8-7

Table C-1. 9000 Series Internal Code. (Sheet 4 of 4)

CARD CODE
GRAPHIC, IF ANY

APPENDIX D. STERLING NOTATION

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. Monetary fields using sterling notation must conform to the specific rules indicated in this section. The RPG uses three basic formats for the representation of sterling amounts.

D.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, shilling, 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. Figure D-1 illustrates Format 1 maximum sizes.

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	d d d d d d d d d d			fff	RPG

£ pound digit
s shilling digit
d penny digit
f penny fractional digit

Figure D-1. Maximum Sizes of Sterling Notation Formats

Refer to Figure D-2 for an example of sterling amounts written in Format 1.

D.2. FORMAT 2

Format 2 is used for both input and output. Input cards are read and output cards 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.

D.2.1. Shillings

BSI Code

A single column is used to represent the shillings field. Amounts of 0...9 are indicated by the punches 0...9. Ten shillings are represented by a 12 punch in the column. Eleven through nineteen shillings are represented by the A-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 the second for the units (digits 0...9).

STERLING AMOUNT

FORMAT	UNSIGNED				NEGATIVE				POSITIVE																
	POUNDS	SHILLINGS	PENCE	DECIMAL	POUNDS	SHILLINGS	PENCE	DECIMAL	POUNDS	SHILLINGS	PENCE	DECIMAL													
	£	:	19	:	11	.	15		-£	425	:	15	:	10	.	14		+£	188	:	0	:	7		
1	0	19	11	15	425	15	10	IM	188	00	0G		2A	0	19	+	15	425	15	-	15	14	18H	00	7
2B	0	19	-	15	425	15	+	15	14	18H	00	7	2C	0	I	-	15	425	E	+	15	14	18H	0	7
2D	0	I	+	15	425	E	-	15	14	18H	0	7	PENCE	239		15	102190		IM	4512G					

Figure D-2. Sterling Amounts Expressed in Available Formats of Notation

D.2.2. Pence

BSI Code

The pence field is represented by a single column. Amounts of 0...9 are indicated by the punches 0...9. Ten pence are represented by a 12 punch; 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...9 are indicated by the punches 0...9. However, ten pence are represented by an 11 punch, eleven pence by a 12 punch. Format 2 may be used in any of the following combinations of BSI and Hollerith code.

D.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

Figure D-1 illustrates the maximum sizes of Format 2. Refer to Figure D-2 for an example of sterling amounts written in Format 2.

FIELDS	PRINTING FORMAT	CARD READING/PUNCHING FORMAT	
	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

Figure D-3. Summary of Sterling Notation Formats

D.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.

D.2.5. Leading Zeros

Leading zeros can be represented by blanks in sterling input fields.

D.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.

Figure D-1 illustrates the maximum size of the pence format. Refer to Figure 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.

UNIVAC

DIVISION OF SPERRY RAND CORPORATION

UP-4106