SPERRY Horst.

PUBLICATIONS UPDATE

Operating System/3 (OS/3)

Sort/3

User Guide/Programmer Reference

UP-8836 Rev. 2-A

This Library Memo announces the release and availability of Updating Package A to "SPERRY Operating System/3 (OS/3) Sort/3 User Guide/Programmer Reference", UP-8836 Rev. 2.

The SORT3 program is compatible with sort programs available with the IBM System/30, System/32, and System/34. This manual is part of a series describing sort/merge facilities; it describes the statements needed to create sort specifications and tells how to run SORT 3.

This update documents changes to the following:

- Program Restrictions and Considerations
- Table 1–2, Comparison of Transfer Rates for Magnetic Tape Devices.

Copies of Updating Package A are now available for requisitioning. Either the updating package only or the complete manual with the updating package may be requisitioned by your local Sperry representative. To receive only the updating package, order UP-8836 Rev. 2–A. To receive the complete manual, order UP-8836 Rev. 2.

LIBRARY MEMO ONLY	LIBRARY MEMO AND ATTACHMENTS	THIS SHEET IS
Mailing Lists BZ, CZ, and MZ	Mailing Lists B00, B02, B04, 28U, and 29U (Package A to UP-8836 Rev. 2, 9 pages plus Memo)	Library Memo for UP-8836 Rev. 2–A
		RELEASE DATE: January, 1985

UNISYS

OS/3 SORT3

Programming Guide

Relative to Release
Level 9.0

Priced Item

August 1987

Printed in U S America
UP–8836 Rev. 2

UNISYS

OS/3 SORT3 Programming Guide

Previous Title: OS/3 SORT3 User Guide/Programmer Reference

Relative to Release Level 9.0

August 1987

Printed in U S America

Priced Item

UP–8836 Rev. 2

NO WARRANTIES OF ANY NATURE ARE EXTENDED BY THE DOCUMENT. Any product and related material disclosed herein are only furnished pursuant and subject to the terms and conditions of a duly executed Program Product License or Agreement to purchase or lease equipment. The only warranties made by Unisys, if any, with respect to the products described in this document are set forth in such License or Agreement. Unisys cannot accept any financial or other responsibility that may be the result of your use of the information in this document or software material, including direct, indirect, special or consequential damages.

You should be very careful to ensure that the use of this information and/or software material complies with the laws, rules, and regulations of the jurisdictions with respect to which it is used.

The information contained herein is subject to change without notice. Revisions may be issued to advise of such changes and/or additions.

FASTRAND, ✦SPERRY, SPERRY✦UNIVAC, SPERRY, SPERRY UNIVAC, UNISCOPE, UNISERVO, UNIS, UNIVAC, and ✦ are registered trademarks of Unisys Corporation. ESCORT, PAGEWRITER, PIXIE, PC/IT, PC/HT, PC/microIT, SPERRYLINK, and USERNET are additional trademarks of Unisys Corporation. MAPPER is a registered trademark and service mark of Unisys Corporation. CUSTOMCARE is a service mark of Unisys Corporation.

PAGE STATUS SUMMARY

ISSUE: Update B – UP-8836 Rev. 2
RELEASE LEVEL: 9.0 Forward

Part/Section	Page Number	Update Level
Cover		B
Title Page/Disclaimer		B*
PSS	1	B
Preface	1, 2	Orig.
Contents	1 thru 3	Orig.
1	1 thru 3	Orig.
	4, 5	A
	6	Orig.
	7	A
	8 thru 10	Orig.
2	1 thru 6	Orig.
3	1 thru 34	Orig.
4	1 thru 7	Orig.
Appendix A	1 thru 9	Orig.
Appendix B	1 thru 7	Orig.
Index	1 thru 5	Orig.
User Comment Form		

*New page

All the technical changes are denoted by an arrow (⇒) in the margin. A downward pointing arrow (⇓) next to a line indicates that technical changes begin at this line and continue until an upward pointing arrow (⇑) is found. A horizontal arrow (⇒) pointing to a line indicates a technical change in only that line. A horizontal arrow located between two consecutive lines indicates technical changes in both lines or deletions.

Preface

This manual is one of a series designed to instruct and guide the programmer in the use of the SPERRY Operating System/3 (OS/3). It specifically describes the SORT3 program available to users of OS/3. The intended audience is the programmer with a basic knowledge of data processing and whose experience is on systems other than those of Sperry. The programmer should also have a basic knowledge of job control language (JCL).

