

7090 DATA PROCESSING SYSTEM BULLETIN

IBM 7090 PROGRAMMING SYSTEMS SHARE 7090 9PAC PART 1: INTRODUCTION AND GENERAL PRINCIPLES

SHARE 7090 9PAC is a business-oriented programming system which facilitates the establishment and maintenance of data files and enables the user to obtain any desired report on this data with a minimum of programming effort, in a timely manner, and in the format which the user specifies.

The 9PAC System will operate on any IBM 7090 or on any IBM 709 equipped with Data Channel Trap. It requires a minimum machine configuration of 32K words of core storage, one on-line printer, one on-line card reader, and 4 tape units on each of 2 channels.

This publication is the first part of a reference manual which describes the 9PAC System and prepares the reader to use the facilities it affords. The reader is assumed to have a basic understanding of the IBM 7090, especially as regards input/output and magnetic tape records; no knowledge of symbolic programming is required except with respect to the use of hand calculations, which is an auxiliary system feature.

This publication provides a general introduction to the 9PAC System and a description and explanation of the use of 9PAC files. Other parts of the manual are:

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

SHARE 7090 9PAC
Part 3: The Reports Generator
Form J28-6168

References in this publication to the other parts of the manual are in terms of part and chapter numbers.

9PAC was initially developed by SHARE members. SHARE members who cooperated in the programming of the initial system were:

General Electric Company, Richland, Washington Chrysler Corporation Dow Chemical Company General Electric Company, Syracuse, New York Northern States Power Company Phillips Petroleum Company Union Carbide Corporation

SHARE 7090 9PAC is currently being maintained and improved by IBM Applied Programming.

TABLE OF CONTENTS

Chapter 1: An Introduction to 9PAC
Program Generation
Program Execution
The File Processor
The Reports Generator
Input/Output Configurations for 9PAC Object Programs
Dictionaries
A Simplified 9PAC Program
Chapter 2: General Characteristics of 9PAC Files
The Logical Characteristics of a File
Records and Fields
Partial Fields
Grouping
Summary Records 1
Sequencing
The "Parent-Offspring" Relationship 1
The Physical Characteristics of a File
File Characteristics
Labels
Attached Dictionary
Mode
Density
Blocking
Block Check Sums
•
File Restrictions
Chapter 3: Format of IOCS Control Cards
The *JOB Card
The *FILE Cards
The *END Card
Appendix A: 9PAC Generation Procedure
The Reports Generator
Appendix B: Format of Dictionary Records
Appendix B: Format of Dictionary Records
Appendix C: 9PAC Collating Sequences
appoint of the comming sequences, and a first the comming sequences, and the committee of t

CHAPTER 1: AN INTRODUCTION TO 9PAC

9PAC is a system designed for data processing of business transactions; it consists of two basic and distinct parts: the File Processor and the Reports Generator.

The File Processor is concerned with the original gathering of facts, organizing the acceptable facts, and storing the facts on tape. In addition, the File Processor provides a means of inserting, deleting, and modifying the facts contained in a file, as well as recording new facts as they become available.

The Reports Generator is concerned with the extraction of information from a file and the use of this information to produce desired reports; this may involve developing new information based on given information (totals, etc.) or may involve editing the information in a file to create a desired report.

Program Generation

Both the File Processor and the Reports Generator are program generators; that is, they accept input parameters (described later) written by the programmer and from these build routines which will accomplish the objectives specified by the programmer. These routines are generated by the File Processor and the Reports Generator and are known as the object program.

The object program is loaded into the IBM 7090, which will operate upon the data, under control of the object program, to produce the output desired by the programmer.

Program Execution

Program generation and object program execution may be accomplished in one of two ways: Generate and Go or Load and Go.

Generate and Go is a method of operation whereby the parameter deck is fed into either the File Processor or the Reports Generator, the object program is generated, and then control is transferred to the object program for immediate execution. At the same time, an object program deck may be produced for later Load and Go.

Load and Go is a method of operation whereby a previously produced object program deck is loaded into core storage and is then executed immediately.

The Generate and Go method provides an easy means of changing parameters whenever an unusual combination of file activity, especially report generation, is desired. This feature facilitates business control through the medium of timely and custom-designed reports. The Load and Go method provides an economical means of doing standard 9PAC jobs, since regeneration is not required.

The File Processor

The File Processor portion of 9PAC is designed primarily to generate object programs which will control file maintenance; however, inherent in the logic of this system is the ability to handle file establishment as a special case of file maintenance. File establishment consists of the initial organization of the master file and its creation from the various available data. File maintenance consists of the subsequent processing of the master file, processing which will keep the file current and maintain any desired historical data. In this process, object programs produced by the File Processor are designed to read the master file and read any number of additional files which contain transactions to be applied against the master file.

These transaction files or change files are merged together and are processed against the master file in order to produce the desired updated new master output file.

The changes to the master file, which are generally specified in the parameters, but which may be specified in the change file itself, are made by means of insertion, deletion, or modification.

<u>Insertion</u> is the placement of new data into the file. For example, a sales file may require insertion of records for new salesmen or for new customers.

<u>Deletion</u> is the removal of information from a file. For example, a payroll file may require the removal of records of employees who have left the company.

Modification involves replacing, adding to, or subtracting from, information in the file. For example, a record which contains marital status requires replacement to reflect marriage; a record which contains year-to-date salary requires addition to reflect current salary; and a record which indicates inventory-on-hand requires subtraction to reflect stock with-drawals.

Just as there are numerous kinds of changes to a file, there are also numerous methods of applying these changes. The File Processor includes three basic methods of making changes to a file during file maintenance. These are: vertical, update, and horizontal.

A <u>vertical change</u> is one which applies in the same way to all records which meet specified selection criteria. For example, all payroll records of master mechanics might need to indicate that a ten cent raise has been granted.

An <u>update change</u> is one which applies in the same way to all records which meet specified selection criteria, but unlike the vertical change, the data may vary. For example, an accounts payable file may need to have new bills payable added to existing balances payable.

A horizontal change is one which applies in various ways to records which meet specified selection criteria. The change data itself tells both the required action and the data to use in making the change. For example, a personnel history report may require the replacement of a new address for an old address for one employee, and it may require the addition of a dependent for another employee. Similarly, an inventory file may require the addition of a quantity of one part number and the insertion of a new vendor's name in another.

