

The IBM logo, consisting of the letters "IBM" in a bold, sans-serif font, is positioned within a dark rectangular box.

**Systems Reference Library**

**Report Program Generator  
IBM 1401 Card and Tape Systems**

This reference publication contains, language specifications for Report Program Generator for IBM 1401 Card and Tape Systems. The descriptions of the specifications sheets and the rules and conventions for writing them constitute the major portion of this publication. Also included, is information about machine processing of the source deck to produce the object program and machine processing of the input data file to produce a report.

For titles and abstracts of associated publications, see the *IBM 1401/1460 Bibliography*, Form A24-1495.

# Contents

<b>Introduction</b> .....	3
<b>First Phase of Report Generating</b> .....	10
General Description .....	10
Correlating the Report Specifications .....	12
IBM 1403 Spacing Chart .....	12
Input Specifications .....	16
Data Specifications .....	22
Calculation Specifications .....	29
Format Specifications .....	34
Control Card .....	44
Special Feature Specifications .....	44
Summary of First Phase of Report Generating .....	44
<b>Second Phase of Report Generating</b> .....	45
Checking the Specifications .....	45
<b>Third Phase of Report Generating</b> .....	46
Program Storage Requirements .....	46
<b>Fourth Phase of Report Generating</b> .....	47
Sense Switches .....	47
<b>Suggestions and Recommendations</b> .....	48
Split Control-Fields in Input Records .....	48
Program Efficiency .....	48
<b>Description of the General Block Diagrams</b> .....	51
Object Program to Process Card Input Files .....	51
Object Program to Process Tape Input Files .....	54
<b>Inserting Subroutines in Object Program</b> .....	58
Subroutine Labels .....	58
<b>Suppressing Total Lines and Calculations</b> .....	59
<b>Sample Report Documentation</b> .....	59
<b>Index</b> .....	74

*Minor Revision, February 1965*

This publication is a reprint of an earlier edition with the format changed to conform to that of the Systems Reference Library. Like the earlier edition, it obsoletes Form J24-0215-1.

Copies of this and other IBM publications can be obtained through IBM Branch Offices.  
Address comments concerning the content of this publication to IBM Product Publications, Endicott, New York 13764.

## Introduction

The IBM 1401 Report Program Generator produces programs that write reports of variable format from card or tape input files. Instead of writing a specific program for each report, the user writes a set of specifications which he supplies to the Generator. The process of producing any desired report can be separated into four phases as shown in Figures 1 and 2.

Object programs can be *produced* on a 1401 card system equipped with a minimum of 4,000 positions of core storage. No special features are required.

The amount of storage required for executing an object program is dependent upon the complexity of the report specifications. An object program can be *executed* on any 1401 system whose storage capacity will accommodate the object program. Some object programs can be *executed* on a 1401 system equipped with less than 4,000 positions of core storage. The printer can be an IBM 1403, Model 1 or Model 2. No special features are required.

During the first phase (descriptive phase) the user describes information relevant to the report according to certain rules and conventions. He then punches the specifications into cards that constitute the source deck for the Generator.

The second phase is generating the object program in the language of the IBM 1401 Symbolic Programming System (see *IBM 1401 Symbolic Programming System*, Form J24-0200). In phase three the 1401 SPS processor or the 1401 Autocoder (see *Autocoder for the IBM 1401*, Form J24-1434) processor converts the symbolic object program into the machine-language instructions required to write the report. Before the object program is converted from symbolic language to machine language, the user can incorporate any of his own subroutines into the program and then assemble the augmented object program.

The fourth (final) phase of report generating is the execution of the object program to produce the report. The input data must be in either card or tape files.

Object programs produced by the Report Program Generator can process magnetic-tape input files having:

1. Fixed-length records with fixed blocking.
2. Variable-length records that are unblocked (single-record blocks).

The maximum allowable block-length for input tape records varies with the size of the object program generated. After an object program is assembled in machine language, the user can readily determine how

much storage remains for the input tape-record block (see *Third Phase of Report Generating: Program Storage Requirements*).

A tape file is the data contained between tape marks. A tape reel can contain any number of tape files. All or only some of the files on a tape may be considered part of the input data to a program. Thus, the ability to process or bypass files as well as the ability to restart the program after an end-of-file condition has been sensed are included in the generated object program. Because the program always halts on an end-of-file condition, input data may be contained on more than one reel of tape.

If the user specifies that there is a header label at the beginning of an input tape, the generated object program bypasses the first record on the input tape, including any tape mark following that record. The user can choose to have the label processed if it meets the specifications for record and block size pertaining to the entire data file.

As mentioned earlier, the output of the object program can be a printed report, punched cards, tape records or some combination of the three. Card input can be processed to produce printed and/or punched output. Tape input can yield tape output as well as printing and punching in any combination. Only unblocked, fixed-length tape records no longer than 132 characters can be produced as output. Although these records must compose a single file, that file can be written on more than one reel of tape. (The generated program halts on an end-of-reel condition to allow a new tape to be mounted.)

This bulletin is concerned mainly with a description of the first phase of report generating, i.e., stating the specifications of the report and compiling a source deck. Accordingly, the descriptions of the sheets on which the specifications will be written and the rules and conventions for writing them constitute the major portion of this bulletin. Also included, however, is information about machine processing of the source deck to produce the object program (phases 2 and 3) and machine processing of the input data file to produce a report (phase 4).

Requests for the Card or for the Tape Report Program Generator program deck and write-up (which includes program listings and operating instructions) should be made to the local IBM sales representative or sales office.

Figure 3 shows a few input cards for a report produced using the IBM 1401 Report Program Generator. Figure 4 shows the report. Description of the specifications for this report appears throughout this bulletin.

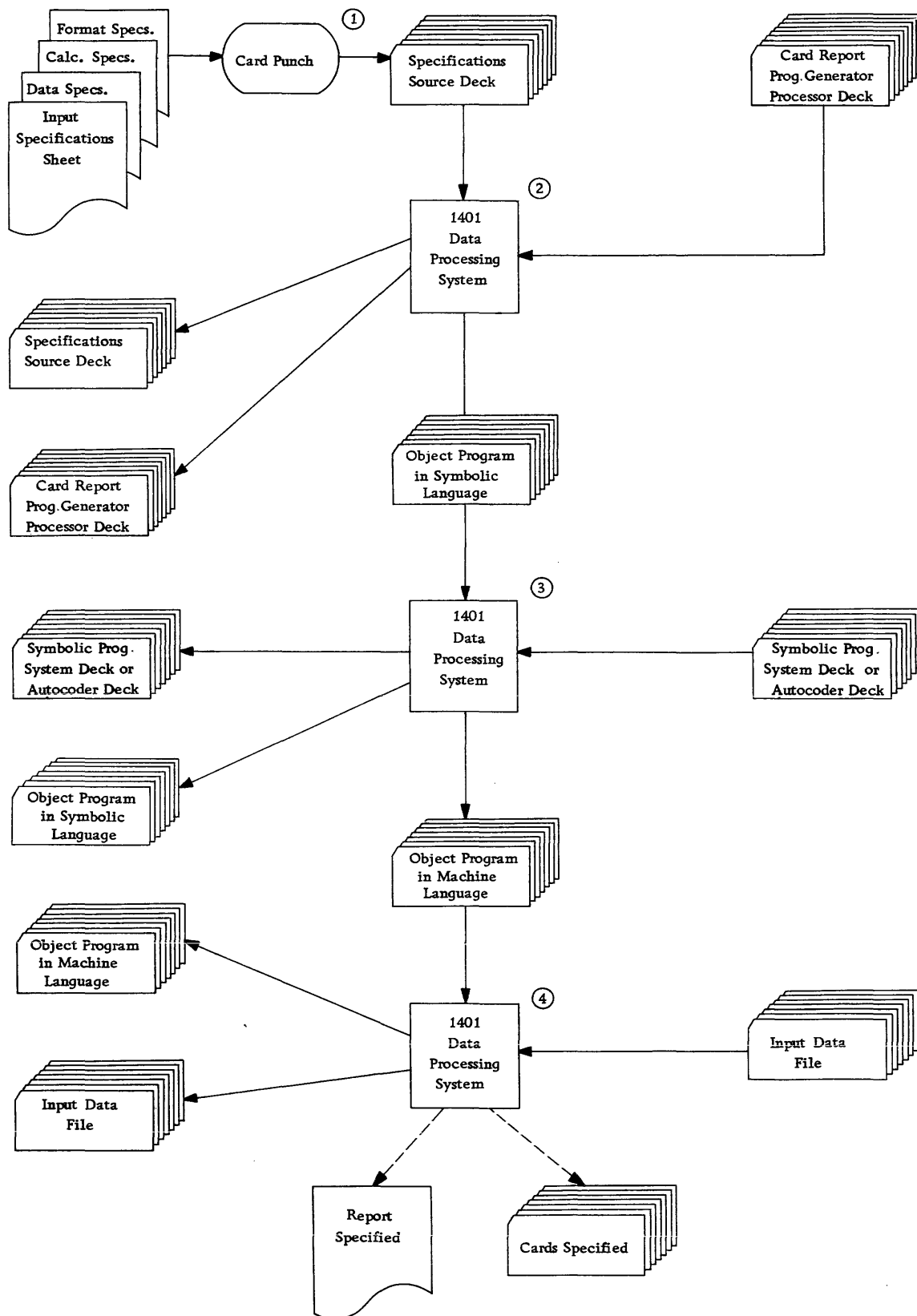


Figure 1. Generating a Report on a 1401 Card System

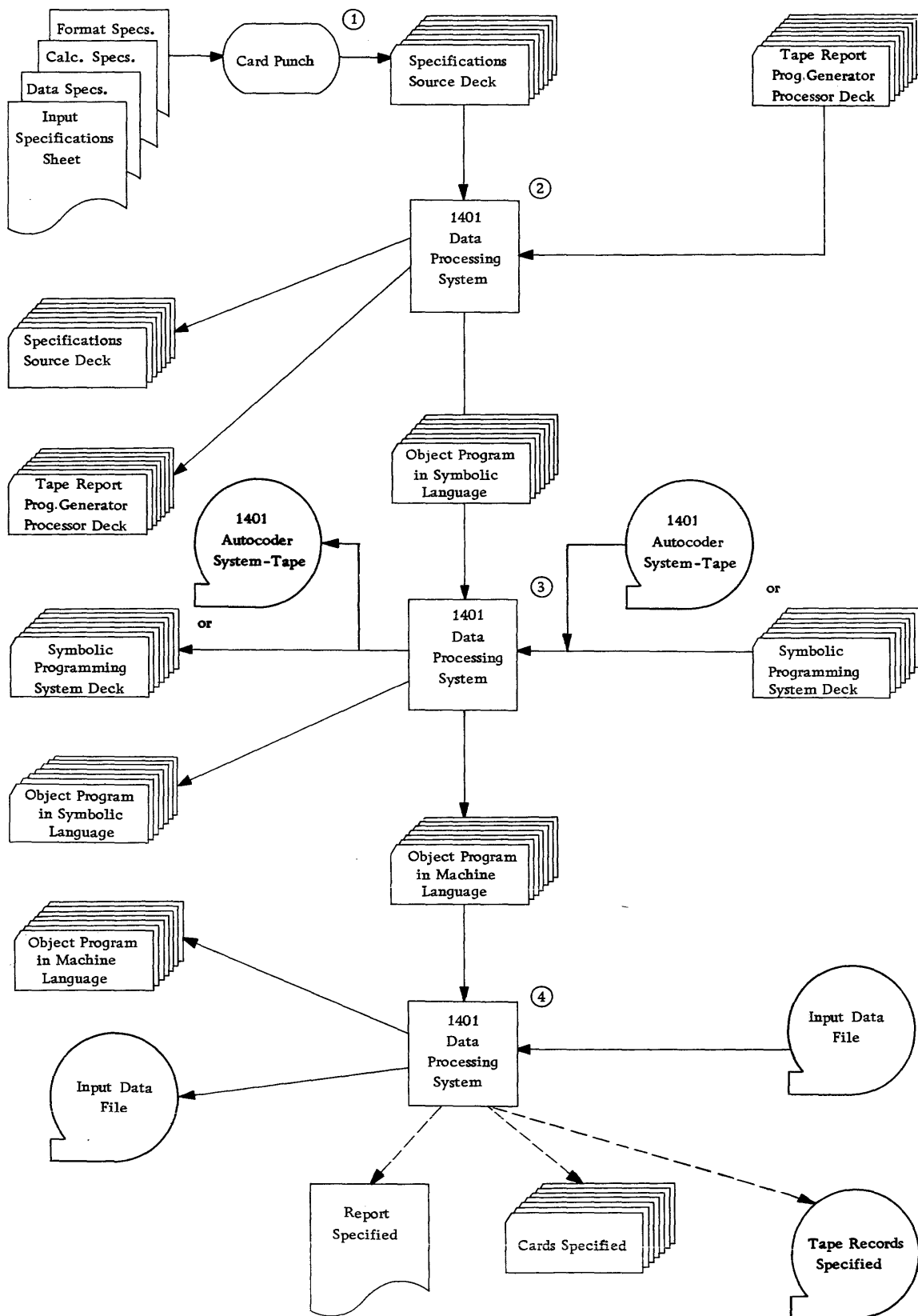


Figure 2. Generating a Report on a 1401 Tape/Card System



# MONTHLY EXPENSE DISTRIBUTION REPORT

REPORT DATE 07-18-60 PAGE 1

	OUR INVOICE NUMBER	DATE MO DAY	AMOUNT	AMOUNT BY ACCOUNT	AMOUNT BY DEPT
*** DEPT. NO. 041					
** GEN. LEDGER NO. 913					
* SUB. LEDGER NO. 660					
	12042	4 07	687.50		
	12084	2 14	721.92		
			1,409.42 *		
* SUB. LEDGER NO. 700					
	12125	11 23	675.95		
			675.95 *		
				2,085.37	
** GEN. LEDGER NO. 915					
* SUB. LEDGER NO. 760					
	12086	12 15	2,119.50		
			2,119.50 *		
				2,119.50	
					4,204.87

Figure 4. Monthly Expense Distribution Report (Part 1 of 3)

# MONTHLY EXPENSE DISTRIBUTION REPORT

REPORT DATE 07-18-60 PAGE 1

	OUR INVOICE NUMBER	DATE MO DAY	AMOUNT	AMOUNT BY ACCOUNT	AMOUNT BY DEPT
*** DEPT. NO. 042					
** GEN. LEDGER NO. 901					
* SUB. LEDGER NO. 615					
	12095	5 08	125.03		
			125.03 *		
* SUB. LEDGER NO. 623					
	12091	6 10	571.00		
	12088	5 16	685.94		
			1,256.94 *		
* SUB. LEDGER NO. 629					
	12080	5 03	24.15		
	12073	5 02	1,631.17		
			1,655.32 *		
* SUB. LEDGER NO. 636					
	12109	7 03	1,725.54		
			1,725.54 *		
				4,762.83	
** GEN. LEDGER NO. 906					
* SUB. LEDGER NO. 643					
	12150	6 08	402.00		

Figure 4. Monthly Expense Distribution Report (Part 2 of 3)

				REPORT DATE 07-18-60	PAGE 2
OUR INVOICE NUMBER	DATE MO DAY	AMOUNT	AMOUNT BY ACCOUNT	AMOUNT BY DEPT	
12090	5 23	331.96			
12106	7 03	95.65			
		829.61 *			
* SUB. LEDGER NO. 650					
12093	8 09	806.00			
		806.00 *			
				1,635.61	
					6,398.44

Figure 4. Monthly Expense Distribution Report (Part 3 of 3)

## First Phase of Report Generating

### General Description

To generate an object program, the IBM 1401 Report Program Generator (RPG) requires certain information. The information answers these questions:

1. What are the characteristics of the file from which the report data is obtained?
2. What type of information is to be extracted from the input file? From which records can these source fields be obtained?
3. What types of calculations are to be performed during the execution of the object program? How are the results of these calculations to be manipulated?
4. What is the format of the report? What headings and constants must it contain? How should the data composing the report be edited?

As shown in Figures 1 and 2, four forms are required for writing the report specifications. The forms contain answers to the preceding questions. The information is punched into cards (Figure 5) with one card punched for each line. These cards comprise the specifications source deck. The RPG program processes the source deck to produce the object program.

Before the actual report specifications can be written, the user must have a clear image of what he wants as the final product. That is, he must know the contents of each line of the report, the spacing between lines, and the positioning of the information within each line of the report. He uses the IBM 1403 *Spacing Chart*, Form X24-6436, before writing specifications. Preparing this chart consists of laying out the complete format of the report to obtain a pictorial representation of the final product. Although no cards for the source deck are punched directly from the entries on this chart, the pictorial representation serves as a guide to completing the four specifications sheets. It thus plays an important role in writing report specifications.

The spacing chart and the four forms required during phase one are listed here in the order in which they are used. A brief description of the functions of each form is also given. Later sections will explain their use in more detail.

#### 1. IBM 1403 SPACING CHART, FORM X24-6436

This chart was described earlier as the form on which the user's image of the report is projected. Define the position of each field on each line of the report and include constant information, headings, and editing symbols, where applicable.

#### 2. INPUT SPECIFICATIONS, FORM X24-1336

A description of the data file, from which the information required for the report will be extracted, must be specified on this form. Describe each type of record in the data file, with its distinguishing record codes and control fields.

#### 3. DATA SPECIFICATIONS, FORM X24-1337

On this form the user lists the data fields necessary for processing the report. These data fields may be output fields or factors in calculations. Each field described is associated with the input record or records that contribute to it. It is also associated with any conditions that govern the processing of those input records. Any number of fields from one or more input records can be listed as the sources of a data field. The input sources can be added and subtracted as well as moved to the data field. Furthermore, the user can state that the status (positive, negative, zero, or blank) of a data field will be needed to govern subsequent processing. For example, a line can be conditioned to print only if a particular data field is positive.

