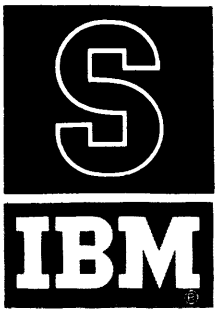


7090 DATA PROCESSING SYSTEM BULLETIN

IBM 7090 PROGRAMMING SYSTEMS
SHARE 7090 9PAC
PART 3: THE REPORTS GENERATOR

This is the third part of a reference manual which describes the SHARE 7090 9PAC System; this publication describes the Reports Generator portion of 9PAC. Other parts of the manual are:

SHARE 7090 9PAC
Part 1: Introduction and General Principles
Form J28-6166

SHARE 7090 9PAC
Part 2: The File Processor
Form J28-6167

References in this publication to the other parts of the manual are in terms of part and chapter numbers. An understanding of Part 1 is required to use this part.

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

The Reports Generator produces object programs that will produce reports and/or records; with one minor exception, all object program output will be written onto tape. In the sense that the terms are used here, reports pertain only to tape files for printing or card punching, and records pertain to any type of magnetic tape file. The distinction is important for two reasons. First, the coding forms used to generate reports and records are different; report coding forms are specifically printer-oriented while record coding forms are not. Secondly, the physical format of the output tape, while completely optional for records, is somewhat limited for reports, since the printer or card punch must have BCD files of fixed length records without dictionaries, check sums, or sequence numbers.

Included in the types of records which the Reports Generator may write are certain specialized types of output. For example, file establishment may be accomplished through the Reports Generator. Whereas the File Processor is limited to single file output, the Reports Generator may establish files while writing other output files. This feature of the Reports Generator does not, however, eliminate the need for File Processor file establishment since the File Processor may have multi-file input to create a master file, whereas the Reports Generator is always limited to single file input.

The Reports Generator includes many automatic features for ease of report and record generation. Chief among these features is the ability of the Reports Generator to operate upon data conditionally and to provide special output editing based on output data. Additional features will be discussed later in this chapter; however, at this point it should be noted that the Reports Generator makes provision for handwritten calculations in a language similar to symbolic machine language; these hand calculations may be used to augment the automatic features of the system.

Input/Output Files

The Reports Generator can produce several output files, although it is restricted to a single input file.

Input Files

Any type of file is acceptable to the Reports Generator as input, however:

1. Only one file may be input to a single Reports Generator program.
2. Sequencing information is not checked.
3. If a parent-offspring relationship exists, it is ignored.
4. If a file contains multiple record types of unequal length, the record type must be designated by the first two BCD characters of the record and the file must have a dictionary.

5. If a file contains records which are all of equal length, the first two BCD characters of the record may or may not contain the record type; in either case, a dictionary is optional.

Output Files

Although Reports Generator object program output is basically on tape, on-line printing is provided for certain types of report output; these are discussed later. Generally, output files may have any of the physical characteristics listed in Part 1, Chapter 2. However, output files for printing or punching may be blocked or unblocked and must conform to the following:

1. They must not have a dictionary, check sums, or sequence numbers.

2. They must be in the BCD mode, either high or low density.

3. They must consist of records of a size acceptable to the printer or punch. Thus, records for off-line printing must be less than or equal to 120 or 132 characters, depending on the printer being used.

Group Printing

When off-line printing is to be done on a printer equipped with a buffer, such as the IBM 720 Printer, it is possible to specify group printing for Reports Generator reports; group marks are generated by the Reports Generator. Use of this facility may reduce output time. An on-line message will indicate whether the records are being grouped by 5 or 10.

Blocking Records or Reports

When desired, Reports Generator output may be blocked. This feature is not generally meaningful for non I-String (see below) report output, since the output file would require deblocking during a subsequent computer run before printing or punching could occur. It does, however, permit tape space and output time economies for certain record output.

I-String Report Output

Reports Generator output may be in the form of I-Strings, which will compose an I-Language file requiring subsequent processing on the IBM 1401. This type of output relieves the 7090 of considerable editing and preparation of the print line, and it is conservative of tape space, thus decreasing 7090 processing and output times with little change in the method of specifying report generation.

CHAPTER 2: REPORT OUTPUT

Specifying desired report output is similar to drawing a picture of the desired output on the required coding forms. In the following discussion, report output will be discussed with reference to the printed page. The concepts presented are equally applicable to punched card output except as noted; of course, report output destined for punching should not exceed 80 columns per line.

Types of Lines

A report can be constructed using three types of lines; the different types control when a line is to be printed relative to the report as a whole. The three types of lines are detail lines, total lines, and heading lines.

Detail Lines

Each detail line is associated with a particular input record type, which is called the associated record type. Whenever a record type meets the acceptance criteria for processing, it causes all detail lines associated with that record type to be printed. Thus, there is a one-for-one correspondence between the printing of each detail line and the reading and processing of its associated record type.

Total Lines

Total lines are associated with one or more control fields and are printed each time a control break occurs on any associated control field. While the associated control field is normally a sequencing field, this is not necessarily the case. Total lines may also be printed when there is a synthetic control break; this will be discussed later.

Each total line is given a level designation, which, in turn, is associated with one or more control fields; there may be many total lines at the same level. The importance of this level designation is that each time a total line is printed, all total lines of a numerically lower level are also printed; that is, the Reports Generator assumes that when a control break occurs on a numerically higher level control field, that a control break has also occurred on each numerically lower level control field. Thus, in general, the primary sequencing field of the input file is used as the highest level control field, and the least significant sequencing field of the input file is used as the lowest level control field.

Heading Lines

There are three types of heading lines: one-time headers, detail-line headers, and total-level headers.

One-time headers are printed as the first lines of a report.

Detail-line headers are printed preceding the first detail line and every nth line thereafter, where n is specified by the programmer. The number of lines per heading line, n, is tested before a detail line is printed; thus,

the detail-line headers may be more than n lines apart if the heading is preceded by total lines. A detail-line header is not automatically placed at the top of a page; this only occurs if the first character of the header line is the proper carriage control character for a skip to a new page.

Total-level headers are associated with total levels and are printed whenever a control break occurs on an associated control field for that level.

Frequency of Output Lines

In order to visualize the appearance of a report, it is necessary to have a knowledge of when the various lines will occur (timing), as well as other information concerning spacing. The following is a summary of timing information:

1. Except with I-String output, a detail line following a heading or total line will be automatically preceded by a double space.

2. A detail line will occur each time its associated record type is read and accepted. Acceptance of an associated record type is based upon use of the Classification Table, which is discussed below; this feature is similar to selection of records by the File Processor.

3. A total line will occur each time its associated control field has a change in value or whenever a total line having a numerically higher level is printed.

4. A detail-line header will occur regularly every nth line unless the nth line precedes a series of total lines, in which case, the detail-line header will precede the next detail line.

5. A total-level heading line occurs whenever a control break occurs on its associated control field(s) or whenever a total-level heading line of a numerically higher level occurs. If the particular control break also causes total lines of equal and lower levels to occur, these lines will all precede the total-level headers.

6. Blank lines, caused by carriage control or printer setting (e.g., Double Space), are counted the same as printed lines.

If a detail line has record type ZZ as its associated record type, then that detail line will be printed every time a record is read and accepted which is not the associated record type of any other line of any report or record in the same Reports Generator job.

Format Line Numbers

Each line specification must be given a format line number. Format line numbers, in addition to providing unique identification for each line format, also contain some of the information used to determine the relative occurence of each line.

The format line number is composed of three characters:

1. The first character specifies the type of line as follows:

   H Heading line
   D Detail line
   T Total line

2. For detail lines, the second and third characters of the format line number serve two functions; they give each line a unique identification code for referencing and they specify the order of printing detail lines when several lines have the same associated record type.

3. For one-time heading lines and detail-line headers, the second character is D and the third character specifies the order in which the lines are to be printed.

4. For total lines and total heading lines, the second character is the control break level. This character must be associated with one or more control fields. For example, if the second character is a 2, the line will be printed whenever a control break occurs on a control field for level number 2 or any greater number. The third character serves the function of sequencing the lines for printing if several lines have the same control break level. If total heading lines and normal total lines have the same control break level, all total lines which will be printed because of this control break will be printed before any of the corresponding total heading lines.

Specifying Line Format

Each line on the printed page and each field of each line, will be specified on the Report Definition form; information may be designated as either literal or variable. Literal information is represented as it is to appear each time the line is printed. Variable information is represented by a symbolic variable field which will subsequently be replaced by data obtained during processing.

The Report Definition form is used to "picture" report output in the following manner.

Literal information may be written in the print positions as it is to appear on the printed report.

Variable information is represented symbolically, as follows:

1. The first character of a variable field must be +.

2. Succeeding characters, if any, must be represented by X.

3. Punctuation may be included in the variable field, as follows:

   a. The lozenge (⌑) will cause a blank to be output.

b. The period (.) or the comma (,) will cause a period or comma, respectively, to be output.

4. Punctuation (␣ . ,) must conform to the following:

a. A variable field may not have more than five punctuation characters not counting a single terminating punctuation character, if one exists.

b. No two pieces of punctuation may be adjacent.

c. Punctuation may not be preceded by more than 15Xs (or 14 Xs and a plus sign).

d. A punctuation character not preceded by a + or X is considered to be a literal.

e. Punctuation immediately to the right of the final X is considered to be part of the variable field.

The actual data field replaces the Xs and the leading plus sign in the variable field. If there are more Xs than there are characters in the data, leading zeros are supplied; if there are more data characters than there are Xs, an error message will be written during generation.

Each variable field is identified within the line format in terms of its rightmost print position, that is, the rightmost character of the variable field (whether it be +, X, or punctuation).

Carriage Control

The method of specifying carriage control varies, depending upon whether the output is being written in I-Language.

Normal Printer Carriage Control

A carriage control character may be stipulated for each line format specification. It is stated as a literal and becomes the first character of the output line. It will be printed in print position 1 unless the printer is set to Program Control, in which case it will be interpreted as a carriage control character and the second character of the output line will be printed in print position 1.

The available carriage control characters and their effect are:

blank	Single space before printing
0	Double space before printing
+	No space before printing
1-9	Skip to printer control channels 1-9

Carriage control is further discussed in the reference manual applicable to the printer being used.

Note: An automatic double space is inserted by the Reports Generator preceding the first detail line of a group (series) of detail lines.

I-Language Carriage Control

If report output is to be in I-Language, characters indicated as normal printer carriage control characters will be printed in the first print position of the print line. The following carriage control characters may be coded to control I-Language output:

Hx — Skip x lines, where x is a two-digit designation of the number of lines to be skipped.

Zx — Skip to line x, where x is a two-digit designation of the line number on which the line is to be inserted. This number must not be greater than the stated number of lines per page. If the number is less than the line at which the page is currently positioned, the printer will skip to a new page and print upon the specified line.

Ibb — Skip to new page.

bbb — Single space.

CHAPTER 3: RECORD OUTPUT

Whereas the Report Definition form provides a method of specifying a picture of output by print position, the form used for record definition is quite similar to the File Processor Dictionary Definition form, in that record types and field numbers are specified within a logical record. Although heading lines, as such, may not be prepared for records, both records and reports are generated on a one-to-one basis with associated input record types.

Types of Output Records

Each output record must be designated as either a detail record or a total record. The primary distinction is that detail records cannot have information summed into them, whereas total records can; as in report lines, a detail record must be associated with an input record type, whereas a total record must be associated with an input control field. As with input records, a unique record type must be assigned to each of these output records in order to facilitate subsequent referencing.

Detail Records

Except for the method of specifying the format of a detail record, a detail record is treated the same as a detail line in report generation and is subject to the same conditions. That is, a detail record must be given an associated record type, which controls the frequency of output of this particular record and which, in general, contains at least some of the data which will be inserted into the output record.

Total Records

A total record is subject to the same rules as to content and frequency that total lines are subject to in report generation. That is, total records may contain fields which are the accumulation of one or more input data fields. Also, the frequency is controlled by one or more associated control fields.

Thus, one can visualize a record generation pass as a specialized report generation pass consisting of only detail and total lines, each having a format composed of various fields rather than specified print positions. Heading lines, lines per page, lines per heading line, carriage control characters, etc., have no significance in record generation.

Format of Output Records

Each output record is composed of fields and, when the output record is to have a dictionary, the first field must be a record type designation consisting of two numeric BCD characters.

Each field specification is made unique by a record type and field number designation. As with the File Processor, the record type is a unique two-digit designation and the field number is a four-digit designation unique within the record type.

The length of the field must be stated in bits and, if the file is to be used subsequently by the File Processor, the sequence level of the field must be stated if the field is to be a sequence field, a "replace only" code must be given, and, if desired, an allowable range for entries may be stated. The field itself must be specified as one of three types: signed, unpacked binary; unsigned, unpacked BCD; or unsigned, packed BCD.

The programmer need not be concerned with the location of the field within the record unless he so desires, since this assignment may be made automatically. If any field within a record type is assigned an absolute location by the programmer, then he must assign absolute locations to all fields within that record type. Regardless of the method of assignment, the record type must always be the first field of the record. If field locations are to be assigned automatically, the field containing the record type must be designated field number 0001. If the field locations are assigned automatically by the Reports Generator, they will be in the following order: unpacked BCD, packed BCD, and binary; and they will be organized by field number within each grouping.

Creation of the Output Dictionary

The programmer may specify that the output file is, or is not, to have the dictionary attached. Further, he may specify (in the case of a file without dictionary) that the record type is to be omitted from the output records.

CHAPTER 4: REPORTS GENERATOR OPERATIONS

The Reports Generator provides for a variety of operations within the automatic framework of the system. These operations may be used alone or they may be augmented by the use of hand calculations to produce desired reports.

Editing of Output Fields

The programmer has two primary methods of controlling the appearance of a report. The first is through specification of the print method to be used in printing the output field and the second is through grouping of fields within detail lines (which may be used to suppress redundant information).

Print Method

A print method must be specified for each data field. Listed below are the various print methods and their effects:

Print Method Code	Description
0	Insert data without editing; this is primarily used for alphameric BCD fields.
1	Suppress leading zeros and punctuation preceded by all zeros. Zeros following a decimal point are not suppressed.
2	Print plus zero as blank; otherwise, edit as indicated by the variable field.
3	Operate in accordance with codes 1 and 2.
4	Same as 1, except that suppressed characters are replaced by asterisks (*).
5	For use with I-Language only; same as 1, with a floating dollar sign ($) to the left of the decimal point or leftmost non-zero digit preceding the decimal point.
6	For use with I-Language only; same as 1, but CR printed rather than a minus sign (-) following a negative number.

Notes: Emitted information (discussed below) must use print method 0. A one-character I-Language field must use print method 0. For all print methods other than 0, a BCD field with an 11-punch over the low-order digit will be considered negative. In this case, a minus sign will be formatted to the immediate right of the variable field.