These types of changes, like many of the other concepts contained in this chapter, will be explained further elsewhere in this manual. Their introduction at this point serves only to orient the reader to the overall operation of the 9PAC System and to develop in the reader an awareness of the power and scope of the 9PAC System.

The output of this processing of the change data against the master file is a new master file, an error file, and an activity report file.

The new master file may be used for subsequent processing; the error file will reflect (1) data that is apparently in error and (2) changes that were not made due to errors in the program. The activity file, which is optional, is a condensed file of selected records as they appear before and/or after change. These records may subsequently be analyzed to obtain a report of activity to the file and may also be used to check that changes were made in the desired manner.

Throughout the system, the programmer need not be concerned with many of the details normally associated with programming data processing applications. Programs generated by 9PAC include a basic logic which controls the input/output of all data, merges multiple change files, selects the proper change record for action against the master file, reports object-time errors such as incorrect change data values, checks to determine that the correct master file tape is mounted, and performs many other routine functions automatically.

The Reports Generator

The Reports Generator portion of 9PAC is designed to generate object programs which will produce printed reports; as an auxiliary feature due to the logic of the system, it may also produce tape files for other purposes such as sorting.

The Reports Generator produces object programs which read the master file or change activity file produced by File Processor programs (or for that matter, any of a variety of files produced from various sources) and operates upon it to produce the desired output without destroying or changing the input file in any manner.

Reports Generator programs select and edit data from the master file; they then "format" this data into print lines for off-line printing or punching. Reports produced may have various types of lines, such as heading lines, detail lines, and total lines.

Heading lines give information about the report. They identify it and comment upon various sections of it.

<u>Detail lines</u> normally are used to list selected information related to a single input record. For example, a detail line may contain an employee's name, his seniority date, and his job classification.

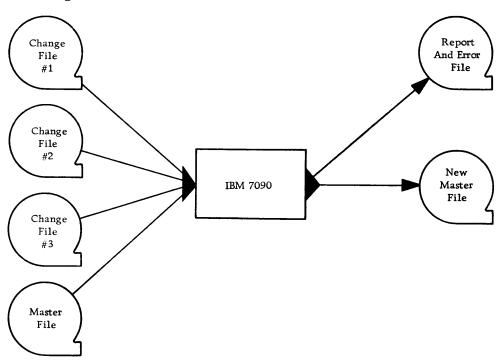
<u>Total lines</u> may be used to produce totals of the various detail lines. If the detail lines list part number, quantity on hand, and value of the stock of that item, total lines might indicate the total value of various groupings of the stock as well as a grand total of all the stock.

Tape files containing records that are not destined for printing may be produced by the Reports Generator; these may include detail and total information but may not contain heading line-type information.

Input/Output Configurations for 9PAC Object Programs

At this point, a brief explanation of 9PAC input/output configurations will help to explain the various types of activities that may be carried out by File Processor or Reports Generator object programs.

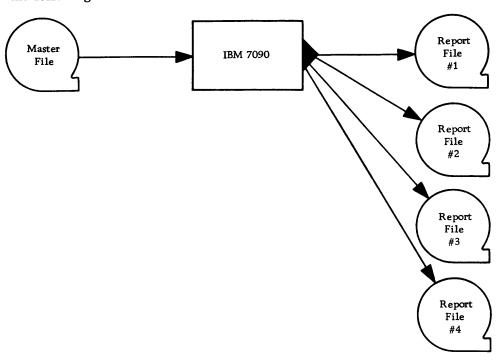
The File Processor typically has an input/output configuration such as the following:



Thus, a File Processor program will have a master file as input (except in the case of file establishment) and will have one or more files which contain change information. These are all input to the IBM 7090 under control of the object program which has been previously loaded into core storage.

The object program controls the combining of the change files into the master file and produces, as output, a single, updated, new master file and a report and error file. These files may then be processed by a Reports Generator program and/or a future File Processor program along with additional change files.

The Reports Generator typically has an input/output configuration such as the following:



Thus, a Reports Generator program has a single file as input and, under control of the object program which has been previously loaded into core storage, produces a number of output tapes, each of which is destined for off-line printing or punching, or for future processing such as sorting.

Dictionaries

As an aid to program preparation and to provide for error detection, 9PAC files have the ability to carry a description of their own data in the form of dictionaries. These dictionaries are located at the beginning of the file and may be used with master files and certain types of change files. Dictionaries fulfill a variety of services. They allow input/output routines to handle records of various lengths automatically. They permit ease of referencing information within the file, since references may be in record type and field number rather than in increment and length. They provide an option whereby certain records need not carry some of their identifying information (thus condensing the file). They define the mode of a field (BCD or binary). They specify maximum and minimum values that a field may have (for automatic error detection when a field exceeds these values). They also perform other functions which are described later.

When programming subsequent applications using a file that has a dictionary, the programmer need only refer to the established dictionary to identify the records and fields with which he is concerned. This alone reduces the programming effort significantly.

The 9PAC Language

Most programming languages are procedural languages; that is, the programmer must first design a flow chart of the program he wishes to compile and then he must translate this flow chart into a series of procedural statements.

9PAC does not use this type of language. In order to describe the 9PAC language, it is helpful to understand the construction of 9PAC object programs. 9PAC object programs have a predetermined logic (or generalized framework) into which are inserted various modules that are designed to do the various functions which the programmer specifies in his source language program. Thus the programmer need not flow chart his problem; he need only specify the functions he wishes his program to perform. Appropriate machine instructions are then generated and are placed within the generalized framework that is pre-established for all 9PAC object programs.

To specify the functions which he wishes his program to perform, the programmer makes entries in specific fields (or columns) of various coding forms. These entries are referred to as parameters, since they serve to vary the size, complexity, number, and sequence of the various standard functions which are built into the generalized framework to produce the desired object program. Entries need not be made for reading and writing, since these functions are automatically built into the object program; however, entries describing the logical and physical characteristics of the files must be made so that the reading and writing routines may be generated with the correct parameters.

The File Processor generator uses four forms to specify the four basic functions. These functions are Dictionary Definition and three types of change functions. The mere selection of the form determines the basic function. Within each basic function are a number of sub-functions. Examples of these sub-functions are:

SEL	Select records on the basis of the contents of specified fields
MTCH	Match change records to master records
FLD	Change the specified field according to the specified action

These and other functions will be described in more detail in Part 2.

The language of the Reports Generator is different from that of the File Processor, although here too the programmer makes parameter entries in specific fields (or columns) of various coding forms.