Programmers with experience on SPERRY systems should refer to the independent sort/merge user guide/programmer reference, UP-8819 (current version). Users of basic assembly language (BAL) should refer to the sort/merge macroinstructions user guide/programmer reference, UP-9072 (current version). An introductory manual, the introduction to sort/merge, UP-8073 (current version), is also available. It briefly describes the general characteristics and facilities of all the sort/merge programs available to users of OS/3.

Other current OS/3 publications referenced in this manual that are helpful when using SORT3 are:

- Data utilities user guide/programmer reference, UP-8834

  Describes the data utility routine.

- System service programs (SSP) user guide, UP-8841

  Describes various system utilities (e.g., librarian, linkage editor).

- General editor user guide/programmer reference, UP-9976

  Describes the OS/3 general editor (EDT).

- Job control user guide, UP-9986

  Describes the job control language used under OS/3.

- Interactive services commands and facilities user guide/programmer reference, UP-9972

  Describes the commands and operating procedures for workstation terminals.

- Consolidated data management macroinstructions user guide/programmer reference, UP-9979

  Describes data management macroinstructions.

- Basic data management user guide, UP-8068

  Describes the effective use of OS/3 basic data management.

The subject matter in this manual is divided into the following sections:

SECTION 1. INTRODUCTION

Describes what SORT3 does for you and gives you some background as to its structure, performance considerations, and usage restrictions. Section 1 also discusses structuring your input/output data and describes the running of a SORT3 job from a workstation.

SECTION 2. BASIC CONCEPTS

Describes the execution of the SORT3 program, its operational phases, and record handling, as well as listing the characteristics of the sorts performed by SORT3.

SECTION 3. SORT REQUIREMENTS YOU SUPPLY

Tells you how to prepare job control statements and sort control specifications.

SECTION 4. PROGRAM AND CONTROL STREAM EXAMPLES

Includes sample sort program control specifications and control streams for a variety of sort applications.

The following appendixes are also included:

Appendix A. Standard EBCDIC and ASCII Collating Sequences

Appendix B. SORT3 Specifications Summary

Contents

3. SORT REQUIREMENTS YOU SUPPLY

4. PROGRAM AND CONTROL STREAM EXAMPLES

APPENDIXES

A. STANDARD EBCDIC AND ASCII COLLATING SEQUENCES

B. SORT3 SPECIFICATIONS SUMMARY

INDEX

USER COMMENT SHEET

FIGURES

TABLES

1. Introduction

1.1. WHY YOU NEED A SORT PROGRAM

Why is it important that you have a sort capability? Well, consider the amount and types of data contained in your files, and the number of ways in which you use that data. You'll probably discover that you seldom use all of the data for every job and that the organization of the data does not always lend itself to efficient methods of processing during certain applications. In general, most files contain a collection of data records, possibly of different types, that have no relationship other than their existence in the same file. Finding records and specific types of data in your files requires a search, and searching takes time. However, less time is expended to search an ordered file than to search an unordered file, and time is directly related to processing efficiency. This is where a good sort program comes into play. It allows you to select the data you need and to organize that data according to criteria such as an employee number, a customer account number, an inventory item, or whatever your particular job application requires. Remember, data is useless for the most part unless it can be related to something real, such as the type of record entries mentioned. A file properly organized and formatted for the job at hand allows the use of techniques that achieve faster searching of your files, faster determination of the presence or absence of the information needed, and faster record retrieval during job execution.

1.2. WHAT SORT3 DOES FOR YOU

The SORT3 program is an easy-to-use, *canned* sort program because it is modular in design, requires a minimum of user programming, and does not need to be assembled or linked to your program. It increases the versatility of the OS/3 sort package by providing you with a program that is compatible with the IBM System/3, 32, and 34 sort. That is, SORT3 accepts, with minor differences, all System/3, 32, and 34 sort specifications and offers all the features of the System/3, 32, and 34 sort that are feasible within the OS/3 operating system. In addition to disk and tape input files, the SORT3 program is capable of processing input data from card and diskette files. It also provides you with added control over the record sequencing, data reduction, and data disposition without the necessity of reverting to user own-code routines.

SORT3 can be initiated through OS/3 job control language (JCL); instructions for running SORT3 under OS/3 JCL are provided in Section 3.

