File Number 1401/1460-28
Form C24-3166-1

IBM Systems Reference Library

Basic 4K Report Program Generator Specifications
IBM 1401
IBM 1460

The IBM 1401 and 1460 Basic 4K Report Program Generator, with *load-and-go* capability, produces programs that write reports of variable format. This publication explains the writing of report specifications and the preparation of source decks, to produce object programs.

The language used for the report specifications is problem-oriented rather than machine-oriented. Therefore, little knowledge of machine-language coding is required.

Preface

Speed in preparing reports is achieved not only by rapid processing of *input data* but also by rapid preparation of *programs* to produce reports. Easier and faster preparation of report programs is the purpose of the IBM 1401 and 1460 Basic 4K Report Program Generator.

The Basic Report Program Generator (BRPG) produces report programs with a minimum of time and effort. Instead of writing a specific program for a report, the user states his problem and solution (the report specifications) in the BRPG language. BRPG processes the specifications, generating a program to write the reports. By relieving the user of the machine coding and most of the program testing, BRPG permits him to concentrate his efforts on the best solution to his problem. Thus, the Basic Report Program Generator is essentially *problem-oriented* rather than *machine-oriented.*

The programs produced by BRPG write reports in varying formats, using source data contained in card files. Output from programs produced by BRPG is prepared in any of these forms:

- Printed report
- Punched cards
- Printed report and punched cards

The reports produced by programs generated by BRPG range from a simple listing of items from the input file to complex reports that incorporate editing and calculations upon the input data. Included are such functions as printing various kinds of lines (heading lines, detail lines, total lines controlled by control-field changes, and offset total lines), crossfooting, and summary punching. Along with the report, exception records can be produced.

Machine Requirements

Object programs can be produced and executed on an IBM 1401 system equipped with a minimum of:

4,000 positions of core storage
One IBM 1402 Card Read-Punch
One IBM 1403 Printer.

Although the High-Low-Equal Compare special feature is not required, BRPG can use it to advantage if it is installed on the 1401 system.

Object programs can be produced and executed on an IBM 1460 system equipped with a minimum of:

8,000 positions of core storage
One IBM 1402 Card Read-Punch
One IBM 1403 Printer.

This is a reprint of an earlier edition and incorporates the following Technical Newsletters:

Form No.	Pages	Dated
N24-0277	23	10/29/64
N21-0029	29	2/15/65

Copies of this and other IBM publications can be obtained through IBM Branch Offices. A form is included at the back of this manual for readers' comments. If this form has been removed, address comments to: IBM Corporation, Product Publications, Dept. 245, Rochester, Minn. 55901.

Contents

Basic 4K Report Program Generator Specifications
IBM 1401
IBM 1460

The IBM 1401 and 1460 Basic 4K Report Program Generator (BRPG) produces programs that write reports of variable format from card input files. Instead of writing a specific program for each report, the user writes a set of specifications (Figure 1). He punches these into cards and prepares the necessary control card. All of these cards make up the source deck for BRPG. The user then places the source deck and the BRPG processor deck in the card reader of the IBM 1401 or 1460.

The next step is determined by the user's choice of one of two options. The two options are:

1. Load-and-go.
2. Load-and-go, with the machine-language program deck punched out for future use.

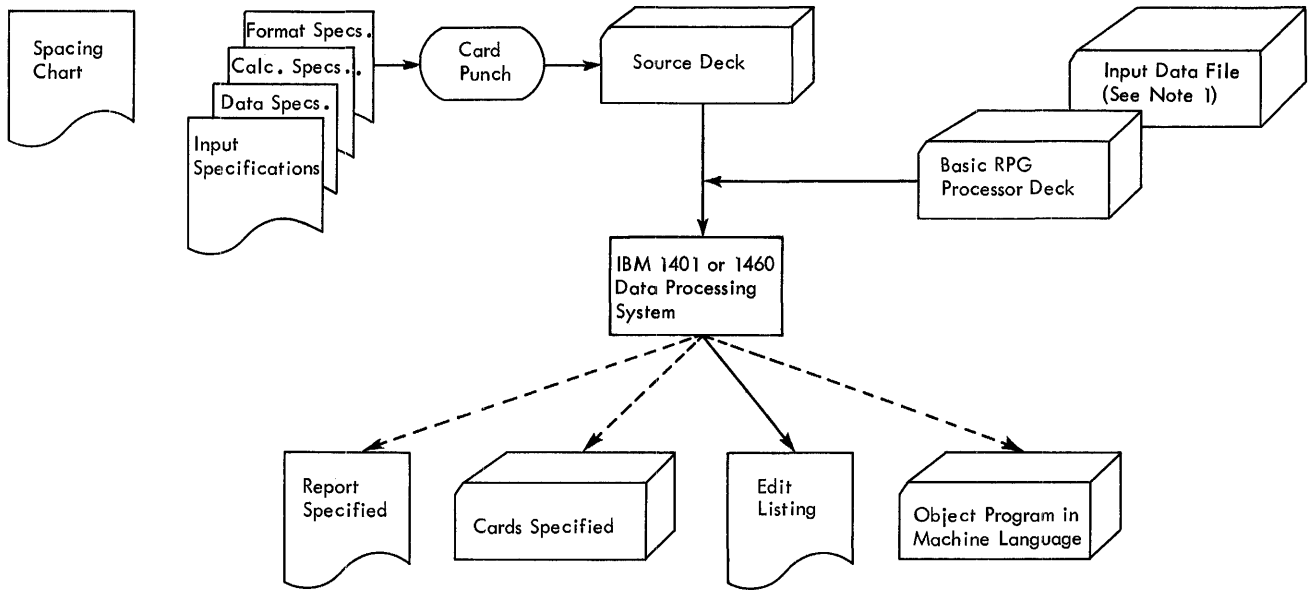
If the user is prepared to run the report program but does not want to keep the program for future reports, he chooses the first option. He places his card input file in the card reader and starts operating the IBM 1401 or 1460 system. BRPG prints an *edit listing* and generates the report program. The edit listing is a printed record of the source deck. Certain kinds of errors (for example, unacceptable entries in the report specifications) will produce error messages.

The report program is a machine-language program, located in core storage. The program is ready to be run, using the card input-data file supplied. The output that the program produces is in printed form, punched-card form, or both.

If the user chooses to generate a report program to produce a report *now* and to retain the program for *future* reports, he selects the second option. He places a deck of blank cards in the punch feed and his card input file in the card reader and starts the IBM 1401 or 1460. BRPG prints the edit listing, generates the report program, and punches that program into cards. The program, which remains in core storage after the punch-out operation, is ready to be run, using the card input file supplied. The output, as in the other option, can be in printed form, in punched-card form, or in both forms.

Input File

The file that a program generated by BRPG can process must be contained in cards. A card file consists of all the cards in the deck. The order in which records are processed is determined by the card order. Processing the last card indicates the end of the file.



Note 1: When a punched object-program deck is specified (to be used to produce this report in future), place blank cards in punch feed.

Figure 1. Producing a Report Using the Basic Report Program Generator

Control Card

The Basic 4K Report Program Generator: IBM 1401 and 1460 requires one control card from which it extracts certain information. In it, the user punches information that identifies the control card, the core capacity of the system, and the number of print positions on the IBM 1403 Printer. The format and contents of the control card are:

Columns	*Contents*	*Explanation*
1-3	CTL	Identifies the control card
4-5	04	Always 04, because neither the BRPG processor nor a generated object program will use more than 4,000 positions of core storage.
6-8	100 or 132	Specifies the number of print positions of the IBM 1403 Printer attached to the IBM 1401 or 1460.
9-12	1401 or 1460	Specifies the system to be used to generate and execute the object program.

Writing Report Specifications

General Description

To generate an object program, the IBM 1401 and 1460 Basic Report Program Generator 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 Figure 1, four forms are required for writing the report specifications. The forms contain answers to the preceding questions. The information is punched into cards with one card punched for each line. These cards comprise the specifications source deck for the BRPG 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 Printer 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 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 Printer 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 X24-6590

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 X24-6591

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 X24-6592

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 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 X24-6593

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) 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 2. 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 (for example, 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 determine many conditions for processing the data to be extracted from the input file.

The fields named on the data specification 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. 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 or punching 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. In the sections that follow, each of the forms will be examined in greater detail. The rules and conventions governing report specifications are presented.

IBM 1403 Printer Spacing Chart

The purposes of laying out the report on the spacing chart are:

1. to establish the positions at which the various data will be printed or punched, 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 par-

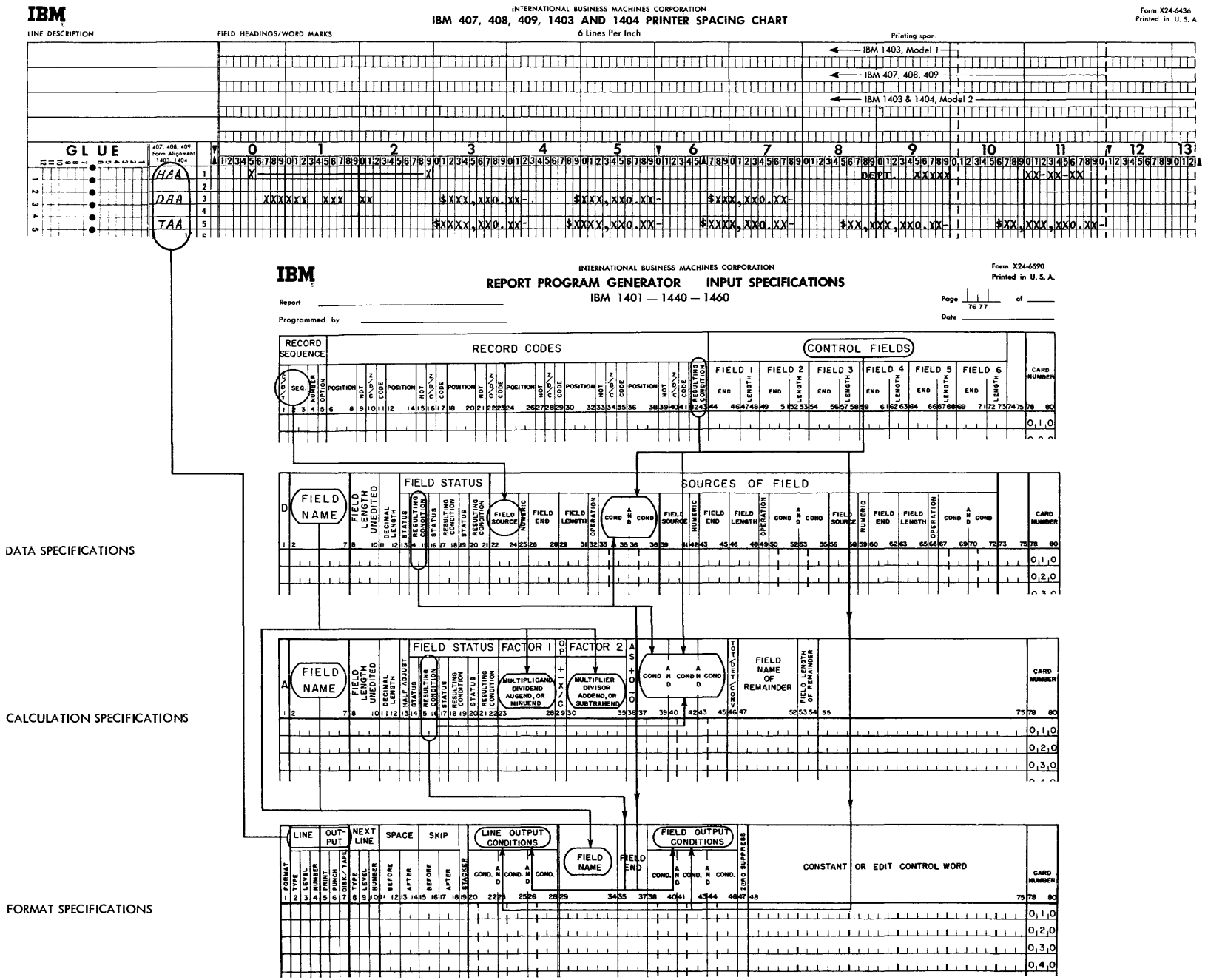


Figure 2. Specifications Correlation Chart

ticular field and makes his notation in the selected positions. In the sample layout (Figure 3) 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.

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.

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.

Classification of Lines

Two methods of classifying lines may be used. Because it does not require a consideration of established order or rank, *alphabetic-level* classification is a quick and efficient method of assigning an identification code to each line. See *Numeric-Level Classification* for a detailed explanation of the hierarchical relationship between line-levels. However, all examples in *Format Specifications* will be given as alphabetic classification.

Alphabetic-Level Classification

Assign letters to heading lines, detail lines, and total lines as shown in Figure 3. The first heading line in Figure 3 is assigned HAA. The second heading line is assigned HBB; the third, HCC, etc. The first detail line is assigned DAA. The first total line is assigned TAA. The second total line is assigned TBB; the third, TCC, etc.

For convenience, these lines are assigned pairs of letters, but if printing occurs on a large number of lines, the lines may be classified as HAA, HAB, HAC, etc.

Input Specifications

This form (Figure 4) 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 (that is, 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. See *Sequence Control* under *Input Specifications* for a discussion of the SCFx entry.

Columns 2-3 specify two numeric or two alphabetic *sequence characters.* 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 numeric 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 sequential records *column 4* indicates the number (either a 1 or N) of that type of record in a control group in the input data file. If there is only one record of a type per group, enter a 1 in column 4. If there is more than one record of a type, enter an N in column 4.

