File No. 7090-33
Form C28-6307

IBM Systems Reference Library

IBM 7090/7094 Generalized Sorting System

7090/7094 Sort

This publication describes the IBM 7090/7094 Generalized Sorting System. The 7090/7094 Sort sorts fixed-length or variable-length records written in either signed or unsigned binary or BCD mode. The records can be sorted using either the commercial or scientific collating sequences, in ascending or descending order. The program operates under the 7090/7094 Basic Monitor (IBSYS).

PREFACE

This publication describes the IBM Generalized Sorting System (7090/7094 Sort). Part I, "Introduction and General Principles," discusses the organization and structure of the program, including a description of the sorting and merging techniques used. Part II, "7090/7094 Sort Operations," gives detailed information for using the program, including general specifications, control card formats, tape record format and file structure, and user modification procedures.

It is assumed that the reader has a basic understanding of the 7090 or 7094, especially as regards input/output devices and magnetic tape records. No knowledge of symbolic programming is required to execute Sort, except in those cases where it is desired to introduce program modifications. The reader may also refer to the following IBM publications:

IBM 7090/7094 Operating Systems: Basic Monitor (IBSYS), Form C28-6248.
IBM 709/7090 Input/Output Control System, Form C28-6100-2.
IBM 709/7090 Programming Systems: FORTRAN Assembly Program (FAP), Form C28-6235.

The minimum machine configuration necessary for the 7090/7094 Sort is as follows:

An IBM 7302 or 7302-1 Core Storage Unit (32K)
Two 7607 Data Channels
One 729 Magnetic Tape Unit (II, IV, or VI) as a System tape unit or one 1301 Disk File
Four 729 Magnetic Tape Units (II, IV, or VI), two units on each channel
An on-line printer
One 729 Magnetic Tape Unit (II, IV, or VI) for control cards or an on-line card reader

This publication, C28-6307, obsoletes and replaces IBM 7090/7094 Generalized Sorting Program, 7090/7094 Sort (729-Fixed Length), Form J28-6217 and the following Technical Newsletters pertaining to the 7090/7094 Systems.

N28-0036, IBM 7090/7094 Generalized Sorting Program
N28-0023, IBM 7090/7094 Generalized Sorting Program, 7090/7094 Sort (729-Fixed Length), Loading from IBM 1301 Disk Storage.

Copies of this and other IBM publications can be obtained through IBM Branch Offices.
Address comments concerning the content of this publication to:
IBM Corporation, Programming Systems Publications, Dept. D91, PO Box 390, Poughkeepsie, N.Y.

CONTENTS

The IBM 7090/7094 Generalized Sorting Program (7090/7094 Sort) is a modular program designed to offer the user the greatest possible flexibility. The fixed-length portion of the Sort accepts, as input to be sorted, fixed-length records written in either binary or BCD, signed or unsigned. The variable-length portion accepts, as input to be sorted, variable-length records written in either binary or BCD, signed or unsigned. These records are sorted in ascending or descending order, using either the commercial or scientific collating sequences. The records may be sorted on any number of control fields, and these control fields may be up to 2,000 words in length, i.e., the maximum size of a logical record or tape record allowed by the program.

The 7090/7094 Sort is designed to operate under the Basic Monitor (IBSYS), which controls the availability of all input/output components. Specifically, 7090/7094 Sort uses IBNUC, IBSUP, and IOEX. The Sort, however, uses its own read/write routines and buffering schemes. (See the IBM Reference Manual, 7090/7094 Operating Systems: Basic Monitor (IBSYS), Form C28-6248-0.)

PROGRAM CHARACTERISTICS

The 7090/7094 Sort is composed of separate subprograms (modules), supplied as a series of blocked card images on the Library tape. Only those subprograms needed for a particular sort run are loaded from the Library tape at execution time. For this reason, each of the subprograms is relocatable.

The editing portion of the system determines which subprograms are required for a given sort run from information supplied in the form of control card parameters. These control cards are described in full in Part II of this manual.

The modular structure of Sort allows the user great flexibility in introducing routines into the Sort program to permit such operations as summarization, deletion, lengthening, and shortening of records. These routines can be located on any of the SYSUNI tapes, and are loaded and handled in the same manner as the subprograms supplied with the 7090/7094 Sort.

The Merge Program

Sort incorporates a complete merge program, thus making a separate merge program unnecessary. When a merge run is executed, only those subroutines that constitute the merge program are loaded from the Library tape. These same subroutines are used during a sort run, as part of the sort program, to merge strings of sorted records. Furthermore, it is possible, in the course of a sort run, to merge previously sorted records with records currently being sorted.

The order of merge to be used, i.e., the number of sequences merged concurrently, is determined by the user, usually on the basis of tape unit availability. An increase in the order of merge is advantageous when it reduces the number of merge passes necessary to complete the sort.

Tape Usage

The Basic Sort requires 2M tapes, where M is the order of merge. An additional tape is required if the system is residing on tape rather than on a 1301 Disk File. If unreadable records and dictionaries are to be saved and checkpoints are to be taken, an additional tape is required and must be attached as SYSCK2. If no tape is attached as SYSCK2, no checkpoints will be taken and unreadable records and dictionaries, if any, will be deleted. If control cards are on tape, an additional tape is required; otherwise, control cards can be read from the on-line card reader. Sort always reads control cards from the unit attached as SYSIN1.

Tape Labels

Sort uses the standard labeling conventions of 709/7090 IOCS (see the publication, IBM 709/7090 Input/Output Control System, Form C28-6100-2), but provides the option of using nonstandard labeling procedures. (See "File Structure and Tape Record Format.")

Optional Program Features

The following options, which are specified by parameters in a control card, are available to the user during execution of a sort and/or merge run:

1. Checksums -- If desired, a checksum will be computed for each logical record of a sort, and will be carried for the entire sort. It is dropped before the final output is written on tape.
2. Variable Blocking -- If this option is specified, short input tape blocks will be accommodated by the fixed-length portion of the program, provided their length is a multiple of the length of the logical input record.
3. Maintaining equal records -- If desired, any equal records encountered by the sort will be written

in the output file in the same order that they appeared in the input file.

4. No checkpoints -- Checkpoints are normally written by Sort (see "Checkpoint and Restart Procedure"), but can be omitted, if desired.

5. Buffers -- Normally, fixed-length records are "scatter read" into core storage. This option, however, allows the reading of records into buffers instead, thereby providing access to the records for modification.

6. Relocate COMMON -- This option provides for relocating, in core storage, the area reserved for the common parameters of the program.

7. Deletions -- If desired, records can be selected and deleted from the file on the basis of control card information.

7090/7094 SORT PHASES

Sort is divided into four distinct phases: the Edit Phase, the Internal Sort Phase, the Merge Phase, and the Final Merge Phase.

In addition to the four phases, the program contains the Sort Monitor, which communicates with IBSYS and initializes the Sort program, and the Post-Processor, which prints out information on the completed sort run and then returns control to the Edit Phase. Edit Phase then checks for another sort run. If another sort run is not pending, control returns to the Basic Monitor.

The Edit Phase

The Edit Phase is the first phase of Sort. It is called by the Sort Monitor, and performs the following functions:

1. Card Analysis -- During Edit Phase, the control cards are loaded into core storage and the parameters are analyzed. Card analysis consists of the following:
   a. The control cards are checked for any errors that may have resulted from mispunching.
   b. The control cards are checked to make sure that all required parameters are supplied to the system. The given parameters are also checked for the proper format.
   c. Messages are printed on the on-line printer to inform the operator of any errors or possible errors in the control cards.
   d. After being checked, the information is placed in the relocatable COMMON tables.

2. Building of Program Lists -- Edit Phase determines which of the subprograms on the Library tape will be required in subsequent phases, and the names of these programs are placed in program lists for the Internal Sort, Merge, and Final Merge Phases. These program lists will later be used by each phase to call the necessary subprograms for that phase.

3. Calculation of Internal Parameters -- The internal parameters used by the other three phases are calculated by the Edit Phase and placed into relocatable tables (the COMMON area, see Appendix B) for later use.

During the calculations, the following message may be printed on-line:

DIVIDE CHECK TEST AT LOCATION XXXXX.
CHECK FOR ILLOGICAL PARAMETER OR INSUFFICIENT MEMORY.

Possible causes of the difficulty are:

a. The blocking factor in the input file is too large, i.e., there are too many logical records per tape record.
b. The amount of storage reserved for modification programs is too large.

Examination of the specific calculation being executed should indicate the cause.

In fixed-length sort jobs, core storage is allocated as follows:

a. Three M tables -- the M tables are partial sort lists. The size of the M table is determined by the number of logical records in each input block. One word is needed for each logical record to be sorted.
b. Two L tables -- the L tables are used to merge the sorted words from the M tables. For this reason, their size must be a multiple of the size of the M tables. This ratio, called R, is in the range 2 through 10 and is computed so as to minimize the sorting time for the given file parameters.
c. Record Storage Area (G table) -- the area in core storage where the actual records are stored.

Example:

Suppose that after all the programs are loaded, 23,000 words are available for building the L and M tables and for record storage. The input blocksize is 2,000 words, and the logical record size is 2 words. Since the control fields are extracted from the logical record and are placed in a word(s) preceding each logical record, this must be taken into consideration when computing the Record Storage Area. Core storage requirements would be:

M tables -- 1,000 words in each table since each input block contains 1,000 logical records.

L tables -- with a ratio (R) of 4, each L table would be 4,000 words long.

The total number of records in core storage at any time is equal to L + 2M.

3 M tables	-(1,000) (3)	= 3,000
2 L tables	-(1,000) (4) (2)	= 8,000
Record Storage	-(6,000) (2+1)	=18,000
(one control field word)		29,000 words

In this example, the capacity of core storage is exceeded. If the input blocking were 1,000 words, core storage allocation would be as follows:

2 L tables	-(500) (4) (2)	= 4,000
Record Storage	-(3,000) (2+1)	= 9,000
(one control field word)		14,500 words

This is within the limits of the available core storage. However, in this case, R would probably be set to 7, and core storage would be allocated as follows:

3 M tables	-(500) (3)	= 1,500
2 L tables	-(500) (7) (2)	= 7,000
Record Storage	-(4,500) (2+1)	=13,500
(one control field word)		22,000 words

For variable-length logical records the Record Storage Area is divided into "bins" of a length calculated to optimize the use of core storage. Records are subdivided into as many bins as are necessary to contain it. The first bin of a record must contain all the control fields being used for sorting. If more than one bin is needed to contain the record, the location of the second bin is given in the last word of the first bin. The process of chaining bins continues until the entire record is accommodated. The format of the last word is:

IOSP Y,,N

where: Y is the address of the next bin, and
N is the number of words used in the next bin.

The following is the table allocation for the Internal Sort, variable-length records:

1. Read Buffer -- The area in core storage into which the input records are read. The minimum size of the buffer is 2,000 words; the maximum size, 5,000 words.
2. Buffer Table -- There is one word in the Buffer Table for every 100 words in the Read Buffer.
3. Locate Table -- There is one word in the Locate Table for every logical record in the Read Buffer.
4. Merge Tag Tables -- There are 2 Merge Tag Tables (L tables), each equal in words of core storage to the number of bins plus three words. GET, PUT, and VXS use the same L tables for their operations. The main control program of the Internal Sort Phase synchronizes the use of these tables.
5. Sort Table -- This is equal in words of core storage to the number of logical records which can be sorted at one time.
6. Block List for PUT -- This is equal in words of core storage to the maximum number of tape blocks written when PUT is executed.
7. Last-Record Holding Area -- This is equal in words of core storage to the length of all control fields in the last record written onto the indexed merge tape.
8. Approximately eight words are used to divide the above areas.

The Internal Sort Phase

The Internal Sort Phase is a one-pass phase. Its function is to arrange the input records in sequenced strings, one or several output blocks long, the output block being as large as can be accommodated by the Merge Phase. The output of the Internal Sort phase is divided among the number of tapes represented by the order of merge.

The method used by this phase in producing the sequenced strings is a modified continuous merge. Records are read into a record storage area, and are sorted in this area. A smaller number of records are then read into an auxiliary area and are sorted and merged with the sorted records in the record storage area. A number of records comparable to the sorted records is now selected and written as output, and an equal number of records is read in to continue the process.

Unless the option of buffering is requested, records are read into the record storage area by a "scatter-read" procedure, and are not moved during the internal sort processing. Instead, the sorting and merging are done using tag tables. When they are selected as output after sorting and merging, the records are written by a "scatter-write" procedure.

The selection of records to be written is made in such a way as to produce the longest possible strings. This internal sort method takes advantage of any sequences already existing in the file to produce unusually long strings. For random records, strings about twice the size of the record storage area can be obtained. Since the number of strings formed in a given file varies inversely with the length of the strings, longer strings definitely reduce the number of merge passes necessary to complete the sort.

The Merge and Final Merge Phases

The Merge and Final Merge Phases are designed to handle as high an order of merge as the machine configuration will allow. Generally, the higher the order of merge, the fewer the merge passes required. An increase in the order of merge, therefore, frequently reduces the total execution time of a sort or merge run. On the other hand, increasing the order

of merge reduces the size of the output tape block for the same available core storage locations. If the size of a tape block is too small, the input/output operation is slowed because of the inter-record gap time. The internal blocksize is calculated by the Edit Phase depending upon the available core storage and order of merge.