In general, SORT3 assists you in producing a tailored output file from your existing input data files. Through the sorting techniques employed in this program, you can reformat a file (rearrange records and selectively include or omit specific record types), reformat records, and summarize record fields. The types of sorts performed include full record sorts, tag sorts, and summary sorts. Specifically, SORT3 can:

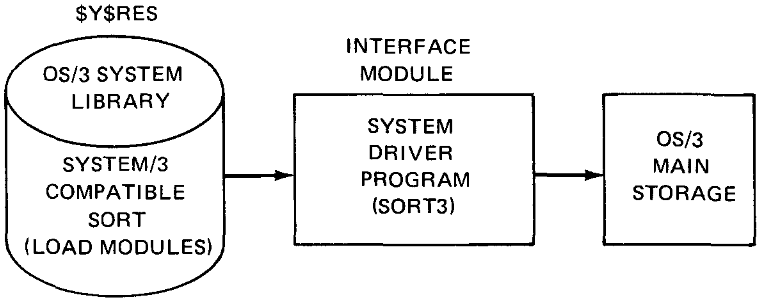
- sort records in ascending or descending sequence;
- sort fixed-length or variable-length records;
- sort blocked or unblocked records;
- sort records with noncontiguous key or control fields;
- recognize key and control fields in the following formats:
  - Character
  - Binary (signed and unsigned)
  - Decimal (signed zoned and unsigned zoned)
  - Packed decimal
  - Leading and trailing sign numeric
  - Overpunched leading and trailing sign numeric
  - EBCDIC data in ASCII collating sequence
  - Floating point (single and double precision)
- sort two or more different characters having the same collating value (multiple character sort);
- sequence files according to user-specified (alternate) collating sequence;
- perform data validity and data integrity checks during sorting; and
- perform restart procedures for tape sorts.

The output produced from your sort job is file formatted according to your instructions to the sort program. You are not, however, automatically provided a copy of the output file produced by the sort. If you want a copy of the sorted file, you can obtain it by running the appropriate data utility routine, as described in the data utilities user guide/programmer reference, UP-8834 (current version). The successful execution of your job results in a *terminated normally* message printed on your job log and a list of the total number of records both included in the sort and deleted during the sort.

1.3. CONCEPT OF MODULAR SORT STRUCTURE

In the process of describing the SORT3 program, we have referred to it as being modular in structure. What do we mean by modular? Modular, as related to the sort programs, refers to the method used to package them. Rather than writing separate sort programs for every conceivable type of sort, we have broken the sort/merge process into a group of interrelated, yet independent, functional subtasks. The subtasks are coded as executable routines and provided to you as load modules residing in the system load library ($Y$LOD). Their implementation into your job is based on the structure you establish in your job stream. That is, you define the type of sort you want performed through parameterized statements in your job control stream, and the sort program will structure the sort/merge process accordingly. One advantage of modular programming is that it conserves main storage space. The sort program loads only those modules needed for the particular sort/merge phase being executed. It also aids in adapting the OS/3 sort programs to the requirements of your installation by increasing programming flexibility.

In addition to the sort modules, SORT3 provides a call module to interface with the system. This call module is the SORT3 system driver program, which resides in $Y$LOD.


$Y$RES
OS/3 SYSTEM LIBRARY
SYSTEM/3 COMPATIBLE SORT (LOAD MODULES)
INTERFACE MODULE
SYSTEM DRIVER PROGRAM (SORT3)
OS/3 MAIN STORAGE

If you want to copy the SORT3 program onto your own user library file, you can use the librarian as described in the system service programs (SSP) user guide, UP-8841 (current version). Be sure to include the SORT3 system driver program and sort load modules beginning with SM$ located in $Y$LOD.

1.4. PROGRAM RESTRICTIONS AND CONSIDERATIONS

Variations in program design, capability, and implementation sometimes restrict the use of a sort program for specific applications or for specific system configurations. Consider the following:

- All sorting is limited to storage-only, disk-only, and tape-only, single-cycle sorts.
- Input files can be disk, diskette, card, or tape, but not mixed.
- SORT3 cannot perform a merge-only operation (merging two or more previously sequenced or sorted data files).
- SORT3 does not support data reduction or record sequencing and checking through the use of user own-code exit routines.
- Auxiliary storage work areas can be disk or tape, but not both, and are limited to six disk files or six tape files.
- Volume of data sorted and merged is limited by the type and physical capacity of the tape or disk space assigned as auxiliary work storage.
- SORT3 deletes duplicate records during summary sort (SORTRS) only.
- The FILTYPE parameter is ignored when the system is generated to support only CDM mode file access.
- If the system supports both CDM and DTF file access, the FILTYPE parameter may be used to specify the file type as IRAM (for MIRAM), NI, or SAM.

  If the FILTYPE parameter is not specified, the output file type will be the same as the input file type. Or, if an input file is not specified, the output file type will be MIRAM.