The following examples will clarify the operation of the various print methods:

PRINT METHOD	SAMPLE FORMAT	DATA	RESULT
0	$+XXXX	12345	$1234N
	$ +XXXX	00000	$ 00000
	$+XX.XX	00123	$000123
	$+XX.XX	00000	$000000
1	$+XXXX	12345	$12345-
	$+XXXX	00001	$ 1
	$ +XXXX	00000	$
	$+XX.XX	00001	$ .01
	$+XX.XX	00000	$ .00
2	$+XXXX	12345	$12345-
	$+XXXX	00001	$00001
	$ +XXXX	00000	$
	$+XX.XX	00001	$000.01
	$+XX.XX	00000	$
3	$+XXXX	12345	$12345-
	$+XXXX	00001	$ 1
	$ +XXXX	00000	$
	$+XX.XX	00001	$ .01
	$+XX.XX	00000	$
4	$+XXXX	12345	$12345-
	$+XXXX	00001	$****1
	$ +XXXX	00000	$ *****
	$+XX.XX	00001	$***.01
	$+XX.XX	00000	$***.00
5	+XXXX	12345	$12345-
	+XXXX	00001	$1
	+XXXX	00000	
	+XX.XX	00001	$.01
	+XX.XX	00000	$.00
6	$+XXXX	12345	$12345CR
	$+XXXX	00001	$ 1
	$ +XXXX	00000	$
	$+XX.XX	00001	$ .01
	$+XX.XX	00000	$ .00

Grouping Within a Detail Line

A variable field within a detail line may, if desired, be caused to appear only in the first detail line following a heading or total line. This is achieved by specifying that the particular output field is to be grouped; the alternative is to print each variable field each time the detail line is printed. If several detail lines have the same associated record type, only fields within the lowest level detail line (the line that is formatted first) may be indicated for grouping; if this line is not printed, the grouped fields will not appear.

Specifying Input Fields

Data for Reports Generator reports and records may come from any type of file. If the file contains a dictionary, the input may be referenced by record type and field number or by partial field within a record type and field number. If the file does not contain a dictionary, the input must be referenced in terms of increment and length from the beginning of the logical record. Increment is defined as the number of characters preceding the field being referenced; thus, the increment of the first character is zero. Length is defined as the number of characters which comprise the field being referenced.

Selection of Input

The system input/output routines perform an automatic selection process, based on input record type, to assure that input records use the correct program modules. The user does not request this selection; it is implicit in the relationship which is established between a detail line and its associated record type.

In addition to automatic selection, there is a high-powered selection facility which can be used to define conditions to be satisifed by the input data and to test for these conditions as a basis for further operations; this selection is accomplished by setting and testing the Classification Table.

The Classification Table

The Classification Table consists of eight columns, each of which may contain at any one time none, any, or all of the integers from 1-9. The process of placing an integer into a column is termed setting the table. The process of determining whether a column does or does not contain a particular integer is termed testing the table.

The use of the Classification Table consists of two steps: setting the table and testing the table. The table can be set as a result of tests on input data; the testing of the table can be used to control such actions as: acceptance or rejection of an entire input record, determining whether a field should be formatted, determining whether a line should be written to the report tape after it has been completely formatted, and determining whether a particular set of hand calculations should be executed.

In the case where the action is to determine whether the input record is to be accepted or rejected, the testing necessary to determine this action will be done before any other testing, thus making it possible to reduce processing time. Tests are divided into two categories called high tests and low tests. All high tests are made prior to the point at which it is determined whether to accept the record; all low tests are made after this point.

Each parameter packet has only one Classification Table regardless of the number of input record types used. Consequently, the usual procedure is to have each input record turn off (reset) the entire table automatically, thereby nullifying the effect of all previous settings on subsequent tests. If the user wishes to have test results available during the processing of more than the immediate input record, he can designate one record type as a master record type. The master is then the only record type which will reset the entire table. Other record types will reset only those parts of the table which they themselves may set.

Functions Used to Classify Input

All classify functions specify a test to be made and an action to be taken. The test can be on an input field or on the Classification Table; the action is always the setting of the table. The tests on input data are completely specified by associating one value or a range of values with an input field. In either case, a test is made on the data field for equality with the associated value(s). The user may specify what integers are to be set into what columns of the table in the event that the specified condition is satisfied.

For functions which perform tests on the table and also specify setting the table, one column of the table must be designed as the column to be tested and the other columns as those to be set.

In all cases of tests on the table, the test can be for a column containing an integer or for a column not containing an integer. To specify that a column is not to contain an integer in order to satisfy the test, a minus sign (11-punch) must be entered in the column in addition to the integer.

There are six classify functions which the programmer may use to set the Classification Table; three for high tests and three for low tests. The three high tests are:

1. CFYH, Classify High. This function is used to make tests on an input field for the purpose of determining whether the input record should be accepted. If the specified condition is satisfied, the Classification Table will be set accordingly. If the condition is not met, the table is not altered.

2. CFNH, Classify Not High. This function is used the same as CFYH, except that the condition is satisfied if the data field is not equal to the value or range specified.

3. RCYH, Reclassify High. This function is used to test a column of the table, and if the test is satisfied, to set the table accordingly. This function can be used to combine and interrelate tests. This function does not cause any resetting of the table. That is, it does not eliminate any integer from any column.

The three low tests are functionally identical to the high tests. The only distinction being the time at which the tests are made relative to the point at which the record is accepted. The low tests are conveniently called: CFYL, Classify Low; CFNL, Classify Not Low; RCFL, Reclassify Low. These tests may be used to determine whether formatting should be done, a line should be printed, or a hand calculation should be executed.

Functions Used to Specify Action

The previous discussion described how input fields can be classified and, accordingly, how the Classification Table can be set. The following discussion will show what actions can be specified as dependent upon these tests and how each is accomplished. As previously mentioned, the purpose of making high tests is to determine if the input record is wanted. In order to specify that this test for acceptance is to be made and what combination of classify high tests are to be used, a function called ACT is available. The ACT function has a record type and a Classification Table test configuration associated with it. This combination of parameters in effect means: test the table and if the test configuration is satisfied, accept the record; if at least one column does not contain its designated integer, reject the record. Only those columns are tested which are specified. As always, the tests on the table can be for one or more columns to not contain particular integers as well as to contain the integers.

When more than one column is specified, they are related in an "and" fashion. If an "or" condition is desired, RCYH can be used to rearrange the table. If the record is not accepted, no more will be done for the current packet. If another packet uses the same record type, control is passed to the section of the program generated by that packet; otherwise, another record is read. If the record is accepted, the contents of the Classification Table are unaltered and the low tests are started. All low testing is done before any subsequent action can be taken.

The low tests are most commonly used to determine whether a field should be formatted and, if so, how it should be formatted. The formatting of a field is accomplished by the association of the format field (variable field) with a particular input data field. The tests to be made to determine the necessary formatting are designated by the association of a Classification Table test configuration with the format field. The association of these three sets of information: a particular input field, a Classification Table configuration, and a particular format field, is basically all that need be specified to place input from the master file into a print line. If no Classification Table test configuration is included, the formatting will be done unconditionally. If a test configuration is included, the formatting is termed conditional formatting and requires that all the specified columns be satisfied in an "and" fashion. A second action which can be controlled by the tests is the printing of a line. This is accomplished by means of a

function called PRT. PRT must have a Classification Table test configuration and a format line associated with it. If this function is used, the Classification Table is tested after all formatting has been done and just prior to writing the print line to an output tape. If the table satisfies all the specified conditions, the line is written; otherwise, the line will not be written. The Classification Table can be used with other functions, such usage will be discussed when those functions are discussed.

Examples of the Use of the Classification Table

EXAMPLE 1

Record Type	Function or Input Field	Value Field or Field Name	Classification Table Columns 0 1 2 3 4 5 6 7	Format Field
10	0003	SALARY		D10026
10	CFYH	10000-20000	2	
10	CFNH	15000	8	
10	ACT		2 8	

These four lines of information will cause the program to examine field number 0003, which is contained in record type 10, and put a 2 into column 2 of the table if the salary found there is between 10000 and 20000, inclusive. A test will also be made to see if the salary is 15000; if so, an 8 will be inserted into column 3 of the table. After all the high testing is done, a test of the table will be made to see if there is a 2 in column 2 and an 8 in column 3. If such is the case, the record is accepted. Thus, records are being accepted for all employees having a salary of $10,000 to $20,000, inclusive, but not exactly $15,000.

A similar report could be generated by the following four lines.

Record Type	Function or Input Field	Value Field or Field Name	Classification Table Columns 0 1 2 3 4 5 6 7	Format Field
10	0003	SALARY		D10026
10	CFYL	10000-20000	2	
10	CFNL	15000	8	
10	PRT		2 8	D10

This method is not recommended, since all lines are formatted but only some are printed; excessive formatting is avoided by using the high tests as shown above.

EXAMPLE 2

Suppose it is desired to prepare a mailing list of all members of department 624 who have worked for the company for more than a year. Suppose also, that it is anticipated that some employees do not have a home address in their records, in which case, their business address is to be used. These conditions can be represented as follows:

Record Type	Function or Input Field	Value Field or Field Name	Classification Table Columns 0 1 2 3 4 5 6 7	Format Field
10	0023	NAME		D10024
10	0046	BUSINESS ADDRESS	1	D10050
10	0146	HOME ADDRESS	1	D10050
10	CFYL		1	
10	0007	DEPT		
10	CFYH	624	2	
10	0008	TIME/CO IN MOS.		
10	CFNH	000-212	2	
10	ACT		2 2	

Note that the functions above are not coded in the order in which they will be executed (and which was described previously). The only necessary order when coding these functions is that the classify functions relating to a specific data field must be coded immediately following the line which specifies that data field.

Emitting Data

The Reports Generator provides a facility for emitting actual data that may be used in the same manner as input data; this is accomplished through the EMIT function. The data obtained in this way can be used in the same manner as data from an input record, except that the print method used with the corresponding format field must be print method 0. This, in effect, means that any desired punctuation must be emitted along with the value. The grouping feature and the Classification Table can be used with the format field. For example, a series of EMITs may be specified; each EMIT may be dependent upon a different table setting. The effect is the same as a table look-up. Following is an example of the conditional use of the EMIT function.

Record Type	Function or Input Field	Value or Field Name	Classification Table Columns 0 1 2 3 4 5 6 7	Format Field
10	0024	STATE CODE		
10	CFYL	023	1	
10	CFYL	018	2	
10	CFYL	006	3	
10	CFYL	106	4	
10	CFYL	074	5	
10	EMIT	N. H.	1	D10015
10	EMIT	MASS.	2	D10015
10	EMIT	N. Y.	3	D10015
10	EMIT	CONN.	4	D10015
10	EMIT	ME.	5	D10015

The above coding will cause the abbreviation of a state name to be emitted, based upon the presence of a specific state code in the input record.

Specifying Associated Control Fields

Each total line must have a total level associated with it and each total level must have one or more associated input fields. These input fields are termed associated control fields for the total line and normally mean that the total line is to be printed each time the value in the field changes. The Reports Generator has extended this concept to apply to ranges of values rather than just individual values. That is, possible values of the associated control field can be grouped into ranges and the frequency of the total line can be controlled by a change of range. These changes in range are termed synthetic control breaks. The function used to define a range is CFYV, Classify Value; it has associated with it a range of values and an alphabetic character to identify the range. The function used to create synthetic control breaks is SCB. It is used to relate the set of alphabetic characters defined by the CFYV function with a total level. Only one synthetic control break is permitted per report packet.

Example:

Suppose it is desired to have a total of company expenditures on every quarter year rather than on every month. The control field might be specified as follows:

Record Type	Function or Input Field	Value or Field Name	SCB Col.	Total Level	Format Field
10	0126	EXPENDITURES			T20026
10	1318	MONTH OF EXP.			
10	CFYV	0-3	A		
10	CFYV	4-6	B		
10	CFYV	7-9	C		
10	CFYV	10-12	D		
10	SCB			2	

A monthly report would be controlled as follows:

Record Type	Function or Input Field	Value or Field Name	SCB Col.	Total Level	Format Field
10	0126	EXPENDITURES			T20026
10	1318	MONTH OF EXP.		2	

Additional Facilities

If a count of the number of records which meet certain selection criteria is desired, the COUN (Count) function may be used. A separtate COUN may be used for each desired set of selection criteria. The COUN function is not used if selection is not involved, since all records of each record type are automatically counted.

If desired, the PAGE function may be used to successively number pages. The PAGE function must be associated with a variable field of a page heading line; the first time the line appears the number 1 will be printed

in that variable field; each succeeding time that the line is printed, the field will be incremented by 1. The PAGE function may be reset to 1 at any total level by associating it with that total level.

The system may perform addition. That is, one field may be added to one or to several total line fields, or several fields may be added into the same total line field. All addition is algebraic. The system also provides for sign reversal; by reversing the sign of a field and adding fields together, subtraction is automatically performed. Fields may be added to and subtracted from the same field.

CHAPTER 5: CODING FORMS

Each Reports Generator report or record is specified by a single packet created from cards punched from a combination of the Reports Generator coding forms.

Coding Forms Used For Reports

The forms used to create a report packet are divided into four major groups as follows:

1. The A01 line.

2. The A02 line.

3. The Report Format lines.

4. The Field Parameter lines.

Types 1, 2, and 3 are prepared using the same coding form (see Figure 1); type 4 uses a separate coding form (see Figure 2). Cards prepared from the forms are ordered as above for input to the Reports Generator.

Each line of each form is punched into columns 1-72 of a single IBM card, with the exception of the format line cards. Format lines are punched as follows:

1. Column 1 through print position 59 are punched into columns 1-72 of the first card.

2. Columns 1-3 are duplicated from the first card and print positions 60-119 are punched into columns 13-72 of the second card.

3. Columns 1-3 are duplicated from the first card and print positions 120-132 are punched into columns 13-25 of the third card.

In all cases, column 4 must be used to sequence the cards. Columns 5-12 need not be punched in the second and third cards. The final card(s) need not be punched if there are no entries in the print positions.

All entries on the coding forms are right-justified unless otherwise specified; numeric fields may have leading zeros or blanks if desired.

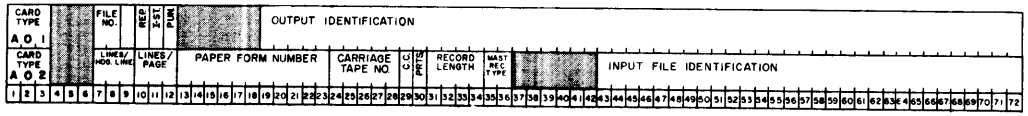
A01 Line

The A01 and A02 lines are used to signify to the Reports Generator that a new report description is to follow; information from these lines will be written either on-line or off-line by the Reports Generator in order to relate any error diagnostics to the correct packet. The decision of where to print the A0 lines or error diagnostics depends on information supplied in the *JOB card.

IBM

X28-6146-0
PRINTED IN U.S.A.