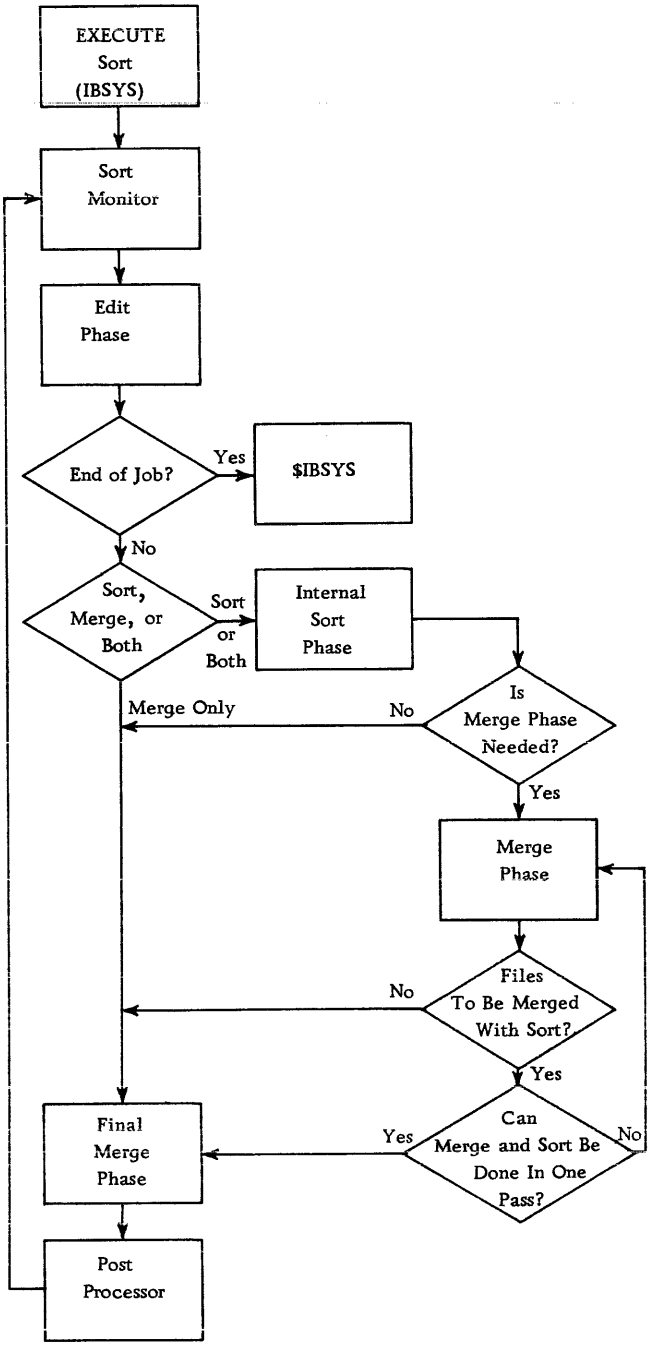
The timing of the merge phases is usually the read/write or tape movement time, since processing (the internal merging of the records) can usually be done within the tape time. The phases are designed so that the reading and writing of records is continuous.

The records in the input blocks have already been arranged in sequence by the Internal Sort Phase. During the Merge Phase, the sequenced strings are merged together in succeeding passes until the number of strings, together with the number of files to be merged, is equal to, or less than, the order of merge. The Final Merge Phase is then loaded, and the last merge pass is performed. The final output is written according to the format specified by the control information.

The Final Merge Phase also handles the merging of previously sorted files, if any, with the records being sorted.

System Flow of Control

The figure at the right shows system flow of control.



System Flow of Control

Storage Allocation in Merge Phases

In order to obtain the maximum overlap of reading, computing, and writing operations in the merge phases, the following technique of assigning record storage areas is used:

For a balanced merge (i.e., one channel used for input and the other for output) of merge order M, 2M + 1 storage areas are needed, each equal in length to the maximum number of words in a tape record on the merge tapes. At least one of these areas is assigned to each tape at any given time, with the extra or "floating" areas assigned as needed to the tape with the highest reading priority, as discussed below. An area becomes a floating area as soon as it is released from its previous assignment.

In the case of a 3-way balanced merge, the number of areas needed is 7. If these are designated by the letters A-G, the Merge Phase starts with the following assignments:

TAPE 1	TAPE 2	TAPE 3	
A	B	C	
FLOATING AREAS			
D	E	F	G

Initially, one tape record is read from each of the three merge input tapes, filling areas A, B, and C. (At this point, areas D-G are floating.) Reading is then continued from the file having the smallest number of areas currently assigned to it, or if more than one file has this same minimum area assignment, priority goes to the file with the lowest number.

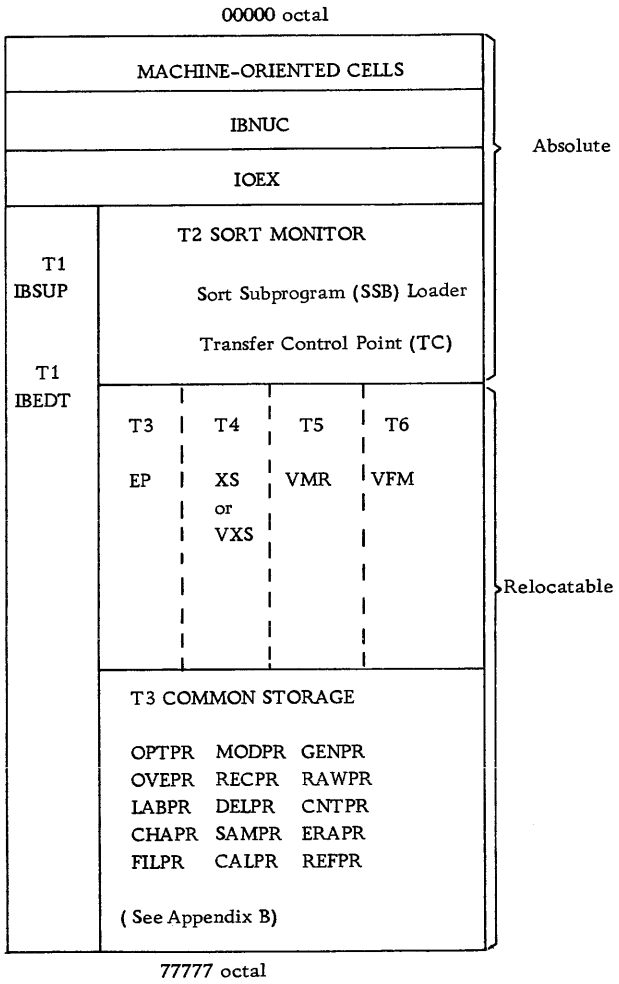
If a record is to be read and no area is available, the program will continue merging and writing until an area is available.

Eventually there will be a sequence break in one of the files. In this case, the area with the new sequence is not used for merging, and stands by until all files have had a sequence break. There will be

three areas in use, each with a block from the new sequence, and the entire process is repeated.

In the final merge pass, two buffers are used for building the output blocks, which may be larger than the merge blocks. Only 2M record storage areas are used, and the priority of reading is determined as in the merge phase.

CORE STORAGE LAYOUT



NOTE: The origin at the upper limit is fixed, but it may be changed using the OPTION Card.

In the diagram above, the symbols T1-T6 indicate the time sequence of the various stages of Sort. The symbols are explained as follows:

T1 -- Calling of 7090/7094 Sort

Sort is called by the Basic Monitor (IBSYS) through a

```
$EXECUTE    SORT
```

control card (see "Basic Monitor Control Cards"). IBSYS defines the computer and provides the necessary information on the availability of input/output components. All I/O availabilities are a function of IBSYS.

T2 -- Loading of Sort Monitor

IBSYS loads the Sort Monitor. At this point, control passes to the Sort Monitor. The Monitor includes the Sort Subprograms Binary (SSB) Loader, which loads the subprograms proper to each phase.

T3 -- Loading of Control Cards and Execution of Edit Phase (EP)

The control cards are always loaded from SYSIN1, the symbolic designation of the IBSYS input function. Edit Phase is then executed. The parameters needed to execute the sort are generated from the control information and placed in COMMON storage starting at location $(77777)_8$. The subprograms loaded in this phase are as follows:

ASSIGN	RESTAR
EP001	SK001
SOP	LABEL
RB01	BTD
RELEAS	CALC

T4 -- Loading and Execution of Internal Sort Phase

The Internal Sort (XS - fixed-length and VXS - variable-length) subprogram is loaded over Edit Phase, along with the subprograms required by this phase. The area in core storage used for sorting extends from the first word after the last subprogram loaded to the last word before COMMON storage. The subprograms loaded in this phase are as follows:

IOBS*	DELETE*	WTFIX	VXS
CKPT*	DEPAD	WRSEL	GET
CKSUM*	EQUALS*	SOP	PUT
DEBLK	FXMOV	RESTAR	XTRACT
IOSS	LABEL	XS	

* optional

T5 -- Loading and Execution of Merge Phase

The Merge Phase (VMR) subprogram is loaded over the Internal Sort Phase, along with the subprograms necessary to this phase. These subprograms are as follows:

BTD	LABEL
IOBS	SOP
CKPT	RESTAR
CKSUM*	WRSEL
DEBLK	VMR

* optional

T6 -- Loading and Execution of Final Merge Phase

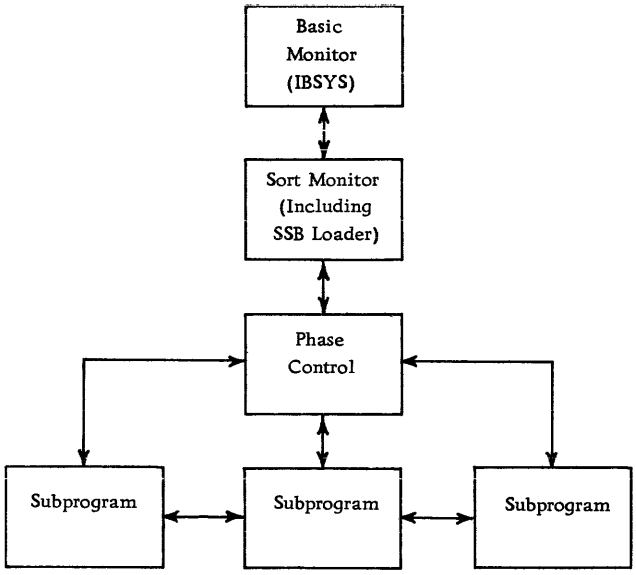
The Final Merge Phase (VFM) subprogram is loaded over the Merge Phase. This phase is identical to the Merge Phase except for its initialization procedures. During this phase, which is effectively a separate merge program, previously sorted files may be merged with the file currently being sorted. The subprograms loaded in this phase are as follows:

BTD	LABEL
IOBS	POST
CKPT	SOP
CKSUM*	RESTAR
DEBLK	WRSEL
DELETE*	VFM

* optional

After the sort run is completed, control passes from the Final Merge Phase to the Post-Processor. The Post-Processor releases all tape assignments; prints counts of records sorted, records deleted, records dumped, padding records added, and reserve units used for intersystem communication; and returns control to the Sort Monitor. The Sort Monitor reloads the Edit Phase, and the next control card is read. If the next control card is a $IBSYS card, control is returned to IBSYS. If another set of Sort control cards is read, sorting is resumed.

PROGRAM COMMUNICATION



1. Interjob control passes to 7090/7094 Sort through the Sort Monitor.
2. Interphase control is a function of the Sort Monitor and the control program for a particular phase.
3. Intraphase (or inter-subprogram) control is a function of the Transfer Vector technique of 7090/7094 Sort. Subprograms can communicate with each other and with the control program for any of the Sort phases. No distinction is made between a user subprogram and a Sort subprogram.

GENERAL SPECIFICATIONS

Input Records

Input to the fixed-length version of 7090/7094 Sort is fixed-length records written in either binary or BCD, signed or unsigned. The maximum size of the input tape records is 2,000 words, not including the checksum and sequence word. The minimum size is three words, in keeping with tape record error recovery conventions now in use on the 7090. The records must be grouped into blocks of equal size, except where the VARIABLE BLOCKING option is specified (see the section "OPTION Card").

High or low padding may be used to fill out any blocks of an input file. The output file may also be padded, either with low padding at the beginning or with high padding at the end, or it may be left unpadded, as desired.

Input to the variable-length version of 7090/7094 Sort is variable-length records written in either binary or BCD, signed or unsigned. A description of the variable-length block format and variable-length record format follows:

Variable-Length Block Format

A block (tape record) may consist of one or more variable-length logical records as long as the total number of words in the block is no more than 2,000. The checksum-block sequence word, which may be appended at the end of a binary tape block, is not included in this total.

Variable-Length Logical Record Format

1. Binary Mode Tape -- Binary logical records must be preceded by a control word containing the length, in words, in the decrement. The length does not include the control word itself. The format for this word is:

   IOCTN **,,n

   The minimum length of a variable-length record is two words plus its associated control word. This will insure a minimum three word tape block as required by the Sort program. An exception to the above is the ability to sort one word logical records if the checksum-block sequence word is appended as part of the tape block, thus insuring the minimum three word tape record.

2. BCD Mode Tape -- BCD logical records must contain, as the first word of the record, a control word containing the number of characters in a record. Included in this number are the six characters of the control word. The record length is expressed as five, left-justified, BCD characters of the control word. Any characters to the left that are not used should be zeros.

   Example: 0 0 0 3 6 C

   The sixth character in the control field may be any valid BCD character. The record length must be at least 18 characters, and must be a multiple of six.

   If the binary input record to the variable-length version of the Sort has the word count in the decrement, and the rest of the word is not in the standard format, the final output will be made to agree with the standard,

   IOCTN **,,n

   When the input record is in BCD and the requested output is binary, the length on the output tape will be expressed by a

   IOCTN **,,n

9PAC Type Records

The 7090/7094 Generalized Sorting Program will not sort 9PAC logical records that are fixed-length within a specific record type and that vary in length from type to type. 9PAC files, however, can be prepared so that they will be sorted by the 7090/7094 Generalized Sorting Program. This preparation is described in the publication, IBM 7090 Programming Systems: SHARE 7090 9PAC Supplement, Form J28-6211-1.