The Reports Generator employs three basic forms. Two forms are used to specify the format of the output file, while the third is used to specify the relationships between the elements of the input file and the elements of the output file.

The format of a report intended for printing is described by a pictorial representation of the desired report. The formats of records not intended for printing are specified by entries which give the logical and physical characteristics of the desired output file.

The parameter entries which specify the relationships between input and output consist of input record and field identification, processing action such as selection, editing, and accumulation, and the identification of the output line (or record) and field associated with the input and the processing These and other functions will be described in more detail in Part 3.

A Simplified 9PAC Program

The following example will illustrate some of the elements of a 9PAC program and some aspects of coding a 9PAC problem solution. The example is highly simplified, as is the coding. Only the major portion of the coding is shown; it is represented schematically in a form which approximates actual coding. The reader need not be concerned if some parts of the sample problem are not completely clear; the example is placed here merely to give him a feeling for the system, rather than to teach him how to code 9PAC problems.

Suppose a program is being written which is to account for machine-time usage by various projects and, within projects, by sub-projects.

Assume that two files exist: a master file and a change file. The master file contains various information which is used to account for machine time usage; information in this file is referenced by record type and field number. The change file contains information which is to be used to change the information in the master file. Information in the change file is referenced in terms of column position; column position itself is indicated by increment and length (which will be explained later).

The master file is organized as follows:

Record type 01: Project record Project number

Field 0006: Estimated hours for the project

Field 0007: Project name

Record type 02: Sub-project record
Field 0004: Project number
Field 0006: Sub-project number
Field 0007: Current hours

Field 0007: Current nour Field 0010: Total hours

The file is ordered on project number; thus each record type must contain a project number.

Each of these record types contains additional fields, but these are not used for this problem.

The change file contains two separate types of change records, as follows:

- 1. New estimate record. This record is identified by a 5 in column 3; it contains the project number in columns 4-10 and the new estimate in columns 11-15.
- 2. Current hours record. This record is identified by either a 6 or 7 in column 3; it contains the project number in columns 4-10, the subproject number in columns 11-13, and the current hours in columns 21-24.

The File Processor portion of this problem involves two file maintenance activities:

- 1. Change the estimated hours in field 0006 of record type 01.
- 2. Record the current hours in field 0007 of record type 02 and add these hours into the previous total in field 0010 of record type 02.

To change the estimated hours, the programmer might code the following:

Function	Master File Record Type
UPDATE	01

Function	Change Location	Change Value
SEL	002001	5

The first line indicates that this is an update change to master file record type 01 (project records). The second line indicates that the proper type of change record for this change (the New Estimate record) is to be identified by a 5 in column 3 of that record. Since columns in the change record are identified by increment (number of columns preceding the field) and length (number of columns in the field), the column which is to contain the 5 (column 3) is stated as 002001. That is, 002 is the increment of column 3 and, since we are dealing with a single column, its length is 001. These are combined as 002001. Fields containing data used in making changes of this type are often identified on 9PAC coding forms as "Change Location."

The next line to be coded might be:

Function	Master File	Master File	Change
	Record Type	Field Number	Location
МТСН	01	0003	003007

This instruction matches the information in record type 01 and field number 0003 (project number) with the information in columns 4-10 (increment 003 and length 007) of the New Estimate change record (project number). This identifying information must match before the change described below can be made (e.g., the master file record for project 31248 can be changed only by a change record for project 31248).

Finally, the following instruction might be given:

Function	Master File Record Type	Master File Field Number	Change Location	Change Action
FLD	01	0006	010005	R

This specifies that the new estimate (columns 11-15 of the New Estimate change record) is to replace the current estimate (master file record type 01 and field number 0006). Replacement is specified by the letter R in the column headed "Change Action."

To record current hours for each sub-project and incorporate them in the sub-project total hours, the following coding might be used:

Function	Master File Record Type
UPDATE	02

Function	Change Location	Change Value
SEL	002001	6-7

These lines indicate that an update change to master file record type 02 is to be made using a Current Hours change record, which is identified by either a 6 or 7 in column 3.

The following might be coded next:

Function	Master File	Master File	Change
	Record Type	Field Number	Location
MTCH	02	0004	003007
MTCH	02	0006	010003

Since master file sub-project records are ordered by sub-project number as well as project number, both of these fields must also be contained in the change record and the change record must be matched against the master file record to determine that both apply to the same sub-project; this is accomplished by the above coding.

The following coding might then be used to indicate the actual change:

Function	Master File Record Type	Master File Field Number	Change Location	Change Action
FLD	02	0007	020004	R
FLD	02	0010	020004	+

This specifies that the current hours contained in the Current Hours change record are to replace the current hours contained in the master file record and are to be added to the total hours in the master file record. The addition is specified by the plus sign in the column headed "Change Action."

After the file has been updated, suppose it is desired to produce a report on the hours used, etc. The master file will contain the information to be used in preparing the report. Assume that the report is specified as follows:

- 1. Each page is to contain a heading line that will label the various columns of information on the page.
- 2. Each Sub-project record in the master file is to cause a detail line to be printed; the detail line will give the sub-project number, the current hours, and the total hours for that sub-project.
- 3. Sub-projects are to be summed into total lines for each project. Total lines will contain the project name, the total current hours and total hours for the constituent sub-projects, and the estimated hours for the project.

The format of the report is specified in terms of <u>literal</u> information (identical information that appears each time the line is printed) and <u>variable</u> information (information from input records — in this case, the master file records — that is inserted into the output print line). The report is pictured on a form which shows all of the various print positions. Each different type of line is "pictured" separately on the same form and is given a line number which describes some of the characteristics of the line.

The report might be pictured as follows:

١	Line]	Pri	at :	Po	sit	ior	1																						
L	Number	1	2	3	4	5	6	7	8	9	10 1	1 1	2 1	3 1	41	51	6 1	71	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	 .	132
	HD1			Р	R	0	J	E	С	Т				T			(٠	บ	R	R	E	N	т		Т	0		D	Α	Т	E			E	ş	Т						
l	D06					+	x	х				1			Ì		1	1		+	x		x	X		+	х	х		X	x											П	
1	T10			+	x	x	x	x	x	x	-	гþ	0	r 4	A	S			+	x	x	٠	x	x		+	x	x		x	x		+	x	x		x	x					

In this "picture," + and X specify column locations where variable data is to be inserted. The other characters represent literal information which is to be printed whenever the line is printed. The line number indicates the type of line and provides certain information to the Reports Generator. The exact rules for specifying report output are given in Part 3.