SHARE 7090 REPORTS GENERATOR
REPORT DEFINITION FORMAT



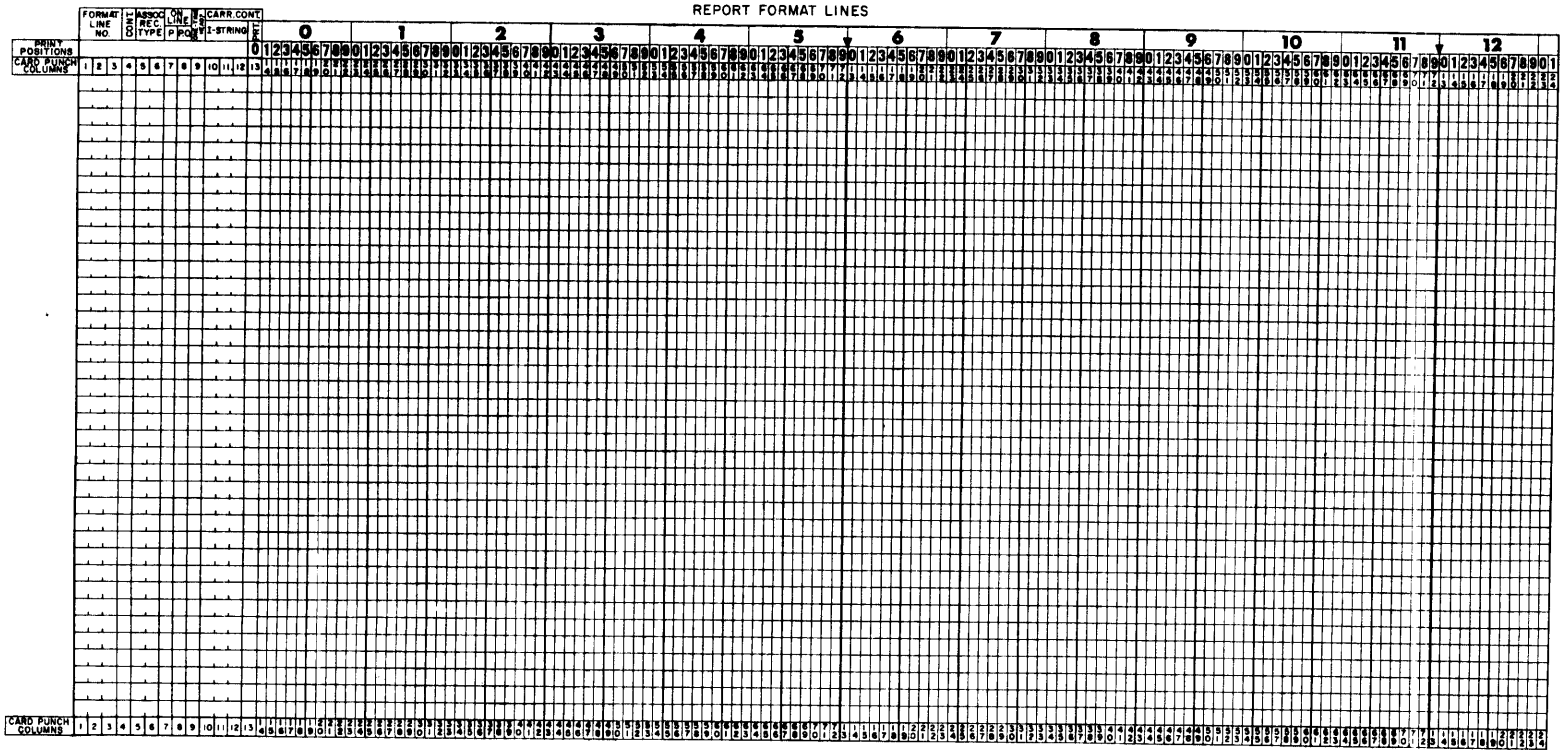


Figure 1. The Report Definition form

X28-6147-0
PRINTED IN U.S.A.

IBM

SHARE 7090 REPORTS GENERATOR
FIELD PARAMETERS

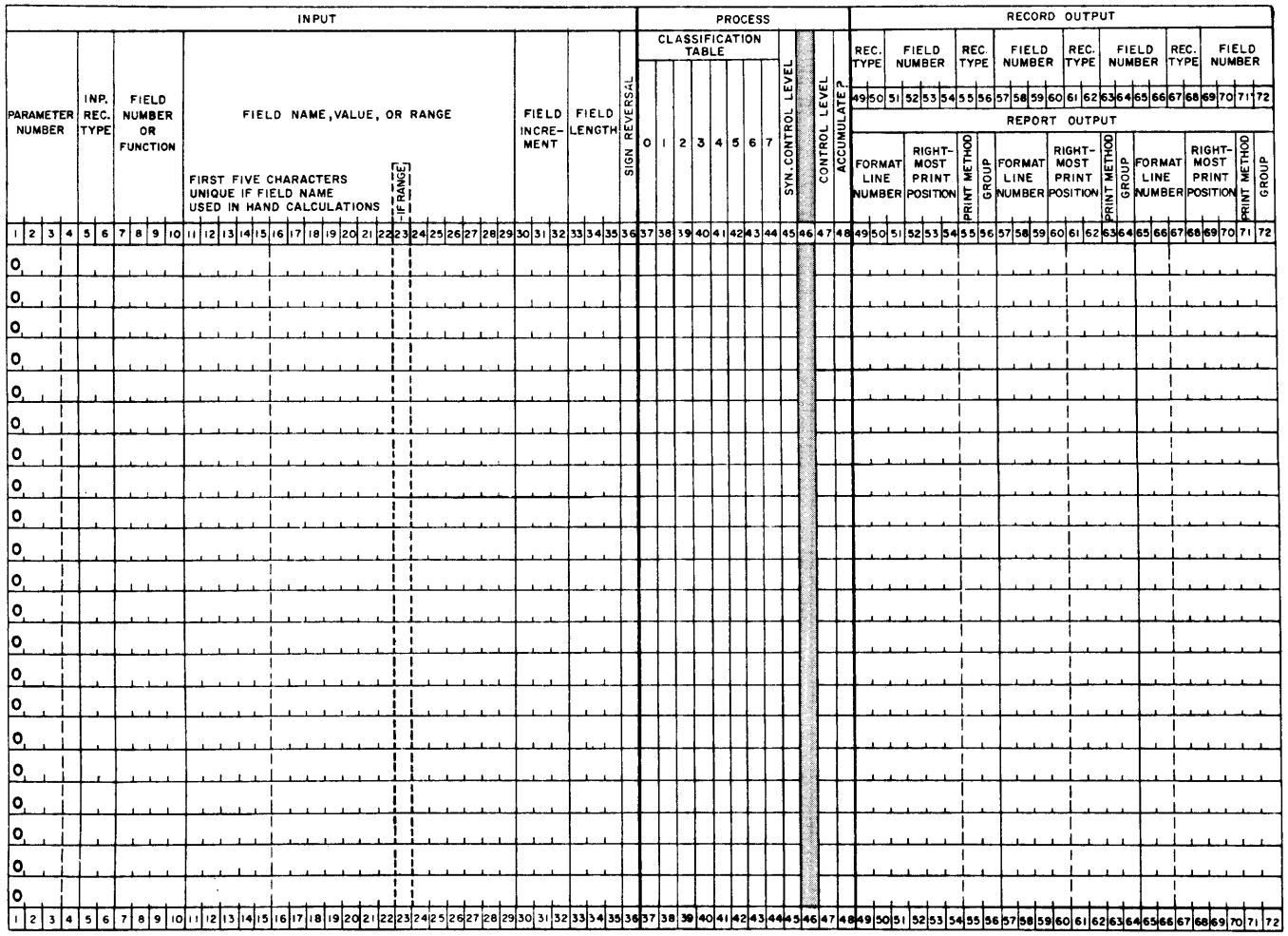
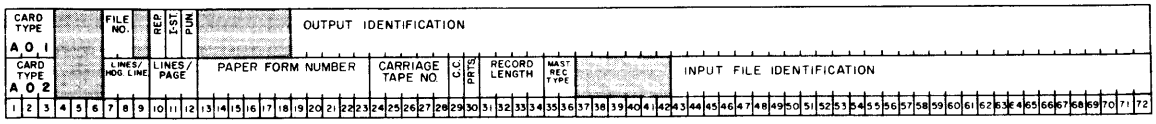


Figure 2. The Field Parameters form



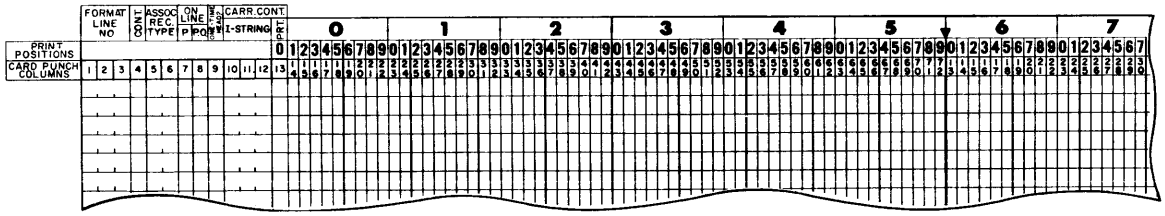
Columns/Contents		Description
1-3	Card Type	Must contain A01.
4-6		Not used.
7-8	File Number	Must contain a file number that is defined by an *FILE card or an S to mean use file number from previous packet, or an N to mean use the on-line printer (N not permitted for I-String output).
9		Must be blank.
10	Report Indicator	Must be non-blank to indicate report generation.
11	I-String Indicator	May be non-blank to indicate I-String output.
12	Punch Indicator	May be non-blank to indicate output is to be punched rather than printed.
13-18		Must be blank.
19-72	Output Identification	May contain any alphameric comments.

A02 Card

Columns/Contents		Description
1-3	Card Type	Must contain A02.
4-6		Not used.
7-9	Lines Per Heading Line	Must specify the number of lines to be printed or spaced before repeating detail line headers (HD-type lines).
10-12	Lines Per Page	For I-String output only; number of physical lines per document page.
13-23	Paper Form Number	May be used to indicate the paper stock on which the report is to be printed or punched.

Columns/Contents		Description
24-28	Carriage Tape Number	May be used to identify the carriage tape to be used with this job.
29	Carriage Control Setting	May contain S, D, or P, to indicate to the operator how the printer is to be set; this has no meaning with respect to I-String output.
30	Parts	May be used to indicate the number of copies desired.
31-34	Input Record Length	If the input file does not contain a dictionary, this field may be coded to indicate the record length in 7090 words; if blank or zero, 14 words will be assumed.
35-36	Master Record Type	May contain a record type number to designate a master record type for this report.
37-42		Must be blank.
43-72	Input File Identification	May contain any alphameric comments.

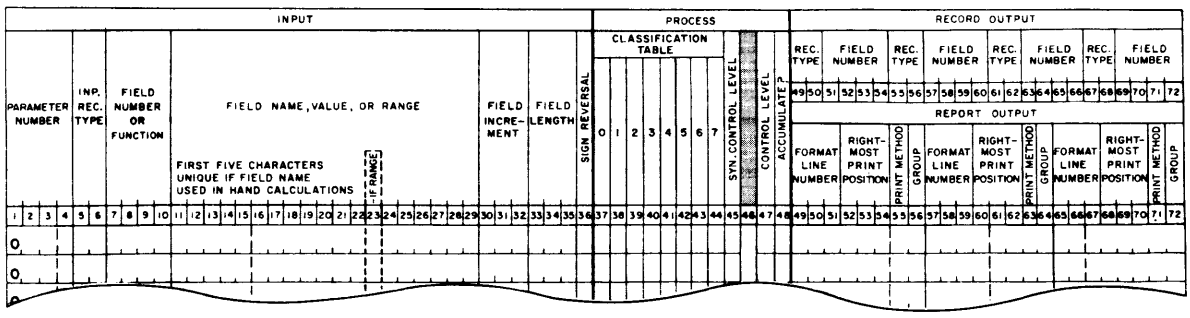
Report Format Lines



Columns/Contents		Description
1-3	Format Line	Must contain a designation in conformance with the following rules, where x always represents a sequence designation (0-9, A-Z) and y represents a control break level designation (1-9). 1. Detail line: Dx where x is a unique two character designation indicating the sequence of printing lines associated with the same record type. 2. Normal heading line: HDx where x is the sequence of printing heading lines. 3. Total-level heading line: Hyx where y is the level of total and x is the order of printing lines within that level. 4. Total lines: Tyx where y defines the level of total and x is the sequence of printing multiple lines for a single level. 5. Final total lines: TFx where x is the sequence of printing multiple lines for the final totals.
4	Continuation	Must contain 1, 2, or 3, depending as it is the first, second, or third card prepared from the coding line.

Columns/Contents		Description
5-6	Associated Record Type	For detail lines only, must contain the record type of the record associated with printing this line.
7-8	On-Line Print or On-Line Print Only	Not applicable to heading lines; if column 7 contains an 11-punch, this line will be printed on-line and written on-tape; if column 8 contains an 11-punch, this line will be printed only on-line; to write the line only on tape, both columns must be blank. On-line printing may be interspersed with system messages.
9	One Time Header	A 1 in this column, combined with an HDx in columns 1-3, will cause this line to be printed only once (at the beginning of the report). An A in this column, combined with a Tyx in columns 1-3, will cause printing of the HDx lines immediately after this level of total (and its related total lines) are printed; this feature may be used to obtain a skip to new page following a desired level of total.
10-12	I-String Carriage Control	For I-String output, may contain one of the following codes: Zx Skip to line x Hx Skip x lines Ibb Skip to new page bbb Single space where x is always a two digit number greater than zero, but less than or equal to the lines per page count from the A02 card.
13	Carriage Control Character	If column 29 of the A02 card contains a P, this is the carriage control character recognized by the printer and will not be printed.
14-72 & 13-72 of card 2, & 13-25 of card 3	Print Position Format	May include any literal and/or variable information prepared in accordance with the rules for specifying report format (see page 5). Where the second and third cards are used, columns 1-3 must be identical to the first card, column 4 must indicate continuation, and columns 5-12 may be blank. When specifying variable fields, a space to the right of the rightmost print position should be reserved for a minus sign when using print methods 1-4.