1.5. ELEMENTS AFFECTING PERFORMANCE OF A SORT PROGRAM

The careful user should be aware of elements affecting the performance of his sort program. These elements are:

- Available main storage
- Number and type of assigned auxiliary storage devices
- Record characteristics
- Input and output data file organization
- Options under which the sort program operates

Remember to be explicit in supplying instructions to your sort program and to be careful in setting up your file and record formats. This results in faster sorts that require less central processor time and reduces the number of I/O operations required. To improve program efficiency, consider these factors during record and file preparation:

- Record size
- File size
- Key or control field size
- Number of key or control fields
- Record format
- File format

As a rule, simplification reduces processing and the time needed to perform a function. By simplifying the key fields and decreasing their number and size, you decrease the number of comparisons and the length of time needed to make each comparison. Sort performance improves when input and output records are blocked. Decrease record size and you increase efficiency because a greater number of records are processed at one time for a given amount of main storage.

To improve processing speed and efficiency:

- Be generous with storage; assign more than one I/O device to the sort for auxiliary storage and more than the minimum amount of main storage.
- Simplify your file and record formats.
- Be explicit in defining your output file requirements to the sort program.

1.5.1. Main Storage Allocation

In general, the more main storage available to a sort program, the more efficient the performance. It decreases the number of I/O functions because fewer passes are needed to produce strings of sequenced data for final merging. Therefore, proper consideration given to these factors when preparing your program reduces processing time and increases program efficiency. The minimum main storage requirements for your sort depend upon which method you use.

SORT3 requires a minimum of 16,000 bytes. However, the more sort specifications you include in your program, the less main storage is available for the actual sort. Each sort specification processed requires 12 bytes of main storage. Also bear in mind that additional main storage is required when using an alternate collating sequence (280 bytes), field specifications for packed data (40 bytes each), and include or omit (or both) specifications for packed data (100 bytes).

Performing large volume sorts is most efficient when 50,000 to 150,000 bytes of main storage are allocated.

1.5.2. Auxiliary Storage Work Area Assignments

Work areas may be assigned as auxiliary storage on tape or disk, but not both. If disk storage is used, all work area disks must be of the same general type, i.e., sectorized or nonsectorized. It is important not to underestimate the amount of auxiliary storage required. When possible, avoid assigning the bare minimum of auxiliary storage needed; otherwise, the sort program must perform a greater number of intermediate merge passes to sequence records. This wastes time and reduces program efficiency. Because the volume of data processed varies with the quantity and type of magnetic tapes or disks assigned as auxiliary storage, selecting auxiliary storage devices with faster data transfer rates results in a faster sort. Data volume doesn't reduce sort performance.

Disk space is assigned by using standard sort work file names DM01,...,DM0n or system scratch space file names $SCR1,...,$SCRn (in consecutive order) on LFD job control statements or by using WORK jproc calls. If one work file is allocated, the file name DM01 or $SCR1 must be assigned; if two are used, the names DM01 and DM02 or $SCR1 and $SCR2 must be assigned, and so forth. The SORT3 program is limited to a maximum of six disk files. The amount of disk space requested must be sufficient to hold the entire volume of data to be sorted, plus 10 to 20 percent additional space for overhead requirements. (An additional 10 to 20 percent space should be requested if data involves variable-length records.) In addition, all disk files used in the sort operation must be the same type; i.e., mixed disk types are unacceptable. Table 1–1 contains a comparison of the direct access storage devices used by SORT3.

Table 1—1. Comparison of Data Capacities and Access Speeds for Direct Access Devices

Characteristics	Disk Subsystem Type							
	8416	8417	8418–92/93	8418–94/95	8419	8430	8433	8470
Maximum data capacity (8-bit bytes per disk pack)	28,958,720	118,270,000	28,958,720	57,917,440	72,396,800	100,018,280	200,036,560	491,520,000
Maximum track capacity (bytes)	10,240	15,360	10,240	10,240	12,800	13,030	13,030	24,576
Minimum cylinder access time (ms)	10	7	10	10	10	7	7	4
Average cylinder access time (ms)	30	35	27	33	33	27	27	23
Maximum cylinder access time (ms)	60	70	45	60	60	50	50	46

When tape is used, the auxiliary storage work areas use labeled or unlabeled tapes. Work files are assigned by using standard tape sort file names SM01 through SM06 (in consecutive order) on LFD job control statements. A minimum of three tape units, and a maximum of six, may be assigned. Each tape work file must be large enough to contain all of the input data; i.e., the volume of data that can be processed in a tape sort is limited to the capacity of the smallest reel of tape assigned to the sort. The speed (rate) of data transfer varies according to the tape density (number of bits recorded across the width of the tape) and tape device. Refer to Table 1–2.