Control Fields

The control fields of a record may be expressed in either bits or characters. If control fields are to be sorted in accordance with the commercial collating sequence, the control fields may be expressed in either characters or bits, but if in bits, the number specified must be a multiple of six.

It is possible to intermix, in the same record, control fields to be sorted in ascending sequence with fields to be sorted in descending sequence. There is no restriction on the number of control fields or their length. However, the number of words has been effectively set at 200 for the distributed version of the program. If necessary, this number can be increased by changing, in the Edit Phase, BES 200 to BES xxxx, where xxxx is the number of control fields

desired, at the following symbolic locations:

FILEN	FILLB
BIEUS	FILNW
FILSW	FILLM
FILSB	FILRM

BASIC MONITOR CONTROL CARDS

Two Basic Monitor (IBSYS) control cards are required to use Sort, and another IBSYS control card may be used with Sort.

1. $EXECUTE SORT
   This card calls Sort, and causes control to be transferred from IBSYS to the Sort Monitor.
2. $IBSYS
   This card transfers control from Sort to IBSYS, and causes the Basic Monitor Supervisor (IBSUP) to be restored from tape.
3. $ID
   The use of this card with Sort is optional. It causes a transfer of control to the installation accounting routine.

SORT CONTROL CARDS

In order for a file to be sorted, certain information must be supplied to the sort program. This includes a description of the logical records, the input and output files, and the control fields on which the records are to be sorted. Other information dealing with options, labels, and modifications may also be needed. Thirteen control cards have been designed with this functional division in mind.

Card	Description
FILE	Defines file characteristics.
SORT	Specifies sort.
RECORD	Defines logical record(s).
MERGE	Specifies merge.
CHANNELS	Specifies the input, merge, and output channels.
LABEL	Used with nonstandard labels.
OPTION	Lists the options desired.
MODIFICATION	Specifies the names of modification programs.
DELETE	Specifies records to be deleted from a file.
OVERFLOW	Used with an overflow sort.
RESTART	Initiates restart procedure.
REMARK	Remarks card.
END	Indicates the end of the control card deck.

In a normal sort run, only the FILE, SORT, CHANNELS, RECORD, and END cards are needed.

The Sort control cards have variable field formats, i.e., the order of the fields in the card is not critical, provided the first field on the card is the card name or identifier. All information pertinent to a job must be contained in columns 7 through 72. However, fields can be eliminated if they are unnecessary for a job.

Control Card Notations

Certain special characters and specified formats are used either to define or to separate fields or subfields.

1. Fields are defined by the first three BCD characters in the word describing the field. The presence of the BCD characters may give sufficient information to the program, or other definitive information may follow.

   Examples:

   SORT in the SORT card need only be SOR;
   MOD effects the same action as MODIFICATION.

2. Slash / -- The slash follows all BCD field definers with which a parameter is associated.
3. Parentheses ( ) -- Parentheses are used when more than one parameter is needed for a BCD field definer. The left parenthesis follows immediately after the slash, and the right parenthesis is used when the end of the parameters is reached.
4. Comma , -- The comma is used in three ways:
   a. To separate one field definer and its parameters from another field definer and its parameters.
   b. To separate one parameter from another when they are enclosed within parentheses.
   c. To separate the name of the card from the definers and parameters on the card.
5. An X is used to indicate a continuation of the control card when it is punched in column 6. This card must immediately follow the control card which it continues.

It is not necessary that the control statements, i.e., a control card and its continuation cards, if any, be arranged in any specific order. The only restriction is that the END card must be the last control card.

FILE Cards

There are FILE cards in the Sort program for both input and output functions. The FILE card for input contains the information describing the physical characteristics and organization of the data which is necessary to read the file into core storage from the input unit. The FILE card for output contains the information from which the processed data will be organized into a file on the output unit.

Input FILE Card

The complete format for the Input FILE card is as follows:

FILE, INPUT/#, REELS/n or L, MODE/B or D, DENSITY/H or L, BLOCKSIZE/n, PADDING/H or L. SERIAL/#, RLSEQ/#, CKPT/S or N, NAME/xxH...,CKSUMS, BLKSEQ, DICT/H or L, LABEL/$\begin{matrix}S & H\\ or & or.\\ N & L\end{matrix}$

where:

INPUT/#

Defines this FILE card as an Input FILE card. The special character # is a number up to four digits in length which is assigned to the input file. The number is used whenever this file is referenced, e.g., in the SORT statement.

REELS/n or L

The n designates the number of reels in the file. L indicates that the file is to be processed under label control. The L feature is operative only in the variable-length version. This field may be omitted if the file is contained on one reel.

MODE/B or D

Indicates the mode in which the file is written; B indicates binary, and D indicates BCD. This field may be omitted if the mode is BCD.

DENSITY/H or L

Density of the file, high or low. This field may be omitted if the density is high.

BLOCKSIZE/n

Designates the number of words per tape block, if the records are fixed length, or the maximum tape block size in words, if the records are variable length. This number does not include the checksum or block sequence word in its block count.

PADDING/H or L

Padding is used only with fixed-length records. The H designates high padding, and the L signifies low padding. Whether a character is high or low depends on the collating sequence being used, and on how the sorting is being done, in ascending or descending order.

SERIAL/#

File serial number. This field is optional, and is omitted if the file is not labeled.

RLSEQ/#

Reel sequence number of the file. This field is optional, and is omitted if the file is not labeled.

CKPT/S or N

If S (standard) is designated, the file carries a checkpoint on every reel except the first. If N (nonstandard) is designated, the file carries a checkpoint on every reel, including the first. If no checkpoint is carried, this field is omitted.

NAME/xxH...

Name of the file. The xx represents the number of alphameric (Hollerith) characters, including blanks, in the file name. This number must not exceed 18. The name of the file immediately follows the character H.

CKSUMS

When specified, a half-word checksum is carried in each tape block. It is permitted only in binary files.

BLKSEQ

When specified, a half-word block sequence is carried in each tape block. It is permitted only in binary files.

DICT/H or L

When specified, a dictionary is carried in the file. If the file is labeled and neither H nor L is designated, the dictionary is assumed to have the density of the label. If the file is not labeled, then H or L must be designated; H indicating high density and L indicating low density.

LABEL/$\begin{matrix}S & H\\ or & or\\ N & L\end{matrix}$

The file carries a standard label in high or low density, or a nonstandard label in high or low density. The four possibilities are SH, SL, NH, and NL. If the file carries no label, the LABEL field should be omitted.

If the standard labeling conventions of IOCS are used, SERIAL number, NAME, and RLSEQ number are given in the label, and are checked by Sort only if they are included in the FILE statement. If they are not included, they are not checked.

The Input FILE statement may, therefore, be considerably condensed. For a one-reel file, it may be as concise as the following:

FILE,INPUT/44, BLOCKSIZE/1000, LABEL/SH

or even,

FIL, INP/44, LAB/SH, BLO/1000

Output FILE Card

The complete format for the Output FILE Card is as follows:

FILE, OUTPUT, MODE/B or D, DENSITY/H or L, BLOCKSIZE/n, PADDING/H or L, SERIAL/#, RLSEQ/#, NAME/xxH..., RETAIN/n , CKSUMS, BLKSEQ, DICT/H or L, LABEL/ S or N H or L.

where:

OUTPUT

Defines this FILE card as an Output FILE card.

MODE/B or D

Mode in which the output file will be written; B designates binary, and D designates BCD.

DENSITY/H or L

Density of the output file; either high or low.

BLOCKSIZE/n

Designates the tape blocksize in words, if the records are fixed length, or the maximum tape blocksize in words, if the records are variable length. This number does not include the checksum or block sequence word in its block size.

PADDING/H or L

Padding is used with fixed-length files only. The first or last tape record may be padded; H indicates high padding, and L indicates low padding. If this field is omitted, no padding will be added to the output file.

SERIAL/#

Designates the file serial number. If standard label is specified, this field should be included.

RLSEQ/#

Designates reel sequence number. If standard label is specified, this number is used as the initial reel sequence number. If this field is omitted, the reels are sequenced starting with the number 1.

NAME/xxH...

Name of the output file. The xx represents the number of characters, including blanks, in the file name. This number must not exceed 18. The name of the file immediately follows the character H. This field is only for files with standard labels.

RETAIN/n

Number of days the file is to be retained. This field is only for files with standard labels.

CKSUMS

When specified, a half-word checksum is carried in each tape block. It is permitted only in binary files.

BLKSEQ

When specified, a half-word block sequence is carried in each tape block. It is permitted only in binary files.

DICT/H or L

When specified, a dictionary is carried in the output file. It is permitted only if the input file contains a dictionary. If the file is not labeled, H or L must be specified for the density of the dictionary.

LABEL/ S or N H or L

Indicates that a file is to have a standard or nonstandard label, in high or low density. The four possibilities are SH, SL, NH, and NL. If no label is desired, this field may be omitted.

If the output file is not to be labeled, the Output FILE statement may be reduced to the following:

FIL, OUT, MOD/B, DEN/H, BLO/n

RECORD Card

The general format of the RECORD card is as follows:

RECORD, TYPE/F or V, LENGTH/(nL1, nL2, nL3), FIELD/(n_1 B or C U or S, n B or C U or S, ..., n_i B or C U or S)

where:

TYPE/F or V

Designates fixed - or variable-length records.

LENGTH/ (nL1, nL2, nL3)

For the fixed-length version, nL1 is the input record length in words, nL2 is the record length of the input to the Merge Phase, i.e., output of the Internal Sort, and nL3 is the record length of the output from the Final Merge Phase. When the length of records is to be modified during the sort, nL2 and nL3 are designated.

Restriction: It is impossible to lengthen records in the Final Merge Phase. If lengthening is desired, it should be done during the Internal Sort Phase. Records can be shortened during either the Internal Sort Phase or the Final Merge Phase.

For variable-length records, nL1 refers to the minimum size variable-length record, nL2 refers to the maximum size record length, and nL3 refers to the major record length.

Example: Suppose that in sorting a file, the smallest record is 5 words long, the largest record is

100 words long and about 80 per cent of the records are 20 words long. The LENGTH definer and parameters would be written:

LENGTH/ (5,100,20)

The restriction for lengthening records applies to the variable-length records as well as the fixed-length records.

FIELD/ (n_1 B or C U or S, n_2 B or C U or S, ..., n_i B or C U or S)

Defines the fields of a logical record.

n - specifies the length of the field.

B or C - specifies either bits or characters.

nB specifies the length in bits, and
nC specifies the length in characters.

U or S - specifies a signed or unsigned field.

Characters are assumed to be the standard units of field measurement; if neither B nor C is given, C is assumed.

If a field is unsigned, it is not necessary to write the U in the field description of the RECORD card. All signed fields, however, must be designated by an S after the length of the field. For signed fields, the sign is included in the length of the field. Unsigned fields are sorted logically, and signed fields are sorted algebraically.

Examples:

Example 1: Fixed length

RECORD, TYPE/F, LENGTH/12,
FIELDS/(12,2,10,8)

This statement identifies a fixed-length record of 12 words with four fields of 12, 2, 10, and 8 unsigned characters.

If only the first and fourth fields are desired for the sort and are signed, the second and third fields can be combined. The RECORD card is then written as follows:

REC, TYP/F, LEN/12, FIE/(12S, 12, 8S)

Example 2: Variable length

REC, TYPE/V, LEN/(5, 100, 20),
FIE/(36B,72B,12S)

This statement identifies a variable-length record with a minimum length of five words. The length of the fields defined cannot exceed the length of the minimum record. In the above example, the defined fields contain 36 unsigned bits, 72 unsigned bits, and 12 signed characters.

SORT and MERGE Cards

The general format of the SORT statement is:

SORT, FILE/#, SEQUENCE/C or S,
ORDER/n, FIELDS/(#1 A or D, ..., #i A or D)

where:

FILE/#

Designates the number of the file as given in the FILE statement. (See "Input FILE Card.")

SEQUENCE/C or S

Designates the collating sequence to be used. (See Appendix A.) C specifies the commercial collating sequence, and S specifies the scientific. If this field is omitted, S is assumed.

ORDER/n

Designates the order of merge.

FIELDS/(#1 A or D, ..., #i A or D)

Specifies the control fields on which the records are to be sorted (see "RECORD Card"), arranged in descending order of importance. The A and the D indicate that a particular field is to be sorted in ascending or descending order. If this designation is not included, A is assumed.

The general format of the MERGE statement is:

MERGE, FILES/(#1,#2,...#i), SEQUENCE/S or C,
ORDER/n, FIELDS/(#1 A or D, ..., #i A or D)

If both a sort and a merge are required in a single run, both SORT and MERGE statements must be used. The MERGE statement, when used in conjunction with a SORT statement, can be abbreviated by writing only the file numbers and eliminating the rest of the fields.

Example:

SORT, FILE/12, SEQUENCE/S, ORDER/5,
FIELDS/(5,1,4,3,2) MERGE, FILES/(2,3)

CHANNELS Card

The CHANNELS card is used to:

1. specify the channel on which the input to be sorted or merged is mounted.
2. specify the channels that are used for merging.
3. specify the output channel.

The actual physical units to be used are determined through the Basic Monitor System and its Availability Table.