Field Parameter Cards



Columns/Contents		Description
1-4	Parameter Number	Column 1 must contain a 0; columns 2-4 may contain a sequence number; if columns 2-3 are identical to the preceding card, the card is considered to be a continuation card. Thus, column 4 may be used to sequence continuation cards. A maximum of 3 continuation cards may be used; these are used in order to provide for additional output fields. For continuation cards, only columns 1-4 and the output specification field beginning in column 49 need be coded.
5-6	Input Record Type	Must contain the record type of the input record which contains the field to be examined.
7-10	Field Number or Function	May contain either a function or the field number of the field to be operated upon; the field number must not be the same as a function name. If these columns contain a classify function (CFYH, CFNL, etc.) columns 5-6 are blank and the function relates to the preceding Field Parameter card which does specify a record type and field number. If the file does not have a dictionary, this field must contain the increment of the field within the logical record, in characters. The length will be specified elsewhere.
11-29 (35)	Field Name or Value	If columns 5-10 specify an input field, columns 11-29 may contain any alphameric description of the field. (The first 5 characters must be unique if the field is to be referenced in a hand calculation.) In the case of the CAL function, the first five characters must be a unique name by which the calculation may be referenced. If columns 7-10 contain a function, columns 11-22 may contain a single test value or these columns may be combined with an entry in columns 24-35 (through an 11-punch in column 23) to provide a range of test values. Columns 11-22 must contain a value algebraically less than that contained in columns 24-35; both entries must be left-justified; negative entries are indicated by an 11-over punch in the least significant position.
30-32	Field Increment	If the file contains a dictionary and it is desired to reference a partial field, these columns may contain the increment of a partial field within the input field, in bits.
33-35	Field Length	If columns 30-32 are used to specify the increment of a partial field, these columns must contain the length of that field in bits. If the file does not contain a dictionary (and 7-10 specify field increment), these columns must specify input field length in characters. If columns 7-10 contain the EMIT function, columns 33-35 must contain the length of the value in the value field (columns 11-29) in bits (which must be a multiple of 6).
36	Sign Reversal	The algebraic sign of a field may be changed by placing an R in this field; otherwise, this field must be blank.

Columns/Contents		Description
37-44	Classification Table	Each column may contain any digit from 1-9 or may be blank. For certain testing situations (when testing for the absence of a condition), any of the columns which contain a digit may also contain an 11-overpunch.
45	Synthetic Control Break	When used with the CFYV instruction, this column may contain an alphabetic character to define a subset of the sequencing information; subsequently, a synthetic control break will occur on change in alphabetic subset character rather than on any other change in sequencing information.
46		Not used.
47	Control Break Level	If it is desired to cause a control break on a change in the field specified in columns 5-10, this column may contain one of the digits 1-9, indicating the level of control break from lowest to highest. Any digit contained in this column must also be associated with at least one total line (i.e., it must be the second character of a total line number). Conversely, all total lines must be associated with a control level specified in this column. Thus, there is a one-to-one correspondence between a level number designation and usage of that level number. If more than one field uses the same control break, a change in sequencing information in any of these fields will cause printing of the same total lines. This column may also be used with the PAGE and COUN functions to cause resetting of the page number or count.
48	Accumulate	May contain a minus sign (11-punch) to cause accumulation (totaling) of this field separately, or may contain an alphabetic character to cause accumulation of this field with all other fields using the same alphabetic character. Accumulation is reset whenever the accumulated value is printed. (Prior to resetting, the value is added to the next higher level total accumulating the same data.)
49-54	Format Line Number and Rightmost Print Position	If the preceding columns describe a field to be output, then columns 49-54 must specify a format line number and rightmost print position of a variable field into which the data is to be inserted. If the data is to be inserted into more than one field of the same output line, a separate Field Parameter card is required for each insertion. If the preceding columns describe a hand calculation to be performed (CAL function), then the insert point for the hand calculations must be written beginning in column 49.
55	Print Method	If a variable field is specified in columns 49-54, this column must contain a print method specification for the variable field (if this field is left blank, print method 0 is used).

Columns/Contents		Description
56	Group Indicator	If the Field Parameter card describes a field to be entered into a detail line, and the detail line is the lowest level of the detail lines associated with the same record type, this column may contain a non-blank character to indicate that the field is to be inserted only into the detail line when it immediately follows a heading or total line. If printing of the line is suppressed, the field indicated for grouping will not be inserted until after the next heading or total line is printed.
57-72		Same as columns 49-56, but for second and third insertion of same field (which, if present, must be into different format lines).

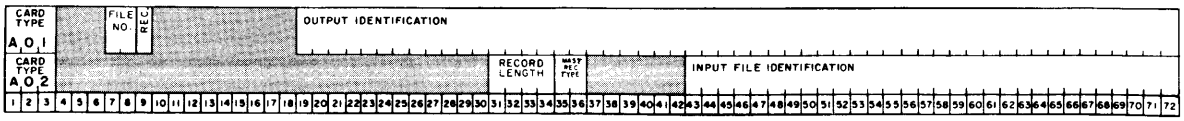
Coding Forms Required For Record Generation

The record definition forms are divided into four major sections as follows:

1. The A01 line.
2. The A02 line.
3. The Dictionary Definition form.
   a. E Record. This defines the record format.
   b. F Record. This defines a field of a record.
4. The Field Parameter cards.

Items 1, 2, and 3 occur on the same coding form (see Figure 3), while item 4 occurs on a separate coding form (see Figure 2, page 21). Cards punched from these coding forms must be arranged in the above order to create a packet which will in turn become a program to prepare records. Each line on the coding forms is punched into columns 1-72 of an IBM card; all entries are right-justified unless otherwise specified; numeric fields may have leading zeros or blanks.

A01 Line



Columns/Contents		Description
1-3	Card Type	Must contain A01.
4-6		Not used.
7-8	File Number	Must contain a file number that is defined by an *FILE card or an S, which means use the file number of the previous packet.
9	Record Indicator	Must be non-blank to indicate record generation.
10-18		Must be blank.
19-72	Output Identification	May contain any alphameric comments.

A02 Line

Columns/Contents		Description
1-3	Card Type	Must contain A02.
4-30		Not used.
31-34	Record Length	May contain the record length of the input record for a file without dictionary; it must be in 7090 words and must be right-justified. If blank or zero, 14 words will be assumed. These columns must be blank for files which contain a dictionary.
35-36	Master Record Type	May contain the record type of a record to designate it as the master record type.
37-42		Must be blank.
43-72	Input File Identification	May contain any alphameric comments to describe the input file.

X28-6148-0
PRINTED IN U.S.A.

IBM

SHARE 7090 REPORTS GENERATOR
DICTIONARY DEFINITION

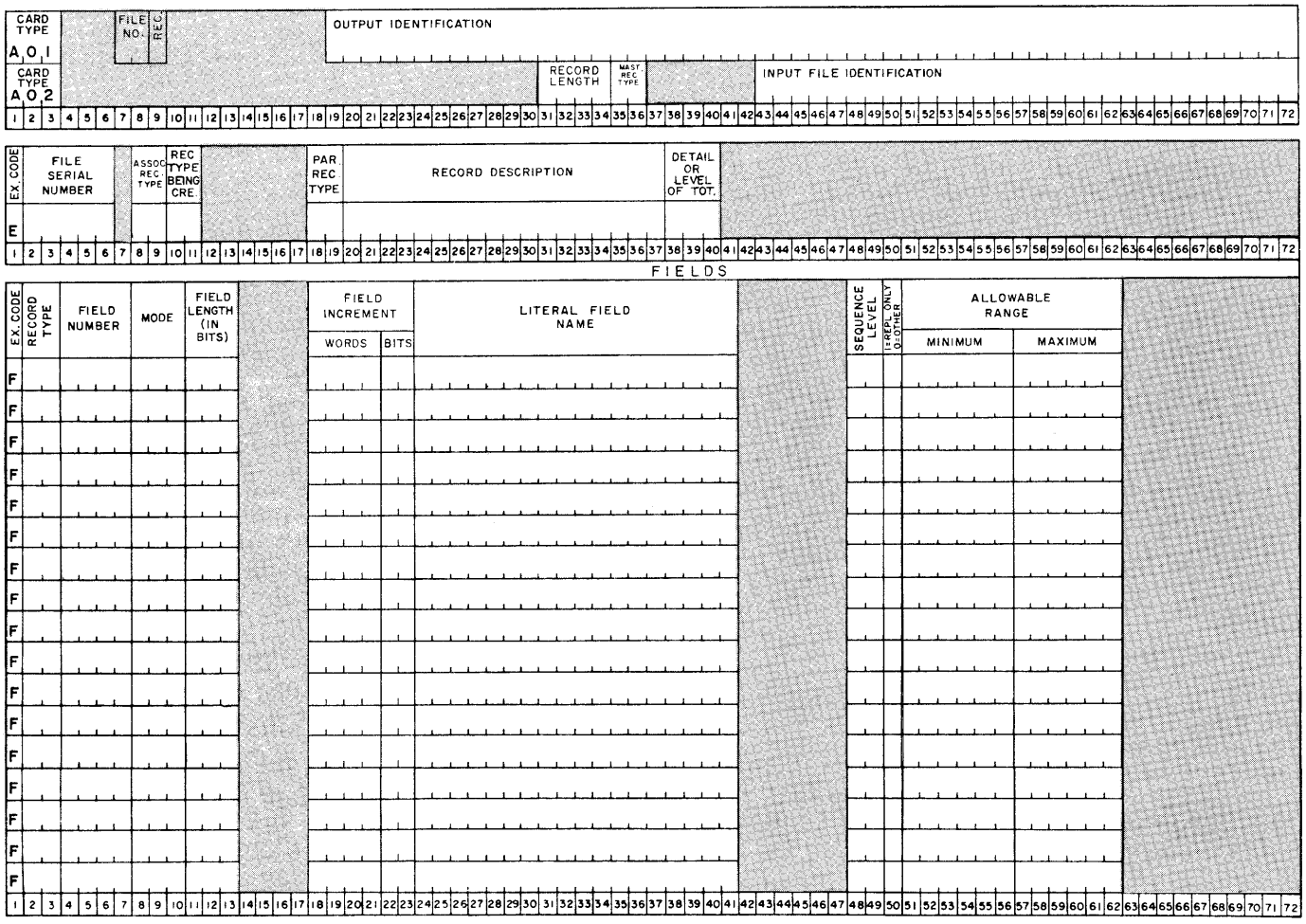
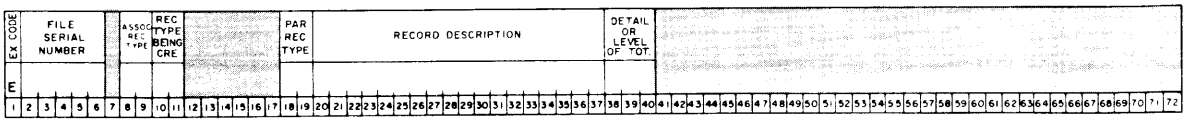


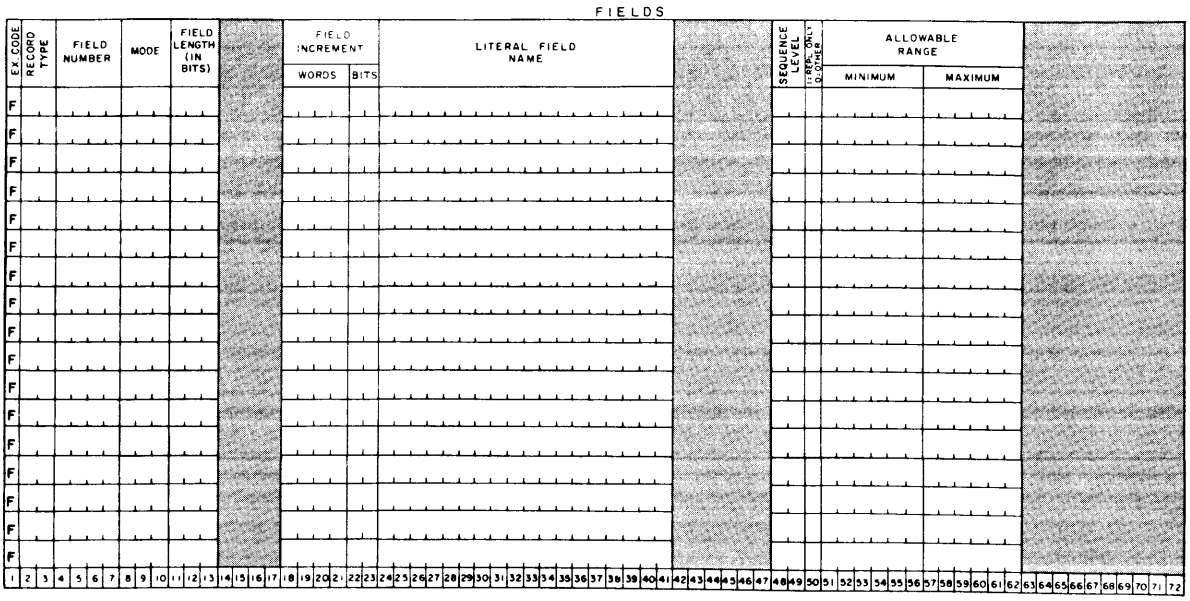
Figure 3. The Reports Generator Dictionary Definition form

E Record



Columns/Contents		Description
1	Executive Code	Must contain an E.
2-6	File Serial Number	Must contain the file serial number (same as columns 44-48 of the *FILE card) of the output file.
7		Must be blank.
8-9	Associated Record Type	Must contain, for each detail record to be created, the record type of the record associated with writing this record. In general, it is the lowest level input record type from which data is to be taken to create the output detail record type.
10-11	Output Record Type	Must contain the record type of the output record being created.
12-17		Must be blank.
18-19	Parent Record Type	If a detail record is being defined, this field may designate the record type which is to be the parent of the record being defined; this information is not used by the Reports Generator.
20-37	Record Description	May contain any desired alphameric information.
38-40	Detail or Level of Total	(Same as format line number for reports.) Column 38 must contain one of the following codes to designate whether the output record is a detail or total record: D Detail T Total For total records, column 39 must contain a total level designation and column 40 may be used to sequence lines within a total level. For detail records, columns 39-40 may be used for sequencing. None of the three columns may be blank.
41-72		Not used.

F Record

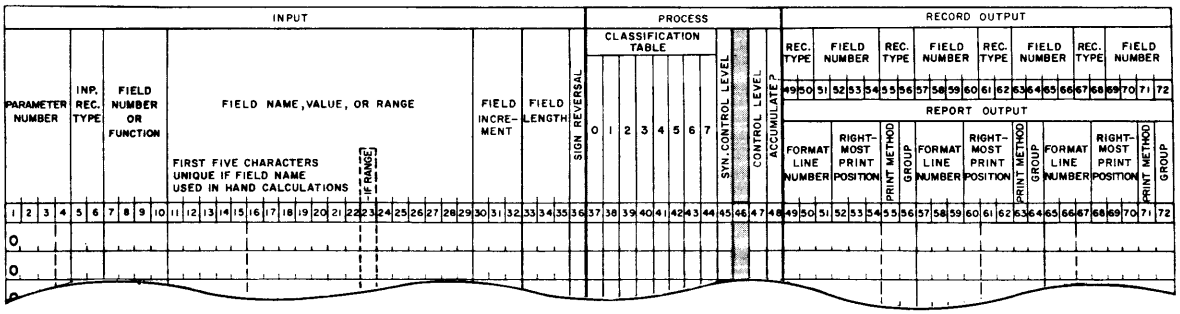


Columns/Contents		Description
1	Executive Code	Must contain an F.
2-3	Record Type	Must contain the record type within which the field is to be located, and must be the same as the entry in columns 10-11 of the preceding E record.
4-7	Field Number	Must contain the four numeric digit name of this field within the record type; this number must be unique within the record type and must not have leading blanks.
8-10	Field Mode	Must contain one of the following codes to designate the mode of the field being output: 100 Unpacked BCD 110 Packed BCD 001 Unpacked Binary
11-13	Field Length	Must contain the length of the output field in bits (right-justified with leading zeros).
14-17		Must be blank.
18-23	Field Increment	May contain the number of whole words (in 7090 words) and partial words (in bits), which precede the field, from the beginning of the record. Entries must be right-justified with leading zeros. If this field is left blank, the Reports Generator will

Columns/Contents		Description
		automatically assign the locations of the fields within the record. All F cards for the same record type must have this field coded or else all must have it blank.
24-41	Field Name	May contain any alphameric information.
42-47		Must be blank.
48-49	Sequence Level	May contain a sequence level for the field, beginning at 01 and increasing to succeeding lower levels of sequence.
50	Replace Only	May contain one of the following codes to indicate how the field may subsequently be modified during a File Processor run: 0 Any type change acceptable 1 May be replaced only
51-62	Allowable Range	May contain a minimum and maximum permissible value for the field; entries are right-justified with leading zeros and are tested against the high-order positions of the field. Either or both values (minimum or maximum) may be left blank, in which case, testing does not occur.
63-72		Must be blank.