Column 4 can be left blank for non-sequential records.

Column 5 must contain the letter X if the presence of a sequential record in the input data file is optional. If a record type is *required* for proper processing, leave *column 5* blank.

Record Codes

An input record can be identified by any number of record codes. All record codes specified for a single record type are considered in an *and* relation. That is, all the codes must be present in the record. Columns 6-41 of the input specifications sheet provide space on one line-entry to specify as many as six record codes per type of input record. It is possible, however, to specify more than six record codes per type by using more than one line-entry:

- 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.

Columns 6-8 state the position number (input card column) that contains the record-code character.

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 10 must have a Z, D, or C, depending upon whether the record-code comparison to be made is that of the zone portion only, the digit portion only, or the full character.

Figure 3. Spacing Chart for Monthly Expense Distribution Report

IBM

INTERNATIONAL BUSINESS MACHINES CORPORATION

IBM 407, 408, 409, 1403 AND 1404 PRINTER SPACING CHART

Form X24-6436
Printed in U. S. A.

LINE DESCRIPTION | FIELD HEADINGS/WORD MARKS | 6 Lines Per Inch | Printing span: IBM 1403, Model 1; IBM 407, 408, 409; IBM 1403 & 1404, Model 2

GLUE

407, 408, 409 Form Alignment 1403, 1404

HAA
HAB 1"
HAC
HAD 2"
HAE
HAF
HBA
HCA 3"
DAA
TBB 4"
TAA 5"
6"
7"
8"

MONTHLY EXPENSE DISTRIBUTION REPORT

REPORT DATE XX-XX-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

XXXXX XX XX XX,XXO.XX

XXX,XXO.XX * X,XXX,XXO.XX XX,XXX,XXO.XX

FINAL AMOUNT XXX,XXX,XXO.XX

DUAL-SPEED CARRIAGE (IBM 1403, 1404)
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" Vertical spacing 1/6"	Min. 3½" Max. 18¾"	3½", 3½", 3⅔", 5½", 6" 7", 7⅓", 8½", 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.

620513MSP

Fold to Here

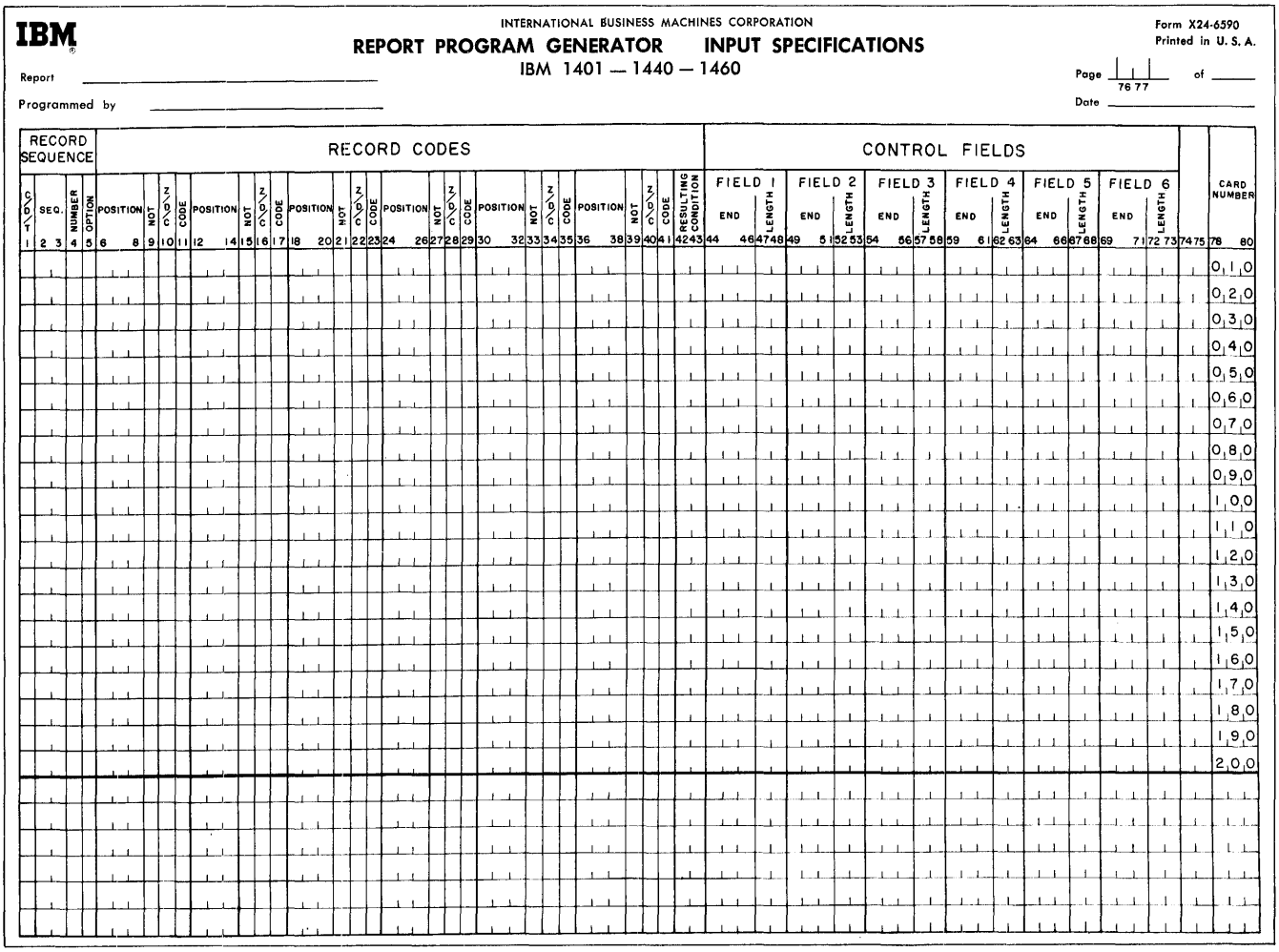


Figure 4. Input Specifications Sheet

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.

Columns 12-41 are used for five additional codes in the same manner.

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, in the range 01-99, arbitrarily chosen by the user to represent a record coded according to his specifications.

Figure 5 shows an input file, and Figure 6 shows the input specifications required by BRPG.

In some applications there is more than one kind of record of a given type. For instance, suppose that in an invoice application the file of input data cards contains 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.

For other reports, the 1-, 2-, and 3-punches might be significant, but in preparing invoices all three kinds of detail cards should be treated as one type. Therefore, they would receive the same sequence designation. Furthermore, suppose that in the input-data file it were possible that for a given customer there may be any number of detail cards of each kind, including 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 7).

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 numeric or alphameric. 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.

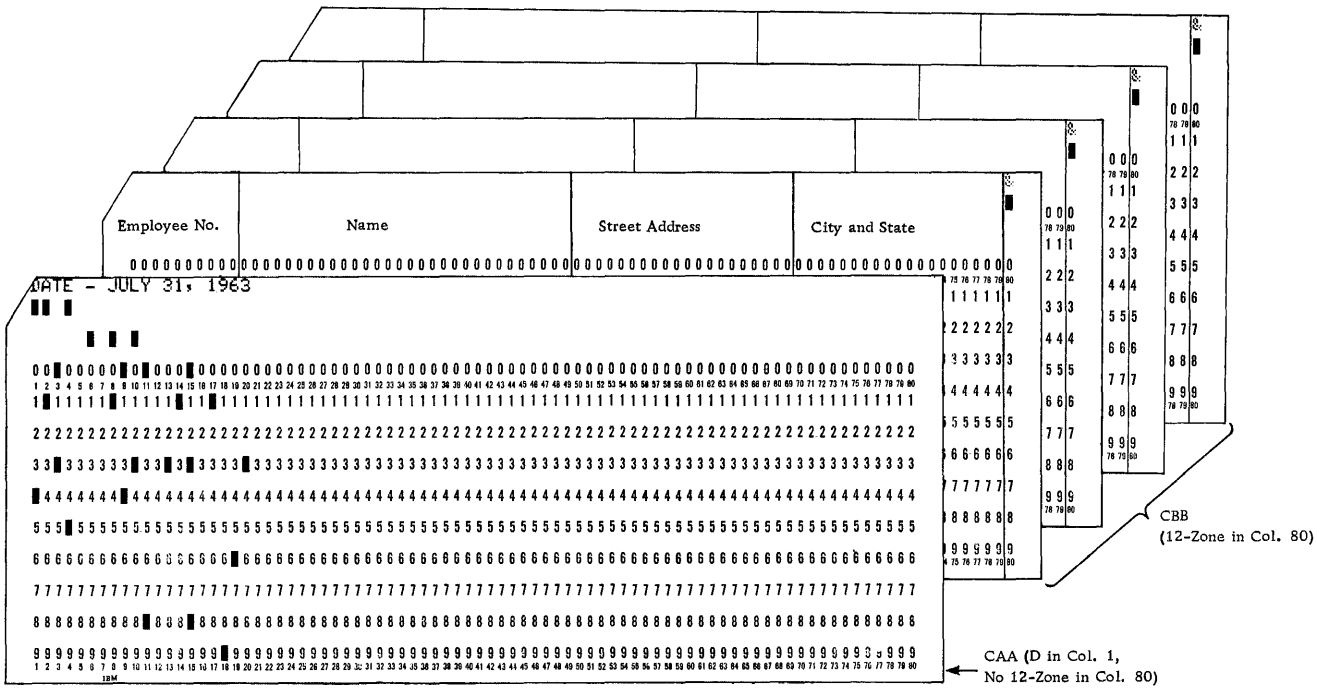


Figure 5. Sample *Name and Address* File

Control Fields

Control fields identify the classifications to which the record belongs. *Columns 44-73* of the input specifications sheet list the control fields present in each type of input data record. Each field is described by its rightmost position in the input record and its length. The fields are specified in ascending sequence of control levels beginning with the entries for the most minor level in Field 1 (columns 44-48) and continuing to the right as far as necessary. Six is the maximum number of control fields that can be specified.

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 control level on the input specifications sheet.

If there are more than six *record codes* in a record type, the *control-field entries* are made only on the last line-entry for that record type.

Sequence Control

A special entry applies when the records of an input file are required to be in a *fixed sequence* in a control group. This is the SCFx entry in columns 1-4, which is used for numerically-designated types of records. The x is the number of the control field governing the sequence of numerically designated types of records.

For example, in a payroll application, suppose that one of each of these records is required for each employee: *Name and Address* card (defined as C01) and an *Hours Worked* (C02) card. The sequence of these two kinds of cards is fixed: The *C01* card must precede the *C02* card for every employee. *Employee Number* is defined as control field 1. Then, the last entry on the input specifications sheet is *SCF1* in columns 1-4. BRPG thus provides for checking the sequence of these two cards for every employee's cards in the file. (See Figure 7.) Every application that has sequential-record specifications must have an SCFx entry at the end of the input specifications.

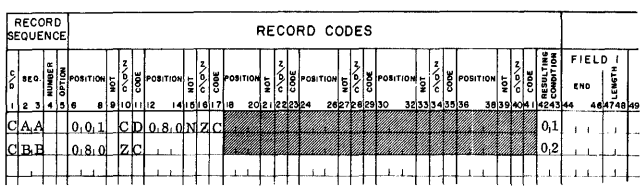


Figure 6. Input Specifications for *Name and Address* File

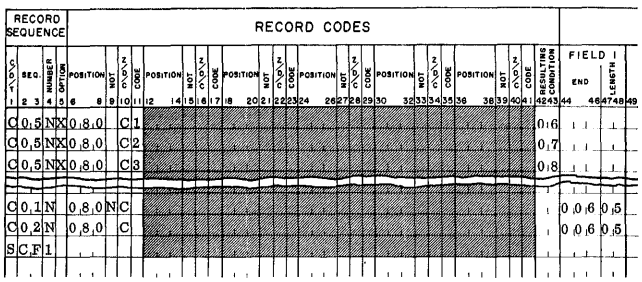


Figure 7. Three Kinds of Input Defined as One Record-Sequence Type, and Fixed-Sequence Input Records

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 prenumbered statements.

Order of Input Specifications

The input specifications must be arranged in this order:

1. Alphabetically designated records (CAA, CBB, etc.).
2. Numerically designated records (C01, C02, etc.).
   When assigning *numeric* sequence designations, be sure to begin with *C01*.
3. SCFx entry, when applicable.

Data Specifications

On this form (Figure 8) the user describes the fields that appear in the output and those used in processing. The following paragraphs describe the entries on the data sheet. The proper *order* of writing the line-entries is given at the end of this section (under *Order of Data Specifications).*

IBM

INTERNATIONAL BUSINESS MACHINES CORPORATION

REPORT PROGRAM GENERATOR DATA SPECIFICATIONS

IBM 1401 — 1440 — 1460

Form X24-6591 Printed in U. S. A.

Report _____ Programmed by _____

Page 76 77 of _____ Date _____

D	FIELD NAME	FIELD LENGTH UNEDITED	DECIMAL LENGTH	FIELD STATUS: STATUS	RESULTING CONDITION	STATUS	RESULTING CONDITION	STATUS	RESULTING CONDITION	SOURCES OF FIELD: FIELD SOURCE	NUMERIC	FIELD END	FIELD LENGTH	OPERATION	COND	AND	COND	FIELD SOURCE	NUMERIC	FIELD END	FIELD LENGTH	OPERATION	COND	AND	COND	FIELD SOURCE	NUMERIC	FIELD END	FIELD LENGTH	OPERATION	COND	AND	COND		CARD NUMBER
1	2 7	8 10	11 12	13	14 15	16	17 18	19	20 21	22 24	25	26 28	29 31	32	33 35	36	36	39 41	42	43 45	46 48	49	50 52	53 55	56 58	59	60 62	63 65	66	67 69	70 72	73 75	78 80
																																		010
																																		012
																																		030
																																		040
																																		050
																																		060
																																		070
																																		080
																																		090
																																		100
																																		110
																																		120
																																		130
																																		140
																																		150
																																		160
																																		170
																																		180
																																		190
																																		200

Figure 8. Data Specifications Sheet

D (Data)

Column 1 must contain a D for every line-entry on the sheet. The D identifies each entry as a data specification.