The general format of the CHANNELS statement is:

CHANNELS, INPUT/A-H or J-Q or UTX, MERGE/(Y1, Y2) or (Y1, Y2, UT),
OUTPUT/A-H or J-Q

INPUT/A-H or J-Q or UTX

Designates the channel on which the input is to be mounted. If A-H is designated, the true channel is assigned for use. The actual physical unit on that channel cannot be specified. It is possible, however, to specify the following:

INPUT/A3

The physical unit A3 is not assigned as the input

unit, but the third unit in the Availability Chain for channel A is selected as the input unit. If J-Q is specified, the input channel is obtained through intersystem reserved units. Normally, a number is associated with the symbolic channel character, i.e.,

INPUT/K5

If no number is specified, zero is assumed. INPUT/Q is equivalent to INPUT/Q0. If UTX is specified, the sort will use, as its input channel, the channel to which the UTX function is assigned. In the case of UTX, the X represents numbers 1 through 4. Caution should be used in assigning UT4 as input, since this tape is used for the IBSYS core storage dump. If more than two units are to be used for input, it is possible to specify the following:

INPUT/(A3, A5) or
INPUT/(UT1, UT3) or
INPUT/(K2, K3, K4, K5, K6)

The latter example is used only with the merge. If INP/A-H is designated and two or more reels are specified on the FILE card, two units will be assigned for input. If INP/UTX is specified with two or more reels of input, only one unit, UTX, will be assigned.

MERGE/$\begin{matrix}(Y1, Y2)\\ \text{or}\\ (Y1, Y2, UT)\end{matrix}$

Y1 and Y2 are real channels (A-H) that are to be used for merging. They cannot be the same channel. If UT is specified, the system may use the utility tapes for merge tapes. The utility tapes must be attached as units on channels Y1 and Y2 before they are used as merge tapes. SYSUT4 will not be used, since this tape is used for IBSYS core storage dumps. There is the possibility that the final output will be written on a tape assigned as a systems utility tape (SYSUTX). Therefore, care should be used so that the final output is removed from the unit before the next stacked job is executed.

OUTPUT/$\begin{matrix}A\text{-}H\\ \text{or}\\ J\text{-}Q\end{matrix}$

Designates the channel on which the output is to be written. Most of the time it will not be necessary to specify the output channel, since it is difficult to determine in advance on what channel the output will reside at the end of a job. If the channel on which the output would normally reside is other than the specified channel, an extra pass will be necessary during the Merge Phase to rectify this. However, if a third channel is specified as the output channel, an extra pass is not necessary.

A-H represent true channel designations, whereas J-Q represent symbolic channels. As with the input, J-Q normally has a number associated with the symbolic channel. When J-Q is specified, the output tapes are rewound and the symbolic designation of the tape unit is stored in the address portion of the first control word of the unit control block for that tape unit. This designation can be referred to by another job.

Use of R on the CHANNELS Card

If the letter R is used after any of the parameters of the INPUT, MERGE, or OUTPUT definers, the input or output tape is rewound. If the R does not appear, the tape is rewound and unloaded.

Example:

CHA, INP/UT1R, MERGE/(A, B), OUT/A.

This example indicates that the input tape is to be rewound, whereas the output tape is to be rewound and unloaded.

If R is used with an intersystem reserved input designation, the unit will be made available after Sort has finished using the input tape.

An R is not necessary following an intersystem output unit designation since the unit will not be unloaded.

If a Sort is being run, the program will assign a merge order number of units on each of the merge channels. If there are not enough units available for merging, the program will check if the units assigned to input can be used for both input and merging. If enough units are still not available, the program will halt and the operator is given two choices: either to interrupt the system and make more units available, or to delete the job. If the output channel is other than a merge channel, the system will assign, depending on availability, a number of output units which is equal to, or less than, the order of merge. At least one unit is required. If a merge is being run, a merge order number of tapes is assigned on the input channel, and the output channel is assigned as described above. The proper format for writing the CHANNEL card for a merge could be:

CHANNEL, INPUT/A, OUTPUT/B or
CHANNEL, INPUT/(M5, M6, M3, ... etc.),
OUTPUT/B or
CHANNEL, INPUT/(M5, M6, M3, ... etc.),
OUTPUT/J3

LABEL Card

The format for the LABEL card is:

LABEL, IDENT/xxH....

where:

IDENT/xxH....

Indicates the length, in characters, of the label to be written on the output tape. The number designated must be not less than 18 and not greater than 84, and must be a multiple of 6. If the number of characters in the label is greater than the number specified, the extra characters are truncated. If the number of characters in the label exceeds the limits of one

card, the rest of the characters must be contained in a continuation card, starting in column 7. If the remainder of the label is all blanks, the continuation card must still be included.

Use of the LABEL card is permitted only when a nonstandard label is specified on the Output FILE card. If a nonstandard label is specified and the LABEL card is not present, it is assumed that the same label that is on the input file is to be written on the output file.

OVERFLOW Card

This card is needed if the number of records to be sorted exceeds the maximum number which can be handled by the program. The maximum number will vary with the order of merge. If an overflow condition occurs, a message is printed explaining the situation and the number of tape records already sorted is designated. The format of the OVERFLOW card is:

OVERFLOW, BLOCKS/n

where:

BLOCKS/n

Designates the number of tape records of the current input tape that have been processed. This number does not include any label, checkpoint, or dictionary records. The number of tape records processed is expressed in decimal notation.

OPTION Card

The format for the OPTION card is:

OPTION, CKSUMS, NOCKPT, EQUALS, RELCOM/n, BUFFER, VARIABLE BLOCKING, MAPS, CARDS, TAPES

where:

CKSUMS

This option, when specified, causes a checksum to be computed for each logical record and retained for the entire sort. The checksum is dropped before the final output is written on tape.

NOCKPT

When NOCKPT is specified, it indicates that no checkpoints are to be written during the sort. Normal checkpoints are, therefore, deleted. If there is no tape assigned as SYSCK2, the NOCKPT option will be assumed. If no checkpoints are taken, restart is impossible except from the beginning of job.

EQUALS

This option, when specified, will instruct Sort to keep all equal records in the same order as they appear in the input file. If records are completely equal, they will not be deleted from the file. Deletion of equal records can be accomplished by a customer modification.

RELCOM/n

This option allows for the possibility of reducing the size of core storage by the number of locations specified. Core storage would be reduced by taking the locations away from the top of core storage and relocating COMMON downward. (See COMMON, Appendix B.)

BUFFERS

The buffer option allows for fixed-length records to be read into buffers instead of being scatter-read into core storage. This option is used if work is to be done on the records before they are sorted.

VARIABLE BLOCKING

The option that is used with the fixed-length input allows a short block to be accepted on the input tape, provided its actual length is a multiple of the length of the logical record.

MAP

Prints on-line, the loading locations of all subprograms, calculations, and between-phase and pass counts.

CARDS

Prints all control cards on-line. The normal mode is not to print any control cards except the REMARK card. Printing of this card cannot be suppressed.

TAPES

Prints on-line, all tapes assigned for sorting and merging. The program then pauses so that all units can be readied. If this option is not specified, the program will not pause.

Example:

Suppose that a user wishes to reserve 1,200 locations in upper core storage for his own use and that he also desires to keep all equal records in the same order as they appear in the input file. In this case, the OPTION statement would be:

OPTION, RELCOM/1200, EQUALS

MODIFICATION Card

The following is the format for the MODIFICATION Card:

MODIFICATION, PROGRAM/xxxxx, CELLS/n, LOCATION/SYSxxx, FORMAT/U or B

where:

PROGRAM/xxxxx

This designates the modification. The xxxxx portion must conform to the modification names defined by Sort. Twenty-five names are available for modifications by this card: 10 names can be used in the Internal Sort Phase, 5 names in the Merge Phase, and 10 names in the Final Merge Phase. The names will have the following format:

Internal Sort Phase	XSM01 through XSM10
Merge Phase	MPM01 through MPM05
Final Merge Phase	FMM01 through FMM10

CELLS/n

This is used to designate the number of core storage locations the program and its parameters will occupy; n must be a decimal integer.

LOCATION/SYSXXX

This informs the Sort program where the modification is located. It may be on any of the SYSUNI functions. If the modification is on tape, it must be relocatable column binary. If the program is in the card reader, it must be relocatable row binary.

If more than one modification is to be used during a phase and several modifications reside on one tape, they must be arranged on that tape in ascending order (XSM01, XSM05, XSM07, etc.). The user must make certain that the tape is correctly positioned so that the requested program can be located. SYSUNI tapes will not be rewound and searched again.

FORMAT/U or B

Informs the program as to whether the modification is blocked as one tape record or is in single card-image format. If the program is blocked, it must have the same format as the regular subprograms on the system tape. The format for the system subprograms is as follows:

IOCT	45000,,n
BCI	1, (name of program)
IOCT	45000,,n
Program Transfer Card	

where n is the number of words in the program.

Example:

MODIFICATION, PROGRAM/XSM03, CELLS/1218, LOCATION/SYSUT1, FORMAT/U

This statement indicates to the program that there is a modification in the Internal Sort Phase which has the name XSM03. When this name is given, the Edit Phase sets an indicator that signifies that XSM03 is to be entered at the appropriate time. The modification occupies 1,218 words.

The program is located on the unit attached as SYSUT1 and is unblocked single card format.

DELETE Card

The DELETE Card has the following format:

DELETE, FIELD/#, IDENT/n$\begin{matrix}O\\B\\H\end{matrix}$ ----

where:

FIELD/#

This is used to specify the field in the record in which the bit pattern will be compared with that given in the IDENT field. Upon an equal comparison the record is deleted.

IDENT/n$\begin{matrix}O\\B\\H\end{matrix}$ ----

This is the specific information used to determine if a record will be deleted; n is the number of bits, characters, or octal numbers that follow; O, B, and H stand for octal, binary, and alphameric (Hollerith) information, respectively. If the given information is found in the specified field in the given record type, then that particular record is deleted from the sort. The given information must cover the full field. The field length, of course, is described in the RECORD Card.

Example:

DELETE, FIELD/2, IDENT/1HZ

This statement directs the Sort to delete records having a Z in field 2.

RESTART Card

The RESTART Card has the following format:

RESTART

The RESTART Card initializes a restart procedure (see "Checkpoint and Restart Procedure").

REMARK Card

The REMARK card has the following format:

REMARK, (any on-line message)

This card prints on-line any desired message.

FILE STRUCTURE AND TAPE RECORD FORMAT

The 7090/7094 Sort accepts as input both signed and unsigned binary or BCD files. The minimum size of a record in words is 3, and the maximum size is 2,000. The output file of 7090/7094 Sort need not be in the same mode or density as the input file. (Note the implications of recording random binary information in the BCD mode as given in the section entitled, "Character Alteration in the BCD Mode," in the publication, 7090 Data Processing System, Form A22-6528.)

Tape labels must be in the BCD mode and must be not less than 3 and not more than 14 words in length. The program uses the standard labeling conventions of 709/7090 IOCS as regards the mode, density, and the EOF on header labels, data blocks, and trailer labels. The program also accepts nonstandard labels (see LABEL Card).

The following conventions apply to all header and trailer labels, the optional checkpoints of the input file dictionaries, the data blocks of the input file, and the EOFs associated with all of these:

Record	Mode	Density	Remarks
Header Label and EOF	BCD	High or Low	The label and the EOF must have the same density; density and mode may, however, differ from that of the file.
Dictionaries and EOF	BCD	High or Low	If a file is labeled, it must be the same density as the label. If unlabeled, density and mode may differ from that of the file. The dictionary and the EOF must have the same density.
Checkpoint and EOF	Binary or BCD	High or Low	The checkpoint and the EOF must have the same density and mode as the data.
Data Blocks and EOF	Binary or BCD	High or Low	The data and the EOF must have the same density.
Trailer Label and EOF	BCD	High or Low	The label and the EOF must have the same density as the data.

The above conventions apply to both input and output files, with one exception: there is no checkpoint file in the output file.

The number of words per tape record must equal the number given in the Input FILE card. The only exception to this rule occurs when VARIABLE BLOCKING is specified in the OPTION card. In this case, the tape record length may be shorter than specified, provided it is a multiple of the length of the logical record (see LENGTH Parameter of the RECORD Card).

The actual tape record length in a binary file may be one word longer than that specified under BLOCKSIZE. This is to accommodate the 18-bit folded checksum (left half of word) and the block sequence number (right half of word) permitted in accordance with the specifications of 709/7090 IOCS. The presence of a checksum and block sequence number is indicated by the CKSUMS and BLKSEQ parameters of the Input FILE card. If CKSUMS and BLKSEQ are specified on the Output FILE card, the program computes an 18-bit folded checksum and a block sequence number and substitutes them in place of those values which existed on the input file.

If either or both the EQUALS and CKSUMS options are specified on the OPTION Card (as distinguished from the CKSUMS which may be specified on the Input and Output FILE Cards), the program modifies the record in the following manner: A number is assigned to every logical record in the file and is placed immediately after the last word of the logical record. The checksum word follows either the sequence number, if it is requested, or the last word of the logical input record, if the sequence number is not requested.

Clockword

On all tapes prepared by the Internal Sort Phase and the Merge Phase, two words are added to the output blocks -- one at the beginning and one at the end. These are called clockwords.

A special bit structure is placed in the first half of the first clockword and the last half of the last clockword. These are used for checking character shifting.

A block sequence number is entered in the last half of the first clockword. The block sequence number is used to assure that the tape is positioned correctly at all times. The block sequence number should, in all cases, agree with the physical record count. If they do not agree, the tape is repositioned to the correct place.

Dictionary

The dictionary is in the first file on the tape. It may be the only information in this file, or it may share the file with the label. In the latter case, its position is between the label and the end-of-file mark. Dictionary records are 14-word BCD tape records. There is no limit as to the number of permissible dictionary records. During the sort, the records are saved on SYSCK2, and are then transferred to the final output file. If no SYSCK2 tape is attached, the dictionary records will be deleted.

SORT SUBPROGRAM BINARY LOADER

The subprograms necessary for each phase of Sort are loaded prior to the initiation of the phase by the Sort Subprogram Binary (SSB) Loader. Thus, storage is not taken up by subprograms used by another phase, and the number of core storage locations used is kept to a minimum for each phase.