#### 4. CALCULATION SPECIFICATIONS, FORM X24-1338

Although a limited amount of calculation is available through entries on the *data specifications* sheet, the *calculation specifications* sheet must be used for more extensive calculations including multiplication, division, and comparing. This form accommodates calculations on data fields described on the *data specifications* sheet, as well as constants and the results of previous calculations. Half-adjusting, position-adjusting, and the conditions governing the performance of a calculation can all be shown on this sheet. Furthermore, the user can define status conditions based upon the sign of the calculated result or the comparison of two fields.

#### 5. FORMAT SPECIFICATIONS, FORM X24-1339

The final step in writing report specifications is describing the format of output lines. Name each line by its type and relation to other lines. Specify the medium of output (printing, punching, writing on tape) as well as the conditions for output. Stacker selection of punched output or forms control of printed output can be specified. Having named a line, list all the constants, data fields, and edit control-words that compose the line. (Control words specify where commas, decimals, and conditional credit CR or minus symbols are to print and where zero suppression is to stop.) Provision is made for description of conditions, if any, governing the inclusion of a field within a line.



## Correlating the Report Specifications

When completed, the five forms are an interrelated statement of the problem being specified as shown in Figure 6. The spacing chart represents the output lines described in the format specifications sheet. The same line names are used on both forms.

The names given to various input records on the input specifications sheet are the same names used as *field sources* on the data specifications sheet to indicate the record from which a data field is taken during report writing. Each input record is assigned a unique two-digit number called a *resulting condition* number. During report processing, this condition is fulfilled whenever a record with the *record codes* specified for that resulting condition is present in the input area. The fulfillment of such a condition can govern the processing of a source field in the data specifications, the performance of a calculation specification, the placement of a field within a line or the output of that line as stated in the format specifications. The fulfillment of a resulting condition can be compared to the transferring of a selector on an accounting machine during the presence of a particular card. It can also be compared to the setting of a programming switch on a stored-program machine to indicate the presence of a particular record type. The change of a *control field* specified for input records can also govern the processing of source fields (e.g., to provide group indication), the performance of a calculation, the printing of lines, and the punching of cards.

Thus, the input specifications describe the kinds of records in the input data file according to the coding and control fields that are significant in these records. These specifications inherently determine many conditions for processing the data to be extracted from the input file.

The fields named on the data specifications sheet can be used as factors in calculations or as fields in lines. The fields named on the calculation specifications sheet can also be placed in lines on the format specifications sheet. Sometimes the status of a field (positive, negative, zero, or blank) is important in the processing of that or other fields. It may be that calculations should not be performed on zero or blank fields or it may be that a field should be printed in different positions depending upon whether it is positive or negative. Perhaps a line should be printed only when a certain field is not blank. Whenever the status of a field is important to processing, that status can be specified on the data sheet or calculation sheet beside the field name. Then, the status is assigned a unique two-digit resulting-condition number to represent it. The fulfillment of that condition during processing can govern further processing, as just indicated.

Thus, fields from the data and calculation sheets contribute to the lines on the format sheet. Conditions representing the presence of a record in the input area, the change in control between that record and the previous record, or the status of certain fields that have been processed can govern the presence and placement of fields within a line or the printing, punching, or writing on tape of that line. Furthermore, a line can be included in the output because the output conditions were met for a previous line. This relationship is described by a *next line* specification on the format sheet.

Considered together, the five forms represent the input file, the significant data fields within that file, the manipulations necessary to obtain the required output fields, and the line formats in which the fields are to appear.

This summarizes briefly the elements that enter into report specifications during the first phase of report generating. In the sections that follow, each of the forms will be examined in greater detail. The rules and conventions governing report specifications are presented. Two sample applications are used to illustrate pertinent parts of the description. For complete documentation of these sample reports, see Figures 52 to 65.

## IBM 1403 Spacing Chart

The purposes of laying out the report on the IBM 1403 Spacing Chart are:

1. to establish the positions at which the various data will be printed, punched, or written on tape, as well as to indicate the spacing between printed lines, and
2. to assign each line a unique identification code representing
  - a. the type of line,
  - b. the level of the line, and
  - c. the number of the line within its level.

## Layout of Lines and Fields

The numbers across the top and bottom of the spacing chart represent the IBM 1403 print positions. The numbers down the left side are line numbers. The user selects the line number and print positions for a particular field and makes his notation in the selected positions. In the sample layout (Figure 7) note that headings and other constant information are spelled out completely in the print positions assigned to them. Variable information is represented by X's and includes, where applicable, credit symbols, punctuation, etc. The position in an amount field where zero suppression ends is indicated by a zero rather than an X.

6 Lines per inch

GLUE

12											
11											
10											
9											
8											
7											
6											
5											
4											
3											
2											
1											

HA1

DA1

T11

1 2 3 4 5

FORM X24-1336  
PRINTED IN U.S.A.

Page 7677 of         
Date                     

## FORMAT SPECIFICATIONS



INTERNATIONAL BUSINESS MACHINES CORPORATION

## IBM 1403 SPACING CHART

6 Lines per inch

Form X24-6436

PRINTED IN U.S.A.

LINE DESCRIPTION

FIELD HEADINGS/WORD MARKS

0 Lines per inch

0										1										2										3										4										5										6										7										8										9										10										11										12										13																																												
<div style="display: flex; justify-content: space-between;"> <div style="width: 15%;"> <p>GLUE</p> <p>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</p> </div> <div style="width: 85%;"> <p style="text-align: center;">MONTHLY EXPENSE DISTRIBUTION REPORT</p> <p style="text-align: right;">REPORT DATE XX-XX-XX    PAGE XXX</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">OUR INVOICE NUMBER</th> <th style="width: 10%;">DATE MO DAY</th> <th style="width: 15%;">AMOUNT</th> <th style="width: 15%;">AMOUNT BY ACCOUNT</th> <th style="width: 15%;">AMOUNT BY DEPT</th> </tr> </thead> <tbody> <tr> <td>*** DEPT. NO. XXX</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>** GEN. LEDGER NO. XXX</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>* SUB. LEDGER NO. XXX</td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td>DAA</td> <td>XXXXX</td> <td>XX XX</td> <td>XX,XXX.XX</td> <td></td> </tr> <tr> <td>T11</td> <td></td> <td></td> <td>XXX,XXX.XX *</td> <td></td> </tr> <tr> <td>T21</td> <td></td> <td></td> <td>X,XXX,XXX.XX</td> <td></td> </tr> <tr> <td>T31</td> <td></td> <td></td> <td></td> <td>XX,XXX,XXX.XX</td> </tr> <tr> <td>TAA</td> <td></td> <td></td> <td>FINAL AMOUNT</td> <td>XXX,XXX,XXX.XX</td> </tr> </tbody> </table> </div> </div>																																																																																																																																		OUR INVOICE NUMBER	DATE MO DAY	AMOUNT	AMOUNT BY ACCOUNT	AMOUNT BY DEPT	*** DEPT. NO. XXX					** GEN. LEDGER NO. XXX					* SUB. LEDGER NO. XXX					DAA	XXXXX	XX XX	XX,XXX.XX		T11			XXX,XXX.XX *		T21			X,XXX,XXX.XX		T31				XX,XXX,XXX.XX	TAA			FINAL AMOUNT	XXX,XXX,XXX.XX
OUR INVOICE NUMBER	DATE MO DAY	AMOUNT	AMOUNT BY ACCOUNT	AMOUNT BY DEPT																																																																																																																																																																										
*** DEPT. NO. XXX																																																																																																																																																																														
** GEN. LEDGER NO. XXX																																																																																																																																																																														
* SUB. LEDGER NO. XXX																																																																																																																																																																														
DAA	XXXXX	XX XX	XX,XXX.XX																																																																																																																																																																											
T11			XXX,XXX.XX *																																																																																																																																																																											
T21			X,XXX,XXX.XX																																																																																																																																																																											
T31				XX,XXX,XXX.XX																																																																																																																																																																										
TAA			FINAL AMOUNT	XXX,XXX,XXX.XX																																																																																																																																																																										

NOTE: This chart is subject to inaccuracies from variations in humidity. Dimensions of form should be calculated from measurements shown and not scaled from this chart.

### Line-Identification Code

The column at the left on the spacing chart is used to assign each line a three-character identification code. This code identifies the line later on the format sheet where each line is described according to type and content.

#### TYPE

The first character of the identification code is H for a heading line, D for a detail line, or T for a total line. All lines must be identified as belonging to one of these categories.

#### LEVEL

The second character of the identification code can be a number from 1-8 or a letter. A heading or total line within a hierarchy is assigned a number to represent its level. A heading or total line that is independent (not in a hierarchy) is assigned a letter. A detail line can be assigned either a number or a letter to represent its level.

#### NUMBER

Lines within a level can be numbered according to their appearance in the output. Lines with *numerical-level* designations must also have numerical line-number designations. Lines with *alphabetic-level* designations can have numerical or alphabetic line-number designations.

### Classification of Lines

The classification of a line by type is usually self-explanatory. Note, however, that total lines differ significantly from heading and detail lines. Heading and detail lines can contain information from the record in the input area at the time when the lines are produced; total lines can not. An input record can effect the control-field change that causes total lines to be written, but that input record cannot contribute data to these lines. A detail line has a direct relationship to the input record in that most or all of the data in a detail line, including calculated fields, comes from the input record. A heading line generally contains constant information, although it can have some information from input records, including the record present at the time the line is assembled.

The concept of line *level* is based upon the relationship of a line to other lines. Heading or total lines that are independent of each other should be given alphabetic-level designations. Heading or total lines that are related in a hierarchy should be given numerical-level designations corresponding to their positions in the hierarchy. A hierarchical relationship can be likened to total operation on an accounting machine, i.e., major lines force minor and inter-

mediate lines. The principle underlying a hierarchical relationship is that lines of higher level govern lines of lower level. Thus, when the object program is running, a total line with a numerical-level designation such as T3x will force T1x and T2x to come before it whenever the output conditions are fulfilled for T3x.

In the *Monthly Expense Distribution Report* (Figure 4) there are three related levels of total. The lowest level is associated with sub-ledger number, the next level with general ledger number, and the highest level with department number. When the general ledger number changes, the sub-ledger total line prints before the general-ledger total line. When the department number changes, the sub-ledger and general-ledger total lines print before the department total line. Because this hierarchical relationship exists, these lines have been given the numerical-level designations T11, T21, and T31. (See the spacing chart in Figure 7.)

Study of the Monthly Expense Distribution Report reveals a difference in the hierarchical relationships for total and heading lines. Total lines appear in *ascending* order by level; heading lines appear in *descending* order by level. On that report the heading lines associated with department number print before the general-ledger number lines which, in turn, precede the sub-ledger lines. Thus, when department number changes, the H3x lines precede the H2x and H1x lines. When general ledger number changes, the H2x lines print before the H1x lines.

In addition to the three related levels of heading and total lines in the Monthly Expense Distribution Report, there are independent heading and total lines. Page headings printed as a result of page overflow, page totals, and final totals are all examples of lines that are independent of the minor, intermediate, and major relations governing a hierarchy. The illustrated report has both page heading lines (HB1-HB4) and a final total line (TAA). All of these have alphabetic-level designations because they do not relate to lines of other levels.

Page heading lines may be printed because of control-field changes, because of page overflow, or because of both conditions. The latter case is true in the Monthly Expense Distribution Report. Just as the IBM 407 Accounting Machine distinguishes between normal programming and overflow programming, the 1401 Report Program Generator considers normal heading lines distinct from overflow heading lines, even when the lines are alike in format. Thus, some of the heading lines in Figure 7 have two names reflecting their status as both overflow and normal page-heading-lines. The normal heading lines are H3x, H2x, and H1x. The overflow lines are HBx. On a change in department number there is a skip to a new page of the report

and the numerical-level heading lines (H3x-H1x) print. On page overflow the alphabetic-level heading lines HBx print.

When there is a single detail-line format in a report, that line can be given an alphabetic-level designation to reflect its independent status. Such is the case in the Monthly Expense Distribution Report in which the detail line is named DAA. Other applications might have any number of detail-line formats which, when they do not relate to one another, are classified alphabetically by level. It is conceivable that some applications might contain hierarchies of detail lines necessitating numerical-level designations.

Note that the *level* of a line is not necessarily equal to the *number of the control field* with which the line is associated. For instance, a total or heading line of level three may not relate to control field three in the input data file. In the invoice operation described later the level-one heading lines relate to a change in control-field two. In other applications, lines with alphabetic-level designations might relate to control fields. Thus, even though six is the maximum number of control fields that can be specified, there can be eight numerical levels for each type of line specified.

The *line number* permits scheduling lines within a level. The Monthly Expense Distribution Report has six heading lines in the highest level. That level is associated with department number. Whenever the department changes, the six heading lines composing level three print in line-number sequence within that level, i.e., H31, H32, H33, H34, H35, and H36. Even though there is only one line in each of the lower levels of heading line in that report (H21 and H11), the lines have numerical line-number designations because they are hierarchical. The same principle applies to the total lines, T11, T21, and T31, in the same report. Application of the line-number concept to hierarchical total lines corresponds to special programming on the IBM 407 Accounting Machine. For instance, four minor total lines could be named T11, T12, T13, and T14.

Analysis of the independent lines in the Monthly Expense Distribution Report reveals that both numerical and alphabetic line-number designations can be used in classifying lines with alphabetic *level* designations. When page overflow occurs, four heading lines (HB1-HB4) print, and the line number reflects the place of each line in the sequence. If there is only one line in an alphabetic level, that line can have a letter as its line number (DAA and TAA in the example report).

Multiple-line-print (MLP) cards might cause three detail lines to print, and these lines could be named DA1, DA2, and DA3 to reflect the place of each line

in the sequence. Two final total lines in a report might well be named TC1 and TC2.

Line classification is explained further in the description of the format specifications sheet.

### Input Specifications

This form (Figure 8) specifies the input file from which the report is to be prepared. The user describes each type of input record in the file. He specifies the record codes (i.e., the characters used to uniquely identify the records) and the control-data fields significant in that record type. Records that must be processed in sequence within a control group can be given numbers representing their place in the sequence. The following paragraphs describe the information entered on the form.

### Record Sequence

*Column 1* must contain a C for every line-entry that specifies a record type. The C identifies the information as an input specification (as opposed to a data, format, or calculation specification).

*Columns 2-3* specify two numerical or two alphabetic *sequence* characters. The Report Program Generator can accommodate a maximum of 20 unique two-character sequence specifications. If, to ensure proper processing, certain types of input data records must be in an established sequence within a control group, columns 2-3 of the input specifications sheet can contain numerical sequence entries in ascending order.

If input data records do not have an established sequence within a control group, or if it is not desirable to halt processing when the records are out of sequence, alphabetic sequence designations should be used. Some applications contain both sequential and non-sequential records. For example, the invoice form shown in Figure 9 is prepared from the file of cards shown in Figure 10. The source of the invoice date is a date card preceding the entire file. Because this date card has no sequence relationship in a control group in the input file, it is given the alphabetic record sequence AA in columns 2-3 of the input specifications sheet (see Figure 11).

For proper invoice preparation the other input data cards must be arranged within customer number in the sequence shown in Figure 10. Therefore, the *Invoice To* header card's record-sequence number is 01, the *Shipped To* header card is 02, the *Shipped Via* is 03, the *Order Data* is 04, and the *Item* detail is 05. Each time the customer number changes, this sequence begins again.

For sequential records *column 4* indicates the number (either a 1 or N) of that type of record in a control



# 1401 REPORT PROGRAM GENERATOR

## INPUT SPECIFICATIONS

FORM X24-1336  
PRINTED IN U.S.A.

## Report

Page 111 of 111

Programmed by \_\_\_\_\_

Date 7677[illegible]

<b>REPRESENTATIVE COMPANY</b>			
<b>ANY CITY - ANY STATE</b>			
INVOICE TO E C BROWN & CO 2364 MICHIGAN BLVD CHICAGO 27, ILL	SHIPPED TO X-Y-Z SERVICE CORP 222 WESTERN AVE INDIANAPOLIS, IND	INVOICE NUMBER 26115	PAGE 1 JUL 25, 60 29045
SHIPPED VIA WESTERN TRUCKERS		PLEASE REFER TO OUR INVOICE NUMBER WHEN REMITTING TERMS: 2% TEN DAYS F.O.B. FACTORY	
ORDER DATE    ORDER NO. 03/15/60    AK-71423 DE		D NORTON	

ITEM NUMBER	DESCRIPTION	QUANTITY & UNIT OF MEAS.	UNIT PRICE	AMOUNT
115/278	LAG SCREWS 1 x 1/2	2 C	5.25	10.50
115/282	LAG SCREWS 3 x 1/2	5 C	5.30	26.50
SBN- 02	HAMMER-ADZE EYE, BELL POLL, RND NECK	12 DOZ	10.50	126.00
369-2HF	HAMMER-BALL PEEN 2/0, 3/4 LB	4 DOZ	12.64	50.56
101-224	WASHING MACHINE	8 EACH	165.00	1,320.00
21-564	NAILS-STEEL WIRE 60-2 IN	6 CWT	8.30	49.80
21-572	NAILS-STEEL WIRE 100-3 IN	4 CWT	7.80	31.20
143-210	BRUSH, FLAT WALL 3 x 2 5/8 x 13/16	6 EACH	.95	5.70
743D217	GROMMETS	720 EACH	.01	7.20
216-210	PAINT, FLAT WHITE	12 GAL	2.85	34.20

<b>REPRESENTATIVE COMPANY</b>			
<b>ANY CITY - ANY STATE</b>			
INVOICE TO E C BROWN & CO	SHIPPED TO	INVOICE NUMBER 26115	PAGE 2
SHIPPED VIA		PLEASE REFER TO OUR INVOICE NUMBER WHEN REMITTING TERMS: 2% TEN DAYS F.O.B. FACTORY	
ORDER DATE    ORDER NO.			