The HD1 line contains literal information only; it will be printed at the top of every page.

The D06 line contains variable information; it will be printed each time a sub-project record (master file record type 02) is read. Fields specified for this line (referenced in terms of their rightmost print position in the coding above) will subsequently be related to input fields. In this case, the variable information from the input file will be inserted as follows:

Variable Information	Rightmost Print Position of Variable Field
Sub-project number	007
Current hours	023
Total hours	030

The T10 line contains literal and variable information; it will be printed following all sub-project detail lines for each project. The variable fields for this type of line are:

Variable Information	Rightmost Print Position of Variable Field						
Project name	009						
Total current hours	023						
Total hours for project	030						
Estimated hours	037						

Literal information requires no further reference; however, each variable field must be associated with the input field which is to supply the data.

The following coding might be used to relate the master file input with the desired printed report:

		Out	put				
Card	In	put		Control			Rightmost
Number	Record Type	Field Number	Field Name	Break	Accumulate	Line Number	Print Position
1	01	0003	PROJECT NUMBER	1			
2	01	0006	ESTIMATED HOURS			T10	037
3	01	0007	PROJECT NAME		1	T10	009
4	02	0006	SUB-PROJECT NUMBER		j	D06	007
5	02	0007	CURRENT HOURS			D06	023
6	02	0007	CURRENT HOURS		x	T10	023
7	02	0010	TOTAL HOURS			D06	030
8	02	0010	TOTAL HOURS		х	T10	030
) °	02	0010	TOTAL HOURS			1	

Line 1 indicates that project number is the <u>control break field</u> to be used with the T10 line (this correspondence is through the 1 in the Control Break column and the middle character of the line number - 1). A control break may be described as a change in the value of a field which is specified as a control field. Specifying the project number as a control break field causes the T10 line to be printed each time the project number changes.

Lines 2-5 and 7 each specify an input field and describe where it is to be located in the output report.

Lines 6 and 8 also specify an input field and describe where this information is to be located for output; they further specify that the input fields are to be summed together prior to output. Summing is indicated by the X in the Accumulate column.

The result of the above coding would be a report as specified on page 10.

CHAPTER 2: GENERAL CHARACTERISTICS OF 9PAC FILES

Any ordered collection of data is a file. However, to be used with the 9PAC System, files must also meet certain logical and physical requirements.

The exact requirements of a 9PAC file vary due to the purpose and use of the file. Files may exist for many reasons. They may contain a record of past transactions (a master file), changes to be made to a master file, a report based on certain operations specified by a programmer, or data to be ordered in some specific fashion.

The following is a general discussion of the types of files which may be used with 9PAC and some of the logical and physical characteristics of these types. The <u>logical characteristics</u> of a file pertain to the organization and content of the <u>logical records</u>. The <u>physical characteristics</u> pertain to the format and arrangement of the physical records on tape.

Specific information covering the use and format of files will be stated where applicable.

The characteristics of a specific file may be described in a file dictionary. The dictionary concept is a special feature of 9PAC which allows each file to carry its own description. The dictionary is located at the beginning of the first reel of a data file and contains a detailed description of the composition and format of every record type contained in the file.

The Logical Characteristics of a File

Most files have certain logical characteristics: they are divided into meaningful units of information; they are ordered in a useful pattern; and they may contain certain summary information which is a composite of some of the individual information in the file.

Subdivisions of a File

The example on the following page is a typical (though highly simplified) file which will be used to illustrate some of the logical characteristics of a file.

The sequence data is a composite of the identifying information in the file. In actual practice, sequencing information is much more complex than shown here; it will be explained later in this manual. It should be noted, however, that the combined sequence data is in ascending order; this must always be true.

Although the file is arranged in a hierarchical manner, this need not be done; records may be all of the same level.

The example will become clearer as the discussion of file subdivisions continues.

	ii o	Seq	uenc		
	0			e Dat	a
Sales File	Product Division	Sales Office	Salesman	Custom er	Record Type
Product Division 1	1	0	0	00	1
Sales Office 1	1	1	0	00	2
Salesman 1	1	1	1	00	3
Earnings	1	1	1	00	4
Sales-Customer 01	1	1	1	01	5
Sales-Customer 04	1	1	1	04	5
Sales-Customer 05	1	1	1	05	5
Salesman 4	1	1	4	00	3
Earnings	1	1	4	00	4
Sales-Customer 02	1	1	4	02	5
Sales-Customer 06	1	1	4	06	5
Salesman 5	1	1	5	00	3
Earnings	1	1	5	00	4
Sales-Customer 03	1	1	5	03	5
Sales-Customer 09	1	1	5	09	5
Sales Office 2	1	2	0	00	2
Salesman 2	1	2	2	00	3
Earnings	1	2	2	00	4
Sales-Customer 07	1	2	2	07	5
Sales-Customer 08	1	2	2	08	5
Salesman 3	1	2	3	00	3
Earnings	1	2	3	00	4
Sales-Customer 24	1	2	3	24	5
Sales-Customer 25	1	2	3	25	5
Product Division 2	2	0	0	00	1
Sales Office 1	2	1	0	00	2
Salesman 1	2	1	1	00	3
Earnings	2	1	1	00	4
Sales-Customer 14	2	1	1	14	5
Sales-Customer 15	2	1	1	15	5

RECORDS AND FIELDS

A file is composed of records, each record containing information about some area of activity in the file. Thus, a record in a payroll file may pertain to an individual employee and his earnings.

More than one type of record may be included in a file. For example, a sales file may contain a record for each customer, a record for each type of item sold to that customer, and a record concerning the past history of sales to that customer. Each record has a different format, and the specific format of a record is designated by the record type. Thus, a record type designation is quite similar to the form number which is often assigned to a typical business form.

Each record may be divided into separate pieces of information called <u>fields</u>. Thus, an accounts receivable record may contain fields for customer name, debits, credits, description of services or articles sold, etc.

Each record type may be divided into fields differently, just as two business forms may be different; however, within each record type, fields must be fixed just as they are on a business form.

In addition, each record type may be a different length, just as a business form may be a different size, but each record of a specific record type is always the same length as other records of that same record type — even though some of the fields in an individual record are left blank, space is always provided for them.

PARTIAL FIELDS

9PAC permits a programmer to refer to part of a field, rather than the entire field, when he so desires. For example, a date field may consist of six columns: two for the month, two for the day, and two for the year. If it is desired to use the year only, it can be used without reference to the month and day.