The SSB Loader loads the main program of each phase along with whatever subprograms that phase must use. The SSB Loader must, therefore, know how many programs are to be loaded, the names of

these programs, and the input devices on which these programs are situated. This information is supplied to the Loader by its calling sequence and the Program List.

Calling Sequence

The SSB Loader is always in core storage as part of the Sort Monitor. Contact is made with the Loader by a three-instruction calling sequence:

```
TSX   LD001, 4
PZE   ORIGIN, , N
BCI   1, Return
```

where:
ORIGIN contains the initial relocation value for each phase, as supplied by Edit Phase. Loading for a particular phase starts at this location.

N indicates how many words immediately follow ORIGIN in core storage. Return is the name of the program to which control is passed when the loading for a particular phase has been completed. This program must be one of those in the Program List.

Program List

The Program List is part of the Sort Monitor. It contains the beginning load address and the names of the programs to be loaded for a particular phase. It also contains the addresses of the input devices on which these programs are situated. Edit Phase supplies the addresses and the names of the programs required by each phase; the addresses and names are defined as follows:

```
ORIGIN   PZE    xxxxx
         PZE    yyyyy, , M
         BCI    1, NAME1
         BCI    1, NAME2
         .
         .
         .
         BCI    1, NAMEM
         PZE    yyyyy, , M
         BCI    1, NAME1
         BCI    1, NAME2
         .
         .
         .
```

The SSB Loader begins loading at location xxxxx and also uses xxxxx as the initial relocation value, augmented each time by the length of the program just loaded. The address yyyyy signals the Loader as to where the following M programs are situated. This address can specify any of the SYSUNI functions.

When the M programs from yyyyy are loaded, the Loader continues to the next set of M programs specified by the next yyyyy.

Library Tape

The Sort Library tape contains all the programs for the Sort operations along with the Sort Monitor.

The Sort program resides on the Library tape and is composed of blocked card-images in column binary form. In this discussion, any references made to "card" signify a card-image on the Library tape. The subprograms which the Loader must load are placed on the Library tape in the following manner:

```
Program Card
  -Program-
Transfer Card
Program Card
  -Program-
Transfer Card
      .
      .
      .
Program Card
  -Program-
Transfer Card
```

Each program must be preceded by a Program Card and followed by a Transfer Card (one of the three types discussed below).

1. Program Card: This is the first card punched out by 709/7090 FAP for every relocatable program. The Program Card, distinguished by a 12-punch in column 1 (of a column binary card), supplies the SSB Loader with the name of the program (columns 13-15) and the program's associated entry point (columns 16-18). This card also contains the length of the program, the length of the transfer vector, and the COMMON break, which is zero if no reference is made to COMMON.
2. Transfer Card: The presence of a Transfer Card in the deck signals the Loader that the end of a particular program has been reached. The Loader recognizes two types of Transfer Cards:
   a. FORTRAN Transfer Card -- This card contains only a 12-punch in column 1.
   b. Relocatable Transfer Card -- This card is not punched by the assembly program; xx must be hand-punched. It need only consist of 11-, 7-, and 9-punches in column 1.
3. Program: The cards of the program must be relocatable binary. Cards to be loaded via the card reader are row binary. All other cards are blocked in column binary form and are placed on tape.

Load List

The SSB Loader maintains a list of 101 words called the Load List. The first word is the TCP (see "Transfer Control Point"). The next 100 words store information on fifty programs in a two-word

scheme as follows:

Word 1: Program name in BCD
Word 2: Decrement -- number of entries in transfer vector
Address -- SSB-given load address

When the programs have been loaded, the Load List is searched to find the load address of the TCP program and the prefixes of the entries in the transfer vectors of the individual programs are changed to TTRs to the load address or, in case the program has not been loaded, the prefix of the entry in the transfer vector is changed to an STR.

Use of TTR and STR in Transfer Vectors

When loading is complete, pass 2 is begun. Each of the loaded programs contains a transfer list of the entry points to any other subprograms called on by the program. During pass 2, each BCD subprogram name in these transfer lists is replaced by a TTR***** if that subprogram was also loaded, or by an STR 0,0,0 if the subprogram was not loaded. If a subprogram that was not loaded is called, the STR will be executed, at which time an IBSYS core storage dump will be taken.

Transfer Control Point

The third word in the calling sequence to SSB Loader is of the type

```
BCI     1,RETURN
```

The BCD name RETURN is placed in the TCP word, the first word of the Load List, as one of the Loader's first functions. When pass 2 is initiated, the name in the TCP word is compared with the names of the loaded programs. When its equal is found, its load address is placed in a TTR instruction. When pass 2 is complete, this TTR instruction is executed and control is passed to the TCP program.

CHECKPOINT AND RESTART PROCEDURE

Except in those cases where the NOCKPT option is specified in the OPTION Card, checkpoints are written at the following points in the execution of a sort:

1. After each reel of input of the Internal Sort Phase
2. At the end of the Internal Sort Phase
3. After each pass of the Merge Phase
4. After each reel of output of the Final Merge Phase

When a checkpoint is taken, core storage is written on the checkpoint tape (SYSCK2) and preserved, and all tape assignments are recorded along with the number of records on each, to allow the repositioning of tapes on restart. To restart the sort run the following control cards are necessary:

```
$EXECUTE    SORT
RESTART
END
```

The Edit Phase of 7090 Sort reads the RESTART Card and transfers control to the Sort Monitor which initiates the restart procedure. SYSCK2, the checkpoint tape, is positioned; the checkpoint file is read into core storage; core storage is restored; and the tapes are repositioned.

ASSIGN AND RELEASE

ASSIGN is an Edit Phase subroutine used for assigning units from the Availability Chain for use by the Sort program. The user may also use this routine to request the assignment of specific units. The calling sequence to the ASSIGN subroutine is:

```
CALL            ASSIGN
PZE             LIST,,LENGTH
Error Return 1  (2,4)
Error Return 2  (3,4)
Normal Return   (4,4)
```

where:

LIST is the location of the first entry in LIST, i.e., the first of those consecutive locations in core storage containing the information necessary to make desired assignments (see below).

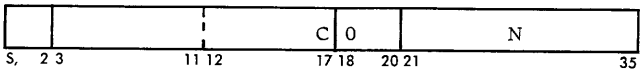
LENGTH is the length of LIST in number of words.

Error Return 1 (2,4) is used by ASSIGN if more units are requested than are available. All available units have been assigned. The number of units still needed is in the accumulator.

Error Return 2 (3,4) is used by ASSIGN when the request is for an intersystem unit (symbolic channels J-Q) and no such unit is found in the scan of the Unit Control Blocks.

Normal Return (4,4) is used if no error conditions are encountered.

Each of the entries in LIST must have the following format:



C is the BCD representation of the channel being requested. When there is an N (number) in the address portion of the entry, the Nth unit in the Availability Chain will be assigned. Otherwise, the first unit in the Availability Chain will be assigned.

If an intersystem communication unit is sought, C must be one of the symbolic channel designations J-Q. The address portion (bits 21-35) of the LIST entry must contain the same bits as are found in the address portion of the first Unit Control Word of that unit.

If the unit to be assigned is not in ready status and the TAPES option has been requested, the message "NOT IN READY STATUS" is appended to the printout of the assigned unit.

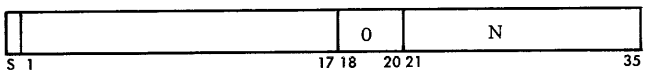
When ASSIGN returns, each entry in LIST corresponding to an assigned unit contains the location of the Unit Control Block of that unit in its address portion.

A unit that has been assigned at the request of the user must also be released by the user. The calling sequence to the RELEASE subroutine is:

```
CALL    RELEAS
PZE     LIST, , LENGTH
        Normal Return
```

where LIST and LENGTH are the same as in the calling sequence to the ASSIGN subroutine above.

Each of the entries in LIST must have the following format:



N is the location of the Unit Control Block of the unit to be released.

If a unit has already been released and placed in the Availability Chain, the request will be ignored.

If a unit was reserved for intersystem communication and is now to be released, the 1-bit must be set to one in the LIST entry. Otherwise, the unit is not released.

USER MODIFICATIONS TO SORT

In general, modifications to the Sort program are subprograms that are added to the list of programs on the Library tape, stored on separate modification tapes, or introduced to the system as cards from the card reader. Modifications can be requested at execution time by the use of control cards (see "Modification Card"). In this way, modifications can usually be incorporated in the program without reassembly. Switches have been included in 7090/7094 Sort at those points where modifications are assumed to be most likely. The request for a modification causes the SSB Loader to load the modification along with the appropriate phase and to set the necessary switch to use it.

Like the other Sort subprograms, the modifications must be assembled in relocatable form, using IBSFAP (located on the IBSYS tape with SORT). In order that they be recognized by the SSB Loader, their names are restricted. The name indicates the phase in which a modification is to be loaded and the point at which it is to be used. The names are as follows:

```
XSM01
  .
  .          Internal Sort Phase
  .
XSM10
MPM01
  .
  .          Merge Phase
  .
MPM05
FMM01
  .
  .          Final Merge Phase
  .
FMM10
```

NOTE. The Edit Phase modifications (EPM01 through EPM05) are treated in a slightly different manner than the above.

The modification switches may be in one of the following forms:

```
            ZET     INDICATOR
            CALL    MODIFICATION
or
            NZT     INDICATOR
            TRA     NEXT
            STL     GENPR-4
            CALL    MODIFICATION
NEXT        BSS     0
```

Switches of the second type are used in a program as follows:

```
Main Program                        Subprogram

    :                  MODIFICATION     :
    .                                   .

    NZT     INDICATOR                   STL     GENPR-5

    TRA     NEXT                        TRA     1,4

    STL     GENPR-4

    CALL    MODIFICATION

NEXT BSS    0
```

GENPR-4 and GENPR-5 are two locations in COMMON. A call to a subprogram is preceded by storing, in location GENPR-4, the location at which control left the main program. A return to the main program is preceded by a storage of the last location of the subprogram in location GENPR-5. This permits a method of tracing in case of processing difficulties.

Each of these switches, or transfer points, has a permanent indicator, such as MODPR-4, and a modification name, such as XSM05. Initially, the indicator is set to zero. However, if the modification is requested, the indicator is set to non-zero. Although the modification name always appears in the transfer vector of the phase in which the above instructions are written, the Loader processes the transfer vector in different ways, depending on whether or not the modification has been requested and loaded from the Library tape.

If an installation executes many different types of sort runs, it is sometimes desirable that one modification switch be used for several different subprograms, each of which is unique to the type of sort run being executed. In order to facilitate this procedure, it is possible to specify the following on the MODIFICATION card:

PROGRAM/XSM01C

where C is any legal BCD character. This allows for the loading of a particular subprogram, depending on the type of run. The modification switch associated with XSM01 is set, and when tested, control is passed to XSM01C.

Example: Suppose in run one, a modification is to be used at modification switch FMM03, and in run two, a different modification also uses modification switch FMM03. The two programs, located on the same tape, could now be named FMM03A and FMM03Z.

The MODIFICATION cards for run one and run two are as follows:

MODIFICATION, PROGRAM/FMM03A, CELLS/20, LOCATION/SYSUT3, FORMAT/B

and

MODIFICATION, PROGRAM/FMM03Z, CELLS/50, LOCATION/SYSUT3, FORMAT/B

Programs FMM03A and FMM03Z would be loaded during runs one and two, respectively.

Programs with names XSM01C, or with similar name formats, cannot be placed on the IBSYS Library tape, upon which the Sort resides. The Sort Editor does not have the ability to edit these programs onto the IBSYS System tape.

One MODIFICATION control card is needed for each modification requested. When the control card is read, the Edit Phase adds the name of the modification to the Program List and sets the appropriate indicator to nonzero status.

Modification programs may obtain their parameters from COMMON storage, assuming that COMMON has been defined as in each of the other Sort subprograms. If additional COMMON storage is needed, a reassembly of the Sort will be necessary. The modification programs may communicate with each other, or with any of the subprograms, through transfer vectors.

It is assumed that modifications to the Edit Phase will be rare. Consequently, a different method of requesting them is provided. After IBSYS reads the

```
$EXECUTE    SORT
```

control card and control is transferred to the Sort Monitor, Sense Switch 5 is interrogated. If the switch is up, the program continues. If the switch is down, the machine halts and prints:

```
SET ENTRY KEYS (31-35) FOR
   EDIT PHASE MODIFICATION
```

After the desired keys are set, the user presses START to continue, and Sort Monitor adds the requested Edit Phase modification names to the Program List.

Entry keys 31 through 35 are assigned as follows:

Key	Modification Name
31	EPM01
32	EPM02
33	EPM03
34	EPM04
35	EPM05

Restriction: At some of the exits to the modification program, valid information is in the registers (e.g., AC, MQ, etc.). This information must be saved before the customer's modification program uses the register.

USE OF COMMON FOR CODING PARAMETER ENTRIES

Since many of the Sort programs and subprograms refer to one or more of the parameter tables, these tables are placed in COMMON by using the IBSFAP COMMON statements. Any reference to a parameter entry results in the decrement or address of an instruction receiving a 10- or 11-bit configuration in columns 7-12 (the relocation bits; see "Relocation Scheme"), depending on whether the decrement or address was above or below the Program Break.