ITEM NUMBER	DESCRIPTION	QUANTITY & UNIT OF MEAS.	UNIT PRICE	AMOUNT
216-418	PAINT, WALL UNDERCOAT	1 GAL	2.95	2.95
	INV TOT			\$ 1,664.61*
	2% DISC			\$ 33.29 CR
	NET AMT			\$ 1,631.32*

---

FIN TOT	JUL 25, 60	\$	74,921.35**
---------	------------	----	-------------

Figure 9. Invoice Example



- The first line has no resulting-condition number in columns 42-43.
- Succeeding lines have a C in column 1 and the additional record codes.
- Only the last line-entry for a single record type has a resulting-condition number in columns 42-43.

*Column 9* must contain an N if a negation condition is intended; otherwise it is left blank. A negation means that the code described is not present in the record specified.

*Column 11* is used for the particular record code. Any valid alphameric character (including a blank) can be entered in column 11. If the entry in column 10 is a C, the character is entered in column 11. If the entry in column 10 is a D, the exact digit is entered in column 11. If the entry in column 10 is a Z, any character containing the desired zone can be entered.

All record-code entries describing one input-record type are represented by one resulting-condition entry entered in *columns 42-43*. The resulting condition is a unique two-digit number, arbitrarily chosen by the user to represent a record coded according to his specifications.

In some applications there is more than one kind of record of a given type. For instance, in the invoice example just given, suppose that the file of input data cards contained three kinds of detail cards:

1. a 1-punch in column 80,
2. a 2-punch in column 80,
3. a 3-punch in column 80.

none of some kinds. Each kind would have N-number within a control group, and each kind would be optional within that group. The specifications for the conditions just stated consist of a separate line-entry for each kind of detail card, each having its own resulting condition (Figure 12).

When records that are coded differently are to be processed alike for the most part, they have the same sequence designation in columns 2-3 whether that designation is numerical or alphabetic. To represent their unique record codes, however, and to distinguish them later, they receive different resulting-condition numbers. Such records are said to be in an *or* relation.

RECORD SEQUENCE		RECORD CODES																																
C	SER. NUMBER	OPTION	POSITION	MOT	CODE	POSITION	MOT	CODE	POSITION	MOT	CODE	POSITION	MOT	CODE	POSITION	MOT	CODE	POSITION	MOT	CODE	POSITION	MOT	CODE	A RESULTING CONDITION	FIELD 1	FIELD 2								
1	2	3	4	5	6	8	9	10	11	12	14	15	16	17	18	20	21	22	23	24	26	27	28	29	30	31	32							
C	0.5N	X	0.8			C	1									262	7	29	30	32	33	4	35	36	36	39	40	41		0.6				
C	0.5N	X	0.8			C	2																							0.7				
C	0.5N	X	0.8			C	3																							0.8				

## Control Fields

It is not necessary that a control field be in the same position in different record-types. It is important, however, that a field be entered under the proper level on the input specifications sheet. None of the heading cards in the invoice example have the lowest level of control, *Item Number*. Therefore, field 1 is left blank for those record specifications. The next level of control, *Customer Number*, is present in those records. It is specified in columns 49-53 of the sheet. The detail cards in that example have both levels of control. Therefore entries are present for fields 1 and 2 (Figure 13).

20

RECORD SEQUENCE											RECORD CODES											C																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																
C	SEQ	NUMBER	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION	KEY	CODE	POSITION</

Figure 13. Input Specifications for Invoice Report

## Sequence Control

A special entry applies when input data records are present in sequence in a control group of the input data file. The number of the control field governing the sequence of numerically-designated types of records must be specified by the entry SCFx in columns 1-4. The x is the number of the control field. In the invoice example the sequence of records specified relates to customer number. Every change in customer number means that the sequence of records begins again with an *Invoice To* record. Because customer number is field 2 (Figure 13), the proper entry for this Invoice Report is SCF2 in columns 1-4 on the line below the last record specification. Every application that has sequential-record specifications must have an SCFx entry at the end of the input specifications.

## Page Number

Columns 76-77 are used for page numbering. The page-number entry is in the upper right-hand corner of the input specifications sheet. The pages are numbered consecutively beginning with the spacing chart as page 01.

## Card Number

A three-character card number in columns 78-80 establishes the order of entries on the specifications sheet. The first 20 lines of the sheet are prenumbered 010-200. The six unnumbered lines at the bottom of the sheet are for the entry of statements inadvertently omitted and for sheet extension. Insertions can be referenced by numbering the statement with the hundreds and tens digit of the statement it is to follow and with any one of the units digits 1-9. This allows up to nine insertions between two consecutive pre-numbered statements.

## Summary of Input Specifications

From knowledge of the records in the input data file, the user writes the input specifications for the Report Program Generator:

COLUMNS	NAME	EXPLANATION
1	C	C for every line-entry that specifies a record type. If any records

must be in a fixed sequence within a control group, enter an S on the line below entry for the last sequential record-type.

2-3	Seq.	Numerical-sequence order if a fixed record-sequence within a control group is required; two alphabetic letters if a fixed sequence within a control group is not required or if there is only one record-type per control group. Enter the letters CF on the line below the last line-entry for sequential record-type.
4	Number	1 if only one sequential record per control group should be present, or N if more than one is permitted. Blank for non-sequential records. If there is an SCF entry in columns 1-3 of this line-entry, enter the control-field number that controls the sequencing.
5	Option	X if the presence of this sequential record-type is optional; blank if its presence is required. Blank for non-sequential records.
6-8	Position	Location within the record of the record-code character whose presence or absence identifies this record-type.
9	Not	N if the <i>absence</i> of the record-code character identifies this record-type; blank if the <i>presence</i> of the character identifies this record-type.
10	Z/D/C	Z, D, or C according to whether the zone-portion, digit-portion, or the full-character of the code specified in column 11 of this line-entry is to be used to identify this record-type.
11	Code	The record-code character whose presence or absence in a record uniquely identifies the record. Any valid letter, number, or special character (including blank) can be used.
12-41	(Other Codes)	Five other record-codes can be specified for a record-type, each in the same manner as described for the first record-code in columns 6-11.



## Field Status

Columns 11-19 are explained later when their purpose is more meaningful.

## Sources of Field

Columns 20-70 specify three sources of the report field. If more than three sources must be specified for a report field, use more than one line.

Columns 20-22 define the first source of the field. Possible sources are input records, serial numbers, page number, and record-count numbers. The entries are Cxx for input record sources, SER for serial numbers, PAG for page number, and RCT for record-count numbers. When the field source is an input record, the appropriate entry must come from columns 1-3 of the input specifications sheet.

Column 23 contains an N if the field is a numerical field that has zoning in positions other than the units position. The object program will then strip the zero-, 11-, or 12-zones from all positions except the units in the field for which an N is specified. If the units position contains no zone, this specification will create a 12-zone in that position.

The other entry permitted in column 23 is an M. This identifies the entry as a month-field conversion specification. (See *Data Specifications: Month-Field Conversion* for an explanation and example of this entry.) Otherwise leave this column blank.

Columns 24-26 contain the location of the low-order position of the field in the source record. (For IBM 1401 Data Processing Systems using 51-column input data cards, see *Special Feature Specifications*.) For fields whose sources are SER, PAG, and RCT, these columns should be left blank.

Columns 27-29 must contain the field-length in the source record only if that length is different from the unedited (report) field-length specified in columns 8-10. For example, in Figure 15 the entry on the data specifications sheet for SOLDTO gives the unedited field-length in columns 8-10 as 020. Because the source field-length is the same, columns 27-29 are left blank for that field. The entry for AMOUNT specifies an unedited field-length of 006 in columns 8-10. This provides sufficient positions for the largest total expected although the source field-length specified in columns

D	FIELD NAME	FIELD LENGTH UNEDITED	FIELD STATUS										SOURCES																							
			STATUS	RESULTING STATUS	CONDITION	RESULTING CONDITION	STATUS	RESULTING STATUS	CONDITION	RESULTING CONDITION	STATUS	FIELD SOURCE	FIELD END	FIELD LENGTH	OPERATION	COND. A	COND. D	FIELD SOURCE	FIELD END	FIELD LENGTH	OPERATION	COND. A	COND. D													
1	2	7	8	10	11	12	13	14	15	16	17	18	19	20	22	23	24	26	27	28	29	30	31	33	34	36	37	39	40	41	43	44	46	47	48	
	D.SOLD TO	020													C.01	026																				
	D.PAGENO	001													PAG																					
	D.AMOUNT	006													C.05	069	005A																			

Figure 15. Data Specifications for Field Sources

27-29 is 005. The source field length should be left blank when the source is SER, PAG, or RCT.

One of the following seven operation characters must be specified in column 30:

- Blank — Move the source field to the data field.
- A — Add the source field to the data field.
- S — Subtract the source field from the data field.
- 0 — Reset the data field to zero before adding the source field to it. (Punch 12, 0 for 0.)
- 0 — Reset the data field to zero before subtracting the source field from it. (Punch 11, 0 for 0.)
- D — Move the numerical portion of the single-character source field to the units position of the data field. The zone portions are undisturbed in both fields.
- Y — Move the zone portion of the single-character source field to the units position of the data field. The numerical portions are undisturbed in both fields.

Figure 16 illustrates the use of the single-character move-numerical and move-zone operations. Each line-entry is explained as follows:

- The first line-entry specifies that the six-character field at position 074 in input record C05 should be moved to the data field, AMOUNT. Furthermore, the zone portion of the single-character source field in column 69 of the same record is to be moved to the units position of the data field, AMOUNT. By using an entry such as this, the high-order position of the source field can contribute the sign to the units position of the data field.
- The second line specifies that the five-column source field in columns 49-53 of record type CBB is to be moved to the five-position data field, FLDR. The zone portion of the single-character source field in column 70 of CBB is to be moved into the units position of the data field, FLDR. By using an entry such as this, a data field can acquire its units-position sign from any position in a record. By choosing a one-position field source containing no zones for the Y operation, an unwanted sign in the units position of the data field can be removed.

D	FIELD NAME	FIELD LENGTH UNEDITED	FIELD STATUS										SOURCES										OF																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
			STATUS	COND.	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS	CONDITION	STATUS	RESULTING STATUS

Figure 16. Single-Character Move-Numerical and Move-Zone Operations

- The third line specifies that the numerical portion of the one-column source field in column 35 of card type C02 is to be moved into the one-position data field, FLDS. By using an entry such as this, a single-column source field that contains an unwanted sign can be moved into the data field without the unwanted sign.

Because the Y and D operations are single-character move operations, there should be no field length specified for them. When Y or D operations are specified on the same lines as another operation, they should follow the other operation. This ensures the proper sequence of operation in the object program. For example, in the first two line-entries in Figure 16, the Y-entries are made to the right of the blank (move) operation entries.

The same principle of ordering source entries on a line applies in some arithmetic specifications. This is because the object program performs the operations specified in the order of entry on the sheet from left to right.

For example, if a field being described is the result of adding or subtracting more than one field from the same source record, and if that field must be reset prior to the first operation, then the source field described first (in columns 20-36) should be the one that is reset-added or reset-subtracted.

Columns 31-33 can be used to specify a condition that is to govern the operation upon the source field. It can be either a positive condition, such as F3 to indicate a change in control field 3, or a negative condition, such as NF3 to indicate no change in control field 3. The conditions specified on this sheet are the resulting conditions of the input specifications sheet, control-field changes (F1-F6), sense switch settings (SB, SC, and SD), resulting conditions specified by other entries on the data sheet, or the negation of any of these when preceded by an N.

In the Invoice Report shown in Figure 9, the minor group-indicative information (Item Number, Description, Unit of Measure, and Unit Price) is to be obtained from the first card of each minor control group. Hence the specification for ITEMNO is condition F1 in columns 32-33 of the data specifications sheet (Figure 17).

This entry specifies that the source field is to be moved to ITEMNO only from the first card after a change in control field 1.

If the field-source entry on a given line is RCT or SER, the entries in columns 31-36 specify the conditions to be met for increasing the record- or serial-count. All input records can be counted by an entry such as the second line of Figure 17. Selective record counting can be done by the third line-entry in Figure 17 where 02 in columns 32-33 represents the resulting-condition number of a record type defined on the input sheet. For the Invoice Report, Figure 13, condition 02 was assigned to a *Sold To* card. Thus, this count in that application will be increased one for each first card in the intermediate group controlled by customer number.

If the field source entry is PAG, the conditions are for reset of page count. In Figure 9 the pages of each customer's invoice are to be numbered at the top of each invoice form. With a control change in customer number (field 2), page numbering is to begin again at one. Therefore the proper entry for PAGENO as shown on the fourth line of Figure 17 includes F2 as the condition for reset. When no condition is described for a PAG source, the object program numbers the report pages consecutively from beginning to end.

Columns 34-36 can be used to specify a second condition governing the operation upon the source field. If two conditions are specified, they are considered in an *and* relation. That is, both conditions specified must be met before the operation specified in column 30 is performed upon the source field.

When the field being specified has more than one source, the second source is specified by entries in columns 37-53 in the same manner as the first source in columns 20-36. The third is specified in columns 54-70. If more than three sources exist for a field, more than one line-entry is required to specify the sources. In such instances, the first line-entry for the field is prepared as previously described. The second and succeeding line-entries for the field have a D entered in column 1. Columns 2-19 are blank. Columns 20-70 are used, as required, to enter the information for the remaining sources of the field. Figure 18 is an example.

D	FIELD NAME	FIELD LENGTH	FIELD STATUS												SOURCES																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
			UNEDITED	STATUS	RESULTING	STATUS	RESULTING	STATUS	RESULTING	STATUS	RESULTING	STATUS	RESULTING	STATUS	RESULTING	STATUS	RESULTING	STATUS	RESULTING	STATUS	RESULTING	STATUS	RESULTING	STATUS	RESULTING	STATUS	RESULTING	STATUS	RESULTING	STATUS	RESULTING																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
1	ITEMNO	0.8																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																									

Figure 17. Conditioning Operations on Source Fields

## Field Status

The user can govern subsequent processing according to the status (positive, negative, zero, or blank) of a particular field. Columns 11-19 of the data sheet are used to assign resulting-condition numbers to represent one, two, or three possibilities for the status of the field. If the field's status is not significant in the control

[illegible]

Figure 18. Specifying Four Field Sources

of processing, leave columns 11-19 blank.

Column 11 is used for the first status character. The acceptable characters and their meanings are:

B – Blank

 $Z = \pm \text{Zero}$ 

N – Negative, including minus zero

P – Positive, including plus zero

NOTE: Minus zero can occur during processing on the IBM 1401 Data Processing System.

*Columns 12-13* are for the assignment of a resulting condition. This is a two-digit number identifying the field status specified in column 11. The number entered here can be any one in the range 00 through 99 with the exception of any number used to define a resulting condition on the input or the calculation specifications sheets. The field status defined by columns 11-13 refers to the condition of the data field after the operation specified by a source field has been accomplished by the object program. For example, if a source field is added into a data field with a zero status defined as Z25 in columns 11-13, condition 25 will be fulfilled by the program whenever the specified addition results in a zero data field. This condition can then be used to govern other operations in the program such as the performance of a calculation.

Referring to the first line of Figure 17, if after moving the source field from C05 to the ITEMNO field that data field is blank, the resulting condition 07 has been met. This condition has a special use in governing total printing when the input file is first read in. See the section *Suppressing Total Lines and Calculations*.

Columns 14-19, if necessary, specify two more sets of status conditions for the data field.

## Page Number

Columns 76-77 are used for page numbering. The entry is made in the upper right-hand corner of the data sheet. The pages are numbered consecutively beginning with the spacing chart as page number 01.

**Card Number**

Although the entries on the data specifications sheet can be in any order, a card number (*columns 78-80*)

aids identification of line-entries. See *Suggestions and Recommendations: Program Efficiency* for a suggestion about the arrangement of data specifications.

### Conversion of Month Field

In many applications a month field is a single character with the months January through September represented by the numbers 1 through 9, and the months October through December each represented by a zone or a combination of a zone and a number. In these applications the single-character representation of the months can be converted automatically at object-program time to the corresponding two character representation. January through September are then 01 through 09; the months October, November, and December are 10, 11, and 12.

Figure 19 shows the specification for month-field conversion used in the Monthly Expense Distribution Report. Columns 8-10 contain 002, the data field-length after conversion. The M in column 23 identifies a month-field conversion specification. Columns 27, 28, and 29 contain the specific single-characters used in the source field to represent the months October, November, and December, respectively. The other entries on the line are as previously explained.

[illegible]

Figure 19. Specification for Month-Field Conversion

### Serial, Record, and Page Numbers

Any number of serial or record-count numbers can be specified in a single application. Each has the source SER or RCT and a unique data-field name in columns 2-7. It is also possible to maintain a single tally for more than one record type by specifying multiple RCT sources to a data field. Each source then must be conditioned by the resulting-condition number of a different record type. For example, all records having

The object program automatically initializes record, serial, and page numbers so that they begin with the number 1. However, if the user wants an initial number other than 1 in an application with card input, a separate card containing the *initial number minus 1* and a unique code to identify that card can be read in ahead of the input data file. The initializing card just mentioned must have been defined on the input specifications sheet, and an entry on the data specifications sheet must have included the proper reference to the source of the *initial number minus 1*. Prepare the initializing card as follows: for each number (record count, page number, or serial number) punch an initial value in the card in arbitrarily chosen columns. Punch a unique code in the card to identify it.

1. One card for initializing both the page and serial numbers is prepared and read in ahead of the remaining input records when a new machine run begins.
2. Specifications pertaining to the initializing card have been included on the input and the data specifications sheets.

The initializing card is assigned two alphabetic record-sequence characters (NN is arbitrarily chosen) for the input specifications sheet in columns 2-3. Figures 21 and 22 show the information pertaining to the

COLUMNS	PUNCHES	REMARKS
5-6	24	The initial serial number desired minus 1.
8-10	050	The initial page number desired minus 1.
80	1 (12- and 9-punch)	Assuming there are no other types of input records with an 1 in column 80.

Figure 20. Example Format of an Initializing Card

To reinitialize a page, serial, or record-count number at any point during a machine run, include a record such as CNN at that point in the input file. The operation code on the data sheet will probably be 0 in such cases.