Note: Columns 48-62 need only be coded if the file will subsequently be used by the File Processor.

Field Parameter Form



Columns/Contents		Description
1-4	Parameter Number	Column 1 must contain a 0. Column 2, 3, and 4 may contain a sequence number; if columns 2 and 3 are identical to the preceding card, the card is considered to be a continuation card. Column 4 may be used to sequence continuation cards. Continuation cards are used to specify additional output fields, and up to 3 continuation cards are permitted; each continuation card must be punched in columns 1-4 but may be blank in columns 5-48.

Columns/Contents		Description
5-6	Input Record Type	Must contain the record type of the input record which contains the field to be examined.
7-10	Field Number or Function	May contain either a function or the field number of the input field to be operated upon; the field number must not be the same as a function name. If these columns contain a classify function (CFYH, CFNL, etc.), columns 5-6 are blank and the function relates to the preceding Field Parameter card which does specify a record type and field number. If the file does not have a dictionary, this field must contain the increment of the field within the logical record, in characters.
11-29(35)	Field Name or Value	If columns 5-10 specify an input field, columns 11-29 may contain any alphameric description of the field. (The first 5 characters must be unique if the field is to be referenced in a hand calculation.) In the case of the CAL function, the first five characters must be a unique name by which the calculation may be referenced. If columns 7-10 contain a function, columns 11-22 may contain a single test value or these columns may be combined with an entry in columns 24-35 (through an 11-punch in column 23) to provide a range of test values. Columns 11-22 must contain a value algebraically less than that contained in columns 24-35; both entries must be left-justified; negative entries are indicated by an 11-overpunch in the least significant position.
30-32	Field Increment	If the file contains a dictionary, and if it is desired to reference a partial field, these columns may contain the increment of a partial field within the input field, in bits.
33-35	Field Length	If columns 30-32 specify the increment of a partial field, these columns must contain the length of that field in bits. If the file does not contain a dictionary and columns 7-10 specify a field increment, these columns must specify the input field length, in characters. If columns 7-10 contain the EMIT function, columns 33-35 must contain the length of the value in the value field (columns 11-29) in bits (however, this must be a multiple of 6).
36	Sign Reversal	The algebraic sign of a field may be changed by placing an R in this field; otherwise, this field must be blank.
37-44	Classification Table	Each column may contain any digit 1-9 or may be blank. For certain testing situations (when testing for the absence of a condition), any of the columns which contain a digit may also contain an 11-overpunch.
45	Synthetic Control Break	When used with the CFYV function, this column may contain an alphabetic character to define a subset of the sequencing information; subsequently, a synthetic control break will occur on a change in alphabetic subset character rather than a

Columns/Contents		Description
		change in any other sequencing information.
46		Not used.
47	Control Level	If it is desired to cause a control break on a change in the field specified in columns 5-10, this column may contain one of the digits 1-9, indicating a level of control break from lowest to highest. Any digit contained in this column must also be associated with at least one total line (i.e., it must be the second character of a total line number). Conversely, all total lines must be associated with a control level specified in this column. Thus, there is a one-to-one correspondence between the designation of a control level number and the use of that number. If more than one field uses the same control break, a change in sequencing information in any of these fields will cause printing of the same total lines. This column may also be used with the PAGE and COUN functions to cause resetting of the page number or count.
48	Accumulate	May contain a minus sign (11-punch) to cause accumulation (totaling) of this field separately, or may contain an alphabetic character to cause accumulation of this field with all other fields using the same alphabetic character. Accumulation is reset whenever the accumulated value is printed. (Prior to resetting, the value is added to the next higher level total accumulating the same data.)
49-54	Record Type and Field Number	If the preceding columns describe a field to be output, these columns must specify the output field to be used; if the preceding columns specify a hand calculation, these columns may specify an output field for the calculation results or columns 49-51 may specify a line number (same as columns 38-40 of an E card) with which the calculation is to be associated.
55-72		Same as 49-54, but for second, third, and fourth insertion of the same field (which if present, must be into different format lines).

CHAPTER 6: REPORTS GENERATOR HAND CALCULATIONS

The basic automatic properties of the Reports Generator can be augmented by the insertion of handwritten programs called hand calculations.

Hand calculations are coded in the SCAT language (see the SOS Reference Manual), in accordance with the rules, and using the additional language facilities, given in this chapter. The instructions themselves are written on the standard SHARE 709 Symbolic Coding form; each line must have a C in column 1. The calculation cards must follow all other cards for the report or record.

In order to indicate where a calculation is to be inserted and whether the result is to go into an output field, a Field Parameter card is prepared as follows:

1. CAL must be coded in columns 7-9 of the Field Number or Function field.

2. Columns 11-22 must contain the name of the calculation to be performed, the first five characters of which must be a unique name by which the calculation may be referenced.

3. The contents of columns 49-54 will vary depending upon the use of the hand calculation:

    a. If the results of the calculation are to be placed in a format field (or in an output record), these columns must specify the line number and rightmost print position of the field (or record type and field number).

    b. If the calculation is to be inserted into the coding which prepares a print line (or output record), the columns need only contain the line number.

    c. If the calculation is to be inserted at some other point within the generated coding, these columns must contain the symbolic insert point.

The symbolic designations are listed below, and must be left-justified in the field.

Calculation Insert and Reference Points

Given below are lists of the insert and reference points which may be used in Reports Generator calculations. Reference points are symbolic locations in the generated object program or are names of subroutines. Most insert and reference points are illustrated in the object program flow charts contained in Appendix A.

The following key will be used in representing variable system symbols:

y Must be a two digit input record type.

r May be a two digit report number (each report in any generation is numbered consecutively starting at 01) or may be coded as RN to designate the packet in which the hand calculation is contained.

x Must be a two digit number corresponding to the second and third characters of a line format number used by a report line or output record.

n Must be a two digit file number from 00 to 15.

Insert Points

Symbol	May Insert Be Conditional?	Location in Object Program
#yAr	No	Follows reading of input record.
#yCr	No	Precedes classify record section.
#yGr	Yes	Follows classify field section and precedes control break section.
#yIr	Yes	Follows control break section and precedes the formatting of the detail line.
Gx	Yes	Precedes the formatting of a group-indicated detail line.
Dx	Yes	Precedes the formatting of a detail line.
#yNx	Yes	Follows the section which formats the detail line.
Tx	No	Precedes the formatting of a total line.
Hx	Yes	Precedes the formatting of a header line.

Reference Points

Symbol	May symbol be referenced from a subsequent hand calculation in the same report packet?	May symbol be referenced from hand calculations in other report packets?	Description
#yAr	Yes	Yes	First location of the section of the program which processes the record type specified for the report number specified.

Symbol	May symbol be referenced from a subsequent hand calculation in the same report packet?	May symbol be referenced from hand calculations in other report packets?	Description
#yCr	No	No	Precedes the section of program which classifies on the specified record type.
#yGr	No	Yes	Follows the classify section and precedes the control breaks.
#yIr	Yes	Yes	Follows the control break section and precedes the section which formats detail lines.
#yFr	No	Only if grouping is used in report being referenced	Follows the section which formats fields specified for grouping.
#yPr	No	Only if there is a single detail line for the associated record type being referenced	Follows formatting of the detail line.
#yZr	Yes	Yes	Follows formatting of all detail lines for the record type and report number specified.
#T+Hr	No	No	Follows the writing of each total or heading line.
#$EOr	Yes	Yes	Location transferred to for end-of-job condition. Precedes writing of all final total lines for designated report.
##READ	Yes	Yes	Routine to get next logical input record.
##WRy	Yes	Yes	Subroutine to output current record for report specified. Note: If reports are stacked, the record that is written is the current record for the output unit rather than for the report.

Symbol	May symbol be referenced from a subsequent hand calculation in the same report packet?	May symbol be referenced from hand calculations in other report packets?	Description
####r	Yes	Yes	Address contains the location of the first word of the current output record for the report specified.
00##y	Yes	Yes	Address contains the location of the first word of the current input record of the type specified.
##INPT	Yes	Yes	Address contains the location of the first word of the current input record (independent of type).
BUFLOC			Location of a 20 word table which gives the location of a buffer for each I/O unit used. The cell for unit n is located at BUFLOC-n.
BUFSZE			Address of BUFSZE-n contains the size of the buffer for file number n.
FILEn			Location of File Control Block for file number n.

Symbols which contain a report number (r) may be referenced from one packet to another as follows:

1. The symbol must be present in the program for the packet being referenced; it will be only if the operation which the reference point precedes or follows is present.

2. In the reference, r must be replaced by the report number of the report in which the symbol is to be located. (When the current packet is being referenced, r may be replaced by RN.)

Example:

If the second report in a Reports Generator job contains instructions which cause record type 20 to use the Classification Table, a hand calculation contained within another report may use the symbol #yCr to reference the point preceding the classification in report number 2, as follows

#20C02

Note: In some instances, a Reports Generator symbol may actually occur several times in the same packet; reference to this symbol in a hand calculation will refer to the first appearance of that symbol. An example may occur with the system symbol #yPr, which follows the formatting of each D line; this symbol would be identical for all D lines in the same report which have the same associated record type.

IOCS Subroutines

The following Input/Output Control System (IOCS) subroutines may be used:

DEFINE	REW
JOIN	WEF
ATTACH	BSR
CLOSE	BSF
OPEN	CKPT
READ	STASH
WRITE	MWR
COPY	

These subroutines are discussed in the reference manual IBM 709/7090 Input/Output Control System, Form C28-6100-1.

The CALC Pseudo-Instruction

The CALC pseudo-instruction is used to relate the calculation cards which follow to the calculation referenced by the Field Parameter card. Columns 7-10 of the first card for each calculation must contain CALC; the address field of that card must contain the calculation name as defined in columns 11-15 of the Field Parameters form.

Location Symbols

With the exception of location symbols for associated storage, described below, location symbols for Reports Generator calculations are formed in accordance with the normal SCAT rules, modified as follows:

1. Location symbols may not be more than five characters in length.

2. Location symbols for other than associated storage must not begin with a C.

3. The C in column 1 is not considered to be part of a symbol; it is a flag only.

4. Location symbols may not be referenced outside the immediate calculation and need not be unique except within the immediate calculation.

Associated Storage Symbols

Associated storage is a block of storage which contains constants, temporary storage locations, masks, and other related out-of-line references. Associated storage locations may be referenced by any calculation routine

in the report packet and must have symbols formed in accordance with the following:

1. Each instruction defining an associated storage location must have a unique location symbol of the form:

   C00x

   where x is a two digit number.

2. An instruction defining an associated storage location may not have a symbol beginning with # in its address.

Examples:

```
C0001          PZE          11
C0009          BCI          2,SAMPLEbENTRY
C0002          DEC          1
```

In addition, the following rules apply:

1. Each hand calculation is limited to 500 words of "C" constants. Expressions of the nature

```
C0005          BSS          100
C0200          BCI          8,yy...yy
```

   count against the allowable "C" constants (in these examples, 100 and 8 words respectively).

2. Each hand calculation is limited to 100 symbols including "C" symbols.

3. No associated storage location may be referenced relative to another storage symbol.

Operation Codes

Most 709/7090 operation codes, and most SCAT pseudo-operations and prefix codes, may be used in Reports Generator hand calculations; a complete list of the codes which may be used is given in Appendix C.

Reports Generator Macro-Instructions

Reports Generator provides three macro-instructions which may be used in hand calculations.

GET

General Form	Examples
"GET x" where x is the first 5 characters of the field name as defined on the Field Parameters form	GET HOURS GET RATEb GET PRICE

The GET macro-instruction will operate as follows:

1. If the calculation is at the detail level, the field will be obtained and placed in the BCD or binary buffer, depending as the field is BCD or binary.

2. If the calculation is inserted at the total level and the Field Parameter card on which the field name was defined is indicated for accumulation, the field will be obtained and placed in the binary buffer.

GET may destroy the previous contents of the MQ if the field being obtained is greater than six BCD characters in length.

SET

General Form	Examples
"SET Lx" or "SET Rx" where L represents the leftmost four columns of the Classification Table, R represents the rightmost four columns, and x is a four numeric digit designation to which these columns are to be set (except that 0 does not cause a change in the table).	SET L5000 (puts a 5 in column 0 of the Classification Table) SET R9000 (puts a 9 in column 4 of the Classification Table)

The SET macro-instruction provides a method of inserting a value in the Classification Table. SET may be used to turn off an entry in the Classification Table by placing a minus sign over the digit to be set; however, it is not possible to turn on and turn off indicators with the same SET.

When the SET macro-instruction is used, the original contents of the AC are lost; the index registers and the MQ are unchanged.

TEST

General Form	Examples
"TEST Lx" or TEST Rx" where L indicates the four leftmost columns of the Classification Table, R indicates the four rightmost characters, and x is a four numeric digit test designation.	TEST L5000 (test column 0 of the Classification Table to see if it contains a 5) TEST R0030 (test column 6 of the Classification Table to see if it contains a 3)

The TEST macro-instruction is provided for testing the Classification Table in the same manner that the SET macro-instruction is provided for setting that table. A test may be made for a condition that is not to be met by placing a minus sign over the digit representing the column to be tested; however, it is not possible to test for a condition to be met and a condition not to be met in the same test instruction.

When TEST is used, the next location is the negative return, and the location after that is the positive return. The TEST macro-instruction destroys the settings of the internal indicators.

Addresses

The address portion of Reports Generator hand calculations follows the normal SCAT language format; that is, an address, tag, and decrement may be specified, and are separated from one another by commas. SCAT conventions are observed in assigning values to fields designated as negative. For example, PZE -4 would result in generation of 000000077774_8. Symbolic addresses (NEXT) are permitted, and relative addresses (*, *+1, NEXT+1) are also permitted. Master file fields may be referenced by means of the first five characters of the field name specified on the Field Parameter form. Symbolic tags and decrements are not permitted.