These parameter tables, which are consecutive, are placed in COMMON as follows:

```
77461    OPTPR    COMMON    8
77451    OVEPR    COMMON    2
77447    LABPR    COMMON    5
77442    CHAPR    COMMON    6
77434    FILPR    COMMON   14
77416    MODPR    COMMON   50
77334    RECPR    COMMON   10
77322    DELPR    COMMON    2
77320    SAMPR    COMMON    7
77311    CALPR    COMMON   50
77227    GENPR    COMMON   10
77215    RAWPR    COMMON   20
77171    CNTPR    COMMON   20
77145    ERAPR    COMMON   30
77107    REFPR    COMMON    1
```

Assumc that program XSM02 makes reference to the parameter tables FILPR, SAMPR, MODPR, and ERAPR. Typical coding and assembly might be:

Coding		Assembly		Load Address
CLA	FILPR-3	0500	77431	77747
ADD	SAMPR-4	0400	77314	77632
SUB	MODPR-10	0402	77404	77722
STO	ERAPR-29	0601	77110	77426

In essence, SSB would add 316 to each of the above addresses. The resulting sum is the desired address, and replaces the assembled address.

Relocation Scheme

When the SSB Loader encounters a zero location bit in columns 7-12 of a relocatable card, it does not relocate the decrement or the address. But when a 10- or 11-bit configuration is encountered, the following relocation scheme is used:

1. 10-bit configuration -- Below the COMMON break (which is the last location used by the program) relocate the address or decrement, relative to the value of the current increment.

   Above the COMMON break, relocate the address or decrement, relative to the parameter table load address.

2. 11-bit configuration -- Below the COMMON break, relocate the address or decrement, relative to the parameter table load address.

   Above the COMMON break, relocate the address or decrement, relative to the value of the current increment.

APPENDIX A. COLLATING SEQUENCES

Sorting Equivalents for the Scientific Sequence	Scientific Sequence	Equivalence Index Scientific/Commercial		Commercial Sequence	Sorting Equivalents for the Commercial Sequence
00	0	00	52	blank	00
01	1	01	53	.	01
02	2	02	54	¤	02
03	3	03	55	≠	03
04	4	04	56	&	04
05	5	05	57	$	05
06	6	06	60	*	06
07	7	07	61	-	07
10	8	10	62	/	10
11	9	11	63	,	11
12	NOTE 1	12	64	%	12
13	#	13	13	#	13
14	@	14	14	@	14
15	NOTE 1	15	65	+0	15
16	NOTE 1	16	66	A	16
17	NOTE 1	17	67	B	17
20	&	20	04	C	20
21	A	21	16	D	21
22	B	22	17	E	22
23	C	23	20	F	23
24	D	24	21	G	24
25	E	25	22	H	25
26	F	26	23	I	26
27	G	27	24	-0	27
30	H	30	25	J	30
31	I	31	26	K	31
32	+0	32	15	L	32
33	.	33	01	M	33
34	¤	34	02	N	34
35	NOTE 1	35	70	O	35
36	NOTE 1	36	71	P	36
37	≠	37	03	Q	37
40	-	40	07	R	40
41	J	41	30	‡	41
42	K	42	31	S	42
43	L	43	32	T	43
44	M	44	33	U	44
45	N	45	34	V	45
46	O	46	35	W	46
47	P	47	36	X	47
50	Q	50	37	Y	50
51	R	51	40	Z	51
52	-0	52	27	O	52
53	$	53	05	1	53
54	*	54	06	2	54

Sorting Equivalents for the Scientific Sequence	Scientific Sequence	Equivalence Index Scientific/Commercial		Commercial Sequence	Sorting Equivalents for the Commercial Sequence
55	NOTE 1	55	73	3	55
56	NOTE 1	56	74	4	56
57	NOTE 1	57	75	5	57
60	blank	60	00	6	60
61	/	61	10	7	61
62	S	62	42	8	62
63	T	63	43	9	63
64	U	64	44	NOTE 2	64
65	V	65	45	NOTE 2	65
66	W	66	46	NOTE 2	66
67	X	67	47	NOTE 2	67
70	Y	70	50	NOTE 2	70
71	Z	71	51	NOTE 2	71
72	≠	72	41	NOTE 2	72
73	,	73	11	NOTE 2	73
74	%	74	12	NOTE 2	74
75	NOTE 1	75	76	NOTE 2	75
76	NOTE 1	76	77	NOTE 2	76
77	NOTE 1	77	72	NOTE 2	77

NOTE 1 -- There is no valid BCD character.
NOTE 2 -- All invalid characters are sorted at the high end of the commercial sequence.

Example: If the commercial collating sequence is specified, and an "A" is the character to be sorted, the following will take place:

1. "A" is equated to $(21)_8$. (See columns 1 and 2.)
2. $(21)_8$ is converted to $(16)_8$. (See columns 3 and 4.)
3. The character that is equivalent to $(16)_8$ in the commercial collating sequence is "Ā". (See columns 5 and 6.)

APPENDIX B. ALLOCATION OF COMMON STORAGE

The COMMON area of core storage contains the parameters of the Sort program as follows:

OPTPR	COMMON	8	Option Card Parameters
OVEPR	COMMON	2	Overflow Card Parameters
LABPR	COMMON	5	Label Card Parameters
CHAPR	COMMON	6	Channel Card Parameters
FILPR	COMMON	14	File Card Parameters
MODPR	COMMON	50	Modification Card Parameters
RECPR	COMMON	10	Record Card Parameters
DELPR	COMMON	2	Delete Card Parameters
SAMPR	COMMON	7	Sort and Merge Card Parameters
CALPR	COMMON	50	Calculated Parameters
GENPR	COMMON	10	General Parameters
RAWPR	COMMON	20	Read and Write Parameters
CNTPR	COMMON	20	Count Parameters
ERAPR	COMMON	30	Erasable Storage Parameters
REFPR	COMMON	1	Referenced Parameters Parameter

Allocation of storage within these areas is as follows:

OPTION Card Parameters

OPTPR	-0	Number of words that COMMON relocated
	-1	CKSUMS option 0 = no 1 = yes
	-2	NOCKPT option 0 = no 1 = yes
	-3	EQUALS option 0 = no 1 = yes
	-4	BUFFER option 0 = no 1 = yes
	-5	VARIABLE BLOCKING option 0 = no 1 = yes
	-6	MAP option 0 - no 1 - yes
	-7	

OVERFLOW Card Parameters

OVEPR	-0	Number of blocks for overflow
	-1	

LABEL Card Parameters

LABPR	-0	Address of area for label construction; the area is 14 words long
	-1	Address of area for reading of label; the area is 14 words long
	-2	Address of LABEL Card, if given
	-3	Length of LABEL Card
	-4	Address of labels used in Merge Phase

CHANNEL Card Parameters

CHAPR	-0	Address of input channel
	-1	Address of output channel UCW Reference Table
	-2	Address of UCW Reference Table for output from Internal Sort Phase
	-3	Address of UCW Reference Table for other merge channel
	-4	Intersystem output mode
	-5	Checkpoint tape attached 0 = yes Nonzero = no

FILE Card Parameters

FILPR	-0	Address of sort input file block
	-1	Address of output file block
	-2	
	.	
	.	Addresses of merge input file blocks
	.	
	-11	
	-12	
	-13	

MODIFICATION Card Parameters

MODPR	-0	XSM01	
	.	.	
	.	.	
	.	.	
	-9	XSM10	
	-10	MPM01	
	.	.	
	.	.	
	.	.	
	-14	MPM05	
	-15	FMM01	
	.	.	
	.	.	
	.	.	
	-24	FMM10	
	-25	DXS01	Customer engineer modification words for Internal Sort
	.	.	
	.	.	
	.	.	
	-34	DXS10	
	-35	DMP01	Customer engineer modification words for the Merge Phase
	.	.	
	.	.	
	.	.	
	-39	DMP05	
	-40	DFM01	Customer engineer modification words for the Final Merge Phase
	.	.	
	.	.	
	.	.	
	-49	DFM10	

RECORD Card Parameters

RECPR	-0	Fixed-length record to Internal Sort Phase Variable-length record minimum length
	-1	Fixed-length record of Merge Phase Variable-length record maximum length
	-2	Fixed-length record from Final Merge Phase Variable-length record major length

-3
-4
-5 Records to be sorted
0 - fixed
1 - variable
-6
-7
-8
-9

DELETE Card Parameter

DELPR -0 Address: address of DELETE control fields
Decrement: number of DELETE statements
-1

SORT and MERGE Card Parameters

SAMPR -0 Sort or merge sequence
0 - Scientific (709)
1 - Commercial (705)
-1 Merge order
-2 Number of files to be merged
-3 Address: address of control field data
Decrement: number of control fields
-4 Sort -- nonzero
-5 Merge -- nonzero
-6

Calculated Parameters

CALPR -0 Date-year, day
-1 Input blocking factor, i.e., number of logical records per block (fixed-length records)
-2 Output blocking factor, i.e., the number of logical records per block (fixed-length records)
Maximum number of records in the output block (variable-length records)
-3 Bin length (variable-length records)
-4 Size of G in words for the Internal Sort Phase (fixed-length records)
Size of the Record Storage Area in words for the Internal Sort Phase (variable-length records)
-5 Size of the L table for the Internal Sort Phase (fixed-length records)
Size of the L table, i.e., the total number of bins plus 3 (variable-length records)
-6 Size of the partial sort list (M tables) for the Internal Sort Phase (fixed-length records)
Maximum length of the sort table (variable-length records)
-7 Ratio (L/M)
-8 Effective record length for the Internal Sort Phase (fixed-length records)
-9 Effective record length for the Merge Phase (fixed-length records)
-10 Effective record length for the Final Merge Phase (fixed-length records)
-11 Sort capacity in logical records (fixed-length records)
-12 Number of words in all control fields (fixed-length records)
Number of words in packed control fields, not including the Equals word (variable-length records)
-13 Maximum merge blocking in words
-14 Maximum merge blocking in number of logical records
-15 Number of merge areas for Merge Phase
-16 Number of merge areas for Final Merge Phase
-17 Grouping factor for the Internal Sort (fixed-length records)
-18 Blocking factor for the Internal Sort Phase (fixed-length records)
-19 Address: address of low padding record (fixed-length records)
Decrement: record length of padding record
-20 Address: address of high padding record (fixed-length records)
Decrement: record length of padding record
-21
-22 Length of longest control field
-23 Available core storage for the Internal Sort Phase
-24 Available core storage for the Merge Phase
-25 Available core storage for the Final Merge Phase
-26 Number of words in input block, including trailer words, if expected (fixed-length records)
Maximum blocksize expected for the Internal Sort Phase (variable-length records)
-27 Total length of extracted control fields
-28 Output file blocksize (fixed-length records)
Output buffer size for the Final Merge Phase (variable-length records)

	-29	
	-30	
	-31	Length of buffer for the Internal Sort Phase (variable-length records)
	-32	Number of words in packed CF, including the Equals word (variable-length records)
	-33	Maximum words/PUT for the Internal Sort Phase (variable-length records)
	-34	Minimum words/PUT for the Internal Sort Phase (variable-length records)
	-35	Length of the Buffer Table for the Internal Sort Phase (variable-length records)
	-36	Length of the Locate Table for the Internal Sort Phase (variable-length records)
	.	.
	.	.
	.	.
	-49	

General Parameters

GENPR	-0	Phase indicator
	-1	Binary/BCD indicator zero = BCD; nonzero = binary
	-2	Pass number (Merge Phase)
	-3	Complement of Index Register 4 for use by RESTART to return control to the main program
	-4	Location of last executed CALL statement
	-5	Location of last return from a subprogram which had been called
	-8	Dictionary switch zero -- no dictionary specified for output file nonzero -- dictionary specified for output file
	-9	No dump switch nonzero -- no more records can be dumped, but the option to continue is taken

Read and Write Parameters

RAWPR	-0	Calling sequence Word 1 for read and write
	-1	Calling sequence Word 2 for read and write
	-2	Read completion indicator
	-3	Read EOF indicator
	-4	Write completion indicator
	-5	Write EOT indicator
	-6	Current buffer read grouping factor
	-7	Address: logical file number of current read Decrement: logical file number of current write
	-8	Sequence break and block length (second word of merge block)
	-9	Write head word (sequence and block word)
	-10	Location of file data block for current read
	-11	Buffer write switch
	-12	External output switch 0 = current output is internal output nonzero = current output is external output
	-13	External input switch 0 = current input is internal input nonzero = current input is external input
	-14	Location of table for read calling sequence word 2
	-15	Location of table for write calling sequence word 2
	-16	Current read mode 0 = BCD nonzero = binary
	-17	Current write mode 0 = BCD nonzero = binary
	-18	Location of first IOSP for current block in Scatter Read
	-19	Address: number of logical records in current short block Decrement: number of words in last logical record of current short block

Count Parameters

CNTPR	-0	Count of records (XS)
	-1	Count of records (VMR)
	-2	Count of records (VFM)
	-3	Count of records dumped (XS)
	-4	Count of records dumped (VMR)
	-5	Count of records dumped (VFM)
	-6	Count of padding records
	-7	Records dumped, this pass
	-8	Records deleted in Internal Sort Phase
	-9	Records deleted, this phase
	-10	Record count, low padding, Final Merge Phase
	-11	Record count, high padding, Final Merge Phase
	-12	Write-table fixer count
	-13	Count of records to date for deblocking
	-14	Count in records of total input to sort or merge

-15
.
.
.
-19

Erasable COMMON Parameters for Internal Sort Phase

Fixed-Length Records

ERAPR	-0	
	-1	
	-2	
	-3	
	-4	
	-5	Address of last record written out
	-6	
	-7	
	-8	
	-9	
	-10	Address: location of first command in read table Decrement: M
	-11	Address: location of first command in write table Decrement: number of record to be written out
	-12	
	.	
	.	
	.	
	-19	
	-20	File number (logical)
	-21	
	.	
	.	
	.	
	-29	

Variable-Length Records

ERAPR	-0	Location and length of read buffer
	-1	Location and length of Buffer Table
	-2	Location and length of Locate Table
	-3	Location and length of available table for GET
	-4	Location and length of available table for PUT to build write table after
	-5	Address of last record written out
	-6	
	.	
	-8	
	-9	Location and length of sort table
	-10	Location of Block Table for PUT and its length
	-11	
	-12	
	.	
	.	
	-29	

Erasable COMMON Parameters for Merge Phase

ERAPR	-0	
	-1	
	-2	
	-3	
	-4	
	-5	Communication word between VMR and VFM EQUAL routines
	-6	
	-7	
	-8	
	-9	Communication between VMR and VFM EXTRACT Routines
	-10	
	-11	Write base for DELETE
	-12	
	.	
	.	
	.	
	-19	
	-20	Address to insert file number in internal label
	-21	Address to insert pass number in internal label
	-22	
	-23	
	-24	
	.	
	.	
	.	
	-29	

Referenced Parameters Parameter

REFPR	-0	Address of last word used by referenced parameters

APPENDIX C. FORMAT FOR CONTROL FIELD INFORMATION

Each of the control fields by which records are to be sorted or merged has a 9-word block of information in storage. This information is developed during the Edit Phase from the parameters on the SORT (or MERGE, in a merge run) control card.