[illegible]

Figure 21. Input Specifications for an Initializing Card

[illegible]

Figure 22. Data Specifications for an Initializing Card

From the spacing chart and the input record formats (see Figure 23) the user (1) assigns an alphabetic field-name, no longer than six characters, to each data field, and (2) determines the source of each field named. The four field sources are input records, page number, serial numbers, and record-count numbers.

On the data specifications sheet the user enters the following information about each field of data:

COLUMNS	NAME	EXPLANATION
1	D	Always a D.
2-7	Field Name	Data field name.
8-10	Length	Data field length, unedited.
11, 14, 17	Status	B, Z, N, or P (representing the status of the data field) in one.



Report Monthly Expense Distribution Report

DATA SPECIFICATIONS

Page 013 of 08

Programmed by \_\_\_\_\_

Date 76 77

FIELD NAME	FIELD LENGTH	FIELD STATUS	SOURCES OF FIELD																CARD NUMBER
			FIELD SOURCE	FIELD END	FIELD LENGTH	OPERATION	COND. N	COND. D	FIELD SOURCE	FIELD END	FIELD LENGTH	OPERATION	COND. N	COND. D	FIELD SOURCE	FIELD END	FIELD LENGTH	OPERATION	
DATE	0.08		CAA	0.08															0.1.0
PAGENO	0.03		PAG																0.2.0
IN VNUM	0.05		CBB	0.27															0.3.0
MONTH	0.02		CBB	0.53	-JK														0.4.0
DAY	0.02		CBB	0.55															0.5.0
AMOUNT	0.07		CBB	0.69															0.6.0
TOTAMT	0.08		CBB	0.69	0.07A														0.7.0
DEPNUM	0.03B03		CBB	0.38															0.8.0
GENNUM	0.03		CBB	0.32															0.9.0
SUBNUM	0.03		CBB	0.35															1.0.0
																			1.1.0

Figure 24. Data Specifications for Monthly Expense Distribution Report

		two, or all three of these columns if a status is to be established. Otherwise, blank.	30	Operation	Blank, A, S, $\bar{0}$ , $\bar{0}$ , D, and Y to describe the operation to be performed upon the source field. After the operation, the result is located in the data field.
12-13, 15-16, 18-19	Resulting Condition	Used only in conjunction with the status columns. A unique two-digit number must represent each status specified.	31-33, 34-36	Cond.	One or two conditions to govern the occurrence of the operation specified in column 30. If the operation is to be performed unconditionally, these columns are blank. Permissible entries are SB, SC, SD, F1-F6, resulting-condition numbers from the input sheet, resulting-condition numbers from <i>other lines</i> of the data sheet, and negations (an N in columns 31 or 34) of these conditions.
20-36	(First source of field)	Information for the first source of the data field.			
20-22	Field Source	One of four possible sources: Cxx, PAG, SER, or RCT.			
23	Numeric	N to remove zone information from all positions of the source field except the units. M to convert from a single-character representation of <i>month</i> to the two-digit representation. Otherwise, blank.	37-53, 54-70	(Other field sources)	Information for the second and third field sources of the data field, if applicable. The explanation of the columns for the first source applies to the corresponding columns of the second and third sources.
24-26	Field End	Location of the units position of the field in the source record. Blank if source is PAG, SER, or RCT.	71-75	(Blank)	Not used.
27-29	Field Length	Length of the source field if different from the data-field length. For a month-field conversion entry (M in column 23), the characters used in the source field that represent the months of October, November, and December. Otherwise, blank.	76-77	Page	Page number, located in the upper right-hand corner of the sheet.
			78-80	Card Number	Card number.

The data specifications for the Monthly Expense Distribution Report are shown in Figure 24.

### Calculation Specifications

This form (Figure 25) specifies calculations that are more extensive than those that can be defined on the data specifications sheet. For example, comparison, multiplication, and division can be specified, as well as arithmetic operations to be performed upon resulting fields from either the data specifications sheet or from prior entries on the calculation specifications sheet.

For 1401 Data Processing Systems not equipped with the multiply-divide optional feature, the Report Program Generator will incorporate subroutines in the object program to enable the user to perform multiplication and division. The following paragraphs explain the entries on the calculation specifications sheet.

### A (Arithmetic)

In *column 1* an A (punched as a 12- and a 1-punch) identifies the entry as a calculation specification as opposed to an input, data, or format specification.

### Field Name

The name of the field that will contain the result of the calculation specified is entered in *columns 2-7*. Numbers and special characters cannot be used. Field names of less than six characters must be left-justified.

For a compare specification (C in column 29), columns 2-7 *must* be blank. These columns can also be left blank if the result of this calculation affects the last data field specified as the result of a preceding calculation.

### Field Length Unedited

Columns 8-10 must contain the unedited length of the field. See the corresponding paragraph under *Data Specifications* for further explanation of unedited field-length.

## Field Status

As in the case of data specifications, the user may want to govern later processing (as defined by format specifications and subsequent calculation specifications) according to the status of a particular field. *Columns 11-19* of the calculation specifications sheet can be used for this purpose. The allowable entries in columns 11, 14, and 17 are those given for the field status of the data specifications sheet (B, Z, N, and P), as well as the following entries which are related to the comparison of two fields:

E – Factor 2 is equal to factor 1.

U – Factor 2 is unequal to factor 1.

H – Factor 2 is higher in the collating sequence than factor 1.

L – Factor 2 is lower in the collating sequence than factor 1.

[illegible]

Figure 25. Calculation Specifications Sheet

As mentioned under *Field Status* for the data specifications sheet, each status entry (columns 11, 14, and 17) should be defined by a corresponding unique resulting-condition number (columns 12-13, 15-16, and 18-19). These resulting-condition numbers can be used to govern other operations.

Columns 20-25 are used to state the field name or literal that is the multiplicand, dividend, augend, minuend, or the field with which another field is compared. A field name thus entered must have been specified either by an appropriate entry in columns 2-7 of the data specifications sheet, by a previous line-entry on the calculation specifications sheet, or by a W-entry on a format specifications sheet (see *Format Specifications*). The alphabetic name of factor 1 is left-justified when it is less than six letters long, the first letter being entered in column 20.

If factor 1 is a literal in the range from 000001 through 999999, the entry in columns 20-25 is the numerical value of that literal with its sign indicated over the units position, as shown above. When punched into a specifications card, the sign of the constant is overpunched in the units position as a 12-punch if plus or an 11-punch if minus. In the absence of a sign overpunch, the constant is treated as a positive number. A constant whose value can be expressed in less than six digits need not be entered with leading zeros. Thus, the constant +1 can be entered as a 1 in column 25, with columns 20-24 left blank. Literals of less than six digits are right-justified, the units position being entered in column 25.

Column 29 can be used to identify the operation to be performed using the two factors. The entries and the operations they represent are:

Columns 30-35 are used for the name of the field (or the numerical value of the constant) that is being specified as one of the following: multiplier, divisor, addend, subtrahend, or the field compared with factor 1. The explanation given for factor 1 applies to factor 2.

Column 39 must contain an A, an S,  $\bar{0}$  (12, 0), or  $\bar{0}$  (11, 0), depending upon whether the result of the calculation being specified by this line-entry is to be added to, subtracted from, or substituted for the previous contents of the field named in columns 2-7 of this line-entry. Figure 26 illustrates various multiplication operations. Plus, minus, and divide operations can be similarly specified. An explanation of the entries in Figure 26 follows:

1. The first line-entry specifies that the product of  $FLDB \times FLDC$  be added to the contents of  $FLDA$  and that the sum be placed at  $FLDA$ .
2. The second line specifies that the product of  $FLDB \times FLDC$  be subtracted from the contents of  $FLDD$  and that the difference be placed at  $FLDD$ .
3. The third line-entry specifies that the product of  $FLDB \times FLDC$  be reset-added into  $FLDE$ .
4. The fourth line specifies that the product of  $FLDB \times FLDC$  be reset-subtracted into  $FLDF$ .
5. The fifth line specifies that the product of  $FLDB \times WORD03$  be subtracted from the contents of  $FLDG$  and that the difference be placed at  $FLDG$ .  $WORD03$  is the field name of a 9-position constant that must be defined on the format specifications sheet by a  $W$ -entry.

[illegible]

1. The first line specifies that the field AMOUNT be added to the contents of TOTAMT and that the sum be placed at TOTAMT.
2. The second line specifies that the field AMOUNT be subtracted from the contents of TOTAL and that the difference be placed at TOTAL.

results of a calculation specified as a total-time calculation. A total calculation is similar to a group total on an IBM 407 Accounting Machine application, where the card at the first reading station that initiates a control change does not contribute to group totals for the previous group.

A detail calculation is performed after the test for a control-field change is made and after information is removed from the record in the input area. The record just read can, therefore, contribute to the results of a calculation specified as a detail-time calculation.

When a detail calculation is to be performed for every record in the input file, the D in column 49 may be sufficient to condition the calculation. Selected detail calculations and *all* total calculations must be governed by specifications in columns 40-48 as well as T or D in column 49.

Figure 27. Progressive-Total SpecificationsFigure 28. Comparison Specification

Columns 40-48 can be used to specify a maximum of three conditions in an *and* relation to govern whether the calculation specified is to be performed. The conditions that can be entered are any of the resulting conditions previously defined on the input, data, and calculation specifications sheets as well as LC, F1-F6, SB-SD, and negations of each of these. For example, if one condition to be specified is *no change in control field number 3*, the entry in columns 40-42 will be NF3.

Column 49 must contain the letter T or D, depending upon whether the line-entry specifies a total or a detail calculation.

### Half-Adjust

Columns 50-51 are used to specify the position of the result that will be half-adjusted (have 5 added to it). The *result* just referred to is the data developed by the calculation specification before the operation specified in column 39 of this specifications sheet. If half-adjusting were specified in the first line-entry in Figure 26, it would be accomplished after multiplying FLDB by FLDC but before adding the result to FLDA. If half-adjusting were specified on the first line in Figure 27, the field AMOUNT would be half-adjusted just before being added to TOTAMT. Whenever the units position of the result is to be half-adjusted, the entry in columns 50-51 is 01; if the tens position, 02; if the hundreds position, 03; etc.

Columns 52-53 are used to specify the number of positions to be *dropped* from the result of the calculation. The *result* referred to is the same as explained in the preceding paragraph. If the units position of the result is to be dropped, the entry in columns 52-53 is 01; if the units and tens positions are to be dropped, 02 is entered; if units, tens, and hundreds, 03; etc.

Half-adjusting, position-adjusting, or both can be specified separately from any other calculation. For example, if field A has the form 31415926 and the user wants to retain only the five high-order digits of field A with the units position rounded (to yield 31416) the appropriate entries are shown in Figure 29.

Figure 29. Half-Adjust and Position-Adjust Specifications

	Decimal Positions in Factor 1						
	0	1	2	3	4	5	6
0	00	00	00	01	02	03	04
1	00	00	01	02	03	04	05
2	00	01	02	03	04	05	06
3	01	02	03	04	05	06	07
4	02	03	04	05	06	07	08
5	03	04	05	06	07	08	09
6	04	05	06	07	08	09	10

Example:  $xxx.xxx \times xx.xx =$   
 Round this position.  
 Drop these positions.

Factor 1 has 3 decimal places;  
 factor 2 has 2 decimal places.

By looking in the vertical row labeled 3 and in the horizontal line labeled 2, the number at the intersection is 03. This is the number to specify in the Half-Adjust and in the Position-Adjust columns of the Calculation sheet.

NOTE: For a three-decimal product rounded to the nearest thousandth, subtract 01 from the table values.

For a one-decimal product rounded to the nearest tenth, add 01 to the table values

Figure 30. Number of Positions to Half-Adjust and Position-Adjust to Yield a Two-Decimal Product Rounded to the Nearest Hundredth

In most commercial applications involving multiplication, the product should be rounded to the nearest hundredth, and positions to the right of the hundredths should be dropped. Figure 30 is a chart to aid the user in selecting the proper half-adjust (columns 50-51) and position-adjust (columns 52-53) entries for the calculation sheet.

### Page Number

Columns 76-77 are used for page numbering. The numbers are punched in the cards from the page-number entry in the upper right-hand corner of the calculation specifications sheet. As previously mentioned, the sheets are numbered consecutively beginning with the spacing chart as page number 01.

### Card Number

The first twenty lines of the sheet have preprinted card numbers (see *Input Specifications: Card Number*). The cards punched from the calculation specifications must be arranged in order by page number (columns 76-77) and card number (columns 78-80).

### Leaving "Field Name" Blank

Figure 31, which illustrates a withholding-tax calculation specification, shows how the field name (columns 2-7) and its length (columns 8-10) can properly be left blank. In that figure, the problem is:

Weekly withholding tax = [gross pay - (tax class  $\times$  13.00)]  $\times$  18%.

The line-entries in Figure 31 perform the following operations at total-time following a change in control field 1:

1. The first line resets to zero the five-position field in core storage labeled WHTAX and adds to the WHTAX field the five-position amount, GROSS.
2. The second line multiplies TAXCL by the literal 1300, subtracts the product from the WHTAX field, and then tests the sign of WHTAX. If the sign is negative (meaning that the employee's non-taxable income for the week exceeds his gross earnings), resulting-condition indicator 10 is set on.
3. If WHTAX is not negative (resulting-condition indicator 10 is not on), the third line multiplies the result of the previous calculation, WHTAX, by the literal 18 to produce a seven-position product. The object program then half-adjusts the tens position of the product, drops the units and tens position of the product (leaving the five high-order positions), resets to zero the five-position WHTAX field, and adds the five-position product to the WHTAX field.

		FIELD STATUS										FACTOR 1		FACTOR 2		COND. N COND. N COND. N COND. N COND. N COND. N COND. N COND. N COND. N										HALF ADJUST		POSITION ADJUST																																																																																																																																																																																																																																																																																																																																																																																																																																																																	
A	FIELD NAME	FIELD LENGTH	UNORDERED	STATUS	UNIT	RESULTING	RESULTING	RESULTING	RESULTING	RESULTING	MULTIPLY DIVIDED	AUGMENT	MINOR	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT	UNIT

Figure 31. Leaving Field Name and Length Blank



52-53	Position Adjust	Highest-order position to be dropped from the result of the calculation. That position and any others to its right will be dropped.
-------	--------------------	---

The calculation specifications for the Monthly Expense Distribution Report are shown in Figure 32.

The entries on this sheet pertaining to the *fields* within a line supply the RPG with information such as field name, the rightmost position of the field in the

34

output line, the conditions under which the field will be placed in the line prior to printing, punching, or writing on tape, the edited field-length, and information controlling zero-suppression and editing.

Much of the information for the entries on the format specifications sheet is taken from the spacing chart previously described.

*Entries for hierarchical line specifications must be in the same order as they will appear on the report. Entries for all of the fields within a line must follow the entry for that line specification. The field entries may, however, be in any order after the line specification.*

### Format

In *column 1* a letter must be entered that designates the entry as a format specification. An L in that column identifies the entry as a format specification for a line.

### Line

*Columns 2-4* identify the line being specified. Entries for these columns are taken from the spacing-chart entries that define the line by *type*, *level*, and *number*, as explained previously under *IBM 1403 Spacing Chart*.

The entry in column 2 specifies the *type* of the line. It must be H for a heading line, D for a detail line, or T for a total line. An important consideration in assigning *type* to a line is the difference between a total line and a heading or detail line with regard to the record in the input area when the line is formed. This difference can be noted from the block diagrams of the object programs generated by the RPG. Figures 48 and 49 illustrate that the test for control-field changes, the performance of total-time calculations, and the formation of total lines precede the function of removing fields from the input record. Thus, an input record that causes a control change *cannot* contribute data to total lines that result from that control change.

The block diagrams also illustrate that detail calculations, heading lines, and detail lines follow the removal of fields from the record in the input area, and thus that record can contribute to these calculations and lines. Therefore, the naming of a line according to type is not arbitrary, particularly with regard to total and detail lines.

Column 3 specifies the level of the line, as explained under *IBM 1403 Spacing Chart*. Column 4 specifies the number of the line within a given level. For example, in the invoice application spacing chart shown in Figure 60, there is one hierarchical (numerical) level of heading lines that includes five lines. Those lines have the line-identification codes H11, H12, H13, H14, and H15. Although these heading lines are associated

with customer number, which is the intermediate control field (F2), the lines are *level one* because there are no heading lines of lower level. This illustrates the principle that *line level* is not necessarily equal to the *number of the control field* with which the line is associated. Because the level is numerical, the line numbers must be numerical to reflect their sequence within the level.

As mentioned in the spacing chart explanation of the *Classification of Lines*, report lines to be used as output when a page-overflow condition occurs must be identified by an alphabetic-level character to reflect their independence from a hierarchy. The overflow heading line in the invoice application (see Figure 9) is similar to the first normal heading line. For this reason there are two line-identification codes in Figure 60 for the first heading line (H11 and HAA). As previously mentioned, the line-number entry in column 4 for a line with an alphabetic-level character, such as HAA, can be a number or a letter. Because there is only one page-heading line rather than a sequence, an alphabetic line-designation was used in the example.

The entries on the format specifications sheet pertaining to lines must be in *descending level-order for heading lines*, and in *ascending level-order for total lines*. This is the normal order of printing related report lines, as can be noted on the spacing chart in Figure 7 and on the printed report in Figure 4.

### Output

*Column 5* must contain an alphabetic X if the line is to be printed; otherwise it is left blank.

*Column 6* must contain an alphabetic X if the line is to be punched into an output card; otherwise it is left blank.

If both printing and punching are being specified for the line, an X must be entered in both columns 5 and 6.

*Column 7* must contain an alphabetic X if the line is to be written on magnetic tape; otherwise it is left blank.

If printing, punching, and writing on tape are being specified, an X must be entered in each of the three columns, 5-7.