The BCD and Binary Buffers

The binary buffer, BINBUF, is one word. The BCD buffer, BCDBUF, is 14 words and is stored backwards in core storage. The last word of the BCD buffer is BCDBUF; the next to last word is BCDBUF-1; the first word is BCDBUF-13. The last word of a field is placed in the last word of the buffer (BCDBUF); the next to last word of a field is placed in the next to last word of the buffer (BCDBUF-1); etc.

Example:

Symbolic Location	BCDBUF-3	BCDBUF-2	BCDBUF-1	BCDBUF
Sample Contents	bbJOHN	bPAULb	JONESb	&bCObb

The contents of the binary and BCD buffers are not saved outside of the hand calculation in which they are used.

Reports Generator Subroutines

The following subroutines may be used in Reports Generator hand calculations. If used, these subroutines may destroy the contents of the AC and the MQ. Reports Generators subroutines may be referenced by means of a TSX instruction; these subroutines are always in storage and available for use.

BCDBIN

This subroutine may be used to convert one or two BCD words to one binary word, in accordance with the following:

1. Store the BCD words to be converted, right-justified, in locations BCDBUF-1 and BCDBUF.

2. Execute the following calling sequence:

```
TSX     BCDBIN,4
PZE     x
```

   where x is 0 if one word is to be converted and is not 0 if two words are to be converted.

3. The binary word will be stored in location BINBUF.

BINBCD

This subroutine may be used to convert one binary word to two BCD words, in accordance with the following:

1. Store the binary word to be converted in location BINBUF.

2. Execute the following calling sequence:

```
TSX     BINBCD,4
```

3. The BCD words will be right-justified in locations BCDBUF-1 and BCDBUF.

Sequence of Hand Calculations

If more than one hand calculation is to be performed at the same insert point, the hand calculation routines must be in the order in which it is desired that they be executed.

To save generation time, the calculation routines should be ordered in the same sequence as the insert points are ordered in the list on page 36.

Machine Indicators

If index registers, sense indicators, or other machine registers (except the AC and the MQ) are to be used by a hand calculation, the hand calculation must make provision for saving and restoring these registers.

Share 709 Symbolic Coding Form

The SHARE 709 Symbolic Coding form is shown in Figure 4; this form is used according to the normal SCAT rules, with the additional requirements that each line must contain a C in column 1 (this column is not considered part of the location field) and that the variable field must begin in column 16.

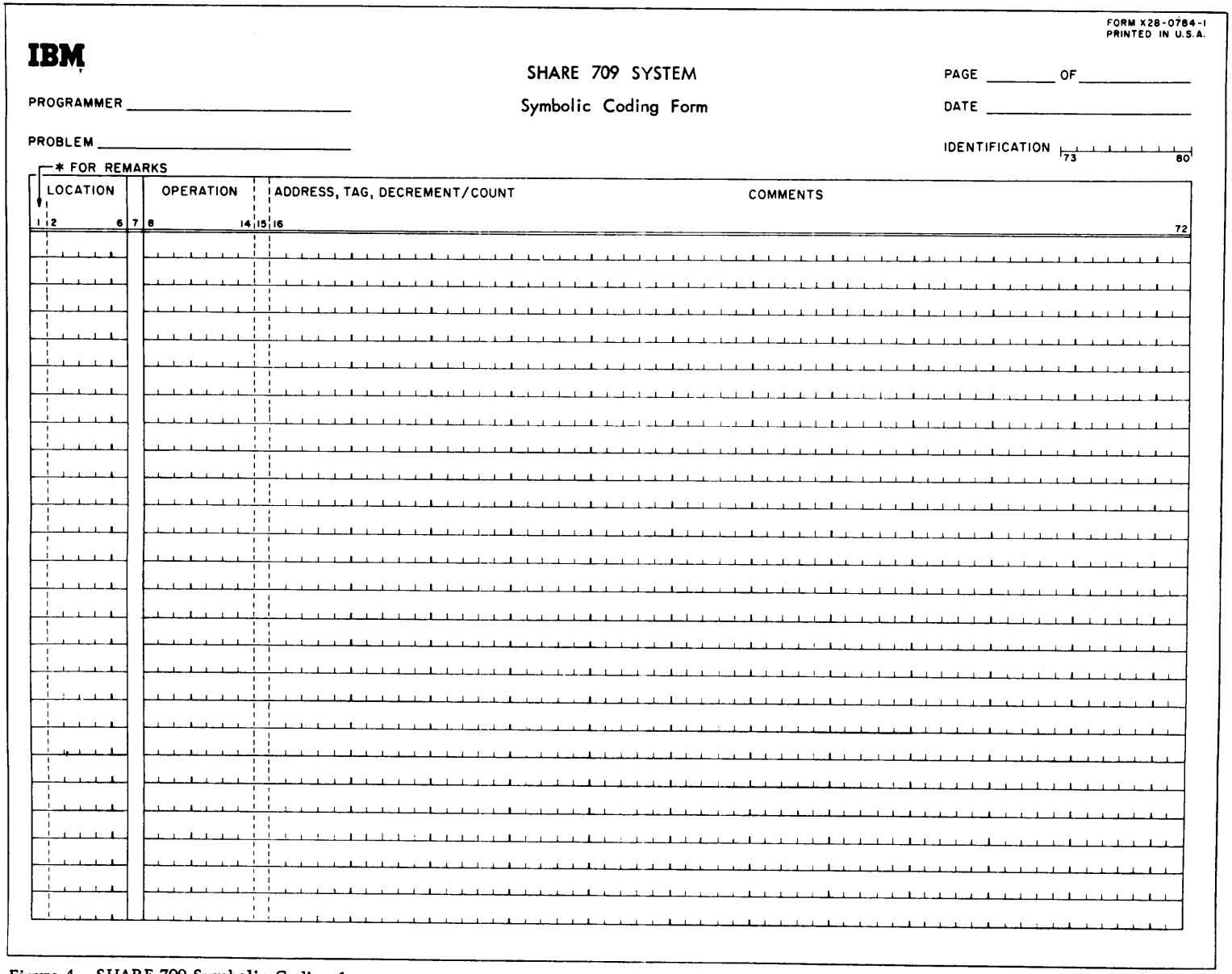


Figure 4. SHARE 709 Symbolic Coding form

CHAPTER 7: DIAGNOSTIC AIDS

As an aid to diagnosing Reports Generator programs, this chapter gives the general format of a generated object program (which is supplemented by object program flow charts in Appendix A) and a discussion of Reports Generator diagnostic error messages.

General Format of a Generated Object Program

The following discussion describes the relationship among packets in a single Reports Generator job and the relationship of a line in one packet to other lines in the same packet and in other packets. When used in conjunction with the object program flow charts given in Appendix A, this information will be useful in analyzing object program core dumps.

The object program is generated by packet, and within a packet by associated record type, and within an associated record type by line number.

Heading and total lines are formatted by subroutines which are located following the coding for all detail lines.

The Classification Table is set before the detail lines are formatted.

The write subroutines (one per packet) are generated after the packets are processed and in the same order as the packets are processed.

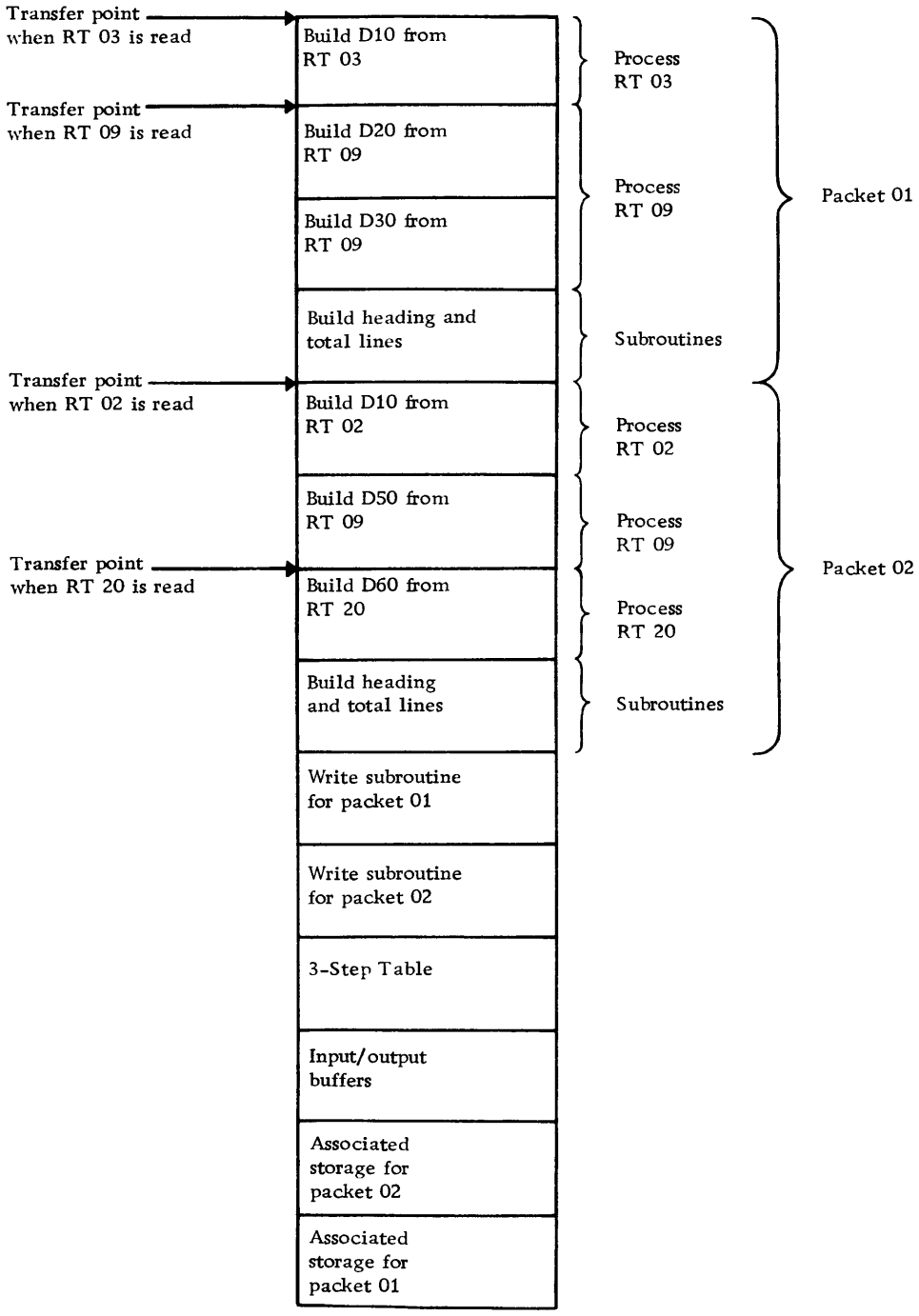
The 3-Step table is located after the write subroutines and is used to determine the transfer point for each input record type. As each input record is read, transfer is made to the first packet using that input record type. After a packet is through using a record, transfer is made to the next packet using that record type. When all packets are through using an input record, transfer is made to the routine that reads the next record.

Example:

Assume that packet 01 has a D10 line for input record type 03, a D20 line and a D30 line for input record type 09, and various heading and total lines.

Assume also that packet 02 has a D10 line for input record type 02, a D50 line for input record type 09, a D60 line for input record type 20, and various heading and total lines.

The following diagram indicates most of the major blocks of coding which would be generated:


Transfer point when RT 03 is read
Transfer point when RT 09 is read
Transfer point when RT 02 is read
Transfer point when RT 20 is read
Build D10 from RT 03
Build D20 from RT 09
Build D30 from RT 09
Build heading and total lines
Build D10 from RT 02
Build D50 from RT 09
Build D60 from RT 20
Build heading and total lines
Write subroutine for packet 01
Write subroutine for packet 02
3-Step Table
Input/output buffers
Associated storage for packet 02
Associated storage for packet 01
Process RT 03
Process RT 09
Subroutines
Process RT 02
Process RT 09
Process RT 20
Subroutines
Packet 01
Packet 02

Unconditional System Messages

Unconditional system messages and object program logical counts are written either on-line or on the system diagnostic tape, depending upon whether there are entries in columns 24 and 25 of the *JOB card. If these columns are blank, the following messages will be written on the diagnostic tape; if non-blank, they will be written on-line:

1. The A01 and A02 cards for each packet.

2. The END OF PHASE 1 message. (The two-phase generation process is described in an appendix in Part 1.)

3. Messages which indicate the core locations used by each report number, as follows:

   REPORT XX USES LOCATIONS XXXXX THRU XXXXX

4. The END OF PHASE 2 message.

5. End-of-job messages which indicate the number of records of each record type used by each report number, as follows:

   REPORT XX USED XXXXXX RECORDS OF RECORD TYPE XX

6. A message that indicates the total number of input records used by all reports in the job, as follows:

   TOTAL NO OF RECORDS USED XXXXXX

7. Messages that indicate the number of logical records written to each symbolic output tape, as follows:

   TOTAL NO OF LOGICAL RECORDS WRITTEN TO XX XXXXXX

Error Conditions and Messages

Error conditions may or may not cause generation to stop immediately; however, if errors are detected, execution is not advisable and the operator will be so notified by a system error message.

In addition to the errors which produce standard error messages (see below), other errors may be detected by the system. These fall into two groups:

1. Those which the system may be able to correct (usually parameter coding errors). An error message will be printed which will describe the error condition and which may describe the action taken by the system. These will not prevent execution of the object program.

2. Those which the system cannot correct (usually because the logic of the program does not conform to the logic of the system) or those caused by exceeding the capabilities of the system. These will result in an error message; execution is not advisable and may be impossible.

Frequently, error messages contain additional diagnostic information, such as the contents of tables, the parameter card at which the error was detected, etc.

Standard Error Messages

Certain error condtions produce the following type of error message:

ERROR CONDITION XXX

XXX is the number of the error condition; these numbers are listed below with an explanation of the error cause. Error numbers that are followed by an asterisk (*) are system or machine errors; they cannot be caused by the programmer.