Field Name

Columns 2-7 must contain a unique name for each of the data fields necessary to process the report. Numbers and special characters cannot be used in a field name. All names must be left-justified.

Note: Any number of line-entries for the same data field (same *Field Name*) is permitted. See *Order of Data Specifications.*

Field Length Unedited

Columns 8-10 contain the unedited field length of the report field. Unedited length means the length not counting punctuation supplied by IBM 1401 and 1460 program editing. For example, if a date for a report were contained in a card as Jul 25,63 and if the desired *form* of the printed date field were the same, the unedited field-length entry for DATE would be 009. If the date were contained in a card as *Jul2563,* and the date field desired in the report were *Jul.25,63* with the period and comma supplied by 1401 and 1460 program editing, then the entry for unedited field-length would be 007.

Decimal Length

Enter in *columns 11-12* the number of positions of the data field that is a *decimal fraction.* (The maximum permitted is 09.) For example, suppose that a data field were of the form xxx.xx. The unedited length would be 5 positions. Of these 5 positions, 2 positions would belong to the right of the decimal point. Hence the entry in columns 11-12 for this data field would be 02. For a numeric data field that contains no decimals, enter 00. For alphabetic or alphameric data fields that contain no decimals, leave these columns blank.

Sources of Field

Columns 22-24 define the source of the field. The entries are Cxx for input record sources, as defined in columns 1-3 of the input specifications sheet.

Column 25 contains an N if the field is a numeric 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 25 is an M. This identifies the entry as a month-field conversion specification. (See *Conversion of Month Field* under *Data Specifications* for an explanation and example of this entry.) Otherwise leave this column blank.

Columns 26-28 contain the location of the low-order position of the field in the source record.

Columns 29-31 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, suppose the unedited field length for a data field were 020. If the *source* field length were the same, columns 29-31 would be left blank.

One of the following seven operation characters must be specified in *column 32:*

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.

$\overset{+}{0}$ — Reset the data field to zero before adding the source field to it. (Punch 12, 0 for $\overset{+}{0}$).

$\bar{0}$ — Reset the data field to zero before subtracting the source field from it. (Punch 11, 0 for $\bar{0}$.)

D — Move the numeric 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 numeric portions are undisturbed in both fields.

Figure 9 illustrates the use of the single-character move-numeric and move-zone operations. Each line-entry is explained as follows:

1. The first line-entry moves the six-character field at position 074 in input record CAA to the data field, AMOUNT.
2. The second line moves the zone portion of the single-character source field in column 69 of the same record 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.
3. The third line moves the numeric portion of the one-column source field in column 35 of card type C01 into the one-position data field, FLDS. By using an entry such as this, a single-column source

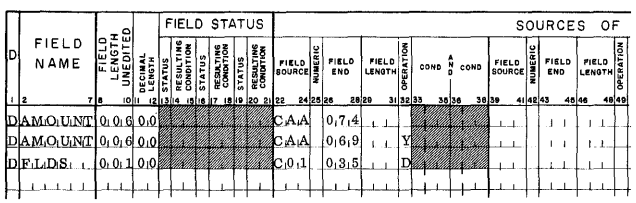


Figure 9. Single Character Move-Numeric and Move-Zone Operations

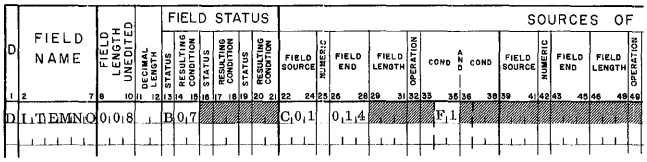


Figure 10. Conditioning an Operation on a Field Source

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 in conjunction with other operations, they should follow the other operations. This ensures the proper sequence of operation in the object program. For example, in the first two line-entries in Figure 9, the Y-entry is made after the blank (move) operation entry.

The same principle of ordering source entries on the data sheet applies in some arithmetic specifications. This is because the object program performs the operations specified in the order of entry on the sheet.

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 should be the one that is reset-added or reset-subtracted.

Columns 33-35 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), resulting conditions specified by other entries on the data sheet, or the negation of any of these when preceded by an N.

In Figure 10, Item Number is to be obtained from the first card of each minor control group. Hence the specification for ITEMNO is condition F1 in columns 34-35. 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.

Columns 36-38 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 32 is performed upon the source field.

Columns 39-72 must be left blank.

Note: The Basic Report Program Generator accepts one entry pertaining to field sources (columns 22-38) per line on the data sheet. If a data field has more than one source, do not use columns 39-72. Instead, use more than one line-entry to specify multiple sources for a data field.

Field Status

The user can govern subsequent processing according to the status (positive, negative, zero, or blank) of a particular field. Columns 13-21 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 of processing, leave columns 13-21 blank.

Column 13 is used for the first status character. The acceptable characters and their meanings are:

B — Blank
Z — ±Zero, Zero, or Blank
N — Negative, including minus zero
P — Positive, including plus zero

Note: Minus zero can occur during processing on the IBM 1401 or 1460 Data Processing System.

Columns 14-15 are for the assignment of a resulting condition. This is a two-digit number identifying the field status specified in column 13. The number entered here can be any one in the range 01 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 refers to the condition of the data field after the operation 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 13-15, 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.

Refer to Figure 10. If after moving the source field from C01 to the ITEMNO field that data field is blank, the resulting condition 07 has been met. This condition can be used in governing total printing when the input file is first read in.

Columns 16-21, 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

Card number *(columns 78-80)* aids identification of line-entries. It helps to maintain the required order of the 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 11 shows the specification for month-field conversion. Columns 8-10 contain 002, the data field-length after conversion. The M in column 25 identifies a month-field conversion specification. Columns 29, 30, and 31 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.

Order of Data Specifications

As previously indicated, a line-entry on this sheet must specify *one* operation upon *one* field source. Data specifications must be arranged in this order:

1. All those pertaining to data fields from the first record type defined on the input specifications sheet.
2. All those pertaining to data fields from the second record type defined on the input sheet.
3. And so on, maintaining the same *Field-Source* order (columns 22-24 on the data sheet) as was established on the *input specifications* sheet in columns 1-3 (Figure 12).

Note: If processing of a particular record type is not required but if the object program must check the sequence of each type of record, then do not make any *data*-specification entries for that record. For example, suppose that in Figure 12 it were not necessary to process record type *C01*, but suppose it were necessary to check the sequence of the C01 and C02 records. Then, the input specifications would be as shown in Figure 12. However, there would be no entries on the data sheet for the type *C01* record.

Calculation Specifications

This form (Figure 13) 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. The following paragraphs explain the entries on the calculation specifications sheet.

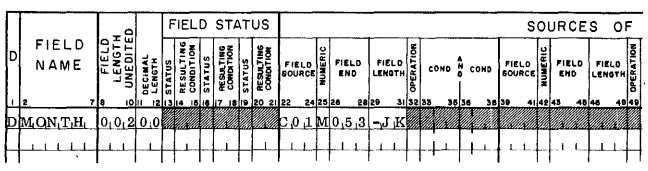


Figure 11. Month Field Conversion

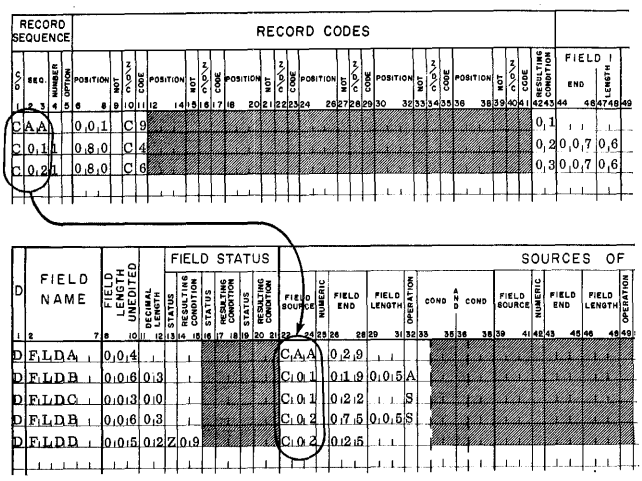


Figure 12. Order of Data Specifications

A (Arithmetic)

In *column 1* an A identifies the entry as a calculation 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.

Note: Any number of line-entries for the same field (same *Field Name)* is permitted. Write the line-entries in the order in which the calculations should occur.

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.

Decimal Length

Specify in *columns 11-12* the number of digits in the decimal portion of the field represented by Field Name. (The maximum permitted is 09.) For example, when a field should contain three digits to the right of the decimal point, specify *03* in columns 11-12. For further explanation of decimal length, see *Decimal Alignment* under *Calculation Specifications*. For a

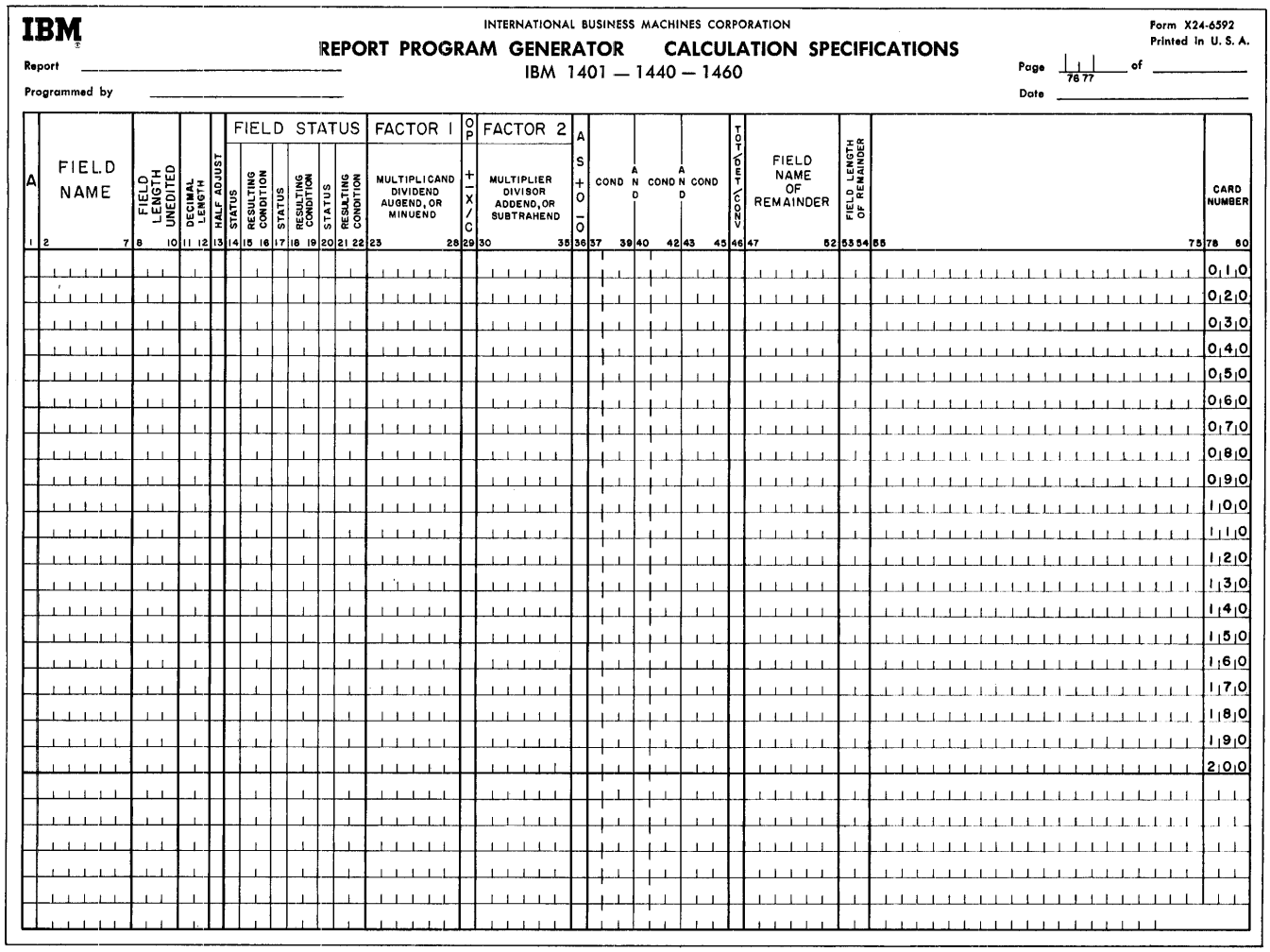


Figure 13. Calculation Specifications Sheet

numeric field that contains no decimals, enter 00. For alphabetic or alphameric fields that contain no decimals, leave these columns blank.

Half Adjust

When the user wants to half-adjust (or round) the result of a calculation, he should enter an alphabetic X in *column 13.* Half-adjusting the result is accomplished automatically by adding 5 to the number in the highest-order position to be *dropped.* Half-adjusting is done just prior to placing the result of a calculation in the Field Name specified. See *Decimal Alignment* under *Calculation Specifications* for an example that illustrates half-adjusting. When no half-adjusting is wanted, leave column 13 blank.