### Next Line

*Columns 8-10* define the next line to be printed, punched, or written on tape. This entry is made only if the *next line* specified in these columns should come unconditionally in the output after *this line* (the line being described in this entire line-entry). Otherwise columns 8-10 should be left blank. When a *next line* is specified in columns 8-10, that next line must be of

in the carriage tape is sensed while printing a total line, all total lines whose output conditions are met print before overflow-skipping occurs.

## Stacker

In *column 19* the entry of a 4 or 8 causes the card containing this line to be selected into the corresponding 1402 punch stacker. When no selection is desired, column 19 must be left blank.

In *columns 13-14* 01, 02, or 03 calls for single, double, or triple line-spacing *after* the printing of the line being specified.

### Line Output Conditions

Columns 20-28 can be used to specify a maximum of three conditions under which the line being specified is to become output. If two or three conditions are entered on one line-entry, they are considered in an *and* relation. The entries in these columns can be any of the resulting conditions defined on the other specifications sheets, as well as OF (overflow), LC (last card), 1P (first page), F1-F6 (a change in control fields 1 through 6), SB-SD (sense switches), and negations of any of these. If a line is referenced by a previous entry as a *next line*, columns 20-28 are left blank. This is because its line-output conditions must be the same as those for the line which referenced it.

In *columns 17-18* entries 01-12 cause skipping to carriage tape channels 1-12, respectively, *after* printing the line being specified.

In preparing reports that require headings at the top of a page, format specifications usually control form-skipping upon overflow. For a simple report, however, no specifications are necessary to cause the forms to be advanced from the last printed line on a page to the first print line of the next page. Thus, if report specifications provide no forms control for overflow, whenever the object program senses a punch in carriage tape-channel 12 while printing, it automatically causes skipping to channel 1. When the 12-punch

A line will appear in the output *only* if:

1. *Line-output conditions* for the given line are specified and fulfilled, or
2. The line was specified as the *next line* of another line for which the output conditions are fulfilled, or

IBM	<b>1401 REPORT PROGRAM GENERATOR</b>	FORM X24-1339 PRINTED IN U.S.A.
Report Invoice Report		Page <u>105</u> of <u>06</u> <small>76.77</small>
Programmed by _____		Date _____

Figure 34. Conditioning a Line to Print upon Reading a Specific Kind of Data Card

- The line is of lower level in a hierarchy than another line of the same type for which the output conditions are fulfilled.

To illustrate the first principle, in Figure 34 the line H11 is specified to be printed when condition 04 is met. Because condition 04 is defined on the input specifications sheet to be the presence in the input area of a record containing an L in column 80, the first heading line will be printed when that record is present.

The second principle is illustrated in Figure 35. Heading line H12 will print only after line H11 prints, because the format specification of H11 contains a *next line* entry of H12. Note that no output conditions are specified for H12.

A line conditioned by OF cannot be associated with any lines not conditioned by OF. Thus, a line conditioned by OF cannot specify a *next line* not conditioned by OF; nor can a line not conditioned by OF specify a next line conditioned by OF.

F	LINE	OUTPUT	NEXT LINE	SPACE	SKIP	LINE OUTPUT CONDITIONS			FIELD NAME	FIELD END	FIELD OUTPUT CONDITIONS			CONST.
						COND.	N COND.	N COND.			COND.	N COND.	N COND.	
1	H11	1	H12	0	1	0	1	0	0	0	0	0	0	
2	H12	2												
3	H13	3												
4	H14	4												
5	H15	5												
6	H16	6												
7	H17	7												
8	H18	8												
9	H19	9												
10	H20	10												
11	H21	11												
12	H22	12												
13	H23	13												
14	H24	14												
15	H25	15												
16	H26	16												
17	H27	17												
18	H28	18												
19	H29	19												
20	H30	20												

Figure 35. Conditioning Lines to be Used as Output

The third principle with regard to line-output conditions relates to the specification of hierarchies of lines described extensively under IBM 1403 *Spacing Chart*. Lines related in a hierarchy must be of the *same type* and must have *numerical-level designations* that reflect their relative positions within the hierarchy. The processing principle underlying a hierarchical relationship is that lines of a higher level govern lines of lower level. Thus, when the output conditions are fulfilled for T3x lines, the object program will force T1x and T2x lines to come before the T3x lines in the output *without regard for the line-output conditions of the T1x and T2x lines*. This is precisely what happens in the Monthly Expense Distribution Report.

In addition, multiple lines within one level in a hierarchy must be referenced by *next line* designations rather than individual line-output conditions whenever there is a higher level of lines in the hierarchy. Thus, H21 must call for H22 as a next line and H22 must call for H23 as a next line to ensure that all three will be present in the output when H3x lines precede them.

In the invoice example, the H15 line has a separate condition from the H11, H12, H13, and H14 lines, and so it is not called for as a *next line* by H14. If there

were heading lines of level two, the H15 line would be missing whenever the level-two lines forced the level-one lines to follow in the output. Such a specification error could be corrected in two ways.

The first solution is to cause *all five* of the level-one heading lines to print when the *Order Data* card is present (condition 05) rather than the first four lines when the *Shipped Via* card is in the input area (condition 04). Then the *next line* reference can be used in the H14 line-entry to call for H15.

The other solution is to put aside the heading line hierarchical relationship by naming the supposed level-two lines HBx and the level-one lines HC1, HC2, HC3, HC4, and HC5. The level-one lines can then have the same output conditions that they now have. The two levels of lines are then independent of one another, and each prints according to its own condition.

If the conditions for both levels were fulfilled simultaneously, the lines would appear according to the order of their entry on the format specifications sheet. Therefore, even if alphabetic-level designations were used, specify the level-two lines before the level-one lines to maintain the descending order of heading lines in the output.

In applications requiring multiple line-levels (minor, intermediate, major) that occur simultaneously, the specification of hierarchically-related lines with applicable next-line designations results in a faster object program than does the specification of independent lines. If, however, the individual conditions of lines of lower level cannot be ignored when the conditions are fulfilled for lines of a higher level within the same type, numerical-level designations should not be used.

For example, assume minor totals to be conditioned to print on F1 and SB, intermediate totals on F2 with no sense switch conditioning, and major totals on F3 and SC. With line names T1x, T2x, and T3x, the minor and intermediate lines print before the major lines, when F3 and SC are ON without regard for the setting of sense switch B. When F3 is ON and SC is OFF, the major lines do not print because their conditions are not met; the intermediate lines print because their conditions are met. They force the minor lines without regard for SB. For correct processing, name these lines with alphabetic-level designations such as TAx, TBx, and TCx. Then, each level will have its conditions checked independently. Again the lines must appear in the specifications in the order in which they should appear in the output.

In summary, the line output conditions are equally as important as the type and level distinctions in the relationship of lines to one another. Careful analysis should be made of these conditions when the lines are named.

CONDITION 1P (FIRST PAGE)

1P is a condition unique to the format specifications. It is fulfilled at the beginning of processing before any input records have been read. The purpose of this condition is to cause the printing of a cover sheet or page heading lines on the first sheet that print as a result of OF on the following sheets. Thus, lines that contain *only constant information* and that appear in the output before any input records are processed can be printed, punched, or written on tape as a result of condition 1P. Because this condition precedes any input record, no information from the input file can appear in a line conditioned by 1P. Thus, a page heading line that contains a date from the first record in a file will not contain that date on the first sheet if the line is conditioned by 1P. The line should be conditioned by the resulting-condition number that represents the date header record.

VARIABLE LINE-OUTPUT CONDITIONS

Some applications require that varying conditions govern the appearance of a line in the output. A line that is governed by *or* conditions must be specified with a separate line-entry for each *or* condition. The first line-entry made in the normal manner will specify the first *or* condition. Each succeeding line-entry consists of an L in column 1 and the *or* condition in columns 20-28 as required.

For example, if a detail line called DAA is to be written on tape when any one of the following three conditions is met: (1) condition 02 and not condition 05; (2) condition 06, or (3) condition 09; the proper entries on the format specifications sheet are as shown in Figure 36.

F O R M A T T I N G	LINE		OUTPUT		NEXT LINE		SPACE		SKIP		LINE OUTPUT CONDITIONS								FIELD NAME		FIELD END		FIELD OUTPUT CONDITIONS								ZERO SUPPRESS		FIELD LENGTH																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																										
	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																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																									
	TYPE	LEVEL	NUMBER	PUNCH	RESERVED	TYPE	LEVEL	NUMBER	BEFORE	AFTER	BEFORE	AFTER	BEFORE	AFTER	BEFORE	AFTER	BEFORE	AFTER	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N

Figure 36. Or Line-Output Conditions

A line whose output is governed by conditions in an *or* relation can be governed by OF only if OF is part of every *or* condition. For example, if a heading line identified as H25 is to be printed upon occurrence of overflow and either conditions 07 or 13, the proper entries on the format sheet are as shown in Figure 37.

NOTE: Columns 29-75 are left blank for line specifications (L in column 1). As explained later, columns 76-80 pertain to both line and field specifications.

FORMATTING	LINE		OUTPUT		NEXT LINE		SPACE		SKIP		LINE OUTPUT CONDITIONS								FIELD NAME		FIELD END		FIELD OUTPUT CONDITIONS								ZERO SUPPRESS		FIELD LENGTH	
	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
	TYPE	LEVEL	NUMBER	PUNCH	RESERVED	TYPE	LEVEL	NUMBER	BEFORE	AFTER	BEFORE	AFTER	BEFORE	AFTER	BEFORE	AFTER	BEFORE	AFTER	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N	COND. N
L	H	2	5	X															0	7														
L																			0	13														

Figure 37. OF as an Or Line-Output Condition

Format

In the description of a field within a line or the definition of a constant *column 1* contains an F, B, K, or W. The meaning of these entries is as follows:

- F — identifies the entry as a *field* specification for a data field that *will not be blanked* after it is placed in an output line.
- B — identifies the entry as a *field* specification for a data field that *will be blanked* after it is placed in an output line. This entry causes processing similar to a readout and reset total operation on an accounting machine.
- K — identifies the entry as a *field* specification that uses a constant.
- W — identifies the entry as a *field* specification that *defines* a constant or an edit control-word.

NOTE: Columns 2-28 are blank in a line-entry that specifies a field or a constant.

Field Name

Columns 29-34 state the name of the field to be inserted in the line whose specification immediately precedes the field entry. An entry with a B in column 1 must contain a field name from columns 2-7 of the data or calculation specifications sheets. An F entry in column 1 can also contain a field named in the data or calculation specifications, or it can have a WORDxx defined by a W-entry elsewhere in the format specifications. Those field names that are less than six characters long must be left-justified. The entry of a K in column 1 requires a blank field name. The entry of a W in that column must have a field name of the form WORDxx, where xx is a number in the range 00-99. W-entries are fully explained later in this section.

Field End

Columns 35-37 contain the number of the rightmost position of the field in the output line as shown on the spacing chart. These columns are left blank for a W-entry.

Field Output Conditions

Columns 38-46 provide for a maximum of three conditions, considered in an *and* relation, under which the field being specified is to be placed in the output line.

[illegible]

### Z (Zero Suppress)

### Constant or Edit Control Word

In the invoice example the total invoice-amount for each customer prints on the next line below the last minor total for that customer. The total is identified by the constant INV TOT on the same line. The amount, which will not exceed 99999.99 dollars, is edited with an edit control-word to print \$xx,xxx.xx\*. The necessary format entries to specify this total line are shown in Figure 39. This figure also illustrates the use of the spacing chart with the format specifications sheet to produce a total line in the desired format.

A constant that is used only once should be specified by a K-entry in the manner illustrated in Figure 39. An edit control-word used only once should be entered beside the field to be edited as shown in the same figure. Both constants and edit control-words used more than once should be defined by a W-entry to produce a more efficient object program. Each W-entry must have a field name of WORDxx, where xx is a

When a constant, defined by a W-entry somewhere on the format sheet, is to be part of an output line, an F-entry is used in column 1. The F-entry has the field name of the constant (WORDxx) in columns 29-34 and the position of the constant in the line in columns 35-37.

As mentioned, but not illustrated previously, an edit control-word can be defined as WORD<sub>xx</sub> by a W-entry on the format sheet. This permits the name, WORD<sub>xx</sub>, to be entered in columns 51-56 to edit a data field. In Figure 41 the third and fifth lines use WORD<sub>22</sub> to edit the two data fields, GROSS and NET. The W-entry, defining WORD<sub>22</sub> as \$bb,bb0.bb is shown on the ninth line.

Page Number

**Card Number**

The first twenty lines of the sheet have preprinted card numbers in *columns* 78-80, as explained in this corresponding paragraph under *Input Specifications*.

Figure 39. Using Constant and Edit Control-Word

Spacing Chart

INVOICE EXAMPLE PAGE 01 OF 06

INVOICE AMOUNT TOTAL

INVOICE EXAMPLE PAGE 06 OF 06

IBM 1401 REPORT PROGRAM GENERATOR

Report \_\_\_\_\_

Programmed by \_\_\_\_\_

FORM X24-1339 PRINTED IN U.S.A.

Page 11 of 76

Date 76 77

LINE	OUTPUT	NEXT LINE	SPACE	SKIP	LINE OUTPUT CONDITIONS	FIELD NAME	FIELD END	FIELD OUTPUT CONDITIONS	CONSTANT OR EDIT CONTROL WORD	CARD NUMBER
1	LINE	1			COND. N COND. N COND. D					75 76 80
2	LINE	2			COND. N COND. N COND. D					0,1,0
3	LINE	3			COND. N COND. N COND. D					0,6,0
4	LINE	4			COND. N COND. N COND. D					0,7,0
5	LINE	5			COND. N COND. N COND. D					0,8,0
6	LINE	6			COND. N COND. N COND. D					0,9,0
7	LINE	7			COND. N COND. N COND. D					1,0,0

Format Specifications Sheet

Printed Report

REPRESENTATIVE COMPANY

ANY CITY - ANY STATE

INVOICE TO E C BROWN & CO

SHIPPED TO

INVOICE NUMBER 26115

PAGE 2

SHIPPED VIA

ORDER DATE ORDER NO.

PLEASE REFER TO OUR INVOICE NUMBER WHEN REMITTING TERMS: 10 TEN DAYS F.O.B. FACTORY

ITEM NUMBER 216-418

DESCRIPTION PAINT, WALL UNDERCOAT

QUANTITY & UNIT OF MEAS. 1 GAL

UNIT PRICE 2.95

AMOUNT 2.95

INV TOT

\$ 1,664.61

This control-word edited the data field, TOTAMT (0166461), suppressing the high-order zero and adding the symbols specified.



**IBM**

FORM X 24-1339  
PRINTED IN U.S.A.

## FORMAT SPECIFICATIONS

Page 76 of 77  
Date \_\_\_\_\_

[illegible]

## Summary of Format Specifications

Having completed the spacing chart showing the layout of the desired lines and the input, data, and calculation (if required) sheets, the user writes the format specifications for the output lines.

The following general rules govern the order of the entries on the format sheet:

1. The first entry must be a format specification for a *line*.
2. An entry made for a given line must be followed by entries for each field that must appear in that line.
3. Entries for lines should be specified in the order in which the lines must appear in the output.

The following is a summary of entries on the format specifications sheet.

COLUMNS	NAME	EXPLANATION
1	Format	Letter (L, F, B, K, or W) to identify the line-entry as a format specification.
Columns 2-28 apply to line-specification entries only.		
2-4	Line	Line-identification code for this line (taken from the spacing chart).
5	Print	X if this line is to be printed.
6	Punch	X if this line is to be punched.
7	Reserved	X if this line is to be written on magnetic tape.
8-10	Next Line	Line-identification code of the next line provided that it is to be used as output under the same output conditions as those governing the line being specified. The next line must be of the same type and level as the line calling for a next line.
11-12	Space Before	Numbers 01, 02, or 03 for single, double, or triple line-spacing <i>before</i> printing this line.
13-14	Space After	Same as columns 11-12, except for spacing <i>after</i> printing.
15-16	Skip Before	01-12 for skipping to carriage-tape channel 1-12 <i>before</i> printing.
17-18	Skip After	Same as columns 15-16, except for skipping <i>after</i> printing.
19	Stacker	4 or 8 to select the card representing this line into either the 4 or the 8 punch-stacker pocket. Blank when not selecting.

20-22 Line Output Condition First condition to govern the appearance in the output of this line. The following can be entered: any resulting-condition number defined on the input, data, and calculation sheets; OF, LC, 1P, F1-F6, SB-SD, and negations of these.

23-28 (Two other conditions) Two other conditions to govern the appearance in the output of this line. Output conditions for a given line are considered in an *and* relation.

A line will be used as output only if: (1) its line-output conditions are specified and fulfilled, or (2) the line is specified as the *next line* in the L-entry of the preceding line, and that preceding line's output conditions are fulfilled, or (3) the line is of a lower level in a hierarchy than another line of the same type for which the output conditions are fulfilled.

Columns 29-75 pertain to field specifications only.

29-34 Field Name If the field is not a constant, the name of the field. That name must appear in columns 2-7 of either the data or calculation sheet or be a WORDxx. If the field is a constant defined by a W-entry, the name is WORDxx. If the field is a constant not defined as WORDxx, these columns are blank.

35-37 Field End Print position of the units position of the field.

38-40 Field Output Condition First condition to govern the inclusion of the field in the output line. The same conditions are accepted here as are accepted in columns 20-22.

41-46 (Two other conditions) Two other conditions with the same function as the first condition in columns 38-40. The field-output conditions for a given field are considered in an *and* relation. If a field must be included unconditionally in an output line, columns 38-46 are blank.

47	Zero Suppress	Z to zero-suppress a field without using an edit control-word.
48-50	Field Length	Length of either an edit control-word used with the field or a constant that is defined in columns 51-75.
51-75	Constant or Edit Control Word	Either a constant (numerical or alphameric) or an edit control-word that is being used or defined.
76-77	Page	Page number of this specifications sheet, located in the upper right-hand corner.
78-80	Card Number	The first twenty lines of each sheet are prenumbered.