GROUPING

A number of records may be grouped together when it is desired that one record carry some information that is common to all of the records in a group. For example, an inventory file may contain a record for each part number. If, for some reason, it is desired to know what plant each part is located in, and information concerning the plant (for example, an address for delivery of replacement parts) is used in conjunction with the inventory records for that plant, the inventory records may be grouped by plant and information concerning the plant may be contained in a plant record. Grouping serves a different function than sequencing, which is described below, but must be consistent with it.

Grouping must be planned in the file, but it does not require special coding other than that the group header (the record being referred to) have a lower record type than the detail records pertaining to it. The File Processor and the Reports Generator automatically recognize records treated as group headers and keep them available for ready "look back" reference. Thus, at any time, reference may be made to the previous record of each lower record type.

SUMMARY RECORDS

9PAC makes provision for a specialized type of record which can accumulate totals from fields of other records. These specialized records are called summary records in the File Processor, total lines or total records in the Reports Generator. Any number of summary records may be used. Each summary record type may contain several fields. Each field of a record type may be totaled into one or more fields of one or more summary records. Alternately, several fields from one or more record types may be totaled into the same field of a single summary record type.

Each summary record type produces totals for all fields in the record at the same time; the totals are produced at the programmer's option but must be dependent upon a sequence field break (see below).

The totaling procedure produces a record in the master file at each point that a total is taken. These total records may then be used by the Reports Generator to formulate a desired report.

Summary records must be located at the points where the totals are to be taken. Thus, in the above example, if a total is to be taken on the sales of each salesman, then a separate summary record must follow the customer sales record of each salesman. This record would be of the same level as the salesman record since a total would be taken on a sequence break (see below) on salesman. Additional summaries, if desired, might be taken on the office level, the product division level, or the entire file level.

If it is not desired to carry these summary records in the master file, Reports Generator total lines may be created. In this case, the totals are created by the Reports Generator and they do not become part of the master file; however, they may be used as desired in the Reports Generator program.

Sequencing

A file is processed serially from beginning to end. Thus, each item is handled sequentially; there is no random access to the file. In addition, each item of information in a file must be capable of being addressed individually. Therefore, each record in a file must be sequenced; that is, it must be in a meaningful order and it must contain information by which it may be placed into that meaningful order.

For example, suppose an inventory file is ordered by plant, section, department, and part number. A record type may be set up as follows:

PLANT NUMBER — SECTION NUMBER — DEPARTMENT NUMBER —
PART NUMBER — QUANTITY IN STOCK — ORDER POINT —
USAGE TO DATE

Thus, there are seven fields in the record, and records are ordered on the first four fields only. In addition, each file is sequenced by record type (this is always the lowest sequence level). In the example on the following page, records marked with an asterisk are out of sequence.

Record 3 is out of sequence on part number; it should be inserted before record 2.

Record 4 is out of sequence on section number; it should be inserted after record 1.

Record 5 is out of sequence on plant number; it should be inserted before record 1.

Record 11 is out of sequence on department number; it should be inserted before record 10.

Record Number for Illustration Only	Plant Number	Section Number	Department Number	Part Number	Record Type	Other Information
1	1	3	1	88	03	Not related to sequence
2	2	2	1	101	03	Not related to sequence
* 3	2	2	1	99	03	Not related to sequence
* 4	2	1	3	104	03	Not related to sequence
* 5	1	2	1	31	03	Not related to sequence
6	3	2	3	222	03	Not related to sequence
7	3	3	2	301	03	Not related to sequence
8	4	1	1	42	03	Not related to sequence
9	4	2	2	88	03	Not related to sequence
10	4	3	2	91	03	Not related to sequence
* 11	4	3	1	76	03	Not related to sequence

Since there is only one record type in this example, records may not be out of sequence on record type. This field, however, is checked for correct sequence to prevent equality when there are several record types and other sequencing information is identical.

These records are shown below in correct order.

Record Number for Illustration Only	Plant Number	Section Number	Department Number	Part Number	Record Type	Other Information
5	1	2	1	31	03	Not related to sequence
1	1	3	1	88	03	Not related to sequence
4	2	1	3	104	03	Not related to sequence
3	2	2	1	99	03	Not related to sequence
2	2	2	1	101	03	Not related to sequence
6	3	2	3	222	03	Not related to sequence
7	3	3	2	301	03	Not related to sequence
8	4	1	1	42	03	Not related to sequence
9	4	2	2	88	03	Not related to sequence
11	4	3	1	76	03	Not related to sequence
10	4	3	2	91	03	Not related to sequence

Records may be of different record types, but they must be sequenced according to the same information in any given file.

Fields which contain sequencing information are called sequence fields. These fields may actually appear in any order in the records. Thus, the primary or first sequence level could be the physically last field of the record. A change in the value of any sequence field is called a sequence field break (or control field break). In the revised example above, there are sequence field breaks on plant number after records 1, 2, and 7. There are sequence field breaks on section number after records 5, 1, 4, 6, 7, 8, and 9. There are also sequence field breaks on department number and part number.

Because records in a file must be handled in sequence, the sequence of a file must be planned carefully in terms of the file use, format of input change data to the file, and types of reports it is desired to produce from the file.

Although the establishment of file sequence is the programmer's responsibility, the File Processor will check to determine that all files are correctly ordered on all fields designated as sequence fields. The File Processor will not operate on a file that is not in proper sequence:

- 1. Each record must contain at least one sequence field in addition to record type.
- 2. Each record must be completely and uniquely sequenced. (The sequence fields themselves need not be unique if the record type varies.)

The Reports Generator does not check files for proper sequencing, nor is it required that Reports Generator input files be properly sequenced. However, the logic of the program will usually require that input files be in proper sequence to make control breaks meaningful, etc.

THE "PARENT-OFFSPRING" RELATIONSHIP

If the records in a file are to be sorted, then each record must contain all of the sequence fields on which sorting is to take place. However, it often happens that once a master file is established, sorting is either not required or is infrequently required. In such cases, considerable tape space (and thus input/output time and internal computer storage) may be saved by establishing a parent-offspring relationship for the records. This relationship may be established only in a master file. Such a relationship merely means that the parent record contains some of the sequencing information for the offspring records.