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 14-22* of the calculation specifications sheet can be used for this purpose. The allowable entries in columns 14, 17, and 20 are those given for the field status of the data specifications sheet (B, Z, N, and P), as well as the following entries that 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 sequence than factor 1.
L — Factor 2 is lower in sequence than factor 1.

As mentioned under *Field Status* for the data specifications sheet, each status entry (columns 14, 17, and 20) should be defined by a corresponding unique resulting-condition number (columns 15-16, 18-19, and 21-22). These resulting-condition numbers can be used to govern other operations.

Note: The H, L, and E entries are valid only if the IBM 1401 Data Processing System used to execute the object program is equipped with the high-low-equal compare special feature.

Factor 1

Columns 23-28 are used to state the field name or the *literal constant* 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 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 23.

Literal constants can be used as factor 1 (see *Constants* under *Calculation Specifications).* When factor 1 is a numeric literal less than seven characters long, or an alphameric literal less than five characters long, enter the literal constant as factor 1. If the literal requires less than six columns, right-justify it (enter the units position in column 28), but do not use leading zeros.

A constant that is too long to be written on the calculation sheet as factor 1 can be used indirectly as factor 1. Such a constant then becomes a *named constant,* because the entry in factor 1 columns (23-28) must be WORDxx.

Op (Operation) + − × / C

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

Operation	*Entry on Specifications Sheet in Column 29*	*Punch in Card Column 29*
Plus	+	12-zone
Minus	−	11-zone
Multiplied by	×	0 and 7
Divided by	/	0 and 1
Compare factor 2 with factor 1	C	12-zone and 3

Plus and minus specifications on this sheet provide for addition and subtraction operations upon data developed by the data specifications sheet or prior entries on the calculation specifications sheet. Factor 1 and Factor 2 of a multiplication or division operation must not contain blanks.

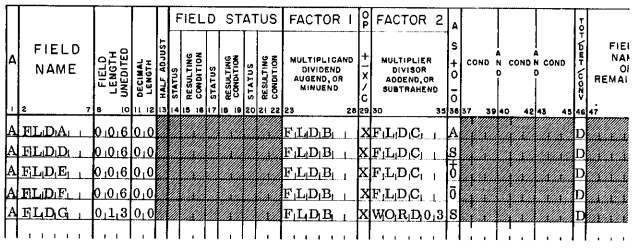


Figure 14. Multiplication Specifications

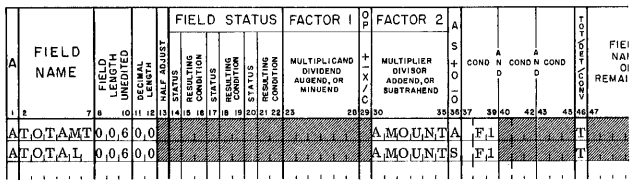


Figure 15. Progressive-Total Specifications

Factor 2

Columns 30-35 are used for the name of the field (or the 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.

AS$\overset{+}{0}\overset{-}{0}$ (Add, Subtract, Reset Add, Reset Subtract)

Column 36 must contain an A, an S, $\overset{+}{0}$ (12, 0), or $\overset{-}{0}$ (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 14 illustrates various multiplication operations. Assume that FLDB and FLDC are specified on the data sheet, having field lengths of 004 and 002, respectively, and no decimals. Plus, minus, and divide operations can be similarly specified. An explanation of the entries in Figure 14 follows:

1. The first line-entry specifies that the product of FLDB × 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 ×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 ×FLDC be reset-added into FLDE.
4. The fourth line specifies that the product of FLDB × FLDC be reset-subtracted into FLDF.
5. The fifth line specifies that the product of FLDB × 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.

Figure 15 illustrates a means of obtaining progressive totals. Note the absence of an entry in the factor 1 field and in the OP field (column 29). This is the explanation of the entries in Figure 15:

1. The first line specifies that the field AMOUNT be added to the contents of TOTAMT and that the sum be placed at TOTAMT on a change in control field 1.

2. The second line specifies that the field AMOUNT be subtracted from the contents of TOTAL and that the difference be placed at TOTAL on a change in control field 1.

Figure 16 illustrates two comparison specifications. The first one compares factor 2, the numeric literal 12345, with factor 1, FLDNAM. FLDNAM must have been defined on either the data sheet or a prior entry on the calculation sheet. An equal result of the comparison can be referred to in other operations as condition 07. The second line compares factor 2, the alphameric literal 6AL5, with factor 1, TUBETP. An equal result of the comparison can be referred to in other operations as condition 12.

Conditions

Columns 37-45 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, 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 37-39 will be NF3.

Tot/Det

Column 46 must contain the letter T or D. The entry depends on whether the line-entry specifies a total or a detail calculation.

A total calculation is performed after the test for a control field change is made, but before information is removed from the record in the input area. The record just read cannot, therefore, contribute to the results of a *total-time* calculation. It is necessary to condition all total-time calculations that specify multiply or divide operations to occur only when the factors are non-blank. 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 46 may be sufficient to condition the calculation. Selected detail calculations and *all* total calculations must be governed by specifications in columns 37-45 as well as T or D in column 46.

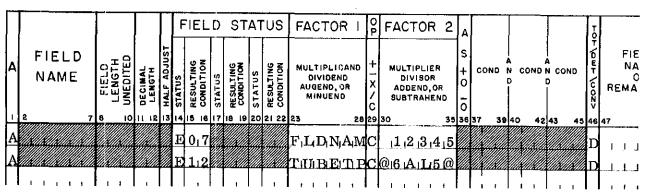


Figure 16. Comparison Specifications

Field Name of Remainder

When the user needs the *remainder* from a divide operation (/ in column 29), he should assign it a *name.* He does this by entering the name in *columns 47-52* of the line-entry specifying the divide operation. BRPG then retains the remainder in the location thus named. It is subsequently available by this name to the user for other operations.

Field Length of Remainder

When the remainder of a divide operation is to be retained for further use, enter the length of the field in *columns 53-54.*

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 sheet must be arranged in order by page number (*columns 76-77* and card number (*columns 78-80).*

Constants

BRPG can use two general types of constants: *literals* and *named constants.*

Literals

Constants whose actual values are entered in the appropriate specifications where they are required are known as *literals.* Literals are not specified by names. BRPG can use two kinds of literals, *numeric literals* and *alphameric literals.*

Numeric Literals. A numeric literal can consist of any combination of the numbers 0 through 9. One decimal symbol and/or one plus sign or one minus sign can also appear in a numeric literal.

Rules for Forming Numeric Literals.

1. Blanks must not appear within a numeric literal.
2. One sign and/or one decimal symbol can appear in a numeric literal.
3. The sign, if present, must be the leftmost character. If a literal is unsigned, it is treated as a posisive literal.
4. The decimal symbol can appear anywhere in the literal except as the rightmost character.
5. Numeric literals must *not* be enclosed in @ symbols.

The following are examples of numeric literals.

162534897	1625.34897
+16253	+162.5
1	—.1625
+1	.1
—1	—.0001

Alphameric Literals. Any set of consecutive characters that is enclosed in @ symbols is treated as an alphameric literal. *Alphameric literals cannot be used for computation.* They can, however, be used for compare operations and as output fields of reports.

Rules for Forming Alphameric Literals.

1. Any character in the IBM 1401 or 1460 character set except the @ symbol (card code is 4 and 8 punches) and the Δ symbol (card code is 11-7-8 punches) can be used in an alphameric literal. Blanks are treated as valid characters and can be used freely.
2. Alphameric literals must be enclosed in @ symbols.

Note: If a literal has the *form* of a numeric literal but is enclosed in @ symbols, it is treated as an *alphameric* literal. It cannot, then, be used for computation.

The following are examples of alphameric literals:

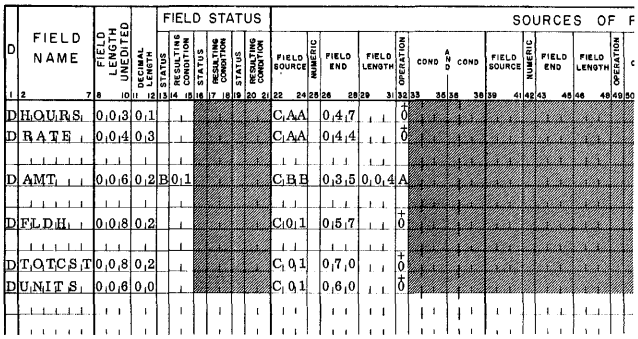
@STOCK INVENTORY REPORT@
@JULY 2, 1963@
@—1.2@

Named Constants

Constants to be used in BRPG can be given names of the form WORDxx. In this form, xx can be any number from 01 through 99. Named constants to be used in computations must have the same *form* as *numeric* literals. Named constants not used in computations must have the form of alphameric literals. As explained under *Format Specifications,* named constants must be defined on the format sheet by a W-entry.

Decimal Alignment

BRPG provides for decimal alignment of computations. This will be accomplished when the resulting



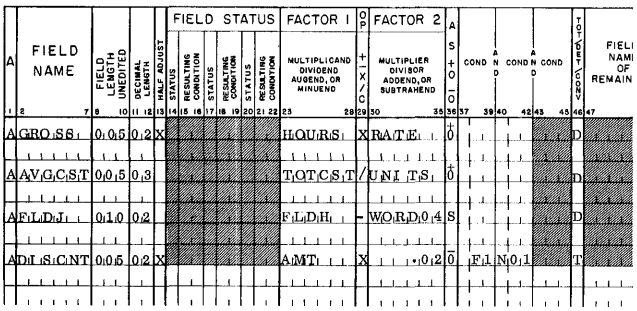


Figure 17. Decimal Alignment

fields, as well as the data fields used as factors in the calculations, contain the appropriate specifications for *field length* and *decimal length.* Numeric literals that include a decimal symbol, when used as factors in computations, also permit BRPG to perform decimal alignment. The same is true for constants named WORDxx that are used in computations (WORDxx defined in the *form* of numeric literals).

Figure 17 shows four computations. *These four calculation specifications are not related to one another.* They serve only as examples illustrating decimal alignment. The first calculation specification multiplies HOURS (xx.x) by RATE (x.xxx) to produce GROSS (xxx.xx), rounded to the nearest cent. The second calculation specification divides TOTCST (of the form xxxxxx.xx) by UNITS (xxxxxx) to develop AVGCST (xx.xxx). The third specification subtracts WORD04 from FLDH (xxxxxx.xx), and subtracts this difference from FLDJ (xxxxxxxx.xx). WORD04 is a named constant, having the *form* of a numeric literal. For this example, assume that it is defined (by a W-entry—see *Format Specifications)* as 12345.67.

The fourth calculation specification multiplies AMT (xxxx.xx) by the literal constant .02, producing DISCNT (—xxx.xx) rounded to the nearest hundredth. The calculation will not take place if AMT is blank.

Note: The data fields in the examples just given do not contain the decimal-symbol character. The illustrations *indicate* the decimal-symbol locations. Programs generated by BRPG check the specified locations of decimal symbols to perform the necessary alignment of the factors used in computations.

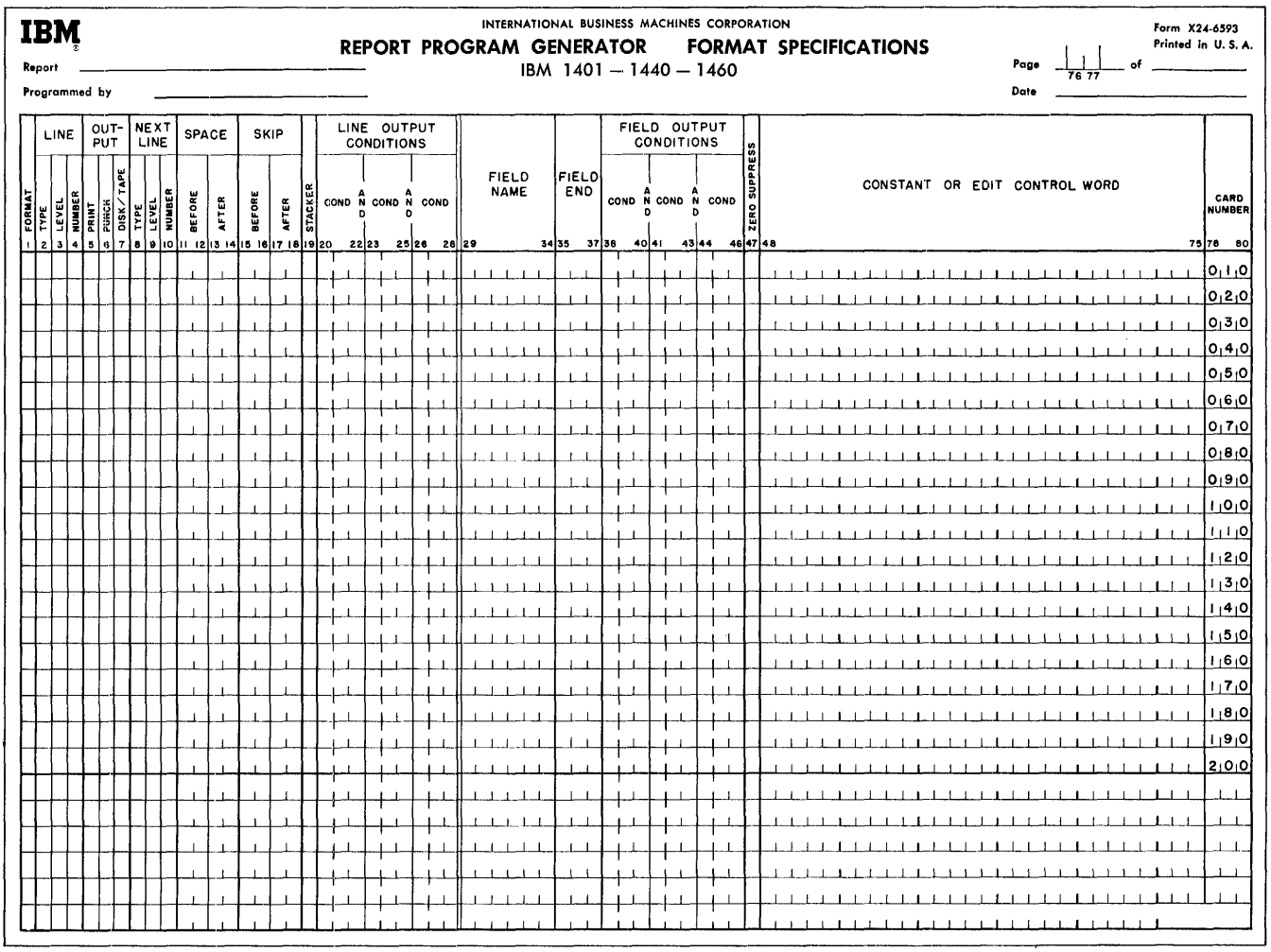


Figure 18. Format Specifications Sheet

Order of Calculation Specifications

All *detail* calculations (D in column 46) must precede all *total* calculations (T in column 46). Within each of these two groups, specify the calculations in the desired order of execution.