COLUMNS	PUNCHES	EXPLANATION
1-4	CNTL	Identifies the card.
5	One of these: 3-6	Core-storage capacity of 1401 system to be used to generate object program. See Note 1.
6	One of these: 1-6	Core-storage capacity of 1401 system to be used to execute object program. See Note 1.
7	1 or blank	1 if sense switches B-G are used; blank otherwise.
8	1 or 2	1 for IBM 1403, Model 1 (100 print positions); 2 for IBM 1403, Model 2 (132 print positions).
9	M or blank	M if multiply-divide special feature is installed on 1401 system used to execute object program; blank otherwise.
10-12	Tape input record-length	Length for fixed-length records, or VVV for variable-length records.
13-14	Tape input blocking factor or blank	For fixed-length, blocked input records, the number of data records per input tape block (input blocking-factor); for variable-length input records, blank.
15	1 or blank	1 if header label on tape input is to be bypassed; blank otherwise.
16	2-9 or blank	Number (2-9) of tape data files to be processed; blank if only one.
17-19	Output tape record length or blank	Length of output tape records; blank if 132 characters.
20-75	Blank	(Not used)
76-80	Identification	If program identification is to be punched in the symbolic object-program deck, that identification; blank otherwise.

NOTE 1: The core-storage capacity of a 1401 system is represented by the following numbers:

1 — 1400	4 — 8000
2 — 2000	5 — 12000
3 — 4000	6 — 16000

Figure 42. Format of RPG Control Card

## Control Card

The Report Program Generator requires one control card. The user prepares the card and places it, as explained under *Second Phase of Report Generating*, in the RPG processor deck. Figure 42 shows the control card format.

## Special Feature Specifications

An IBM 1401 Data Processing System equipped with a special feature to read 51-column cards reads either standard 80-column or 51-column cards, depending upon the operating mode. When operating in the 51-column mode, columns 1-51 correspond to columns 15-65 of an 80-column card. Thus, in any specifications of card columns for 51-column input data file cards, the entry is the column number plus 14. For example, if a minor control field in the 51-column cards of the input data file is punched in columns 2-7, the proper entry in columns 44-46 of the input specifications sheet is 021.

## Summary of First Phase of Report Generating

After the information relevant to the desired report has been described on the spacing chart and the four specifications sheets, the user prepares the cards that constitute the source deck for the Report Program Generator. He punches one card for each line-entry on the specifications sheets. The control card, prepared as described, is placed ahead of the specifications cards. The sections of the source deck are now in the order:

1. control card,
2. input specifications cards,
3. data specifications cards,
4. calculation specifications cards, and
5. format specifications cards.

## Second Phase of Report Generating

The second phase is developing the object program in symbolic language. This is done by a machine pass through the IBM 1401 of the specifications source deck and either the Card or the Tape Report Program Generator processor deck. Both the Card and the Tape RPG processor decks contain comments cards in five appropriate places in each deck. They identify where the user must insert both the control card and each group of specifications cards. For example, in the first part of each deck there are two comments cards punched *\*Place Control Card After This Card* and *\*Place Control Card Before This Card*. At four other places in each deck there are two comments cards between which the user places the appropriate specifications cards.

The output of this machine pass is

1. the specifications source deck, which is selected into the 1 stacker pocket,
2. the Card or the Tape RPG processor deck, which falls into the NR stacker pocket, and
3. the object program deck in symbolic language, which falls into three pockets. The first part of the object program falls into the NP stacker pocket, the second part falls into the 4 pocket, and the last part of the program falls into the 8/2 pocket.

Because proper order of the cards in the object program deck must be maintained, these cards should be removed from the stackers and placed face-down in the following order: NP stacker, stacker 4, and stacker 8/2.

## Checking the Specifications

During the generation of the object program certain errors, if present in the specifications source deck, cause the RPG processor to halt. Because a halt necessitates the correction of the error and restart of the generation phase from the beginning, check the specifications carefully for the following errors before beginning generation.

KIND OF SPECIFICATION	REASON FOR PROGRAM HALT
Control Input	Control card missing. Column 1 not C, S, or *. Resulting condition (columns 42-43) blank.
Data	Column 1 not D or *. Field source not defined in the input specifications. Status (column 11, 14, or 17) not B, Z, N, or P.
Calculation	Status (column 11, 14, or 17) not B, Z, N, P, U, L, E, or H. Total/Detail (column 49) not T or D.
Format	First format card not a line specification (column 1 not an L). Column 1 not L, F, B, K, or W. Type of line (column 2) not H, T, or D. Output medium (columns 5-7) not specified. Final (normal) halt.

### Third Phase of Report Generating

The third phase of report generating converts the output of phase two, which is the object program in symbolic language, to an object program in 1401 machine language. Either a Symbolic Programming System deck or a 1401 *Autocoder* system tape can be used to perform the assembly. When using 1401 *Autocoder*, the user must provide the proper control cards to enter the SPS phase.

If the user wishes to incorporate his own symbolic subroutines in the object program, he places them in the proper place in the object deck before phase three begins. See *Inserting Subroutines in Object Program*.

### Program Storage Requirements

After the object program has been assembled in machine language, the user can determine from the assembly listing how much storage the program requires. With regard to programs that process tape input records, the maximum block length that the object program can accommodate is directly related to the size of the program. After assembly, the user can subtract the address associated with the label RILSS1 (located two lines before the END card) from the maximum storage address minus two. For a machine equipped with 4,000 positions of core storage, the maximum storage address minus two is  $3999 - 2$ , or 3997. The result is the amount of available storage for an input tape block.

## Fourth Phase of Report Generating

The fourth (final) phase of report generating is the execution of the object program to produce the desired output. The input is a data file of cards in the 1402 read feed or magnetic tape on tape unit 3. The output produced is the output the user has specified. The following list includes the output that can be produced when processing card-input data files:

- A printed report.
- A report punched into cards.
- A printed report and a report punched into cards.
- A printed report with selected information punched into cards.
- A printed report with summary totals punched into cards.

For processing tape-input data files, the following list describes the output that can be produced:

- A printed report.
- A report punched into cards.
- A report written on magnetic tape mounted on tape unit 4.
- A combination of any or all three listed above.
- A printed report with certain selected information punched into cards and other selected information written on magnetic tape.
- A printed report with summary totals punched into cards and exception records written on magnetic tape.

## Sense Switches

The object program produced by the RPG permits using certain sense switches that have specified functions. For programs that process card files, only sense

switch A (last card) is necessary. Tape object programs require sense switches E, F, and G. Sense switches B, C, and D can be used to govern processing at the user's discretion. Except for switch E in a tape application, the object program tests the switch settings at the beginning of the run and maintains that setting throughout the run. Manually changing a switch other than E during processing is not recognized by the object program. In tape object programs there is a halt after each file has been processed. If the next file is to be processed, sense switch E must be set OFF (down) and the start key must be pressed. If the next file is to be bypassed, sense switch E must be set ON (up) and the start key pressed.

Sense switches F and G are used in tape object programs to control the action of the tape error routine. If an error in reading a tape record occurs, the object program automatically attempts to read the record ten times. If the tape validity error persists, the program tests switch F. If F is ON, the program processes the record, disregarding the validity error. If F is OFF, the program halts. If switch G is ON, the user can press the 1401 start key for nine more read attempts. If G is OFF, the user can attempt to correct the error record manually by setting the IBM 1401 console tape select switch to the D-mode and pressing the start key. The record will be read once more. After scanning storage to locate the erroneous character, the user can attempt to correct the error from the console. When the correction has been made and the tape select switch has been returned to the N-position, processing can be continued by pressing the console start key.

A tape validity error while writing a tape record results in a maximum of fifty attempts to write the record. If the error persists, the program halts. A new tape should be mounted and the program restarted.



49

[illegible]

If a data field has both a status condition and multiple field sources, storage space and execution time are saved by arranging the field sources so that the same sources with the same conditions are together (Figure 47).

### Calculation Specifications

When possible, specify detail-time calculations (those with a D in column 49) consecutively, and total-time calculations (T in column 49) consecutively, to save both storage space and execution time. Also, whenever order permits, calculations governed by the same conditions in columns 40-48 should be specified together.

[illegible]

D	FIELD NAME	FIELD LENGTH UNEDITED	FIELD STATUS																SOURCES OF FIELD																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																
			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	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271	272	273	274	275	276	277	278	279	280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300	301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320	321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337	338	339	340	341	342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360	361	362	363	364	365	366	367	368	369	370	371	372	373	374	375	376	377	378	379	380	381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400	401	402	403	404	405	406	407	408	409	410	411	412	413	414	415	416	417	418	419	420	421	422	423	424	425	426	427	428	429	430	431	432	433	434	435	436	437	438	439	440	441	442	443	444	445	446	447	448	449	450	451	452	453	454	455	456	457	458	459	460	461	462	463	464	465	466	467	468	469	470	471	472	473	474	475	476	477	478	479	480	481	482	483	484	485	486	487	488	489	490	491	492	493	494	495	496	497	498	499	500	501	502	503	504	505	506	507	508	509	510	511	512	513

Data fields should not be blanked after use unless blanking is essential. Unnecessary use of a B instead of an F in column 1 wastes storage capacity.

If the Next-Line specification in columns 8-10 is *applicable*, using it will save both execution time and storage space.

[illegible]

Figure 47. Data Specifications with Same Field Sources Arranged Together

## Description of the General Block Diagrams

### Object Program to Process Card Input Files

The following explanation applies to Figure 48, the block diagram of an object program generated by the Card Report Program Generator. The paragraph numbers correspond to the circled numbers in that figure.

1. One read instruction is generated in the entire object program. This instruction (READ1) is followed by an instruction labeled ENTRY1. This is the first point where a user can insert his own subroutine in the symbolic deck before assembly. Peculiar manipulation of data fields, for example, might be more efficiently done at this time by inserting hand-coded instructions. The instructions to combine the portions of a split control field into one field located at storage location 0090, for instance, could be inserted at ENTRY1. This field could then be specified on the input sheet as one field ending in 090.
2. For each non-sequential record (i.e., a record assigned an alphabetic sequence designator, such as CAA on the input specifications sheet), a test routine exists in the object program. The first such routine is labeled DA0100; the second, DA0200; and so on. As many of these routines are generated in a given object program as there are non-sequential records specified for the input file. The function of each routine is to check the record in the input area for the record codes specified in one entry of the input sheet. If the first routine finds that the codes in the input record are different from the codes for which the routine is checking, it branches to the second routine, where a similar test is made using the next set of record codes specified. This testing process continues until either the input record is found to be one of the non-sequential records specified, or it becomes necessary to check for one of the sequential records specified. If a non-sequential record is present, the routine sets a corresponding resulting-condition indicator ON and modifies SWICH4 to branch to the proper routine that removes the data fields from the record (see block 8 in Figure 48). The resulting-condition indicator remains ON until another input record is read. If testing is to begin for a sequential record, the program enters SWICH1.
3. For each sequential record (i.e., a record assigned a numerical sequence-number on the input sheet), a routine exists to which SWICH1 branches. The first portion of the routine is labeled RNxxxx. This sets SWICH1 to branch to the routine for the next record in sequence. This is followed by instructions labeled DNxxxx, which test for the presence in the input record of the record codes specified for this record on the input sheet. If the record is in its proper place in the numbered sequence within its control group, SWICH4 is modified to branch to the routine to remove the fields of data from this type of record. If, however, the input record is not acceptable to the object program at this point, a check is made to determine if there is another record specified in an *or* relationship. For instance, should two C01s be specified, if the routine for the first one did not find the proper record codes, it would branch to check for the second C01 record. If, however, only one C01 record is specified and that record is not present after a control-field change, the program comes to a halt (ERR5EQ). All other errors in sequence come to the same halt to allow the user to insert a common error routine under the label ERR5EQ, if he desires. In the invoice application the second record in the sequence, C02, is an optional record. The routine for this type of record includes a branch to a routine that properly processes optional record-types not present. This operation is described in paragraph 14.
4. When the object program has determined which record is in the input area, it branches to a routine to test for control-field changes between that record and the previous record. The routine that checks for control-field changes in a non-sequential record is labeled TAxxxx. The routine that checks for control-field changes in sequential records is labeled TNxxxx. When the control fields in all non-sequential records are alike, there will be only one such routine for all non-sequential records that are consecutively specified on the input sheet. The same is true for all sequential records.
5. Calculations specified to be performed at total time are routines labeled TCALxx. Total calculations do not contain any information from the record in the input area. This is because total calculations are performed before information is removed from that record. Note that ENTRY2 precedes the total-calculation routine. This permits the user to insert under this label in the symbolic program any calculation routine that he prefers to hand-code rather than to specify on the calculation specifications sheet.
6. Total calculations are followed by the printing or punching of total lines whose conditions are fulfilled. Numerical-level total lines (total lines in a hierarchy) whose routines are labeled T11Txx precede alphabetic-level total lines (total lines not in a hierarchy) whose routines are labeled TTxxxx. Page overflow is tested during the printing of total

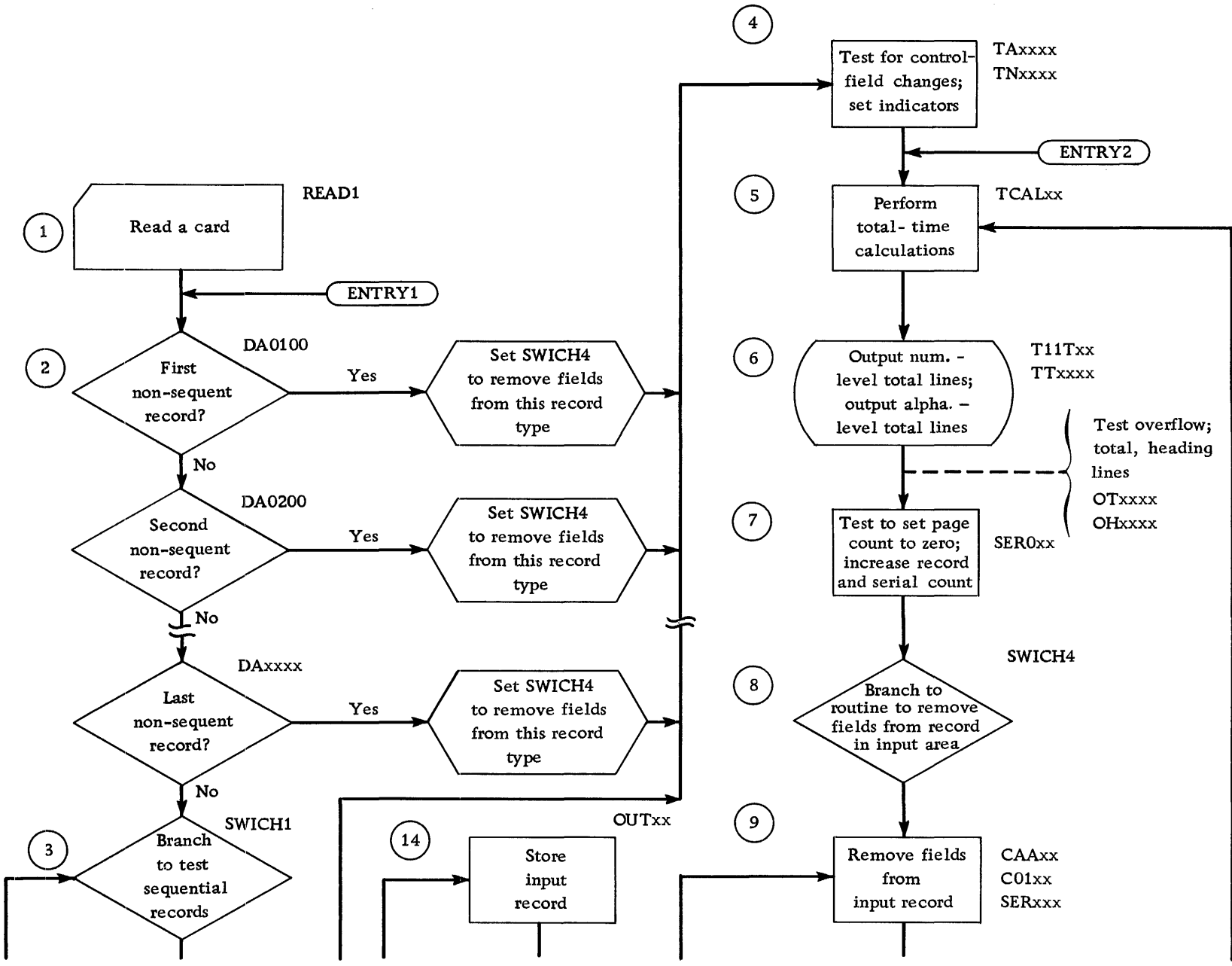
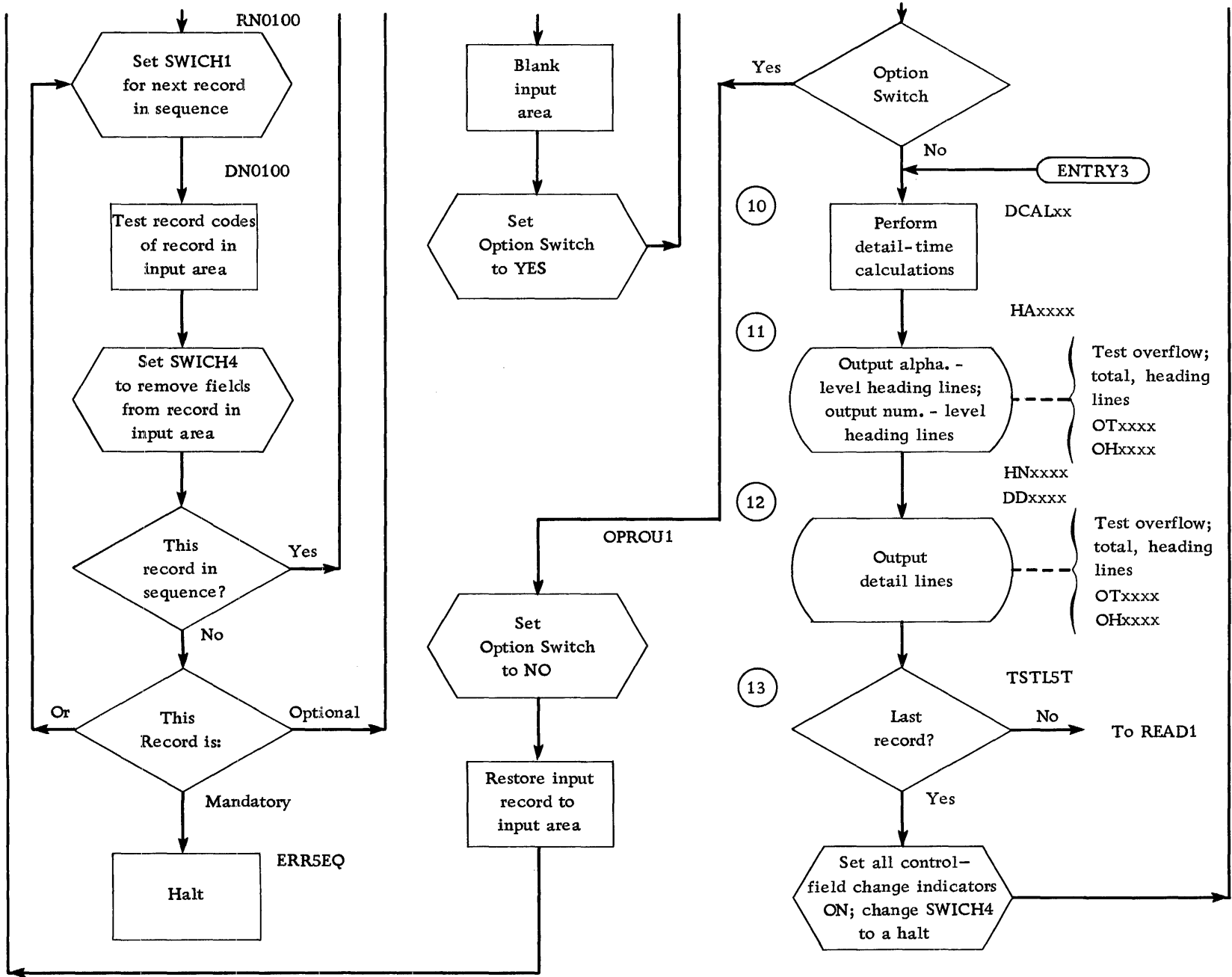