Table 1–2. Comparison of Transfer Rates for Magnetic Tape Devices

Tape Density (bpi*)	**Data Transfer Rate (bps**)**				
	UNISERVO 10	**UNISERVO 22**	**UNISERVO 24**	**UNIVERSO 26**	**UNIVERSO 28**
9-track (phase encoded) 1600	40,000	120,000	200,000	120,000	200,000
9-track (NRZI) 800	20,000	60,000	100,000	60,000	100,000

* bpi = bits per inch
**bps = bits per second

1.5.3. I/O Data File Organization

Data file organization begins with record layouts. Assuming you have a fixed number of records, a file of large records takes longer to sort than a file of smaller records. Also, larger control fields and more control fields per record increase sort time because lengthier comparisons are needed.

Record sizes that exceed one-half track in length may require up to 100 percent more space or twice the normal space calculated by multiplying the number of records to be sorted by the record size.

1.5.4. SORT3 Options Affecting Performance

There are two header specification entries that can affect SORT3 performance:

- VERIFY OPTION

  The VERIFY OPTION specification can be used to improve program performance by requesting SORT3 not to verify the data written to the work files during the sort.

- ALT COLLATING SEQUENCE

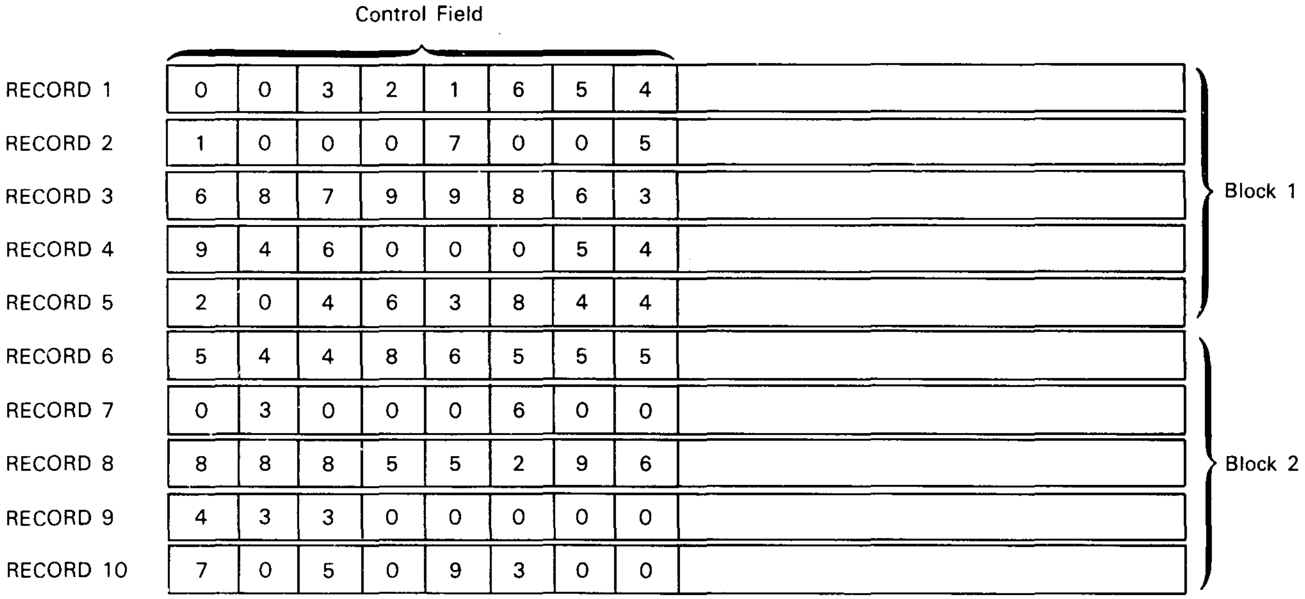
  If you use the ALT COLLATING SEQUENCE field to specify an alternate sequence, you increase sort performance time.

1.6. STRUCTURING YOUR INPUT/OUTPUT DATA

When you first consider the problem of sorting data, you may be faced with a large volume of information that may or may not be organized into workable units. Dividing information into records, blocks, and files helps you and the computer both identify where the data is located and control the changes or manipulations you want performed. After carefully examining the nature and content of the input data and determining the record layout and block size that best suit your needs, you must indicate, via your control stream, what size records and blocks you intend to input for processing and output after the sorting operation is completed.

Records can be divided into smaller units called *fields.* Specific fields, called *control fields,* are used for comparing records to arrange them in the order you want. To tell the sort program which control fields to use, you must specify their size and position within records.