Format Specifications

This form (Figure 18) is used to describe the lines and the fields constituting the output. The two main classifications of format specifications are *line* and *field.*

The entries on this sheet pertaining to each *line* supply the Basic Report Program Generator with information such as the identification of the line, the form of the output (whether printed or punched), the next line of output, carriage form spacing and skipping, the stacker into which punched output cards are to be selected, and the conditions under which the line will become output.

The entries on this sheet pertaining to the *fields* within a line supply the BRPG with information such as field name, the rightmost position of the field in the output line, the conditions under which the field will be placed in the line prior to printing or punching, the values of constants, 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.

Write the entries for alphabetically designated lines in the same order in which they appear on the report.

Format (For a Line)

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.*

If a line is identified as HAA, write an H in column 2, an A in column 3, and an A in column 4. If a line is identified as TAB, write a T in column 2, an A in column 3, and a B in column 4.

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. 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.

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

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 not be used for BRPG.

Next Line

Columns 8-10 define the next line to be printed and/or punched. This field is primarily for use with hierarchically related lines and should be left blank when using alphabetic-level classification of lines.

Space

In *columns 11-12* center the number of line spaces to be taken *before* printing the line. The 01, 02, or 03 specifies single, double, or triple line-spacing before the printing of the line being specified.

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

Skip

In *columns 15-16* the entries 01-12 designate skipping to carriage tape channels 1-12, respectively, *before* printing the line being specified.

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 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.

Line-Output Conditions

A line will appear in the output only if line-output conditions for the given line are specified and fulfilled. *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), and negations of any of these.

Figure 19 shows that lines HAA and HBB are to be printed when condition 04 is met. If condition 04 is defined on the input-specifications sheet to be the

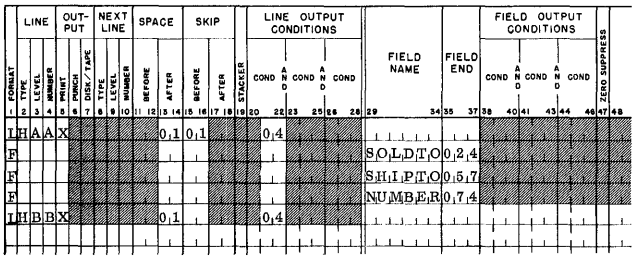


Figure 19. Conditioning Lines to Be Used as Output

presence in the input area of a record containing an L in column 80, the heading lines HAA and HBB will print.

All lines that are to be printed when condition 04 is met will be printed in the order in which they appear on the format-specifications sheet.

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.

Thus, lines that contain *only constant information* and that appear in the output before any input records are processed can be printed or punched 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. Overflow, when it is used as one of two or more *or* conditions, must be written as the last of the *or* conditions. A next-line entry cannot be made for a line that has both an overflow (OF) condition and an *or* relationship with some other line-output condition.

If a detail line called DAA is to be punched when any one of these 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 20.

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.

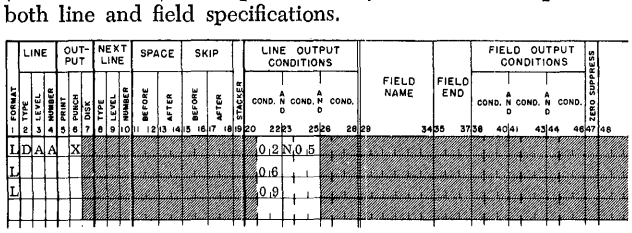


Figure 20. *Or* Line-Output Conditions

Format (For a Field)

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 or a WORDxx 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 read-out and reset total operation on an accounting machine.

K — identifies the entry as a *field* specification that uses a literal 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

State in *columns 29-34* 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. (A *literal* is not named.) 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 under *Constant or Edit Control Word.*

Field End

Place in *columns 35-37* 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.

The same conditions that can govern line output are acceptable in these columns. If several conditions in an *or* relation govern the output of the field, separate entries must be made, each giving all of the information required for the field as well as each *or* condition.

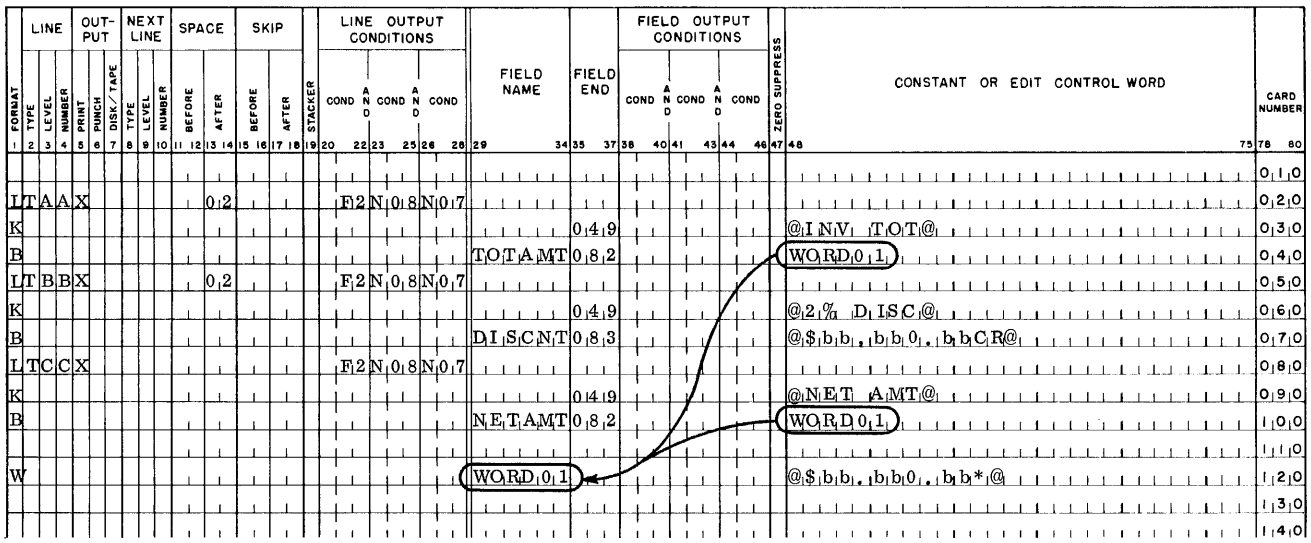


Figure 21. Using Constants and an Edit Control Word

Z (Zero Suppress)

A Z is entered in *column 47* to zero-suppress a field on the output line without using an edit control-word. This entry causes the object program to suppress high-order zeros and strip off the zone in the units position of the field.

Constant or Edit Control Word

A field within a line may consist of a *constant.* Some fields may require an *edit control word.* A constant, used elsewhere in another specification by its name, WORDxx, must be *defined.* In each of these cases, enter the constant or the edit control word in *columns 48-75,* left-justified.

Literal Constants as Output Fields

Literal constants can be included as *fields* of output lines. They are specified as follows: enter a *K* in column 1, the proper *Field End* in columns 35-37, and the *constant* (either a numeric or an alphameric literal) in columns 48-75. Figure 21 shows how the alphameric literal, 2% DISC, is specified (line 060). Although not shown, a numeric literal can likewise be specified. The difference between the specifications for numeric and alphameric literals is only in the *form* of each literal. (Each is explained in *Constants* under *Calculation Specifications.)*

Edit Control Word

When an amount field is to be edited, the user can include the edit control word in the same format specification he writes for the field to be edited. In Figure 21, line 070, the control word $bb,bb0.bbCR edits the field DISCNT to print $xx,xxx.xxCR. Note that the edit control word is written, starting in column 48, in the *form* of an alphameric literal.

Named Constant (WORDxx)

A constant or an edit control word that is used only once should be specified as described previously. However, when a constant or an edit control word is to be used *more than once,* enter WORDxx in columns 48-53 each time it is required. Then, define WORDxx by a W-entry, as explained under *Naming a Constant.* Figure 21 illustrates the use of one edit control word, WORD01, to edit the two amount fields TOTAMT and NETAMT (see lines 040 and 100). Note in each line-entry that the name WORD01 is *not* enclosed in @ symbols.

Naming a Constant

As previously mentioned, both constants and edit control words can be used in calculation and format specifications by *names* of the form *WORDxx.* When so used, each *constant* and edit control word must be

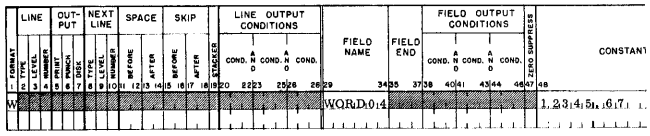


Figure 22. Defining a Constant Used in a Computation

named (or *defined)* by a format specification. This specification must contain a *W* in column 1, *WORDxx* in columns 29-34, and the *constant* or the *edit control word* in columns 48-75. A constant defined as WORDxx that is to be used in computations must be written in the *form* of a numeric literal. For example, if WORD04 were the constant 12345.67 to be used as factor 2 in a calculation specification (see Figure 17), the proper format specification would be as shown in Figure 22. Constants that are not to be used in computations, as well as all edit control words, must be written in the *form* of alphameric literals. In Figure 21, the edit control word $bb,bb0.bb* (used twice by its name WORD01) is defined as WORD01 on line 120. In all applications, W-entries (if used) must follow all other format specifications.

Page Number

Columns 76-77 are used for page numbering in both line and field specifications. The page-number entry is in the upper right-hand corner of the format sheet. The pages 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 in *columns 78-80,* as explained in this corresponding paragraph under *Input Specifications.*

Order of Format Specifications

The first entry on the format sheet must be a format specification for an output *line* (L in column 1).

Line Specifications (L in Column 1)

The required order of format specifications for *lines* is:

1. Alphabetic-level heading lines.
2. Numeric-level heading lines (if used — see *Numeric Level Classification.*
3. Detail lines
4. Numeric-level total lines (if used — see *Numeric Level Classification)*
5. Alphabetic-level total lines

Within each of these groups, lines must be specified in the same order desired in the output (printed or punched output).

Field Specifications (F, B, or K in Column 1)

All field specifications that have an F, B, or K in column 1 must follow the L-entry for the output line in which these fields will appear. These *field* entries can be in any order after the *line* specification.

W-Entries

All field specifications that *define constants or edit control words* (W in column 1) must be written *last* on the format sheet(s).

Numeric-Level Classification

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 can be given numeric-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 intermediate lines. The principle underlying a hierarchical relationship is that lines of higher level govern lines of lower level.

Line-Level Relationships

Of the three types of report lines (H, D, and T), heading and total lines can be related in an established order or rank. Such lines, known as *hierarchical* lines, are assigned a *number* from 1 through 8 as the second character of the line identification code. This number represents the rank or *level* of a line in its relationship to other lines within the hierarchical structure. Heading and total lines in the lowest level must be assigned a *level number* of *1;* the next higher level number assigned must be 2; then 3; and so on, through 8 for the highest level.

For both heading and total lines that are given numeric *level* designations (that is, hierarchical heading and total lines), there may be more than one print line that belongs to a given level. These lines should be numbered, beginning with the number 1 for the first line of that level. For example, suppose that a report requires three heading lines; two of these should print on an intermediate control-field change, and the remaining one should print on a minor control-field change. In this example, the three heading lines would be assigned these line-identification codes: H21 and H22 (the first and second heading lines of the intermediate level), and H11 (the first heading line of the minor level — there is only one minor heading line in this example).

Suppose that, in the foregoing example, there are five total lines that should print; two on a change in the intermediate control field and three on a change in the minor control field. These five total lines should be given these line-identification codes: T11, T12, and T13 (the first, second, and third total lines of the

minor level), and T21 and T22 (the first and second lines of the intermediate level).

Heading lines in a hierarchy (identified by such numeric *level* designations as H21, H22, and H11) are printed in this order:

1. All the heading lines in the highest level, in line-number order.
2. All the heading lines in the next lower level, in line-number order.
3. And so on. The heading lines of the lowest level are printed last, in line-number order.