The control field information facilitates easy extraction of the associated control field. In a sort or merge, all of the control fields will be extracted and placed in front of the record. The RQL and LGL

instructions needed in the extraction subprogram (XTRACT) are initialized by the Edit Phase.

Word	Bit	Contents
1	3-17	Numbers of words control field extends over
	18-20	0 - logical ascending 1 - logical descending 2 - algebraic ascending 3 - algebraic descending 4 - signed BCD ascending 5 - signed BCD descending
2		RQL
3		LGL
4		LGL
5		LGL
6		LGL
7		LGL
8		LGL
9		LGL

Each DELETE control field also has a corresponding storage block of a minimum of 4 words, as follows:

Word	Bits	Contents
1	3-17	Number of words control field extends over
	21-35	Starting word of control field
2		Left mask
3		Right mask
4-n		Information to compare against DELETE control field

APPENDIX D. INPUT AND OUTPUT FILE DATA BLOCK

Information on the input and output files of a sort or merge run is taken from the parameters of the FILE control cards of that run. This information is stored in a 25-word block, arranged as follows:

Word	Contents	Used Input	Used Output
1	Mode 0 = decimal (may be omitted) nonzero = binary	X	X
2	Density 0 = high (may be omitted) nonzero = low	X	X
3	Padding 0 = no padding 1 = low padding 2 = high padding	X	X
4	Label (must be given) 0 = standard 1 = nonstandard 2 = no label	X	X
5	Label density 0 = high (may be omitted) nonzero = low	X	X
6	Blocksize (must be given)	X	X
7	Serial number (checked if given)	X	X
8	Reel sequence number (checked if given)	X	X
9	Checksums 0 = no nonzero = yes	X	X
10	Block sequence number	X	X
11	Dictionary 0 = no nonzero = yes	X	X
12	Name (checked if given)	X	X
13	Name	X	X
14	Name	X	X
15	Input number (must be given)	X	
16	Number of input reels (1 if not given)	X	
17			
18	Checkpoint 0 = standard (may be omitted) 1 = nonstandard 2 = no checkpoint	X	
19	Retain		X
20	Checksums or block sequence 0 = neither nonzero = yes	X	X
21	Grouping factor	X	X
22	Number of reels processed to date	X	
23	Blocksize plus checksum or sequence word	X	X
24	Blocking - Blocksize/ Record Length		
25			

APPENDIX E. THE SORT SUBPROGRAMS

Following is a list of the 7090/7094 Sort subprograms, giving their symbolic names, their functions, and the phases for which they will be loaded.

Symbolic Name	Function	Phase(s) Loaded
RESTAR	Restart	EP, XS, VMR, VFM
SOP	Open-Close	EP, XS, VMR, VFM
LABEL	Label	XS, VMR, VFM
BTD	Binary to Decimal	EP, VMR, VFM
CALC	Calculations	EP
EP001	Edit Phase	EP

Symbolic Name	Function	Phase(s) Loaded
RB01	Card Image to BCD	EP
RELEAS	Release Tapes	EP, VFM
SK001	Scan Control Cards	EP
VXS	Internal Sort	VXS
XS	Internal Sort	XS
IOBS	Buffer-Scatter	XS*, VMR, VFM
XTRACT	Extract	XS, VMR, VFM
GET	Get	VXS
PUT	Put	VXS
IOSS	Scatter-Scatter	XS
WRSEL	Write Select	XS, VMR, VFM
DEBLK	Deblock and Dump	XS, VMR, VFM
DEPAD	Depad	XS, VFM
VFM	Final Merge	VFM
VMR	Merge	VMR
POST	Post-Processor	VFM
FXMOV	Fixed Move	XS*
CKPT	Checkpoint	XS, VMR, VFM
CKSUMS	Checksums	XS*, VMR*, VFM*
DELETE	Delete	XS*, VFM*
EQUALS	Equals	XS*
ASSIGN	Tape Assignment	EP
WTFIX	Write Table Fixer	XS

*optional

APPENDIX F. LOADING THE SYSTEM

1. SYSLB1 - IBSYS Systems
2. SYSIN1 - Input Unit

SSW1 Down	RDA Attached as SYSIN1	Control Card Loading Unit
Yes	Yes	IBSYS control cards and Sort control cards in card reader.
Yes	No	IBSYS control cards in card reader; Sort control cards on tape.
No	Yes	IBSYS control cards on tape; Sort control cards in card reader.
No	No	IBSYS control cards and Sort control cards on tape.

3. To start, press Load Tape button. If IBSYS is on disk, and a start card is in the card reader, press the Load Card button.

APPENDIX G. SENSE SWITCHES

Switch	Function
1	Used by IBSYS (see Appendix F, paragraph 2).
2	Used by Sort Monitor when Sort Edit is desired (see Appendix H).
3	Used by Sort Edit when putting Sort on a system tape initially.
5	Used to interrupt Merge Phase.
5	Used by Sort Monitor to determine if Edit Phase Modifications are desired. If it is down, the machine halts so entry keys can be set telling which modifications are desired.

Some error messages require that a sense switch be set to correct the error. The required switch setting will be printed on-line when the error message is printed.

APPENDIX H. UPDATING AND EDITING 7090/7094 SORT

The 7090/7094 Generalized Sorting System occupies two consecutive files on the IBSYS system tape. The position of these two files on the tape is not critical as long as they keep their positions relative to each other. The first file consists of one program, named Sort, which is assembled in absolute form and has its origin at SYSORG. Sort contains the Sort Monitor and the Sort Editor and is loaded by IBSYS when the $EXECUTE SORT control card is read. Sort Monitor loads the appropriate subprograms from the second file for each phase of the Sort and passes control to the main program of each phase. The second file contains several subprograms; 30 are available to the user for modifications to the Sort System. The file is referred to as the Restart file because RESTART is the first subprogram in the file.

All subprograms in the Restart file are assembled in relocatable form. This is done to provide maximum space in core storage for sorting at object time. It is only after the parameters from the control cards have been analyzed that decisions are made as to what subprograms should be loaded during each phase.

The IBSYS Editor, IBEDT, does not allow modifications to be made to relocatable subprograms. Because the Restart file is relocatable, it was necessary for Sort Editor to be written so that the updating and editing of the System tape could be accomplished.

To update one or more of the subprograms in the Restart file it is necessary to:

1. Assemble the subprograms. IBSFAP will place the relocatable card-images of the assembled programs on SYSPP1.
2. Set Sense Switch 2.
3. $EXECUTE SORT.

When Sort Monitor finds Sense Switch 2 down, it passes control to Sort Editor, which reads the first newly assembled subprogram from SYSPP1 and the first subprogram from the Restart file on SYSLB1. If the programs are not the same, the program from SYSLB1 is written on SYSUT2 and the next program is read from SYSLB1 and compared with the one read from SYSPP1. When the programs are the same, the one read from SYSPP1 is provided with header words and a transfer card is written as one tape record on SYSUT2. This reading, comparing, blocking, and replacing continues until a new Restart file has been built on SYSUT2 with modified subprograms from SYSPP1 replacing their obsolete counterparts. If the subprogram read from SYSPP1 is not in the list of programs acceptable for the Restart file, it will be ignored and the next program from SYSPP1 will be read. Sort (1SM) is not in this list of acceptable subprograms, so it may be assembled without fear of its being included in the new Restart file. When Sort Edit has built the new Restart file on SYSUT2, it returns control to IBSYS. Then a standard IBSYS edit run with a

FILE *REPLACE RESTAR,SYSUT2

card will build a new IBSYS tape with Sort updated. If all subprograms for the Restart file have been reassembled, making it unnecessary to compare the newly assembled programs against those on SYSLB1, Sense Switch 3 should be put down along with Sense Switch 2. When Sort Editor finds Sense Switch 3 down, it reads from SYSPP1 only, adds header words and transfer cards, and blocks the programs on SYSUT2.

The symbolic tape distributed with the Sort package contains symbolic card images blocked 16 cards per block, except for the END cards, which are unblocked. Normal rules for doing a FAP update are found in the publication, IBM 709/7090 Programming Systems: FORTRAN Assembly Program (FAP), Form C28-6235. The symbolic tape has subprograms arranged on it so that the serializations beginning in column 73 have the following order:

Card Identification	Routine
1SM	Sort Monitor and Editor
ARS	Restart
ASO	Sort Open and Close
BLB	Label
CAS	Assign
CBT	Binary to Decimal
CCA	Calculations
CEP	Edit Phase
CRB	Control Card Conversion
CRE	Release
CSK	Scan
CVX	Internal Sort (variable-length records)
CXS	Internal Sort (fixed-length records)
DBS	I/O (buffer-read)
DEX	Extract
DGE	Get
DPU	Put
DSS	I/O (scatter-read)
DWS	Write Select
DWT	Write Table Fix
EDD	Deblock and Dump
EDP	Depad
EVF	Final Merge
EVM	Merge
FPP	Post Processor
GFX	Fixed-length Move
HCK	Checkpoint
HCS	Checksum
HDE	Delete
HEQ	Equals
K01	Modification EPM01
K02	Modification EPM02
K03	Modification EPM03
K04	Modification EPM04
K05	Modification EPM05
L01	Modification XSM01
L02	Modification XSM02
L03	Modification XSM03
L04	Modification XSM04
L05	Modification XSM05
L06	Modification XSM06
L07	Modification XSM07
L08	Modification XSM08
L09	Modification XSM09
L10	Modification XSM10
M01	Modification MPM01
M02	Modification MPM02
M03	Modification MPM03
M04	Modification MPM04
M05	Modification MPM05
N01	Modification FMM01
N02	Modification FMM02
N03	Modification FMM03
N04	Modification FMM04
N05	Modification FMM05
N06	Modification FMM06
N07	Modification FMM07
N08	Modification FMM08
N09	Modification FMM09
N10	Modification FMM10

If one wishes to correct and assemble program

NNN, it is recommended that he use these control cards:

```
1       7          16                          73
        *FAP
        UPDATE     X
  (Correction Cards)
        END                                    NNNZZZZZ
```

To space the symbolic tape up to subprogram NNN (which we will assume follows subprogram MMM), the following control cards may be used:

```
1       7          16                          73
        *FAP
        UPDATE     X,,,D
        ENDUP                                  MMMZZZZZ
```

In order to facilitate the release of modifications to the field, the procedure will be as follows:

All modifications will be distributed in the form of symbolic card corrections. This will require that the user reassemble the subprogram(s) being corrected and update the Sort System on the IBSYS System tape. Therefore, the user must use the Sort symbolic tape that is associated with the current version of the Sort.

Assembly will be done without renumbering columns 73-80. This will provide the user with an up-to-date listing of the reassembled subprogram, and will provide a common communication medium so that several modifications can be made to the same subprogram within a given version.

Each modification distributed for a subprogram will contain all the symbolic corrections ever made for that subprogram. This will allow the user to up-date, using the symbolic tape received with the current version.

When a new version of IBSYS is distributed, all Sort modifications will be reflected on the new IBSYS Library tape even though the version level of Sort is unchanged. The user will need to update Sort when he receives a new IBSYS version only if he is using modification programs on the System tape.

The modification letter that accompanies the symbolic correction cards will contain a list of all subprogram names and the previous modification level of each subprogram.

The following examples will demonstrate how:

1. one subprogram can be updated.
2. all subprograms can be updated simultaneously to produce an up-to-date listing.
3. Sort Monitor can be updated.

The following cards may be used to update the subprogram named LABEL.

```
1       7          16                          73
$EXECUTE           IBSFAP
        *FAP                          SOP
        UPDATE     8,,,D
        ENDUP                                  ASOZZZZZ
        *FAP                          LABEL
        UPDATE     8
  (Correction Cards, if any)
        END                                    BLBZZZZZ
```

A list of control cards is given that will (1) assemble all subprograms of the Sort system; (2) perform Sort Edit on the subprograms just assembled; and (3) build a new System tape using IBEDT. Sort (1SM) will be assembled, but Sort Edit does not replace it on the system tape. Another example will show how Sort can be replaced.

Each installation must determine whether or not the set of control cards in the examples is appropriate to meet the requirements of the installation, and must make required changes to the cards.

For these examples we will assume the following SYSUNI table and FAP logical tape numbers.

Function	Unit	Density	FAP Logical Tape Number
SYSLB1	A1	High	
SYSOU1	B1		
SYSIN1	A2		
SYSPP1	B2		
SYSCK1	None		9
SYSCK2	None		10
SYSUT1	A3	High	
SYSUT2	B3	High	8
SYSUT3	A4	High	
SYSUT4	B4	High	

Attached units not assigned or reserved:

A5
B5
Etc.