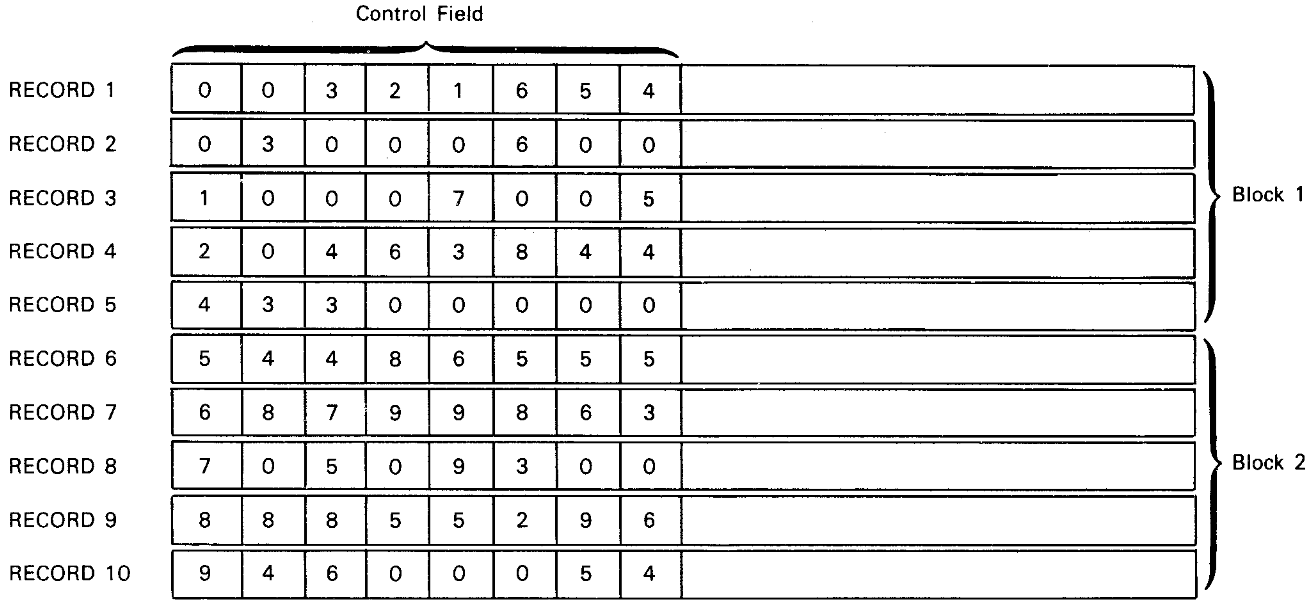
Figure 1-1 illustrates what the data contained in control fields of the first two input data record blocks might look like before the sort.



	Control Field									
RECORD 1	0	0	3	2	1	6	5	4		Block 1
RECORD 2	1	0	0	0	7	0	0	5		
RECORD 3	6	8	7	9	9	8	6	3		
RECORD 4	9	4	6	0	0	0	5	4		
RECORD 5	2	0	4	6	3	8	4	4		
RECORD 6	5	4	4	8	6	5	5	5		Block 2
RECORD 7	0	3	0	0	0	6	0	0		
RECORD 8	8	8	8	5	5	2	9	6		
RECORD 9	4	3	3	0	0	0	0	0		
RECORD 10	7	0	5	0	9	3	0	0		

Figure 1—1. Input Data Records before Sort

Of course, your volume of data is much larger than the two 400-byte record blocks shown in Figure 1-1, but the results of sorting the records in ascending order by control fields should be as shown in Figure 1-2.



	Control Field									
RECORD 1	0	0	3	2	1	6	5	4		Block 1
RECORD 2	0	3	0	0	0	6	0	0		
RECORD 3	1	0	0	0	7	0	0	5		
RECORD 4	2	0	4	6	3	8	4	4		
RECORD 5	4	3	3	0	0	0	0	0		
RECORD 6	5	4	4	8	6	5	5	5		Block 2
RECORD 7	6	8	7	9	9	8	6	3		
RECORD 8	7	0	5	0	9	3	0	0		
RECORD 9	8	8	8	5	5	2	9	6		
RECORD 10	9	4	6	0	0	0	5	4		

Figure 1—2. Data Records after Sort

1.7. RUNNING YOUR SORT3 JOB FROM A WORKSTATION

OS/3 provides you with the capability of running your SORT3 job interactively. This means two things:

1. You can build a control stream to execute the SORT3 job at a workstation, as opposed to punching it on cards or writing it to a diskette.

2. You can initiate the running of the control stream from the workstation, as opposed to asking the system operator to run your job for you.

The easiest way to build a job control stream from a workstation is by using the general editor. This allows you to key in your control stream statements and have them stored on a library file. Then, later you can initiate the running of the program by simply keying in the appropriate system command, that is, RV for an OS/3 control stream. ←

If you are not familiar with job control, use the job control dialog for assistance. The job control dialog is an interactive facility of OS/3 that allows you to describe your job requirements to it in English, in response to a series of questions. Then the job control dialog produces as its output the job control stream needed by OS/3 to run your SORT3 job. The control stream produced by the job control dialog is virtually identical to the control stream that you would have to produce if you were running your job in a batch environment. Only now, you do not have to be concerned with the intricacies of the job control language. The job control dialog eliminates this requirement on your part.