In the previous example, the order of printing the heading lines (upon a change in the intermediate control field) is H21, H22, and H11.

Total lines in a hierarchy (identified by such numeric level designations as T11, T12, T13, T21, and T22) are printed in this order:

1. All the total lines in the lowest level, in line-number order.
2. All the total lines in the next higher level, in line-number order.
3. And so on. The total lines in the highest level are printed last, in line-number order.

In the previous example, the order of printing the total lines (upon a change in the intermediate control field) is T11, T12, T13, T21, and T22.

Line-output conditions should be assigned only to the *first* line of each level that should print. Succeeding lines of any *numbered* level will follow the first line of that level. For example, condition F1 should condition heading line H11 to print, but not H12. The reason is that when H11 is printed, H12 will automatically follow, because it is the second line of level *1*.

In accordance with the proper order of writing the format specifications (see *Order of Format Specifications),* programs generated by BRPG print report lines in this order:

1. Alphabetic Level Heading Lines
   Printed in the order of entry on the format specifications sheet
2. Numeric Level Heading Lines
   Printed in high-to-low-level order according to their places in the hierarchy
3. Detail Lines
   Printed in the order of entry on the format specifications sheet
4. Numeric Level Total Lines
   Printed in low-to-high-level order according to their places in the hierarchy
5. Alphabetic Level Total Lines
   Printed in the order of entry on the format specifications sheet

Hierarchical treatment is given only to *numeric*-level *heading* and *total* lines.

When the object program is running, a total line with a numeric-level designation such as T3x will force T11 and its subsequent lines, and T21 and its subsequent lines, to come before it whenever the output conditions are fulfilled for T3x.

Figure 23 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.

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 Figure 23 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.

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. It is possible that *level-one* heading lines might relate to a change in control-field *two*. In some 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 numeric levels for each type of line specified.

The *line number* permits scheduling lines within a level. In Figure 23 there are 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 the line-number sequence within that level; that is, 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 numeric 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. When assigning line numbers, always start with the number 1.

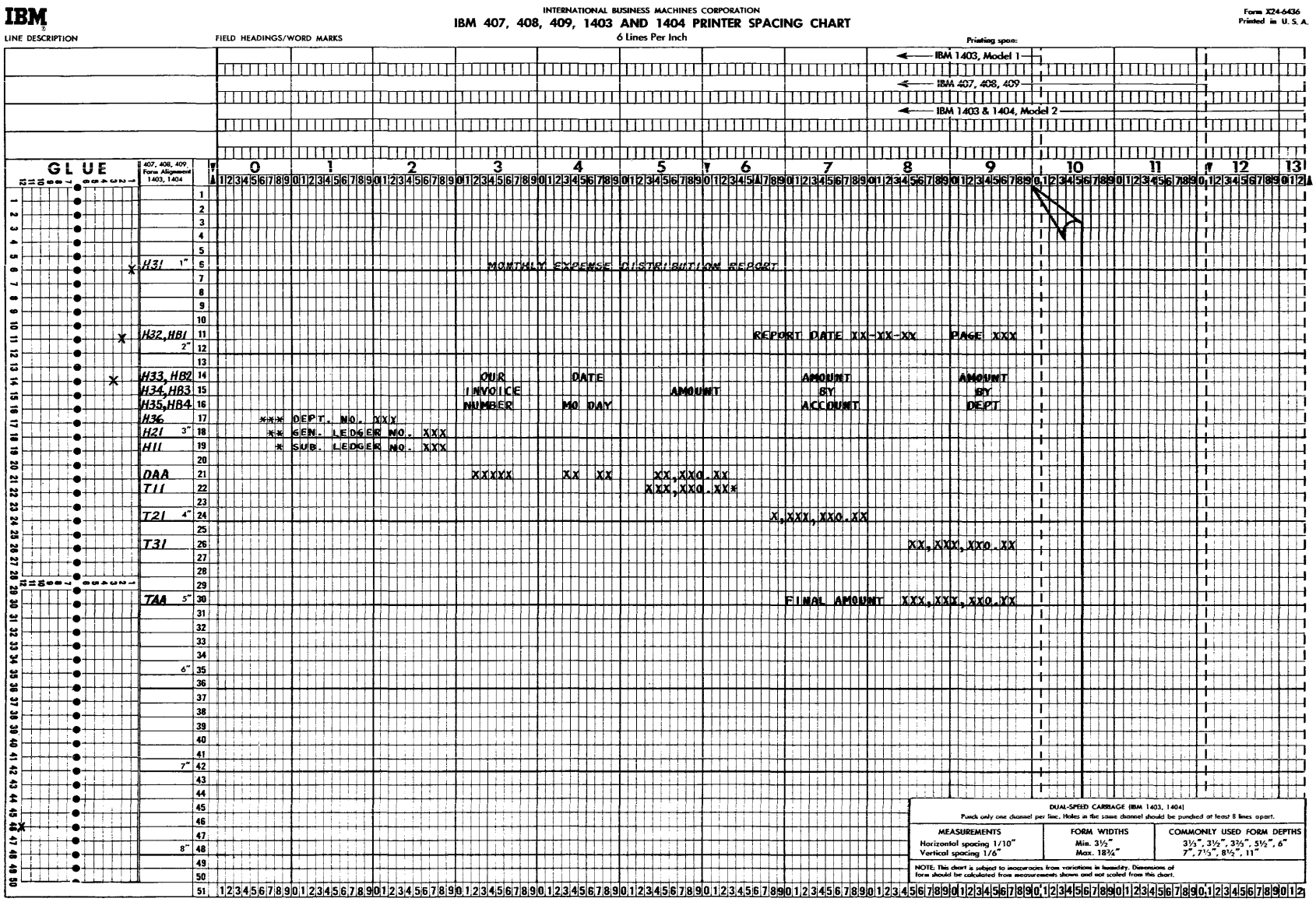


Figure 23. Spacing Chart for Monthly Expense Distribution Report, Numeric-Level Classification

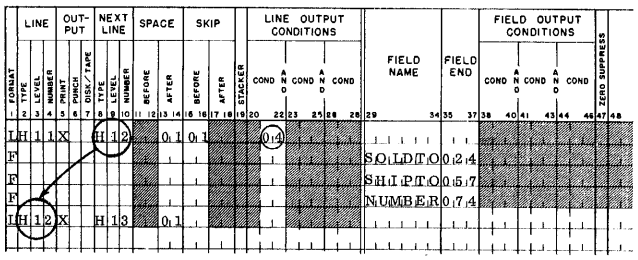


Figure 24. Conditioning Lines to Be Used as Output

Numeric-Level Format Specifications

Format specifications for numeric-level classification closely follow those for alphabetic-level classification, and reference must be made to the previous section dealing with format specifications. However, three specifications are different:

Line

The entry in column 2 specifies the type of the line, as in alphabetic level classification.

Column 3 specifies the level of the line. Column 4 specifies the number of the line within a given level. When lines of a level are to be numbered, be sure to assign the number 1 to the first line.

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.

Next Line

Columns 8-10 define the next line to be printed and/or punched. 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 the *same type and level* as the line being specified. That is, columns 8-9 must be the same as columns 2-3.

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), 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 that referenced it.

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
3. 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, see Figure 24. The line H11 is specified to be printed when condition 04 is met. If 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 also illustrated in Figure 24. 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.

The third principle with regard to line-output conditions relates to the specification of hierarchies of lines.

Lines related in a hierarchy must be of the *same type* and must have *numeric-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. For example, assume that a report has two major total lines (T31 and T32), two intermediate total lines (T21 and T22), and three minor total lines (T11, T12, and T13). Then, when the output conditions are fulfilled for the T31 line, the object program will force T11 and its next lines (T12 and T13), and T21 and its next line (T22), to come before the T31 and T32 lines in the output *without regard for the line-output conditions of the T1x and T2x lines.*

Furthermore, 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 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.

Compressed Specifications

The BRPG program compresses each specifications entry into a string of characters that is used directly by the BRPG program in generating the desired report program. The total number of characters resulting from compressing the specifications entries must not exceed 1900. If the total does exceed 1900, the BRPG program will print the message PROG TOO BIG and halt.

The user can determine the number of characters into which each specifications entry will be compressed. Each type of entry compresses into a number of characters that can be calculated by a simple arithmetic formula. The formula to be used for each type of entry is shown below.

Entry Type	*Formula*
L entry	28
F entry	17 + Literal length + (9 × No. of conditions)
B entry	20 + Literal length + (9 × No. of conditions)
K entry	10 + Literal length + (9 × No. of conditions)
W entry	10 + Literal length
Data entry	24 + (6 × No. of conditions) + (9 × No. of field status tests)
Calculations entry	57 for divide; otherwise 49
Input entry	10 + (6 × No. of record codes) + (5 × No. of control fields)

Performance Data

The time required to generate a Basic 4K Report Program Generator program for an IBM 1401 or 1460 varies from 4.0 minutes to 4.9 minutes with 3.5 to 4.1 minutes for compile time and .5 to .8 minutes for punching the object deck.

The minimum number of statements in BRPG is four (one input, one data, and two format). The maximum number of statements allowed in BRPG depends upon the amount of core needed to compress the user's specifications. This compression area is increased beyond the normal requirements by the following factors.

Format Specifications

1. Literal length on B, F, K, and W entries
2. Field conditioning on B, F, and K entries.

Data Specifications

1. Field-status entries
2. Source-field conditioning.

Calculations Specifications

Use of divide operation.

Input Specifications

1. Control-field entries
2. Multiple-record code entries.

The following specifications were used to run a suggested maximum program of 70 statements.

Type	*No.*	*Description*
Input	5	One record code each; two control fields for each input card.
Data	30	Three with field status.
Calculations	9	Five multiplications and four divisions.
Format:		
Heading	4	Two K entries of 10 characters each and one F entry.
Detail	10	Four B and four F entries.
Total	12	Four B and five F entries with two edit words of eight characters each.

The case shown in Figures 25-33 was run on an IBM 1401 with an IBM 1402 Card Read-Punch, Model 1, and an IBM 1403 Printer, Model 2. It contains 65 statements and requires 4.5 minutes of machine time.

Sample Report Documentation

Documentation for one sample report is presented in this section. Figure 25 shows a portion of the card-input file for the Monthly Expense Distribution Report. Figure 26 is the printer spacing chart. Figures 27-32 are the specifications sheets and Figure 33 shows the printed report.

Control Card

Assuming that the user has an IBM 1401 system with an IBM 1403 Printer with 100 print positions, the control card will have the following format:

Columns	*Contents*	*Explanation*
1-3	CTL	Identifies the control card.
4-5	04	Neither the BRPG processor nor the generated object program will use more than 4,000 positions of core storage.
6-8	100	Specifies the size (number of print positions) of the printer.
9-12	1401	Specifies the system to be used to generate and execute the object program.