In the example above, there were four sequencing fields and each record contained all of the sequencing information for that record. Redundant sequencing information may be eliminated by specifying in the dictionary a parent-offspring relationship as follows:

Record		Parent
Type	Sequence Field	Record Type
03	Plant Number	-
04	Section Number	03
05	Department Number	04
06	Part Number	05

The effect of the above record establishment is to consider that each record of record type 04 pertains to the plant identified by the preceding record of record type 03. Similarly, each record type 05 pertains to the preceding record type 04 and each record type 06 pertains to the preceding record type 05.

Alternately, a parent record type could be established which would contain the first three sequence fields and the detail records could then contain the fourth and final sequence field.

Each record type may contain other fields which are not sequence fields.

A parent-offspring relationship may be established on virtually any possible combination of sequence fields, provided:

- 1. Each record type contains at least one sequence field in addition to the record type.
- 2. All records have a complete and unique sequence identification based either on complete sequence fields or on a parent-offspring relationship.
- 3. When a parent-offspring relationship is established, each offspring is sequentially arranged and is grouped immediately following its respective parent record.

The effect, when using a file, is always the same; each record is uniquely sequenced.

As indicated earlier, a file which utilizes a parent-offspring relationship may not be sorted; this is due to the fact that a record which is to be sorted must contain all sequence fields.

The Physical Characteristics of a File

A file has certain physical characteristics which have no logical significance but which do affect the manipulation of the file by 9PAC. For example, the file may have identifying information, it must be on tape, it may be in the BCD or binary mode.

In 9PAC, the physical characteristics of a file are largely determined by the purpose of the file. Thus, one type of file may be suitable for a change file while another type may be required for a master file.

File Characteristics

All 9PAC files are allowed the following optional characteristics, except as noted in the section on "File Restrictions," see below.

LABELS

Files may be labeled or unlabeled. A labeled file is one which contains control information at the beginning of a reel of tape (header label) and/or at the end of a reel of tape (trailer label). The trailer label may indicate either end of reel or end of file. All labels must be standard IOCS labels. (For this and succeeding references to IOCS, the reader is referred to the IBM Reference Manual 709/7090 Input/Output Control System, Form C28-6100-1.)

ATTACHED DICTIONARY

Files may be preceded by a dictionary which describes the logical and physical characteristics of the file. Thus, the dictionary describes each record type, and its division into fields, sequence fields, parent and offspring records, etc. Records contained in a file which has a dictionary must have their record type designated by their first two characters.

MODE

9PAC files may be written either in the binary or BCD mode. Care should be taken to avoid writing fields containing binary numbers in the BCD mode.

DENSITY

9PAC files may be written in either high or low density.

BLOCKING

Files may be blocked or unblocked. If blocked, the block size may be any desired length.

BLOCK CHECK SUMS

Binary files may have standard IOCS block check sums if block sequence numbers are present.

BLOCK SEQUENCE NUMBERS

Binary files may have standard IOCS block sequence numbers.

File Restrictions

File Processor input master files must have a dictionary attached.

Reports Generator report output <u>must not</u> have attached dictionaries, block check sums, or block sequence numbers.

Any file may have mixed record types providing each record type has a fixed length. If no dictionary is attached, all record types must have the same length; if a dictionary is attached, each record type may have a different length.

CHAPTER 3: FORMAT OF IOCS CONTROL CARDS

In order to specify the physical characteristics of a file, it is necessary to prepare three types of IOCS (Input/Output Control System) cards: *JOB, *FILE, and *END.

The formats of these cards are given below. The *JOB card and the *FILE card are prepared on a single 9PAC coding form (see Figure 1); no coding form is provided for the *END card. All entries must be left-justified. Note: The symbol "b" is used to represent blanks.

The cards for each 9PAC job must be arranged in the following order:

- 1. The *JOB card.
- 2. *FILE cards as required.
- 3. An *END card.
- 4. File Processor or Reports Generator parameter packets (discussed in the File Processor and the Reports Generator sections).

The *JOB Card

The *JOB card provides certain required information about the job to the 9PAC System. It is always printed on-line when it is read.

15.79 (6.7		1 1 1 1 1	WITCHES		
FUNCTION	JOB TYPE	BITS	3 4 5 6 7 8	JOB NAME	COMMENTS
		- 6 6 1 6	3 4 3 6 7 6		
*,J,O,B		\bot	+++++	 	-+ + +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
1 2 3 4 5 6 7 8 9 10 11 12 13	14 15 16 17 18 1	19 20 21 2223 24 2	5 26 27 28 29 30 3	1 32 33 3 4 35 3 6 37 38 39 40 4 1 42 4 3 44 45 46 4 7 48	49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72

Columns/Contents		Description							
1-6		Must be blank.							
7-10	Function	Must contain *JOB.							
11-12		Must be blank.							
13-17	Job Type	Must contain one of the following codes to indicate job type: RGGEN Reports Generator — Generate and Go RGBIN Reports Generator — Load and Go FPGEN File Processor — Generate and Go FPBIN File Processor — Load and Go							
18		Must be blank.							
19	Tape Indicator	Normally this column is blank; however, for Load and Go when the *JOB card is to be read on-line and the binary object program is on tape, this column must contain a T.							

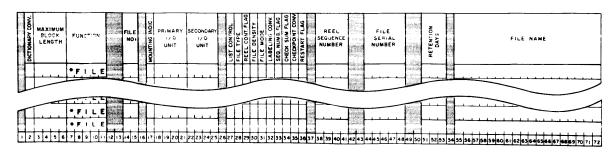
Columns/Contents	Description
20 Bits Indicator	Normally this column is blank to indicate that increments and lengths are in characters; however, if it is desired to specify increments and lengths in bits, this column must contain a B.
21 Punch Indicat	Normally this column is blank; however, if during generation it is desired to punch an object program deck, this column must contain a P.
22 List Indicator	Normally this column is blank; however, if it is desired to produce an object program listing (File Processor only), this column must contain an L.
23-30 Switches 1-8	These switches are used to choose system options; the normal mode is OFF; any non-blank character turns the switch ON. Unless otherwise noted, the switches pertain to both File Processor and Reports Generator. Switch 1. ON: Generation-time error diagnostics printed on-line. OFF: Generation-time error diagnostics printed off-line. Switch 2. ON: Object-time logical counts are printed on-line. OFF: Object-time logical counts are printed off-line. Switch 3. ON: Unconditional system messages (e.g., GENERATION COMPLETE) are printed on-line. OFF: Unconditional system messages are printed off-line. Switch 4. ON: Use 709 collating sequence (File Processor only). This results in faster 9PAC operation, however, the commercial (702) collating sequence is required for off-line sorting equipment. OFF: Use commercial collating sequence.
91 40 1-1-37-	Switches 5-8. Not used.
31-48 Job Name	May contain any alphameric comments; this entry is included in system messages.
49-72 Comments	May contain any alphameric comments.