Figure 48. Block Diagram Guide to Object Program Generated by Card RPG

Figure 48. Block Diagram Guide to Object Program Generated by Card RPG. (Continued)



- lines. Overflow total lines (labeled OTxxxx) are printed after normal total lines if an overflow condition exists. Note that all normal total lines whose conditions are fulfilled *precede* overflow total lines.
7. If a page number must be reset on the basis of a control-field change, that operation occurs at this time. Furthermore, record counts and serial numbers are increased at this point if the conditions for increasing them are met. Record counts are increased because of the type of record in the input area. Serial numbers are usually increased because of control-field changes.
  8. SWICH4 is a branch instruction, modified in the routines for both the non-sequential and the sequential records, to branch to the proper routine for processing the record in the input area.
  9. For each record specified on the input specifications sheet there exists a routine whose first label might be CAAxx for a non-sequential record and C01xx for a sequential record. Furthermore, there is a routine labeled SERxxx to process page number, record counts, and serial numbers. When a sequential record is optional, its absence in the sequence will cause a branch to the optional routine labeled OPROU1. Otherwise the program will branch to perform detail calculations.
  10. Just as ENTRY2 provides for a calculation subroutine before total calculations, ENTRY3 provides this facility before detail calculations. Detail calculations are routines labeled DCALxx. Calculations that require entry to multiply or divide subroutines have a linkage to these subroutines, which are incorporated when necessary in the generated calculation routine.
  11. Because heading lines sometimes require information from the record in the input area, they are printed or punched at this point in the object program. The routine for alphabetic heading lines (heading lines not in a hierarchy) is labeled HAXxxx. The routine for numerical heading lines (heading lines in a hierarchy) is labeled HNxxxx. When block 11 is compared with block 6, it is apparent that the order in which heading lines are produced as output is the reverse of that for total lines. As previously stated, when the conditions are fulfilled for heading lines not in a hierarchy, those lines are produced before heading lines in a hierarchy whose conditions are fulfilled. Heading lines in a hierarchy appear in descending level-number order, but in ascending line-number order within each level.
  12. Detail lines whose output conditions are met follow heading lines. Both heading and detail line-output routines include tests for overflow. The overflow-heading-lines routine is labeled OHxxxx.
  13. The test for last-card condition is labeled TSTL5T in the symbolic program. When the result of this test is YES, the object program sets all control-field-change indicators ON, modifies SWICH4 to a halt and branch to READ1. The program branches to perform total calculations, prints or punches total lines, including total lines containing LC output conditions, and halts at SWICH4. When the result of the test for last card is NO, the program branches back to READ1.
  14. For each type of sequential record that is specified as optional (e.g., record C02 for the invoice application), there is an optional routine labeled OUTxx. If, upon checking for the presence of an optional record the object program does not find it, the program branches to OUTxx. This routine stores the record contained in the input area, blanks the input area, and sets the option switch below block 9 to YES. Then the object program branches to block 9, which removes the fields from the optional record-type in the input area just as if it were present. The fields thus removed consist of blanks. In the invoice application the optional routine causes the *Shipped To* information from the previous invoice to be set to blanks in storage when the present invoice has no *Shipped To* information (when record C02 is missing). This operation prevents the printing of the previous customer's *Shipped To* information in the heading of an invoice that has no such information of its own. After removing the blank fields, the object program branches on the YES setting of the option switch to restore to the input area the record saved (record C03) in the optional routine, OUTxx. After setting the option switch to NO, the object program branches back to SWICH1 to test for the record that should be next in the sequence. The next record in the sequence of the invoice application is C03, which should be the record present in the input area.

### Object Program to Process Tape Input Files

The following explanation applies to Figure 49, the block diagram of an object program generated by the Tape Report Program Generator. The paragraph numbers correspond to the circled numbers in that figure.

#### 1. For Blocked Input Tape Records:

The routine to get the next record, labeled NX7-REC, causes either a tape read instruction (READ1) to be executed or the next sequential record in the block to be positioned for processing, depending upon where the next record is located. After each tape read instruction, the test for end-of-file is made. When the last record has been processed, the program branches to the end-of-file routine, described in paragraph 13.

1. For Unblocked Input Tape Records:  
The instruction labeled NX7REC is the tape read instruction (there is no instruction labeled READ1). The end-of-file test is made. If the last record has been processed, the object program branches to the end-of-file routine, described in paragraph 13. If the last record has not been processed, the object program proceeds to block 2. The function, *Position Next Record for Processing*, does not exist for an object program to process unblocked tape records.
- 2-12. Program operation is the same as in Figure 48, except that after block 12 in Figure 49 the object program returns to block 1.
13. After processing the last record of a given file, the object program enters the end-of-file routine (EOF). The program determines whether all input files have been processed. If not, the object program halts to permit the user to mount the next reel if the files are contained on more than one reel. At this point the user can bypass the next file by first setting sense switch E ON and then pressing the 1401 start key. If he wants to process the next file, he leaves sense switch E OFF and presses the 1401 start key. The program branches to NX7REC to resume processing. However, if all files have been processed, the object program sets all control-field-change indicators ON, modifies SWICH4 to a halt, and branches to block 5 to perform total-time calculations (TCALxx).

Figure 49. Block Diagram Guide to Object Program Generated by Tape RPG

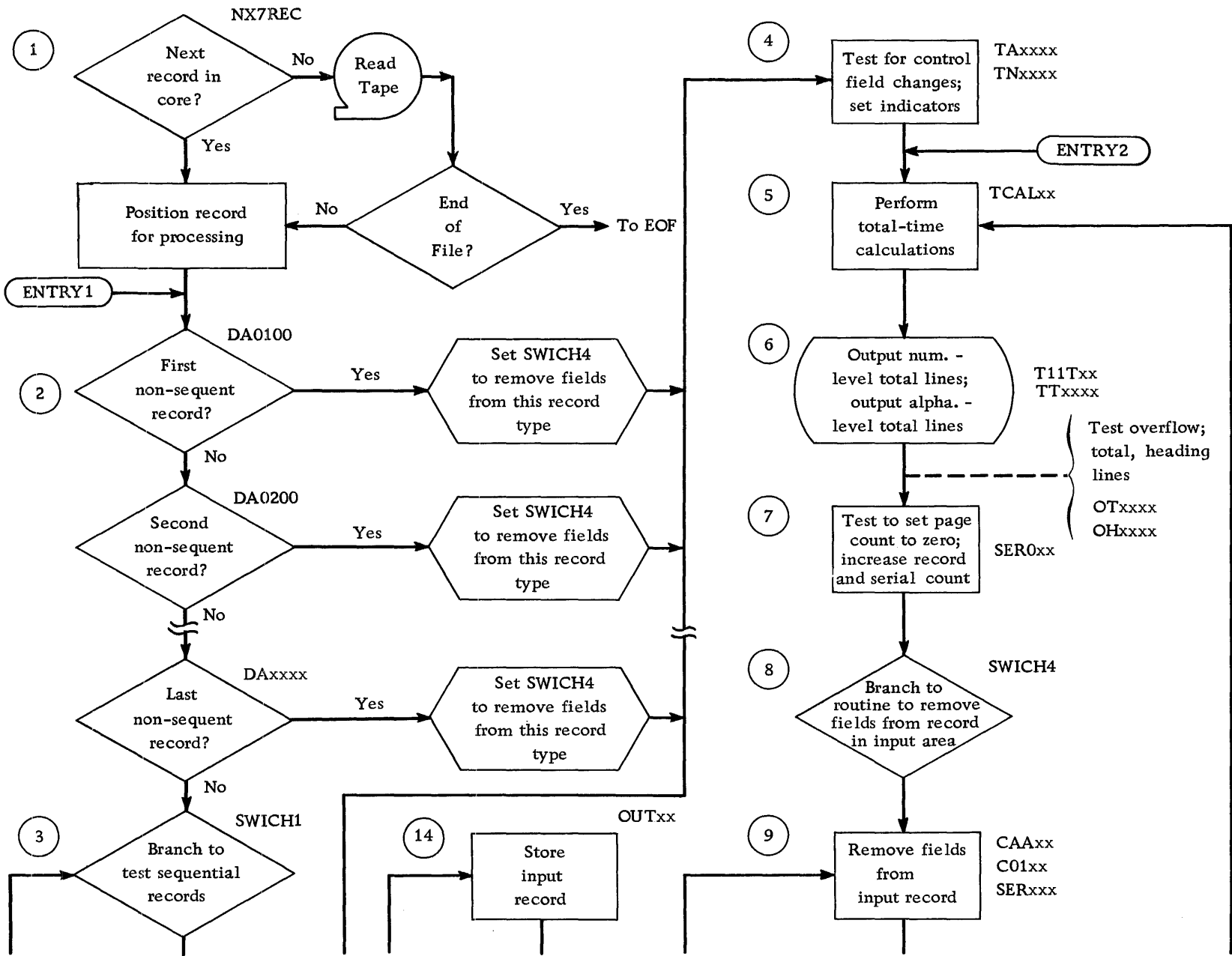
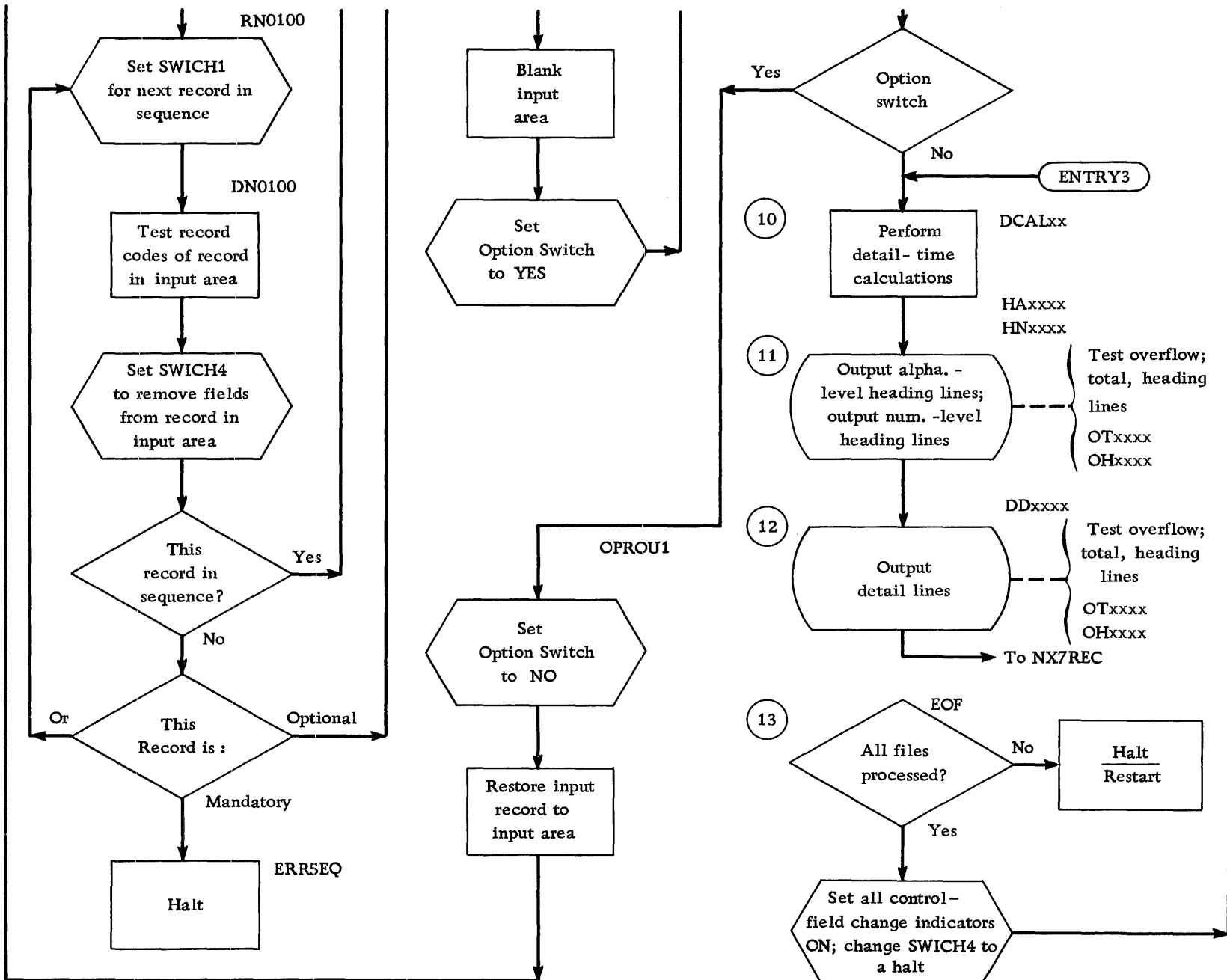


Figure 49. Block Diagram Guide to Object Program Generated by Tape RPG (Continued)



## Inserting Subroutines in Object Program

After the generation of the object program in the language of the IBM 1401 Symbolic Programming System, subroutines written in SPS can be inserted in the object deck before assembly begins. The three insertion points for subroutines are ENTRY1, ENTRY2, and ENTRY3, as indicated in Figures 48 and 49. These entry points appear in the symbolic object program in the form shown in Figure 50.

LABEL	OPERATION	(A) OPERAND			
		ADDRESS	±	CHAR.	ADJ.
8	13 14	16 17	23		
ENTRY1		CONF1			
ENTRY2		BTCL00			
ENTRY3		BTCL00			

Figure 50. Entry Points in Symbolic Form

To insert a subroutine at ENTRY1, the user must change the instruction into two instructions. ENTRY1 should be a branch to the beginning of the subroutine. The instruction to clear a word mark in CONF1 should follow ENTRY1. If the first instruction to be executed in the subroutine had the label SUBROU, the instructions at ENTRY1 would be as shown in Figure 51.

LABEL	OPERATION	(A) OPERAND			
		ADDRESS	±	CHAR.	ADJ.
8	13 14	16 17	23		
ENTRY1		SUBROU			
		CONF1			

Figure 51. Modifying ENTRY1 Instruction

The instruction that exits from the subroutine branches to ENTRY1 + 004. This allows a return to the main line of the program. The subroutine, punched in cards, should be placed immediately in front of the END card in the symbolic object-program deck before assembly (phase three).

A subroutine inserted in the program at ENTRY2 requires simply that the instruction labeled ENTRY2 be changed to branch to the beginning of the subroutine. The exit from the subroutine should be a branch to TCAL00. ENTRY3 requires a similar change with a branch to DCAL00 as the exit from the subroutine.

ENTRY1 allows manipulation of data fields immediately after a record is read. Split control fields can readily be combined at this point. ENTRY2 and ENTRY3 allow for complicated calculations such as square-root functions or a table-searching operation. It is not necessary to provide multiply or divide subroutines; the RPG provides either or both in programs that require them.

### Subroutine Labels

Use caution in assigning labels in subroutines. The labels must be *strictly alphabetic*. Do not duplicate any of the field names assigned on the data and calculation specifications sheets. Because all labels produced by the RPG processor are field names from those sheets or are numerical in at least one position, processing errors can be avoided if this principle is followed.

## Suppressing Total Lines and Calculations

Analysis of the block diagrams (see Figures 48 and 49) shows that total calculations, total lines, and heading lines conditioned by control-field changes will be processed whenever a record causes those changes. Ordinarily such processing results in proper operation, but there are instances when control-field changes alone are not sufficient to govern calculations and line output. For example, at the beginning of a run a record whose control fields are compared with blanks causes changes in control. Although it is usually proper to process heading lines at the beginning of a run, it is not desirable to have total lines produced as output. Further, it may not be desirable to perform total calculations on blank fields. Thus, it is necessary to doubly-condition total-line output and calculations that must not be performed due to a control-field change on the run-in.