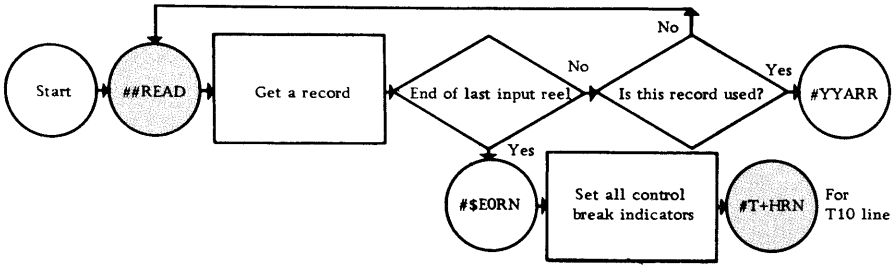
Error Condition	Cause
202*	The first record on the scratch tape for the program should have been the Each Report Table, but was not.
203*	Same as 202 but for Line Format Table.
204*	Same as 202 but for Control Break Table.
206*	The associated record type for the current entry in the Parameter Table is less than that of the previous entry.
210	There is an extra parameter record, representing a particular function, in the Parameter Table. This can be caused by the overuse of some particular feature. For instance, this error will arise if fields which are in a line other than the lowest level detail line for any one associated record type are designated to be grouped.
211	The user has specified an output line to be used with the PRT function and the Line Format Table is empty. This error is highly improbable, but the line format cards can be checked to see if the line number is correctly punched in columns 1, 2, and 3.
212	Classification Table tests specified by a single parameter line result in more than 10 machine instructions, which exceeds the system capabilities. This error may imply that too many tests are made on the Classification Table for this particular parameter line.
213*	More than 12 hand calculation instructions are being processed at one time.

Error Condition	Cause
214*	A field calculated as having a length equal to one word is also calculated as having a length of less than 36 bits. Machine failure.
215	GET package is too large to handle.
216	Object program parameter module has an operation code which is illegal; the module should have been for the ACT function. Check to see if the packet requires an ACT but has none.
217	Too many run-in switches are required. This may be caused by designating more than 26 record types as having control break fields.
230*	The STEPI subroutine is requested to build more than 24 instructions with a single entry, which is more than it can handle at one time.
231*	The STEPI subroutine is requested to select a subroutine via an illegal subroutine code.
232	The object program has exceeded available core storage. Try to reduce size of job. Check to see if there is more than one ACT on the same input record type.
251*	The report number assigned to the current packet is zero.
252	This error implies that there is a total line for which there is no control break specified in column 48 of the Field Parameter cards. Every unique second character (control break level) of every total line number must be entered in column 48 of some Field Parameter card.
253*	The Line Format Table is out of sequence.
254*	Same as error 253 but for Control Break Table.
255*	Illegal line format number. Check columns 1, 2, and 3 of field format cards and columns 38, 39, and 40 of the E record dictionary definition cards.
256	There is a total line having a control break level in excess of 9 but not equal to F (final total). Check total line field format cards for one having a control break level other than 1-9, inclusive, or F.

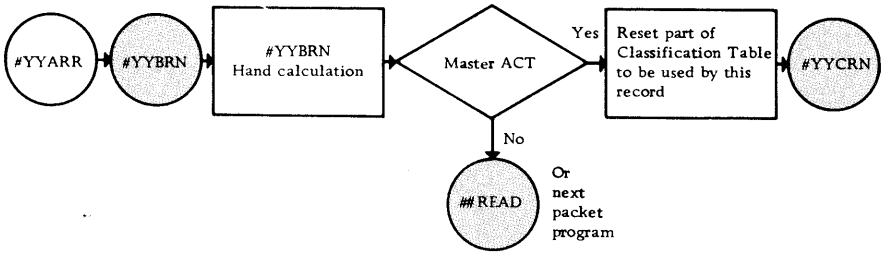
Error Conditions	Cause
257	No reset package built for the current associated record type. This error can be caused by using indicative information from an input record type which has no control break field specified. The system assumes that all higher level sequence fields that are referenced are also designated as control break fields.
260*	The TID Table is out of sequence on associated record type. (This is logically impossible since the table is sorted internally by the system.)
261	Illegal line type, not H, D, or T. Check field format card columns 1, 2, and 3, and E record card columns 38, 39, and 40.
307*	Packet number is incorrect. This error cannot be caused by the user since the packet number is assigned to each packet automatically by the system. Check for read error on hand calculation scratch tape.
310*	Packet numbers are out of sequence. This error is logically impossible for reasons similar to error 307.
330*	Illegal instruction key used with the STEPI subroutine.
626*	Packet number is too large. This error cannot be caused by the user for reasons stated for error 307.

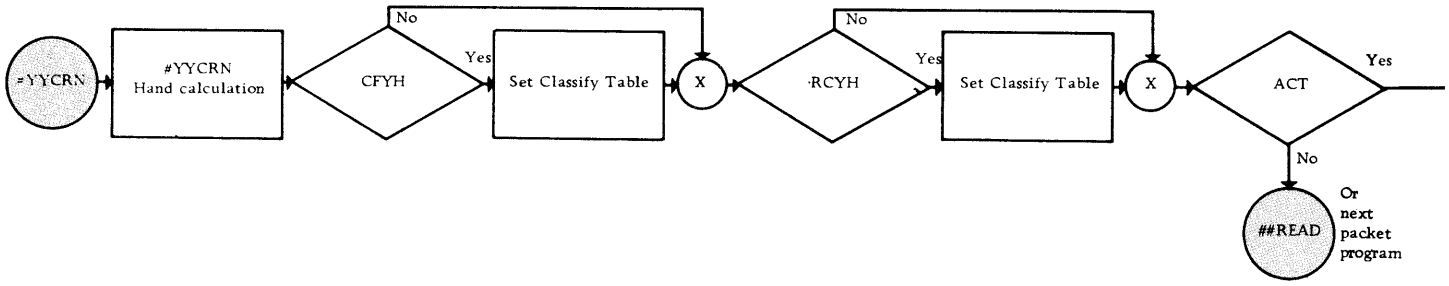
APPENDIX A: OBJECT PROGRAM FLOW CHARTS

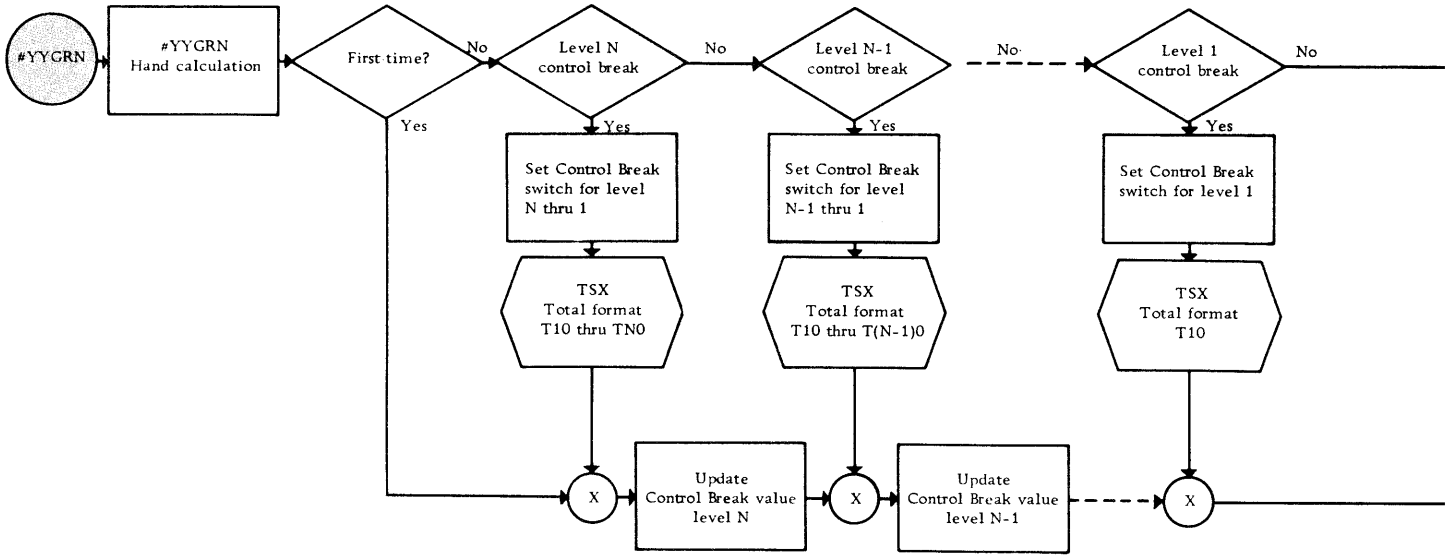
The following flow charts are representative of Reports Generator object program flow charts.

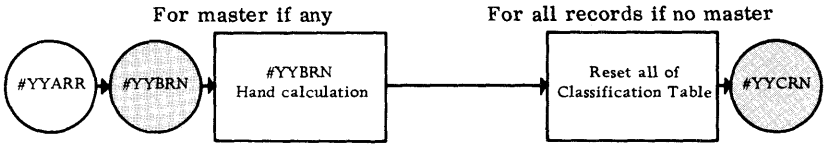


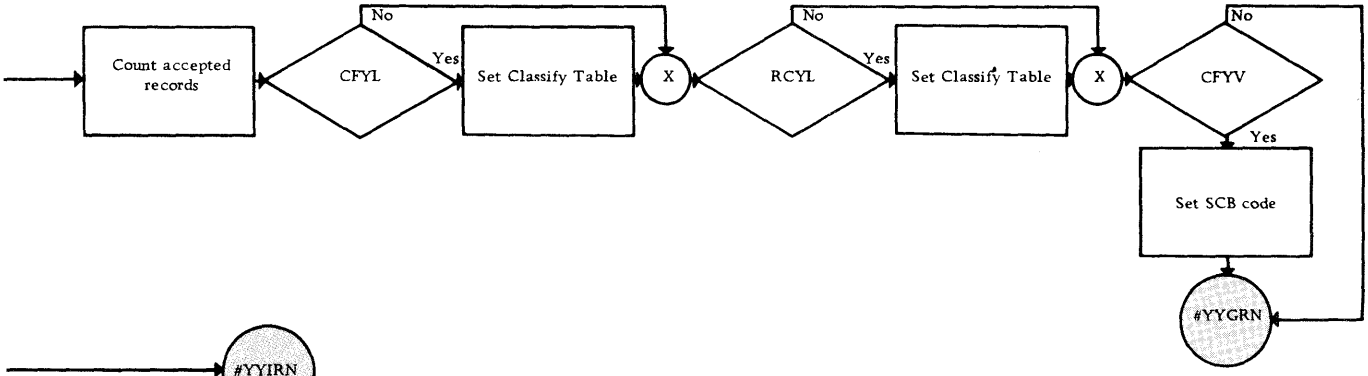
For non-master records if any master

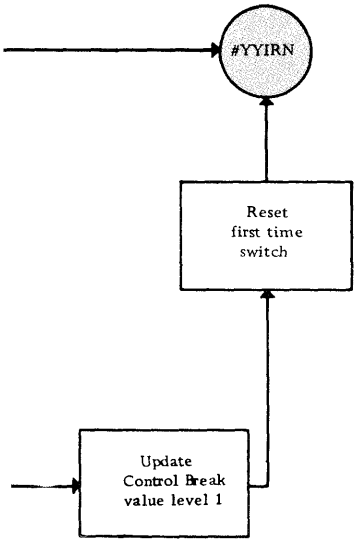




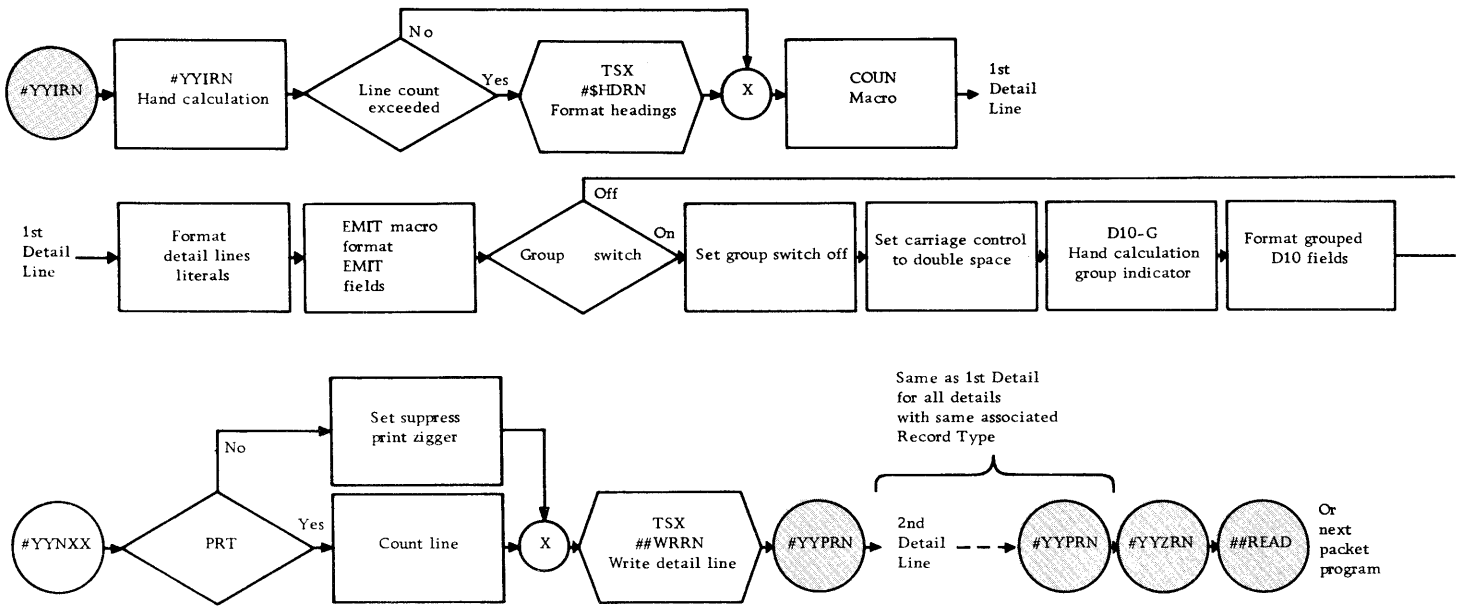



For master if any
For all records if no master
#YYARR
#YYBRN
#YYBRN
Hand calculation
Reset all of
Classification Table
#YYCRN

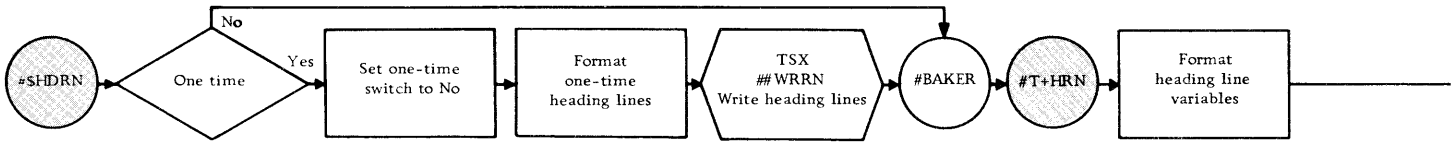

Count accepted
records
CFYL
No
Yes
Set Classify Table
X
RCYL
No
Yes
Set Classify Table
X
CFYV
No
Yes
Set SCB code
#YYGRN