After you have answered the questions presented to you by the job control dialog, it builds a control stream and stores it in a permanent library file for you. From here, you can initiate its running by simply keying in the appropriate system RUN command; or if you'd rather, you can change the contents of the control stream by using the general editor.

The procedures for activating the general editor are detailed in the general editor user guide/programmer reference, UP-9976 (current version).

The procedures for activating the job control dialog and initializing the running of a job are detailed in the job control user guide, UP-9986 (current version).

Although the sample job control streams in this manual are shown on cards, the rules for preparing your sort control statements and specifications also apply to entries keyed in at a workstation.

2. Basic Concepts

2.1. GENERAL

The SORT3 program assists you in sorting and merging your data files with only a minimum amount of user program intervention. Simplicity is the word that best describes the use of this program because, basically, it does all the work for you. It reads your data files, selectively sorts and merges the data according to your specifications, and then writes the data to your output file. You do not have to concern yourself with opening and closing files, supplying read and write routines, passing data to the sort program or retrieving sorted data for the output. Your responsibility is to provide the data files to be sorted, and to prepare the control stream necessary to define the sort and execute the program. In relation to the entire job, your program responsibility, as shown in Figure 2-1, is involved only with the input step of running SORT3. Details for preparing both control statements and sort specifications are covered in Section 3.

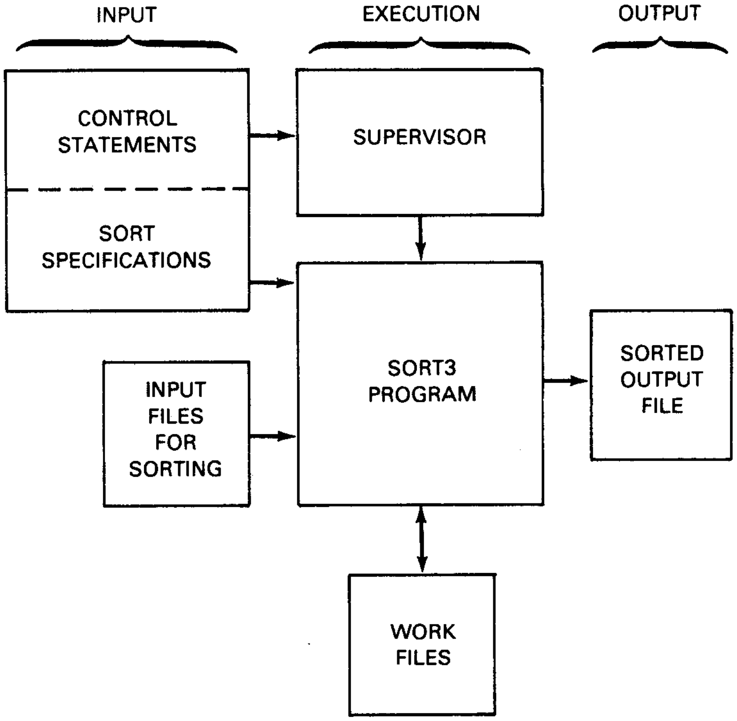


Figure 2—1. Functional Divisions of a SORT3 Job

2.2. EXECUTION OF THE SORT3 PROGRAM

Execution of the SORT3 program takes place after the system input device has read your control stream. In the discussion of program execution, take note of the interplay of activities between your control stream input, the system, and the SORT3 program. The entire sort/merge operation centers around the elements supplied by both you and SORT3.

Program execution begins when the EXEC statement (JCL) is read from the control stream you submitted to the system for running your job. This statement signals the system to load the system driver program (Figure 2–2) for SORT3 into main storage. The system driver program provides the interface between the system and the remaining SORT3 program modules. The first action taken by the driver program is to call into main storage the sort modules needed to initialize the sort process. The loading of the modules signals the sort program to accept your sort specifications and execute the first phase of the program: initialization and assignment. As explained in Section 1, the sort program is modular and the various modules used in the sort process reside in the system load library file ($Y$LOD) on the SYSRES volume. When one phase is completed, it signals the driver program to load the next group of modules into main storage and execute the next phase in the sort process.