The suggested way to suppress these total operations is to govern their performance by the condition that a field is not blank as well as a control-field change. If a blank status is specified for a field that will contain valid information only after the first record has been processed, total operations can be suppressed on the run-in when that field is blank.

Any object program produced by the RPG begins processing by checking the heading lines (block 11 in the diagrams) to determine whether any of them should become output on the condition 1P. The program then continues through the last-record test (block 13) to the instruction that reads a record. This first record causes control-field changes for all the fields specified on the input sheet for this particular record. If the first record is a date header it may not have any control fields, and so no control-field changes can occur. This is true in both the Monthly Expense Distribution Report and the Invoice Report illustrated in this bulletin.

In the Monthly Expense Distribution Report the second record in the input file is an item record having three control fields. All three fields change when that record is compared with blank control fields preceding it in processing. Thus, all total calculations, total lines, and heading lines that are conditioned by only control-field changes will be processed. In the case of heading lines this is proper processing, but no total calculations or total lines are desirable at this point.

Because no fields have been removed from the first item record when total operations occur, the blank status of a field whose source in the data specifications is that record can condition these operations. DEPNUM is blank before the processing of the first item record, and it is a significant field throughout the remainder of the run. Therefore the blank status of this field (condition 03) is used to prevent total lines from printing at the beginning of the run. All total lines are conditioned by their respective field changes and N03 (DEPNUM not blank).

In the invoice application the problem of total suppression is more extensive. The date header at the beginning of the run has no control fields. The second record in the input file is the SOLD TO header, which has only field 2 specified as a control field. Comparison of field 2 in that record with the blank field preceding it in processing produces a control change in field 2 (and incidentally, in field 1). If the total lines on the invoice were conditioned only by control-field changes, those lines would be printed before the heading lines of the first invoice. The total lines are, however, also conditioned by the blank status of the field ITEMNO (condition 07), so that the lines can be printed only when that field is not blank (N07). Because this field is blank until the first item record is processed, no total lines can be printed at the beginning of the run.

Suppressing totals at the beginning of an invoicing run is only part of the problem. For each invoice there is a change in control field 1 whenever an item record appears with a new item number. Between item-number groups this change causes the total line for the last group to print. A problem in total suppression occurs, however, when the first item record of an invoice causes a change in control field 1. If T11 were conditioned only by the field change, a T11 line would appear when the first item record was checked for control changes. To prevent this unacceptable operation, the T11 line is also conditioned by N07 (ITEMNO not blank) and ITEMNO is blanked in the format specifications to ensure that it will be blank at the beginning of a new invoice. ITEMNO is blank, therefore, until after data fields are removed from the first item record of an invoice. These fields are removed (block 9 of the diagrams) after the comparing for control-field changes (block 4) and the processing of total lines (block 6).

---

## Sample Report Documentation

Figures 52 through 59 show the specifications for the Monthly Expense Distribution Report. A few cards of the input file are shown in Figure 3, and the printed

report is shown in Figure 4.

Figures 60 through 65 show the specifications for the Invoice Report. A few cards of the input file are shown in Figure 10, and the report is shown in Figure 9.



INTERNATIONAL BUSINESS MACHINES CORPORATION

### IBM 1403 SPACING CHART

6 Lines per inch

Form X24-6436

PRINTED IN U.S.A.

LINE DESCRIPTION

FIELD HEADINGS/WORD MARKS

6 Lines per inch

GLUE

MONTHLY EXPENSE DISTRIBUTION REPORT

REPORT DATE XX-XX-XX PAGE XXX

OUR INVOICE DATE AMOUNT AMOUNT AMOUNT  
NUMBER MO DAY BY BY BY  
DEPT. NO. XXX GEN. LEDGER NO. XXX SUB. LEDGER NO. XXX

XXXXX XX XX XX,XXX.XX  
XXX,XXX.XX X,XXX,XXX.XX  
XX,XXX,XXX.XX

FINAL AMOUNT XXX,XXX,XXX.XX

DUAL-SPEED CARRIAGE

Punch only one channel per line. Holes in the same channel should be punched at least 8 lines apart.

MEASUREMENTS	FORM WIDTHS	COMMONLY USED FORM DEPTHS
Horizontal spacing 1/10"	Min. 3 1/2"	3 1/2", 3 3/4", 3 1/2", 5 1/2", 6"
Vertical spacing 1/6"	Max. 18 3/4"	7", 7 1/2", 8 1/2", 11"

NOTE: This chart is subject to inaccuracies from variations in humidity. Dimensions of form should be calculated from measurements shown and not scaled from this chart.



## INPUT SPECIFICATIONS

Page 02 of 08

Date 76 77

RECORD SEQUENCE					RECORD CODES																				CONTROL FIELDS																									CARD NUMBER																															
C	SEQ.	NUMBER	OPTION		POSITION		NOT	Z/C	CODE	POSITION		NOT	Z/C	CODE	POSITION		NOT	Z/C	CODE	POSITION		NOT	Z/C	CODE	POSITION		NOT	Z/C	CODE	POSITION		NOT	Z/C	CODE	RESULTING CONDITION		FIELD 1		FIELD 2		FIELD 3		FIELD 4		FIELD 5		FIELD 6																																		
1	2	3	4	5	6	8	9	10	11	12	14	15	16	17	18	20	21	22	23	24	26	27	28	29	30	32	33	34	35	36	38	39	40	41	42	43	44	END	LENGTH	49	51	52	53	54	END	LENGTH	59	61	62	63	64	END	LENGTH	69	71	72	73	74	75	76	77	78	79	80																	
C	B	B			0	8	0	D	4																									0	1	0	3	5	0	3	0	3	2	0	3	0	3	8	0	3																											O	I	0	.	
C	A	A			0	8	0	N	D	4																									0	2																																				O	I	0	.						

# 1401 REPORT PROGRAM GENERATOR

**Report Monthly Expense Distribution Report**

## DATA SPECIFICATIONS

Page 03 of 08  
76 77

Date 76 77

[illegible]

FORM X24-1338  
PRINTED IN U.S.A.

Page 04 of 08  
Date \_\_\_\_\_  
76 77

Programmed by \_\_\_\_\_

[illegible]

# 1401 REPORT PROGRAM GENERATOR

Date \_\_\_\_\_

[illegible]

**IBM**

# 1401 REPORT PROGRAM GENERATOR

FORM X 24-1339  
PRINTED IN U.S.A.

**Report Monthly Expense Distribution Report**

## FORMAT SPECIFICATIONS

Page 06 of 08  
76 77Date 16/11

Programmed by \_\_\_\_\_

[illegible]

**IBM****1401 REPORT PROGRAM GENERATOR**FORM X24-1339  
PRINTED IN U.S.A.Report Monthly Expense Distribution Report

FORMAT SPECIFICATIONS

Page 07 of 08  
76 77

Programmed by \_\_\_\_\_

Date \_\_\_\_\_

F O R M A T	LINE		OUTPUT				NEXT LINE		SPACE		SKIP		LINE OUTPUT CONDITIONS					FIELD NAME	FIELD END	FIELD OUTPUT CONDITIONS					ZERO SUPPRESS	CONSTANT OR EDIT CONTROL WORD							CARD NUMBER																																																																																																																																																																																																																																																																																																																																																																																																																																																																											
	TYPE	LEVEL	NUMBER	PRINT	PUNCH	RESERVED	TYPE	LEVEL	NUMBER	BEFORE	AFTER	BEFORE	AFTER	STAGGER	COND.	N	COND.			N	COND.	COND.	N	COND.		N	COND.	N	COND.	N	COND.	N		COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.	N	COND.

**IBM**

# 1401 REPORT PROGRAM GENERATOR

FORM X24-1339  
PRINTED IN U.S.A.

## Report Monthly Expense Distribution Report

## FORMAT SPECIFICATIONS

Page 08 of 08  
76 77

Date \_\_\_\_\_

[illegible]



INTERNATIONAL BUSINESS MACHINES CORPORATION

### IBM 1403 SPACING CHART

6 Lines per inch

Form 824.6436

PRINTED IN U.S.A.

LINE DESCRIPTION	FIELD HEADINGS/WORD MARKS	6 Lines per inch
GLUE		
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	<p>INVOICE TO</p> <p>SHIPPED TO</p> <p>INVOICE NO.</p> <p>INVOICE PAGE NO.</p> <p>INVOICE DATE</p> <p>CUSTOMER NO.</p> <p>SHIPPED VIA</p> <p>ORDER NO.</p> <p>Salesman</p> <p>ORDER DATE</p> <p>ITEM NO.</p> <p>DESCRIPTION</p> <p>QUANTITY TOTAL</p> <p>UNIT OF MEASURE</p> <p>UNIT PRICE</p> <p>ITEM AMOUNT TOTAL</p> <p>INV TOT</p> <p>2% DISC</p> <p>NET AMT</p> <p>INVOICE AMOUNT TOTAL</p> <p>TAA</p> <p>FIN TOT</p> <p>FINAL TOTAL</p>	

**DUAL-SPEED CARRIAGE**

Punch only one channel per line. Holes in the same channel should be punched at least 8 lines apart.

MEASUREMENTS	FORM WIDTHS	COMMONLY USED FORM DEPTHS
Horizontal spacing 1/10"	Min. 3 1/2"	3 1/8", 3 1/2", 3 3/4", 5 1/2", 6"
Vertical spacing 1/6"	Max. 18 3/4"	7", 7 1/2", 8 1/2", 11"

FORM X24-1336  
PRINTED IN U.S.A.

Page 02 of 06  
76 77Date 7/6/77[illegible]



# 1401 REPORT PROGRAM GENERATOR

FORM X 24-1337  
PRINTED IN U.S.A.

Report Invoice Example

## DATA SPECIFICATIONS

Page 03 of 06  
76 77

Date \_\_\_\_\_

[illegible]

Programmed by \_\_\_\_\_

71

# 1401 REPORT PROGRAM GENERATOR

Report Invoice Example

## FORMAT SPECIFICATIONS

Page 105 of 06  
Date 76 77

Programmed by \_\_\_\_\_

[illegible]

**IBM**®

# 1401 REPORT PROGRAM GENERATOR

FORM X 24-1339  
PRINTED IN U.S.A.

Report Invoice Example

## FORMAT SPECIFICATIONS

Page 06 of 06  
76 77Date 18/11[illegible]

# Index

A (Arithmetic)	29	Format Specifications	10, 12, 13, 34, 50
Addend	30	Forms Control, Spacing and Skipping	36
Alphabetic Level of Output Lines	15, 35, 37, 51, 54	Fourth Phase of Report Generating	47
Alphabetic Record Sequence	16, 51		
ASÖÖ (Add, Subtract, Reset Add, Reset Subtract)	30	General Description, First Phase of Report Generating	10
Augend	30	Generating Report Programs, Machine Requirements for	3
Autocoder for the IBM 1401	3, 46	Group Indication	12, 24
		Group Total	31
Blank Field Status Used to Suppress Total Lines on Run-in	59		
Block Diagrams	52, 56	Half-Adjust	10, 31
Block Length, Tape Input Records	3, 46	Halts (Program) During Processing Source Deck	45
Blocking of Input Tape Records	3, 44, 46, 54, 55	Header Label, Tape Input	3, 44
C - Record Sequence	16	Input Specifications	10, 12, 13, 16, 48
Calculation Specifications	10, 12, 13, 29, 50	Inserting Subroutines in Object Program	58
Calculations, Suppressing	59	Introduction	3
Card Number, Specifications	21, 25, 32, 39	Invoice Report - Calculation Specifications	71
Cards for RPG Specifications	10, 11	Invoice Report - Data Specifications	70
Checking the Specifications	45	Invoice Report Documentation	68
Classification of Lines	15, 35	Invoice Report - Format Specifications	72, 73
Code - Record Codes	20	Invoice Report - Input File	19
Compare Specification	30, 31	Invoice Report - Input Specifications	21, 69
Condition IP (First Page)	38, 59	Invoice Report - Printed Report	18
Conditioning a Calculation	10, 12, 31	Invoice Report - Spacing Chart	68
Conditioning Operation upon Source Field	10, 12, 24		
Conditions	10, 12, 24, 31, 36	Layout of Lines and Fields	12
Constant or Edit Control Word	10, 39	Leaving Field Name Blank	32
Control Card	44	Length of Input Tape Records	3, 19, 44, 46
Control Fields	10, 12, 20, 48	Length of Output Tape Records	3, 44
Conversion of Month Field	25	Length (Unedited) of Data Field	22
Correlating the Report Specifications	12, 13	Length, Control Field	20
Counting Records	23, 24, 25	Length, Source Field	23
		Level of Output Line	15, 35, 37
D (Data)	22	Levels of Control Fields	20
Data Field Name	22	Line (Output)	35
Data Specifications	10, 12, 13, 22, 48	Line Classification	15, 35
Description of the General Block Diagrams	51	Line Level	15, 35, 37
Detail Calculation	31	Line Number	15, 35, 37
Differently-Coded Records Processed Alike	20	Line Output Conditions	10, 12, 36
Dividend	30	Line Type, Level, and Number (Output)	15, 35, 37
Division Specification	30	Line - Identification Code	15, 35
Divisor	30	Literal	30, 32, 39
Documentation, Sample Report	59		
		Machine Requirements for Executing Generated Programs	3
Edit Control Word	10, 39	Machine Requirements for Generating Report Programs	3
Editing	39	Magnetic - Tape Output, Medium	35
End, Control Field	20	Minuend	30
End, Source Field	23	Month - Field Conversion	25
Executing Generated Report Programs, Machine Requirements for	3	Monthly Expense Distribution Report - Calculation Specifications	33, 63
		Monthly Expense Distribution Report - Data Specifications	28, 62
Factor 1	30	Monthly Expense Distribution Report Documentation	60
Factor 2	30	Monthly Expense Distribution Report - Format Specifications	64 - 67
Field End, Output	38	Monthly Expense Distribution Report - Input File	6, 27
Field End, Source	23	Monthly Expense Distribution Report - Input Specifications	61
Field Length	20, 22, 23, 29, 30, 33, 39	Monthly Expense Distribution Report - Printed Report	7
Field Length Specification for Multiplication and Division	33	Monthly Expense Distribution Report - Spacing Chart	14, 27, 60
Field Length Unedited	22, 29, 30, 33	Multiple Input Records in a Control Group	19
Field Name	22, 29, 32, 38	Multiplicand	30
Field Output Conditions	10, 12, 38	Multiplication Specification	30
Field Source, Data	10, 12, 23	Multiplier	30
Field Status	10, 12, 24, 29		
Field 1 through Field 6 (Control Fields)	20		
First Phase of Report Generating	10		
Format	35, 38		

Negation Condition, Input Record Code	20	Sense Switches	47
Next Line	12, 35, 36, 37	Seq. (Sequential) - Record Sequence	16
Non - Sequential Input - Record Types	16	Sequence Characters, Input Record	16
Not - Record Code	20	Sequence Checking	16, 48, 51
Number of Output Line	15, 35, 37	Sequence Control	21
Number, Record Sequence	16	Serial Numbers, Report	25
Numeric	23	Serial, Record, and Page Numbers	25
Numerical Level of Output Lines	15, 35, 37, 51, 54	Single Input Records in a Control Group	19
Numerical Record Sequence	16, 51	Skip (Forms Control)	36
Object Program to Process Card Input Files - Block		Source Deck for RPG, Specifications	3, 10
Diagram Description	51	Sources of Field	10, 12, 23
Object Program to Process Tape Input Files - Block		Space (Forms Control)	36
Diagram Description	54	Spacing Chart, IBM 1403	10, 12, 13
OP (Operation) + - X / C	30	Special Feature Specifications	44
Operation (Upon Source Field)	10, 23	Split Control - Fields in Input Records	48, 51, 58
Option - Record Sequence	19	SPS, IBM 1401	3, 46, 58
Optional Input Records	19, 54	Stacker	36
Output Conditions	10, 12, 36, 38	Status, Field	24, 29
Output Medium	10, 35, 47	Subroutine Labels	58
Overflow Printing	15, 35 - 38, 54	Subroutines, Multiply and Divide	29, 54, 58
Page Number, Report	23, 24, 25	Subtrahend	30
Page Number, Specifications	21, 25, 32, 39	Suggestions and Recommendations	48
Position - Adjust	10, 31	Summary of Calculation Specifications	33
Position - Record Code	20	Summary of Data Specifications	26
Printed Reports, Examples	7, 18	Summary of First Phase of Report Generating	44
Printed - Output Medium	35, 47	Summary of Format Specifications	43
Program Efficiency	48	Summary of Input Specifications	21
Program Storage Requirements	46	Suppressing Total Lines and Calculations	59
Progressive Totals	31	Symbolic Programming System, IBM 1401	3, 46, 58
Punched - Output Medium	35, 47	Tape File	3, 47
Record Blocking, Input Tape	3, 44, 46, 54, 55	Tape - Output Medium	35, 47
Record Codes	10, 12, 19, 51	Third Phase of Report Generating	46
Record Length of Input Tape Records	3, 19, 44, 46	Total Calculation	31
Record Length of Output Tape Records	3, 44	Total Lines and Calculations, Suppressing	59
Record Sequence	16	Total Operations Suppressed on the Run - in	59
Record - Count Numbers	25	Total/Detail (Calculations)	31
Report Generating, First Phase	10	Type of Output Line	15, 35
Requests for Card or Tape RPG Deck and Write-Up	3	Variable Line - Output Conditions	38
Required Input Records in a Control Group	19	W - Entry on Format Sheet	30, 38
Resulting Condition	12, 20, 24, 30	WORDxx	38, 39, 42
Sample Report Documentation	59	Z/D/C - Record Code	20
SCFx	21	Zero Suppress	39
Second Phase of Report Generating	45	51 - Column Cards (Special Feature Specifications)	44



**International Business Machines Corporation**  
**Data Processing Division**  
**112 East Post Road, White Plains, N. Y. 10601**  
**[USA Only]**

**IBM World Trade Corporation**  
**821 United Nations Plaza, New York, New York 10017**  
**[International]**