#YYIRN
Reset
first time
switch
Update
Control Break
value level 1

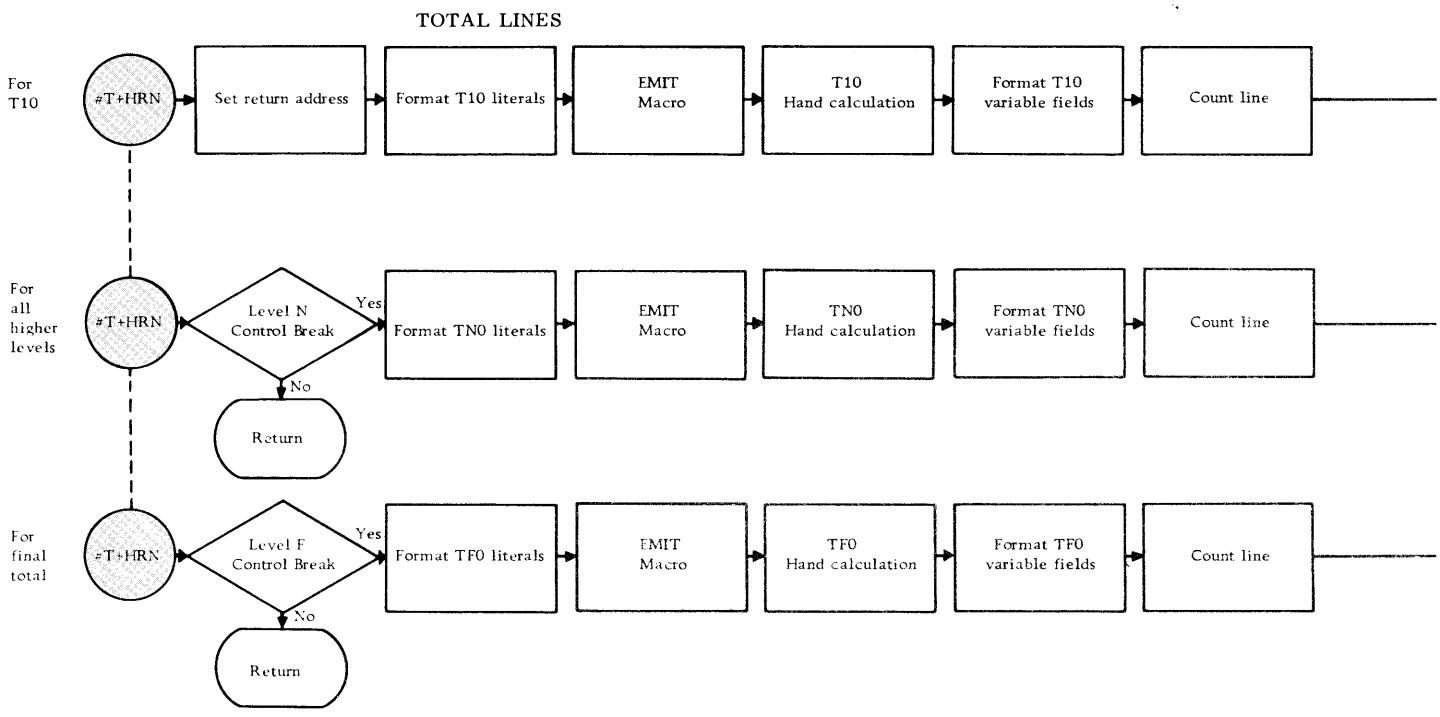
DETAIL LINES

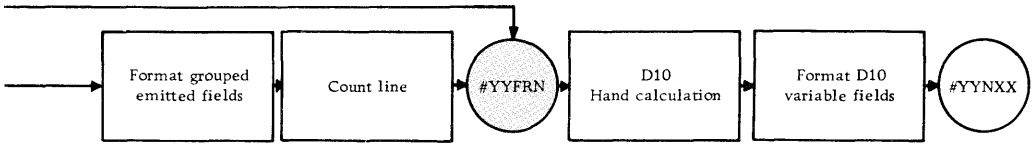


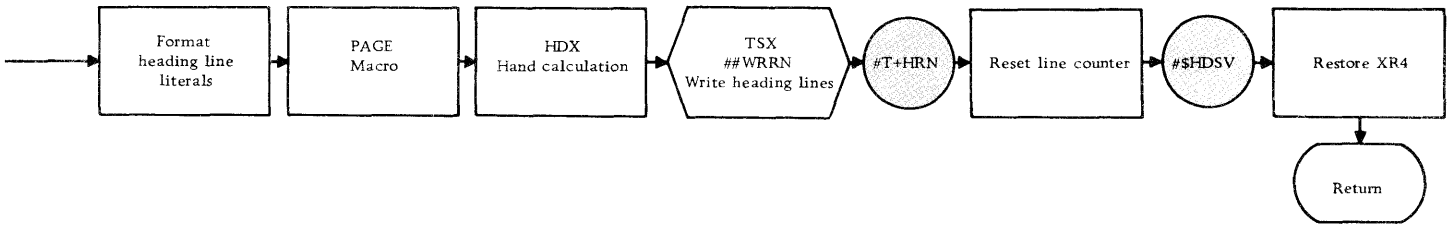
HEADING LINES

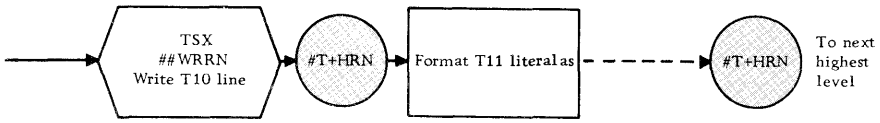


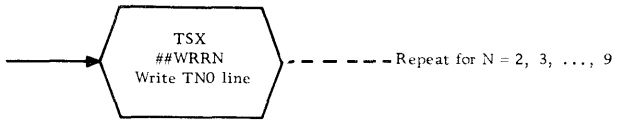
TOTAL LINES

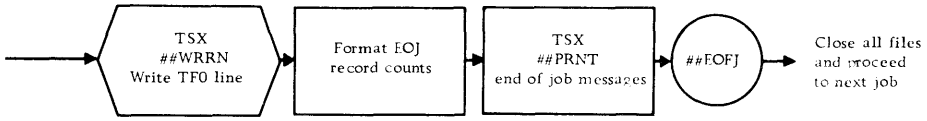



Format grouped emitted fields
Count line
#YYFRN
D10 Hand calculation
Format D10 variable fields
#YYNXX


Format heading line literals
PAGE Macro
HDX Hand calculation
TSX ##WRRN Write heading lines
#T+HRN
Reset line counter
#$HDSV
Restore XR4
Return


TSX ##WRRN Write T10 line
#T+HRN
Format T11 literal as
#T+HRN
To next highest level


TSX ##WRRN Write TN0 line
Repeat for N = 2, 3, . . . , 9


TSX ##WRRN Write TF0 line
Format EOJ record counts
TSX ##PRNT end of job messages
##EOFJ
Close all files and proceed to next job

APPENDIX B: REPORTS GENERATOR MACRO-INSTRUCTION EXPANSIONS

Given below are the expansions of the various Reports Generator macro-instructions. Where a lower case letter (e.g., x or y) appears, it represents a quantity or a location that will be supplied by the system.

The SET Macro-Instruction

1. CAL y

2. Either a or b, depending upon whether the macro-instruction contains a negative indicator to specify that the digit is to be turned off rather than turned on.

   a. With negative indicator

      ANS x

   b. Without negative indicator

      ORS x

The TEST Macro-Instruction

1. LDI y

2. Either a or b, depending upon whether the macro-instruction contains a negative indicator to specify that the test is for a condition not present.

   a. With negative indicator

      OFT x

   b. Without negative indicator

      ONT x

The GET Macro-Instruction

1. Either a or b, depending upon whether the field is being accumulated.

   a. Field is being accumulated

      CLA x
      STO BINBUF

   b. Field is not being accumulated - either (1) or (2), depending upon whether the field is packed.

(1) Field is not packed

```
CLA x,1
```

and either (a), (b), (c), (d), or (e), depending upon whether the field is to be formatted, the mode of the field, and whether its sign is to be changed.

(a) Field is to be formatted and the field is a binary field not having its sign changed.

```
STO BINBUF
```

(b) Field is to be formatted and the field is a binary field having its sign changed.

```
CHS
STO BINBUF
```

(c) Field is to be formatted and the field is a BCD field not having its sign changed.

```
SLW BCDBUF
```

(d) Field is to be formatted and the field is a BCD field having its sign changed.

```
SLW BCDBUF
LDI  BCDBUF
IIR  000040
RIR  000020
STI  BCDBUF
```

(e) Field is not to be formatted and is a BCD field having its sign changed.

```
PAI
IIR  000040
RIR  000020
PIA
```

(2) Field is packed — either (a), (b), or (c), depending upon the length of the field.

(a) Field is larger than 1 word — either ((1)), ((2)), or ((3)), depending upon whether the field is right-justified and whether an extra word is needed; in any case, ((4)) will be present next if the field is a BCD field having its sign changed.

((1)) Field is right-justified.

```
TSX   GET001,4
```

((2)) Field is not right-justified and an extra word is needed.

```
TSX    GET003,4
```

((3)) Field is not right-justified but an extra word is not needed.

```
TSX    GET002,4
```

((4)) Either ((a)) or ((b)), depending upon whether the field is to be formatted.

((a)) The field is not to be formatted.

```
PAI
IIR    000040
RIR    000020
PIA
```

((b)) The field is to be formatted.

```
LDI    BCDBUF
IIR    000040
RIR    000020
STI    BCDBUF
```

(b) Field is equal to one word — either ((1)), ((2)), or ((3)), depending upon the type of field and whether it is right-justified. ((1)) and ((2)) are next followed by the coding for 1b(1)(a), (b), (c), or (d), whichever is appropriate.

((1)) Field is binary and is right-justified.

```
CLA    x,1
```

((2)) Field is BCD and is right-justified.

```
CLA    x,1
```

((3)) Field is not right-justified.

```
CAL    x,1
LDQ    x+1,1
LGL    y,1
```

and either ((a)), ((b)) or ((c)), depending upon the type of field and whether it is to be formatted. In any case, this coding may be followed by 1b(1)(c) or (d).

((a)) Field is BCD and is to be formatted.

```
SLW    BCDBUF
```

((b)) Field is binary and is to be formatted.

```
SLW    BINBUF
```

((c)) Field is not to be formatted.

```
ANA    m
```

(c) Field is smaller than one word.

```
CAL    x,1
ANA    m
```

and either ((1)), ((2)), ((3)), or ((4)), depending upon the type of input and whether the input overlaps into another word. In any case, these instructions may be followed by 1b(1)(a), (b), (c), or (d).

((1)) Input overlaps word and is BCD.

```
LDQ    x+1,1
LGL    c
```

((2)) Input overlaps word and is binary.

```
LDQ    x+1,1
LGR    a
CLM
LLS    b
```

((3)) Input does not overlap word, field is binary.

```
XCL
CLM
LGL    d
LLS    e
```

((4)) Input does not overlap word, field is BCD and is not right-justified.

```
LGR    f
```

APPENDIX C: HAND CALCULATION OPERATION CODES

The following operation codes may be used in either File Processor or Reports Generator hand calculations. Those operation codes preceded by an asterisk (*) are IOCS commands; those preceded by a double asterisk (**) are SCAT language prefix codes; those preceded by a triple asterisk (***) are SCAT language pseudo-operations; all others are 7090 machine instructions.

ACL	FDP	LRS	**PZE	SXA
ADD	FMP	LTM	RFT	SXD
ADM	**FOR	LXA	RIA	*TCH
ALS	FRN	LXD	RIL	TIF
ANA	FSB	**MON	RIR	TIO
ANS	FSM	MPR	RIS	TIX
ARS	**FVE	MPY	RND	TLQ
AXC	HPR	MSE	RNT	TMI
AXT	HTR	**MTH	RQL	TNO
***BES	IIA	**MTW	SBM	TNX
***BCI	IIL	**MZE	SIL	TNZ
***BSS	IIR	NOP	SIR	TOV
CAL	IIS	NZT	**SIX	TPL
CAQ	*IOCD	OAI	SLF	TQP
CAS	*IOCP	***OCT	SLN	TRA
CHS	*IOCT	OFT	SLQ	TSX
CLA	*IORP	ONT	SLT	TTR
CLM	*IORT	ORA	SLW	TXH
CLS	*IOSP	ORS	SSM	TXI
COM	*IOST	OSI	SSP	TXL
CRQ	IOT	PAC	STA	TZE
CVR	LAC	PAI	STD	UAM
DCT	LAS	PAX	STI	UFA
***DEC	LBT	PBT	STL	UFM
DVH	LDC	PDC	STO	UFS
DVP	LDI	PDX	STP	USM
ENK	LDQ	PIA	STQ	VDH
ERA	LFT	**PON	STR	VDP
ETM	LGL	PSE	STT	VLM
FAD	LGR	**PTH	STZ	XCA
FAM	LLS	**PTW	SUB	XCL
FDH	LNT	PXA	**SVN	XEC
		PXD	SWT	ZET

INDEX

PUBLICATIONS

Following is a list of IBM publications which may be of interest to the reader:

REFERENCE MANUALS

Form Number	Title
A22-6528-1	IBM 7090 Data Processing System
A22-6536	IBM 709 Data Processing System
A22-6616	7340 HYPERTAPE Drive
C28-6036	Generalized Sorting Program for the IBM 709 Sort 709
C28-6052	Generalized Merging Program for the IBM 709 Merge 709
C28-6100-1	IBM 709/7090 Input/Output Control System

GENERAL INFORMATION MANUALS

D22-6508-2	IBM 709/7090 Data Processing System
F28-8001	Sorting Methods for IBM Data Processing Systems
F28-8043	IBM Commercial Translator
F28-8053-1	COBOL
F28-8074	FORTRAN

BULLETINS

G22-6505-1	IBM 7090 Data Processing System
G22-6634	7340 HYPERTAPE Drive
J28-6043-1	Sort 709: Sorting Times for the IBM 7090
J28-6059	Addenda and Errata to the Sort 709 Manual
J28-6061	Addenda to the Merge 709 Manual
J28-6080	IBM 709 Utility Programs
J28-6138	Sort 709: Sorting Times for the IBM 7090 with IBM 729VI Magnetic Tape Units
J28-6152	IBM 7090 with IBM 7340 HYPERTAPE Drives: Programs and Programming Systems
J28-6153	IBM HYPERTAPE Input/Output Control System for 7000 Series Data Processing Systems
J28-6156	IBM 7090 Generalized Sorting Program Using IBM 7340 HYPERTAPE Drives

SOS REFERENCE MANUAL

X28-1213	SOS Reference Manual - SHARE System for the IBM 709 (loose-leaf binder and index tabs)
328-1219	SOS Reference Manual Distribution No. 1
328-1262	SOS Reference Manual Distribution No. 2
328-1377	SOS Reference Manual Distribution No. 3
328-1395	SOS Reference Manual Distribution No. 4
328-1406	SOS Reference Manual Distribution No. 5

IBM
International Business Machines Corporation
Data Processing Division, 112 East Post Road, White Plains, N.Y.
Printed in U.S.A. J28-6168