BI	M											IC					9PA0				T								PRINTED IN	J. S.
		FUNCTION		JOB TYPE		B-BITS B-PINCH		2 3 4	Ц.	6 7	ш		1	4136	13613		B NAME	42 4 3 4	44 45 4	16 4 7 41	8 49 50	51 52	53 54	55 56		9 60 61 OMME N.		65 66 67	68 69 70 7	
Š	MAXIMUM BLOCK LENGTH	FUNCTION	F		ပ္	RIMARY I/O UNIT	Γ	DARY 0	IST CONTROL	REEL CONT. FLAG	ILE DENSITY	LABELING CONV.	SEQ. NUMB. FLAG	CHECKPOINT CONV	RESTART FLAG	SE	REEL QUENCE UMBER		FI SE	LE RIAL ABER		RETENTION	33				FILE N			=
P		* F L E * F L E * F L E				<u> </u>	1.4	<u>-</u>		+		+																111	1111	
		* F L E * F L E * F L E						-1				+		1									-							
		* F L E		1								+											1		1 1 1					
		* F L E		- -			1																							
	1-1-1-	* F L E		-1-			+-				+						<u> </u>							E						
	1 1 1	* F L		-1-				L			+									——————————————————————————————————————										1
		*F1L1	E E					1 1 1			+		+			- - -				1 1			- 		_					<u></u>
1 2	345	# F I L			3	. I . I	- 	t 1		++	_		_	\sqcup	-	4	-11	14	1	1.1.		++	-1-1			roles		505565	67 66 50	+

Figure 1. IOCS Control Cards form

The *FILE Cards

The *FILE cards specify the physical characteristics of the files; one card is required for each input or output file.



Colum	ns/Contents	Description	Description									
1		Not used by pro in binary decks	ogrammer; reserved for column binary indication									
2	Dictionary Conventions	Must be blank i must contain or density: H High L Low	If the file does not contain a dictionary; otherwise, ne of the following codes to indicate the dictionary									
3-6	Maximum Block Length	Must specify th Any Number blank Gbbb Nbbb	e maximum block length, as follows: Block length (in 709 words) including the block sequence word, if present. 0500 assumed. Reports Generator report output is to be grouped for 720 printing. Reports Generator report output is to be unblocked for printing.									
7-11	Function	Must contain *I	FILE.									
12		Not used.										
13	Close File Indicator	Must contain on to be taken when U R or b N S	te of the following codes to indicate the action in the file is closed: Rewind and unload. Rewind only. Do not rewind. Do not rewind; no EOF mark or trailer labels written. S may be used only for files used in hand calculations.									
14-15	File Number	Must contain a t 00 07 08	nique number from among the following: File Processor master input file. File Processor master output file. File Processor Change and Error Report file.									

Columns	/Contents	Description							
14-15	File Number (Continued)	09 10 01-06 and 11-15	File Processor horizontal input. Reports Generator input file. Any other File Processor or Reports Generator file. A maximum of 16 files are permitted for any single 9PAC job.						
16		Not used.							
17	Tape Mounting Indicator	May be blank to indicate that the file will be mounted when required (as indicated by an unconditional system message) or may contain an * to indicate that the file is to be mounted at start of generation (this would include labeled files and/or files with dictionaries).							
18-21	Primary Input/ Output Unit		primary I/O unit as follows, where x represents ation A-D and y represents a channel designation Card Reader Card Punch Printer May only be used in files used in hand calculations. Tape Unit						
22-25	Secondary Input/ Output Unit	If a tape unit is specified in the same format as in columns 18-21, tape unit alternation will occur; otherwise, these columns must be blank. For alternation, the first reel of the file must be mounted on the primary tape unit. In either case, physical unit number is assigned by the system at execution time.							
26		Not used.							
27	List Control	File number File name I/O unit Starting reel seq Tape mounting in If a secondary un peated for that un	dicator it is assigned, the above information is re- nit. This column may be blank to cause this d in the file list, or may contain an N to omit						
28	File Type	C Checkpoint I Input T Total block P Partial blo only) 1 Input — oct 2 Input — BC 3 Total block	of the following codes to indicate the file type: c output (standard for most output) ck output (for I-Language and hand calculations tal 7's padding present CD 9's padding present c output — octal 7's padding to be added c output — BCD 9's padding to be added						

Columns/Contents		Description
29	Reel Control Flag	For input files only, may contain one of the following codes: M Multi-reel unlabeled file (a pause will occur at each EOF) L Search for label on OPEN (hand calculations only) b Single reel unlabeled or single or multi-reel labeled file (the file will be closed at EOF)
30	File Density	Must contain one of the following codes to indicate the density of the file text: H High L Low
31	File Mode	Must contain one of the following codes to indicate the mode of the file text: D BCD B Binary Files containing binary numbers (e.g., if any field is specified as binary in the dictionary) must use B.
32	Labeling Conventions	Must contain one of the following codes to indicate labeling conventions: H Labels in high density L Labels in low density b No labels Trailer labels are always written in the same density as the file text. Dictionaries, if present, must be the same density as header labels.
33	Block Sequence Number Flag	For binary files only, may contain an S to indicate block sequence numbers; otherwise, must be blank.
34	Block Check Sum Flag	For binary files with block sequence numbers, may contain a C to indicate block check sums; otherwise, must be blank.
35	Checkpoint Conventions	Must contain one of the following codes to indicate checkpoint conventions: F Write checkpoints at each reel switch (labeled output files only) C Write checkpoints at each reel switch (any file) b No checkpoints
36	Restart Flag	May contain a code to indicate positioning for restart.
37		Not used.
38-41	Sequence Number of First Reel	For labeled files, must contain the sequence number of the first reel; otherwise must be blank.
42-43		Not used.
44-48	File Serial Number	For labeled files only, may contain a file serial number to identify the file.

Columns/Contents		Description
49-50		Not used.
51-53	Retention Days	May contain the number of days the file is to be retained (not checked by system).
54		Not used.
55-72	File Name	May contain any alphameric comments; these columns will be printed in appropriate messages.

The *END Card

The *END card is used to signal the end of the IOCS cards; it must precede all 9PAC parameter cards.

The *END card consists of *END in columns 7-10; all other columns must be blank.