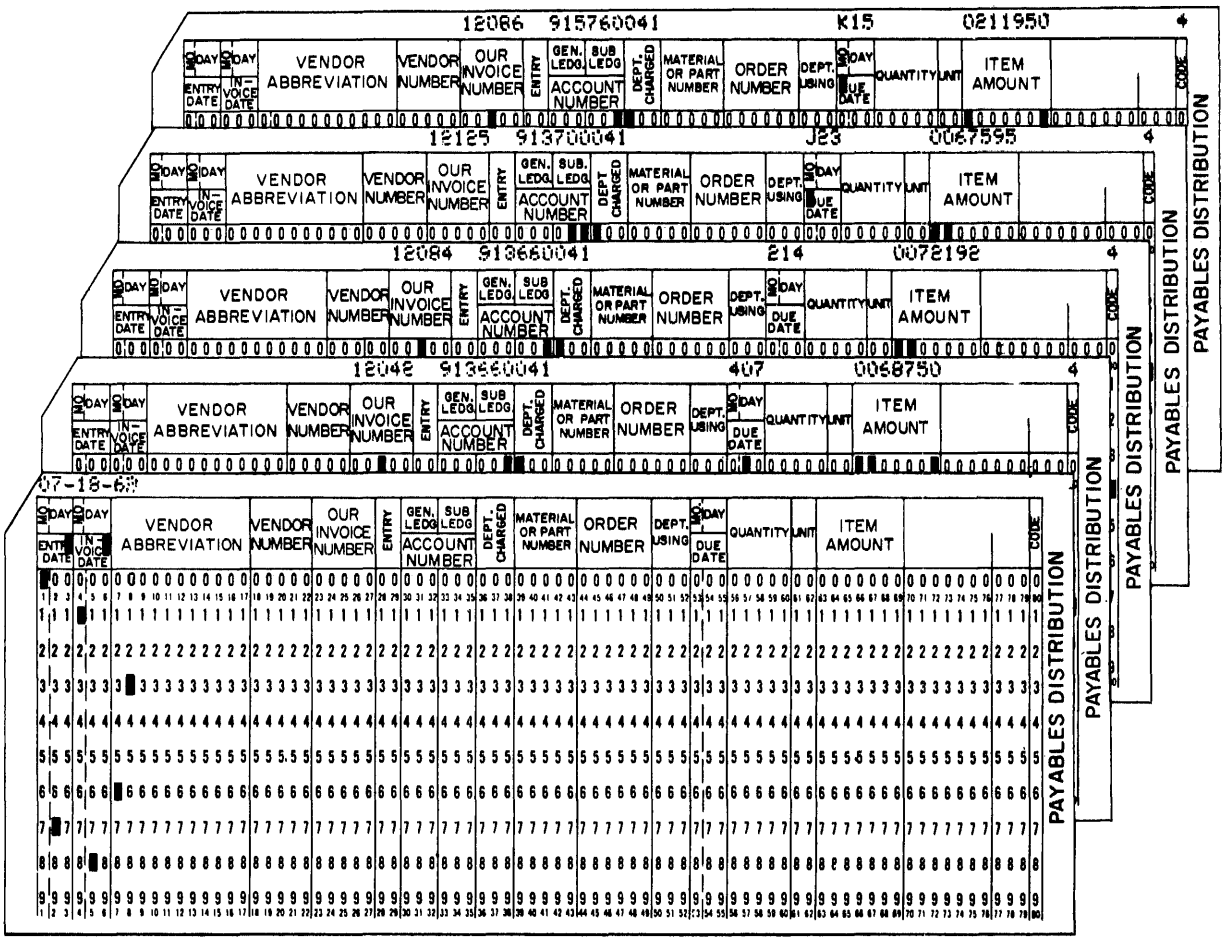


Figure 25. Cards from Input File for Monthly Expense Distribution Report

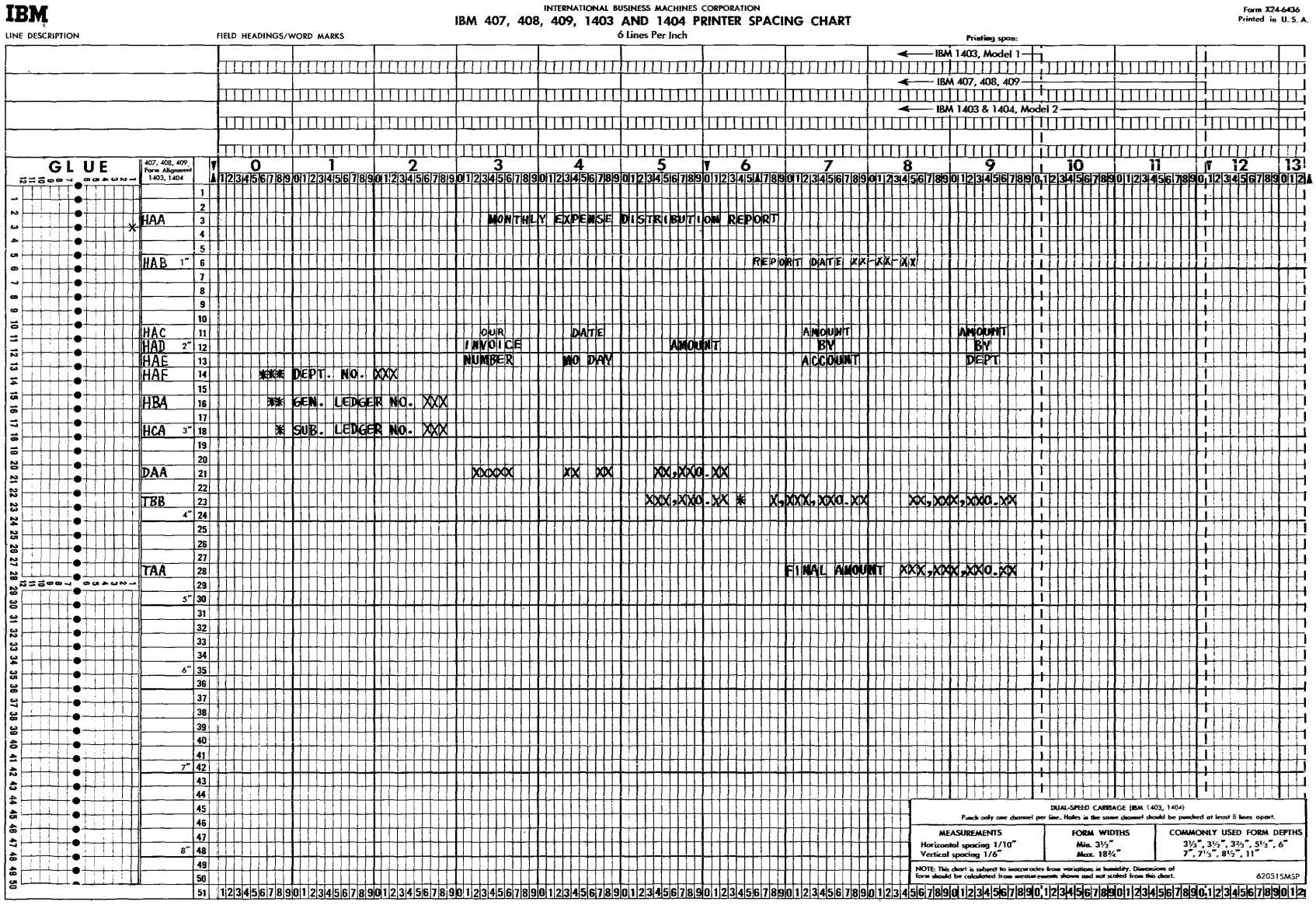


Figure 26. Spacing Chart for Monthly Expense Distribution Report

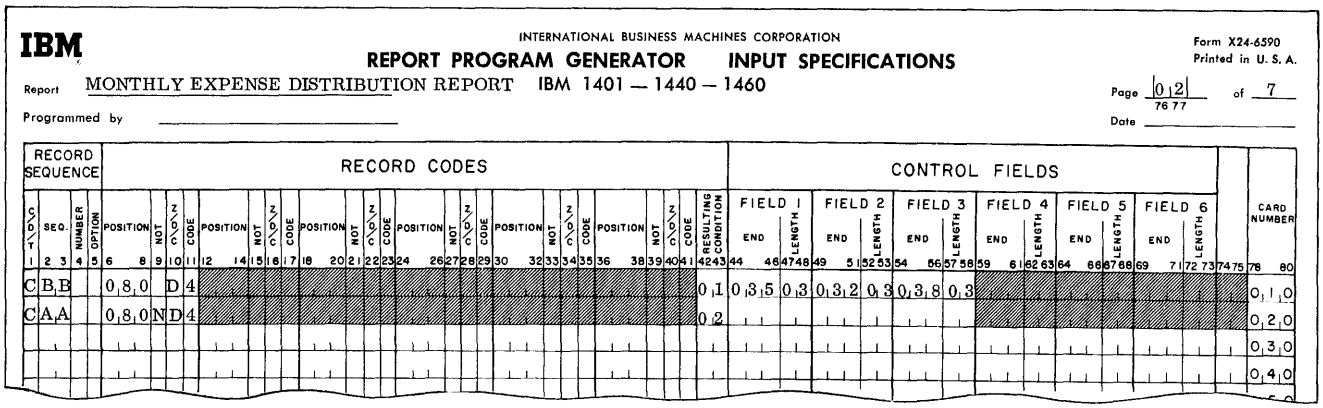


Figure 27. Input Specifications for Monthly Expense Distribution Report

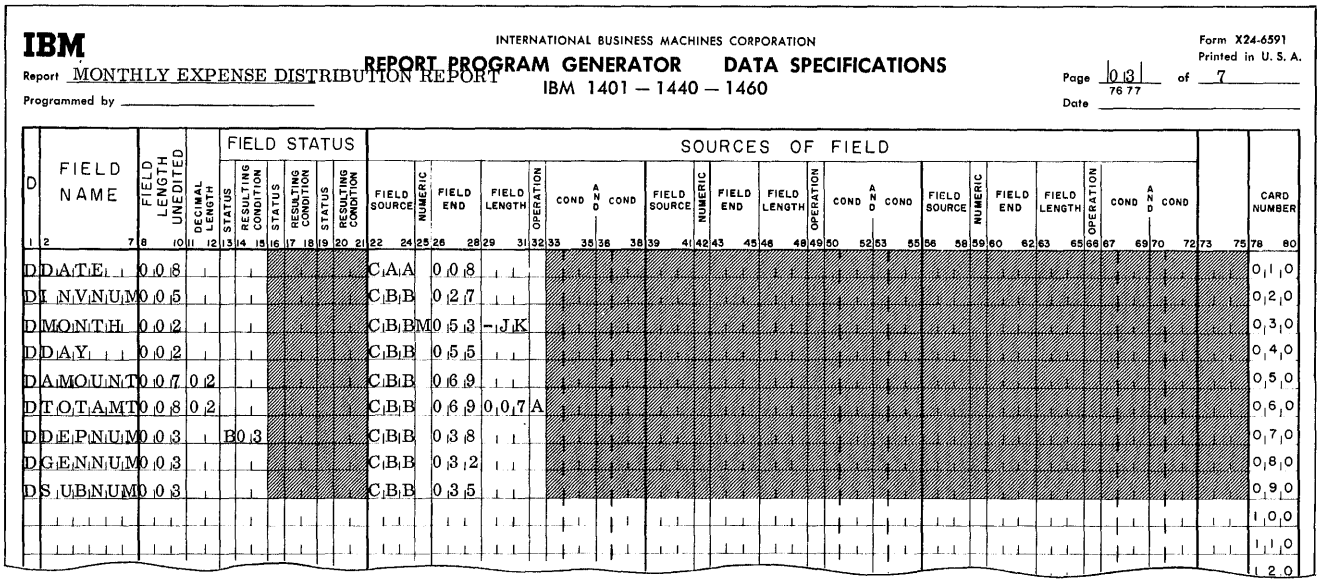


Figure 28. Data Specifications for Monthly Expense Distribution Report

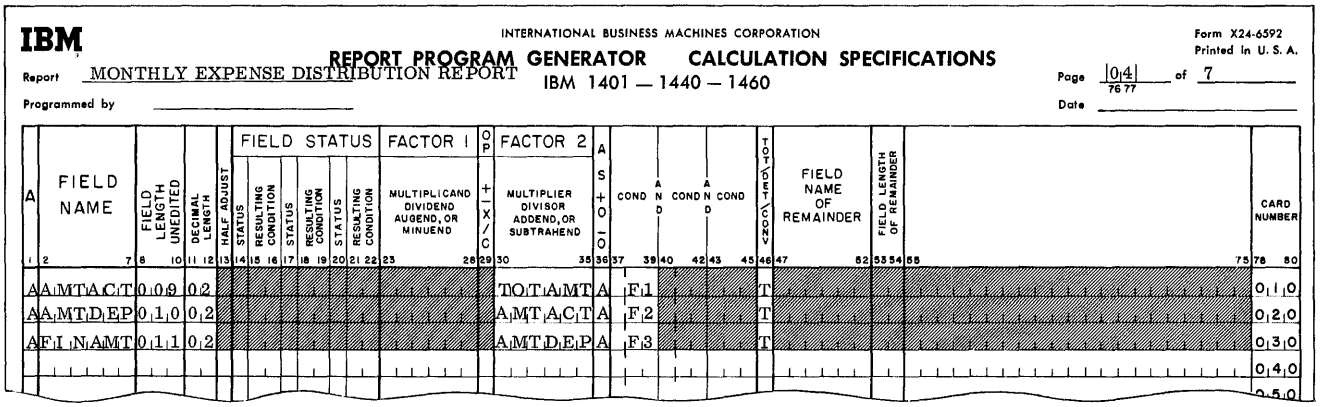
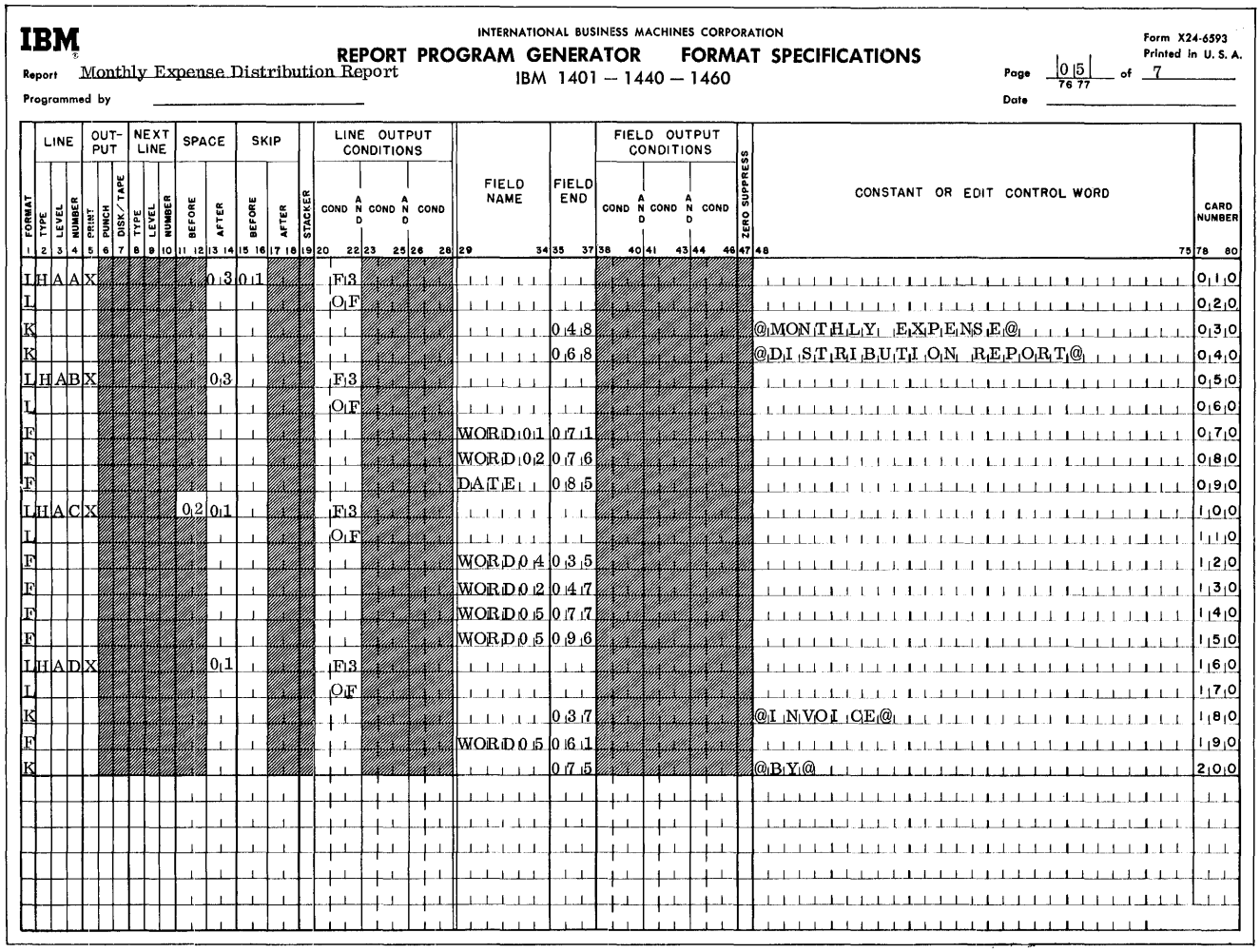


Figure 29. Calculation Specifications for Monthly Expense Distribution Report



IBM

INTERNATIONAL BUSINESS MACHINES CORPORATION

REPORT PROGRAM GENERATOR FORMAT SPECIFICATIONS

IBM 1401 — 1440 — 1460

Form X24-6593 Printed in U. S. A.