The control cards are as follows:

```
1       7          16                          73
$*  (Identification)
$DATE              MMDDYY
$EXECUTE           IBSFAP
        *FAP                          1SM
        UPDATE     8
        END                                    1SMZZZZZ
        *FAP                          RESTAR
        UPDATE     8
      (Correction Cards, if any)
        END                                    ARSZZZZZ
        *FAP                          SOP
        UPDATE     8
      (Correction Cards, if any)
        END                                    ASOZZZZZ
        *FAP                          LABEL
```

```
1      7          16                          73
         UPDATE  8
(Correction Cards, if any)
         END                                   BLBZZZZZ
         *FAP                ASSIGN
         UPDATE  8
(Correction Cards, if any)
         END                                   CASZZZZZ
         *FAP                BTD
         UPDATE  8
(Correction Cards, if any)
         END                                   CBTZZZZZ
         *FAP                CALC
         UPDATE  8
(Correction Cards, if any)
         END                                   CCAZZZZZ
         *FAP                EP001
         UPDATE  8
(Correction Cards, if any)
         END                                   CEPZZZZZ
         *FAP                RB01
         UPDATE  8
(Correction Cards, if any)
         END                                   CRBZZZZZ
         *FAP                RELEAS
         UPDATE  8
(Correction Cards, if any)
         END                                   CREZZZZZ
         *FAP                SK001
         UPDATE  8
(Correction Cards, if any)
         END                                   CSKZZZZZ
         *FAP                VXS
         UPDATE  8
(Correction Cards, if any)
         END                                   CVXZZZZZ
         *FAP                XS
         UPDATE  8
(Correction Cards, if any)
         END                                   CXSZZZZZ
         *FAP                IOBS
         UPDATE  8
(Correction Cards, if any)
         END                                   DBSZZZZZ
         *FAP                EXTR
         UPDATE  8
(Correction Cards, if any)
         END                                   DEXZZZZZ
         *FAP                GET
         UPDATE  8
(Correction Cards, if any)
         END                                   DGEZZZZZ
         *FAP                PUT
         UPDATE  8
(Correction Cards, if any)
         END                                   DPUZZZZZ
         *FAP                IOSS
         UPDATE  8
(Correction Cards, if any)
         END                                   DSSZZZZZ
         *FAP                WRSEL
         UPDATE  8
(Correction Cards, if any)
         END                                   DWSZZZZZ
```

```
1      7          16                          73
         *FAP                WTFIX
         UPDATE  8
(Correction Cards, if any)
         END                                   DWTZZZZZ
         *FAP                DEBLK
         UPDATE  8
(Correction Cards, if any)
         END                                   EDDZZZZZ
         *FAP                DEPAD
         UPDATE  8
(Correction Cards, if any)
         END                                   EDPZZZZZ
         *FAP                VFM
         UPDATE  8
(Correction Cards, if any)
         END                                   EVFZZZZZ
         *FAP                VMR
         UPDATE  8
(Correction Cards, if any)
         END                                   EVMZZZZZ
         *FAP                POST
         UPDATE  8
(Correction Cards, if any)
         END                                   FPPZZZZZ
         *FAP                FXMOV
         UPDATE  8
(Correction Cards, if any)
         END                                   GFXZZZZZ
         *FAP                CKPT
         UPDATE  8
(Correction Cards, if any)
         END                                   HCKZZZZZ
         *FAP                CKSUM
         UPDATE  8
(Correction Cards, if any)
         END                                   HCSZZZZZ
         *FAP                DELETE
         UPDATE  8
(Correction Cards, if any)
         END                                   HDEZZZZZ
         *FAP                EQUALS
         UPDATE  8
(Correction Cards, if any)
         END                                   HEQZZZZZ
         *FAP                EPM01
         UPDATE  8
(Correction Cards, if any)
         END                                   K01ZZZZZ
         *FAP                EPM02
         UPDATE  8
(Correction Cards, if any)
         END                                   K02ZZZZZ
         *FAP                EPM03
         UPDATE  8
(Correction Cards, if any)
         END                                   K03ZZZZZ
         *FAP                EPM04
         UPDATE  8
(Correction Cards, if any)
         END                                   K04ZZZZZ
         *FAP                EPM05
         UPDATE  8
(Correction Cards, if any)
```

```
1       7         16                          73
        END                                   K05ZZZZZ
        *FAP                  XSM01
        UPDATE  8
(Correction Cards, if any)
        END                                   L01ZZZZZ
        *FAP                  XSM02
        UPDATE  8
(Correction Cards, if any)
        END                                   L02ZZZZZ
        *FAP                  XSM03
        UPDATE  8
(Correction Cards, if any)
        END                                   L03ZZZZZ
        *FAP                  XSM04
        UPDATE  8
(Correction Cards, if any)
        END                                   L04ZZZZZ
        *FAP                  XSM05
        UPDATE  8
(Correction Cards, if any)
        END                                   L05ZZZZZ
        *FAP                  XSM06
        UPDATE  8
(Correction Cards, if any)
        END                                   L06ZZZZZ
        *FAP                  XSM07
        UPDATE  8
(Correction Cards, if any)
        END                                   L07ZZZZZ
        *FAP                  XSM08
        UPDATE  8
(Correction Cards, if any)
        END                                   L08ZZZZZ
        *FAP                  XSM09
        UPDATE  8
(Correction Cards, if any)
        END                                   L09ZZZZZ
        *FAP                  XSM10
        UPDATE  8
(Correction Cards, if any)
        END                                   L10ZZZZZ
        *FAP                  MPM01
        UPDATE  8
(Correction Cards, if any)
        END                                   M01ZZZZZ
        *FAP                  MPM02
        UPDATE  8
(Correction Cards, if any)
        END                                   M02ZZZZZ
        *FAP                  MPM03
        UPDATE  8
(Correction Cards, if any)
        END                                   M03ZZZZZ
        *FAP                  MPM04
        UPDATE  8
(Correction Cards, if any)
        END                                   M04ZZZZZ
        *FAP                  MPM05
        UPDATE  8
(Correction Cards, if any)
        END                                   M05ZZZZZ
        *FAP                  FMM01
        UPDATE  8
(Correction Cards, if any)
```

```
1       7         16                          73
        END                                   N01ZZZZZ
        *FAP                  FMM02
        UPDATE  8
(Correction Cards, if any)
        END                                   N02ZZZZZ
        *FAP                  FMM03
        UPDATE  8
(Correction Cards, if any)
        END                                   N03ZZZZZ
        *FAP                  FMM04
        UPDATE  8
(Correction Cards, if any)
        END                                   N04ZZZZZ
        *FAP                  FMM05
        UPDATE  8
(Correction Cards, if any)
        END                                   N05ZZZZZ
        *FAP                  FMM06
        UPDATE  8
(Correction Cards, if any)
        END                                   N06ZZZZZ
        *FAP                  FMM07
        UPDATE  8
(Correction Cards, if any)
        END                                   N07ZZZZZ
        *FAP                  FMM08
        UPDATE  8
(Correction Cards, if any)
        END                                   N08ZZZZZ
        *FAP                  FMM09
        UPDATE  8
(Correction Cards, if any)
        END                                   N09ZZZZZ
        *FAP                  FMM10
        UPDATE  8
(Correction Cards, if any)
        END                                   N10ZZZZZ
$IBSYS
$REMOVE         SYSUT2          SAVE SYMBOLIC TAPE.
$ENDFILE        SYSPP1          REWIND SYSPP1 FOR SORT EDIT.
$REWIND         SYSPP1
$SWITCH         SYSUT2,SYSUT4
$PAUSE SET SENSE SWITCH 2 AND PRESS START TO ENTER EDIT.

$EXECUTE        SORT            GOES TO SORT EDIT.
```

(Note: No IBSYS card is required since Sort Edit returns directly to the Basic Monitor.)

```
$IBEDIT
        *EDIT
FILE    *REPLACE RESTAR,SYSUT2
(EOF Card, 7-8 punch in column 1)
$IBSYS
REWIND          SYSUT2          REWIND SCRATCH TAPES.
```

At this point we are finished, and the next control cards may be:

```
1       7         16                          73
$RESTORE
$SWITCH         SYSUT1,SYSLB1
FOLLOWED BY THE NEXT JOB, OR
$PAUSE  THIS IS THE END OF SORT UPDATE AND EDIT RUN.
```

followed by instructions to the operator.

The following cards will reassemble Sort, the first file, and build a new system tape on which the newly assembled Sort replaces the old one.

```
1       7           16                          73
$*  (Identification)
$DATE           MMDDYY
$EXECUTE        IBSFAP
      *FAP                      1SM
      UPDATE    8
 (Correction Cards)
      END                                       1SMZZZZZ
$IBSYS
$REMOVE         SYSUT2          SAVE SYMBOLIC TAPE.
$ENDFILE        SYSPP1          REWIND SYSPP1 FOR IBEDT.
$REWIND         SYSPP1
$SWITCH         SYSPP1,SYSUT2
$IBEDT
       *EDIT
TAPE *REPLACE SORT
 (EOF Card)
$IBSYS
$REWIND         SYSUT2
```

At this point we are finished, and the next control cards may be:

```
$RESTORE
$SWITCH         SYSUT1,SYSLB1
```

followed by the next job, or

```
$PAUSE  THIS IS THE END OF THE SORT UPDATE RUN.
```

followed by instructions to the operator.

If it is necessary to update both Sort files and this is accomplished as two stacked jobs, it is recommended that Sort be updated and replaced before the Restart file. If these jobs are stacked, certain control cards require changes for ease of operation. If A3 is attached as SYSLB1, another unit, such as A5, must be attached as SYSUT1 between jobs.

APPENDIX I. MODIFICATION EXITS

Internal Sort Phase

Fixed Length

Exit Name	Location in Listing	Logical Condition in Program
XSM01	MODPR	After Alternate Scatter-Read or, if an Alternate Buffer-Read, after moving records from the buffer. If a Scatter-Read is performed, the address portion of ERAPR-10 contains the address of the Read List. If a Buffer-Read is performed, the address portion of RAWPR-1 contains the buffer address. In both cases, the decrement of each word contains the number of logical records.
XSM02	MODPR-1	Immediately after the testing of the MODPR switch. If a Scatter-Read is performed, the address portion of ERAPR-10 contains the address of the Read List. If a Buffer-Read is performed, the address portion of RAWPR-1 contains the buffer address. In both cases, the decrement of each word contains the number of logical records.
XSM03	MODPR-2	Immediately after the return from XTRACT, and immediately before going to the Shell Sort routine.
XSM04	MODPR-3	Immediately after deleting and depadding records and setting up the Write List, and immediately before the exit to the Write subprogram. The address portions of RAWPR-1 and ERAPR-11 contain the address of the Write List; the decrements contain the number of records.
XSM06	MODPR-5	Immediately after the initial Scatter-Read, or immediately after the initial Buffer-Read. If a Scatter-Read is performed, the address portion of ERAPR-10 contains the address of the Read List. If a Buffer-Read is performed, the address portion of RAWPR-1 contains the buffer address. In both cases, the decrement of each word contains the number of logical records.

Variable Length

Exit Name	Location in Listing	Logical Condition in Program
XSM03	MODPR-2	Immediately after the return from XTRACT and before exiting to the Shell Sort routine. The address and decrement of ERAPR-9 contain the starting address of the Sort Table and the number of entries in the Sort Table, respectively.
XSM04	MODPR-3	Immediately after setting up the PUT calling sequence and immediately before exiting to the PUT subroutine. ERAPR-10 contains, in its address, the location of the Block List and, in its decrement, the number of entries to the List. (See the beginning of the VXS listing for a more complete explanation of the Block List.)
XSM06	MODPR-5	Immediately after the initial GET. ERAPR-9 contains, in its address, the location of the Sort Table and, in its decrement, the length of the Sort Table.
XSM06	MODPR-5	Immediately after the alternate GET. ERAPR-9 contains, in its address, the location of the Sort Table and, in its decrement, the length of the Sort Table.

Exit Name	Location in Listing	Logical Condition in Program
XSM10	MODPR-9	Immediately after the compare loop of the two-way merge subroutine. It is set up as a test for any summarization subprogram. This test is made only when the EQUALS option is not specified and there are equal records.
XSM10	MODPR-9	Immediately after the compare loop of the Shell Sort routine. It is set up as a test for any summarization subprogram. This test is made only when the EQUALS option is not specified and there are equal records.

Merge Phase

Exit Name	Location in Listing	Logical Condition in Program
MPM01	USRM1	Immediately after entry to the Internal Merge Phase.
MPM02	USRM2	Prior to moving the address of a record to the Write Table.
MPM03	USRM3	After initiating a read.
MPM04	USRM4	When the control field of a record to be merged is found to be equal to the control field of the last record merged.
MPM05	USRM5	Prior to writing.

Final Merge Phase

Exit Name	Location in Listing	Logical Condition in Program
FMM01	FINIT	Immediately before initialization is completed.
FMM02	SCTIX+1	Immediately after the sequence check. The location of this exit is convenient for summarization.
FMM03	SORT	At the beginning of the subroutine which sorts the control fields of the initial records.
FMM04	NOM04	Prior to either (a) deleting a record, (b) taking a checksum, or (c) converting the record's length word, and moving the record to the output buffer. This is a convenient exit for omitting records.
FMM05	NOM05	Just before entering Read-Write Provider routine.
FMM06	R1+2	Upon entry to Read subroutine; the last read has been completed, but updating in area status, etc., has not yet been done.
FMM07	NOM07	Upon return from the Merge subroutine.
FMM08	REF+7	Immediately before executing the calling sequence to IOBS to read in a tape record.
FMM09	GOWRT+6	Immediately before executing the calling sequence to IOBB to write a tape record.
FMM10	NOM10	Immediately before going to the Merge subroutine.

Printed in U.S.A. C28-6307

IBM®
International Business Machines Corporation
Data Processing Division, 112 East Post Road, White Plains, N.Y.