APPENDIX A: 9PAC GENERATION PROCEDURE

This appendix contains a brief description of the program logic used by the File Processor and the Reports Generator.

The File Processor

The File Processor is a two phase generator for a file maintenance program in which the logic is predetermined. In the first phase, the parameter deck is read and a symbolic program is generated in program modules, or logical blocks, and is written on a scratch tape (file number 09). In the second phase, these modules are read, absolute addresses are assigned, and the program is placed in core storage ready for execution as a single phase object program if no horizontal change packets are present.

If, however, horizontal change packets are present, the generated object program is written as a scratch file behind the file dictionary of the change and error report file (file number 08) and the main object program is written as a scratch file behind the master file dictionary. The horizontal change data is then read, a twelve-word dictionary is inserted in place of each change field number, and the expanded record is written on the tape which is to contain the processed horizontal data. This file becomes the input, along with update change files, to the main object program for file maintenance.

Since generation occurs as the parameter deck is read, and since the object program logic is predetermined, the order of execution of the object program is not the same as the order of program generation.

The Reports Generator

The logic of the Reports Generator is the same as that of the File Processor except that the Reports Generator contains no provision for any situation comparable to the handling of horizontal changes by the File Processor.

APPENDIX B: FORMAT OF DICTIONARY RECORDS

Given below are the formats of the various dictionary records; they are identified by the executive code in the first character position.

The possible dictionary records are:

B Record Beginning of dictionary record.

P Record End of dictionary record; this format is not shown since it

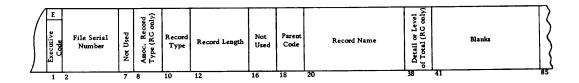
is identical to the B record except that P is the executive code.

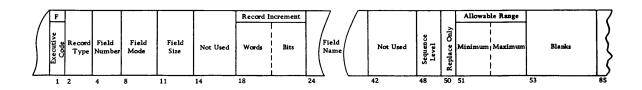
E Record Record type header record.

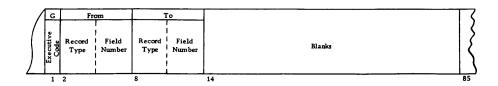
F Record Field description record.

G Record Summary definition record.









APPENDIX C: 9PAC COLLATING SEQUENCES

Listed below are the two collating sequences which may be used in the 9PAC System.

Commercial Sequence		709 Sequence		
<u>Character</u>	Card Code	Character	Card Code	
11 1		0	0	
blank	12-8-3	0 1	0 1	
•	12-8-4	2	2	
) +	12	3	3	
\$	11-8-3	4	4	
*	11-8-4	5	5	
_	11	6	6	
/	0-1	7	7	
·	0-8-3	8	8	
, %	0-8-4	9	9	
#	8-3	#	8-3	
<i>"</i> @	8-4	<i></i> @	8-4	
+0	12-0	+	12	
A	12-1	Α	12-1	
В	12-2	В	12-2	
c	12-3	c	12-3	
D	12-4	D	12-4	
E	12-5	E	12-5	
F	12-6	F	12-6	
G	12-7	G	12-7	
Н	12-8	Н	12-8	
I	12-9	I	12-9	
-0	11-0	+0	12-0	
J	11-1		12-8-3	
K	11-2)	12-8-4	
L	11-3	, -	11	
M	11-4	- J	11-1	
N	11-5	K	11-2	
0	11-6	L	11-3	
P	11-7	M	11-4	
Q Q	11-8	N	11-5	
Ř	11-9	0	11-6	
Record Mark	0-8-2	P	11-7	
S	0-2	Q	11-8	
T	0-3	R	11 -9	
U	0-4	-0	11-0	
v	0-5	\$	11-8-3	
W	0-6	*	11-8-4	
X	0-7	blank		
Y	0-8	/	0-1	
Z	0-9	S	0-2	
0	0	T	0-3	
1	1	U	0-4	
2	2	v	0-5	
3	3	W	0-6	
4	4	X	0-7	
5	5	Y	0-8	
6	6	Z	0-9	
7	7	Record Mark	0-8-2	
8	8	,	0-8-3	
9	9	%	0-8-4	

INDEX

Activity File 3	JOB Card 21
	Job Types 21
Basic Logic 3	
Bits Indicator 22	Labels 19, 32
Block Check Sums 20, 26	Length 8
Block Sequence Numbers 20, 26	List Indicator 22
Blocking 20, 24	Literal Information 10
-	Load and Go 1
Change Files 2, 4	Logical Characteristics of a File 13
Checkpoint Conventions 26	Look-Back 15
Close File Indicator 24	
Coding Forms 6	Master File 2, 3, 4
Collating Sequences 22, 30	Message Printing Options 22
Control Break 12, 17	Mode 20
(See also: Sequencing)	Modification 2
(See also. Sequencing)	
Deletion 2	9PAC Language 6
	5171C Language 0
Density 20	Object Program 1
Detail Lines 4	Object Program 1
Dictionaries 5, 13, 20	D 111 Cl
Dictionary Density 24	Padding Characters 25
Dictionary Record Formats 29	Parameters 1, 6
	Parent-Offspring Relationship 18-19
END Card 21, 27	Partial Fields 15
Error File 3	Physical Characteristics of a File 13, 19
Execution 1	Predetermined Logic 6
	Primary Input/Output Unit 25
Field Number 7	Program Modules 6
Fields 14, 15	Punch Indicator 22
File 13	
density 26	Record Type 7, 14
establishment 2	Records 14
list 25	Reel Control Flag 26
maintenance 2	Report and Error File 4
mode 26	Report Files 4
name 27	Reports 3
number 24	Restart Flag 26
restrictions 20	Retention Days 27
serial number 26	recention Days 2,
type 25	Secondary Input/Output Unit 25
FILE Card 21, 24	
	Sequencing 16-18
Format 3	break 16-17
Functions 6	data 13
	field 17
Generalized Framework 6	Summary Records 15
Generate and Go 1	
Generation 1, 28	Tape
Grouping 15	indicator 21
	mounting indicator 25
Heading Lines 4	unit alternation 25
Horizontal Change 3	Total Lines 4, 15
	Transaction Files 2
Increment 8	
Input/Output Configuration 4	Update Change 2
Input/Output Control System 19	
control cards 21	Variable Information 10
Insertion 2	Vertical Change 2
AUGCICIOII &	relical change =

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			
T00 0150	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			
000 1010	(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			