Report Monthly Expense Distribution Report

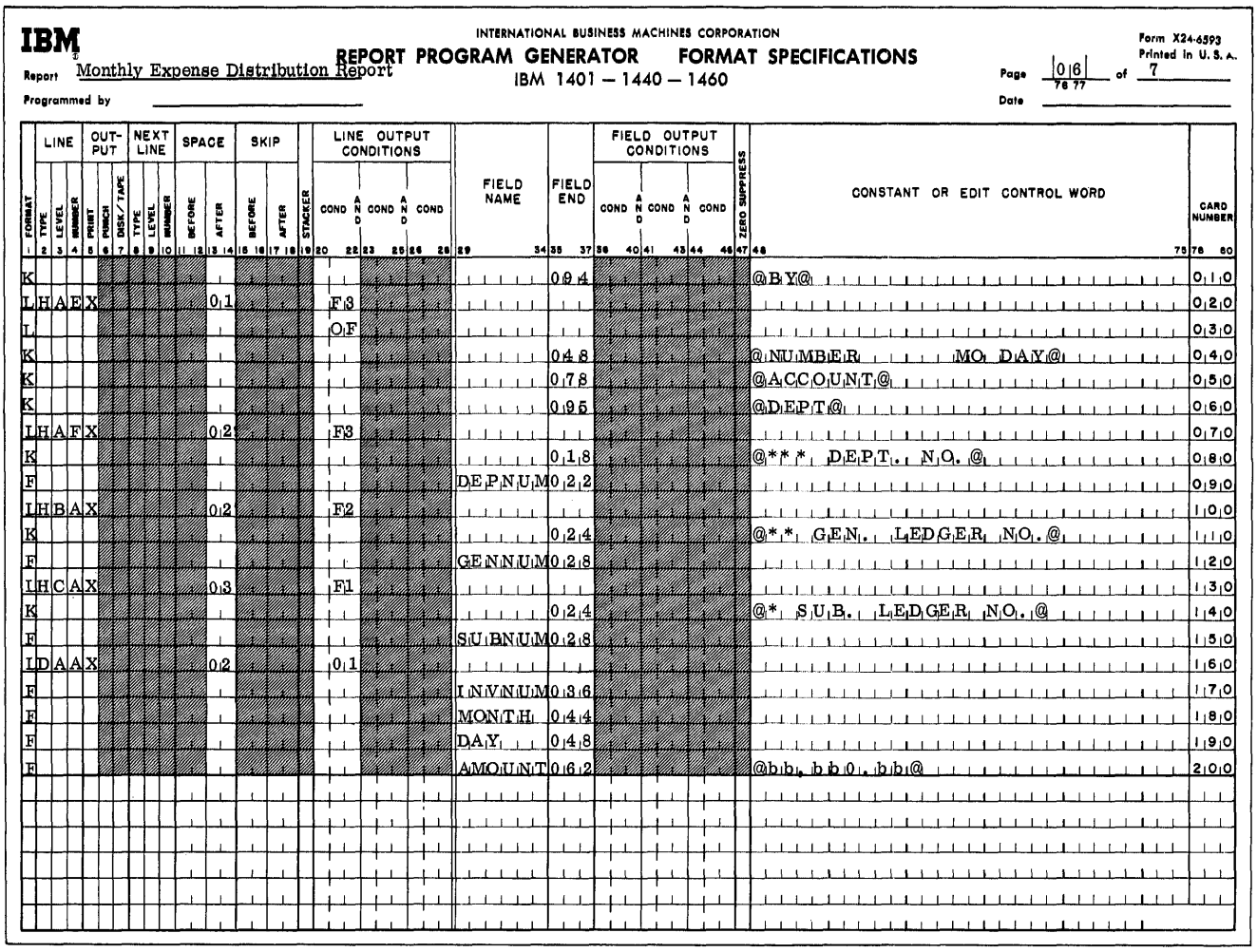
Programmed by

Page 05 (76 77) of 7

Date

FORMAT (1)	LINE TYPE (2)	LINE LEVEL (3)	LINE NUMBER (4)	OUTPUT PRINT (5)	OUTPUT PUNCH (6)	OUTPUT DISK/TAPE (7)	NEXT LINE TYPE (8)	NEXT LINE LEVEL (9)	NEXT LINE NUMBER (10)	SPACE BEFORE (11-12)	SPACE AFTER (13-14)	SKIP BEFORE (15-16)	SKIP AFTER (17-18)	STACKER (19)	LINE OUTPUT CONDITIONS COND (20-21)	AND	COND (22-25)	AND	COND (26-28)	FIELD NAME (29-34)	FIELD END (35-37)	FIELD OUTPUT CONDITIONS COND (38-40)	AND	COND (41-43)	AND	COND (44-46)	ZERO SUPPRESS (47)	CONSTANT OR EDIT CONTROL WORD (48-75)	CARD NUMBER (78-80)
L	H	A	A	X							03	01			F1		03												010
L															O1		F												020
K																					048							@MONTHLY EXPENSE@	030
K																					068							@DISTRIBUTION REPORT@	040
L	H	A	B	X							03				F1		03												050
L															O1		F												060
F																				WORD01	071								070
F																				WORD02	076								080
F																				DATE	085								090
L	H	A	C	X						02	01				F1		03												100
L															O1		F												110
F																				WORD04	035								120
F																				WORD02	047								130
F																				WORD05	077								140
F																				WORD05	096								150
L	H	A	D	X							01				F1		03												160
L															O1		F												170
K																					037							@INVOICE@	180
F																				WORD05	061								190
K																					075							@BY@	210

Figure 30. Format Specifications for Monthly Expense Distribution Report



IBM

INTERNATIONAL BUSINESS MACHINES CORPORATION

REPORT PROGRAM GENERATOR FORMAT SPECIFICATIONS

IBM 1401 — 1440 — 1460

Form X24-6593 Printed in U. S. A.

Report Monthly Expense Distribution Report

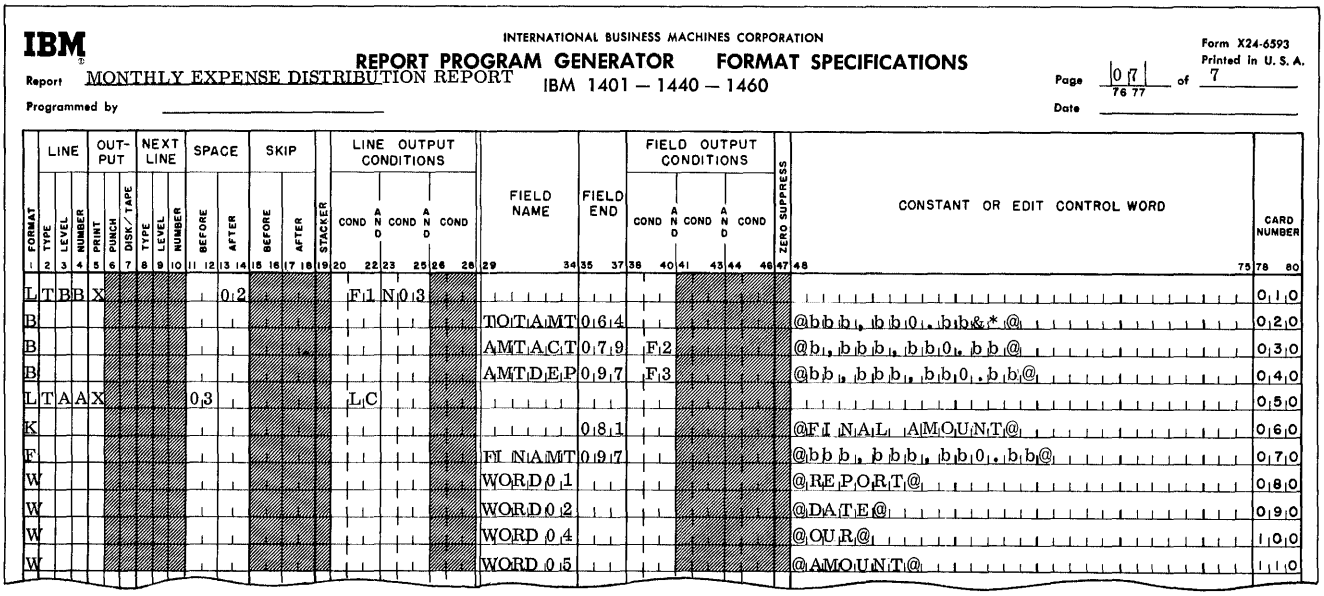
Page 016 (76 77) of 7

Programmed by

Date

FORMAT	LINE: TYPE	LINE: LEVEL	LINE: NUMBER	OUTPUT: PRINT	OUTPUT: PUNCH	OUTPUT: DISK/TAPE	NEXT LINE: TYPE	NEXT LINE: LEVEL	NEXT LINE: NUMBER	SPACE: BEFORE	SPACE: AFTER	SKIP: BEFORE	SKIP: AFTER	STACKER	LINE OUTPUT CONDITIONS: COND	AND	COND	AND	COND	FIELD NAME	FIELD END	FIELD OUTPUT CONDITIONS: COND	AND	COND	AND	COND	ZERO SUPPRESS	CONSTANT OR EDIT CONTROL WORD	CARD NUMBER
K																				094							@BY@	010
L	H	A	E	X							01				F3													020
L															OF													030
K																				048							@NUMBER MO DAY@	040
K																				078							@ACCOUNT@	050
K																				095							@DEPT@	060
L	H	A	F	X							02				F3													070
K																				018							@*** DEPT. NO. @	080
F																			DEPNUM	022								090
L	H	B	A	X							02				F2													100
K																				024							@** GEN. LEDGER NO. @	110
F																			GENNUM	028								120
L	H	C	A	X							03				F1													130
K																				024							@* SUB. LEDGER NO. @	140
F																			SUBNUM	028								150
L	D	A	A	X							02				01													160
F																			INVNUM	036								170
F																			MONTH	044								180
F																			DAY	048								190
F																			AMOUNT	062							@bbb,bb0.bb@	200

Figure 31. Format Specifications for Monthly Expense Distribution Report



IBM

INTERNATIONAL BUSINESS MACHINES CORPORATION

REPORT PROGRAM GENERATOR FORMAT SPECIFICATIONS

IBM 1401 — 1440 — 1460

Form X24-6593 Printed in U. S. A.

Report MONTHLY EXPENSE DISTRIBUTION REPORT

Page 017 (76 77) of 7

Programmed by

Date

FORMAT	LINE: TYPE	LINE: LEVEL	LINE: NUMBER	OUTPUT: PRINT	OUTPUT: PUNCH	OUTPUT: DISK/TAPE	NEXT LINE: TYPE	NEXT LINE: LEVEL	NEXT LINE: NUMBER	SPACE: BEFORE	SPACE: AFTER	SKIP: BEFORE	SKIP: AFTER	STACKER	LINE OUTPUT CONDITIONS: COND	AND	COND	AND	COND	FIELD NAME	FIELD END	FIELD OUTPUT CONDITIONS: COND	AND	COND	AND	COND	ZERO SUPPRESS	CONSTANT OR EDIT CONTROL WORD	CARD NUMBER
L	T	B	B	X							02				F1	N	03											010
B																			TOTAMT	016	4							020
B																			AMTACT	079		F2						030
B																			AMTDEP	097		F3						040
L	T	A	A	X						03					LC													050
K																				081							@FINAL AMOUNT@	060
F																			FINAMT	097								070
W																			WORD	01							@REPORT@	080
W																			WORD	02							@DATE@	090
W																			WORD	04							@OUR@	100
W																			WORD	05							@AMOUNT@	110

Figure 32. Format Specifications for Monthly Expense Distribution Report

MONTHLY EXPENSE DISTRIBUTION REPORT

REPORT DATE 07-18-63

	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	1	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	2,085.37
					FINAL AMOUNT	2,085.37

Figure 33. **Printed Report of Monthly Expense Distribution Report**

Index

IBM 1401/1460 Printed in U. S. A. C24-3166-1

IBM

International Business Machines Corporation

Data Processing Division

112 East Post Road, White Plains, N. Y. 10601