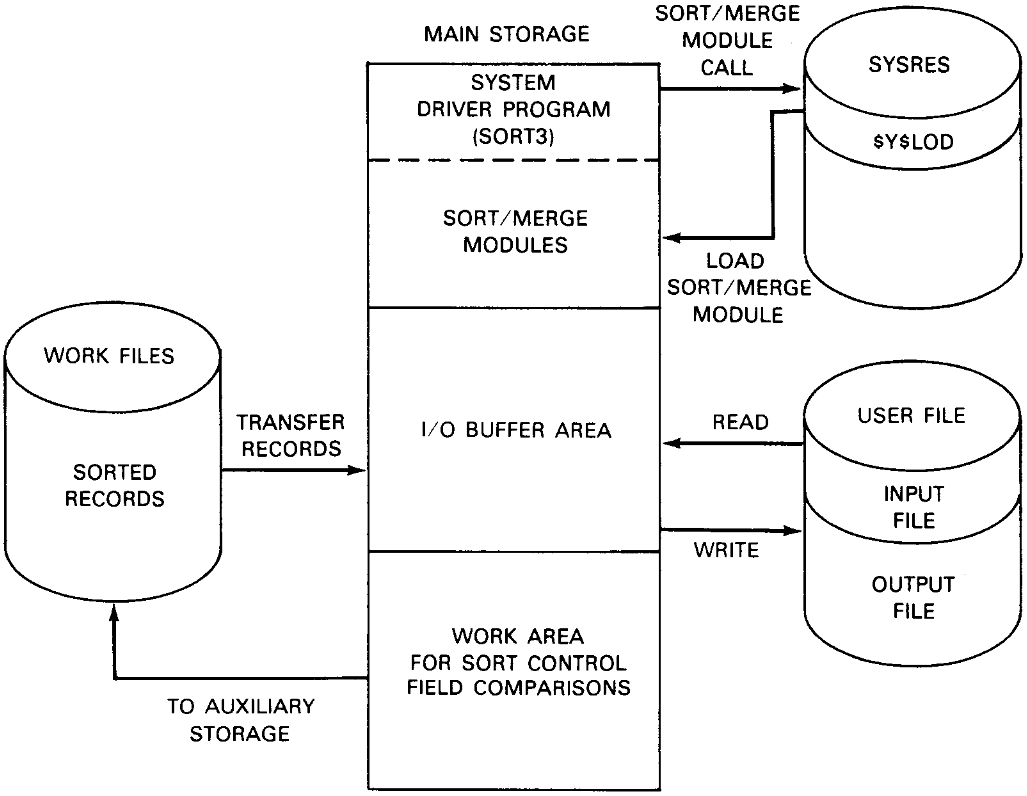


Figure 2—2. Execution of SORT3 Program

2.3. SOFTWARE FRAMEWORK

SORT3 consists of the following four operational phases, which are normally executed in sequence:

1. Sort Initialization and Assignment (Phase 1)

   This phase initializes the sort process by reading sort control statements from the job control stream. It validates both the content and syntax of these statements, and examines your parameters to determine the sort function to be performed. In addition, it builds a parameter table, sets up compare routines, and structures the SORT3 processor to perform only the sort functions you have specified.

2. Data Input and Internal Sort (Phase 2)

   This phase initiates an input routine that opens your input files, validates file labels, and reads the data records one at a time. (The location of your data files is determined by the device assignment sets in your control stream.) Before a record is passed to the internal sort routine of this phase for initial sorting, it is checked against the criteria of your sort specifications to determine whether it is to be included in the sort. If the record is to be included, it is reformatted into a sort record (according to your specifications) and passed to the internal sort routine. (The details of sort record handling are described in 2.4.)

   Internal sorting is performed in main storage and produces strings of sequenced data that are written as intermediate files to auxiliary storage devices (tape or disk). If the number of data strings produced during the internal sort are few enough to be merged in one final merge, the preliminary merge is bypassed and control passes to the final merge and output phase. Otherwise, strings of sequenced data must be continuously merged into larger and larger data strings until only one final merge operation is required to produce an output file sequenced in the order you specified.

3. Preliminary Merge (Phase 3)

   In this phase, the data strings produced by the internal sort are continuously merged, with each successive merge producing longer and longer sequenced data strings. When only one final merge pass is needed to create a single sequenced string (final output string), control passes to the final merge phase.

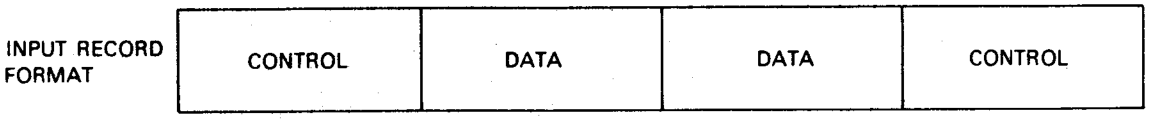
4. Final Merge and Output (Phase 4)

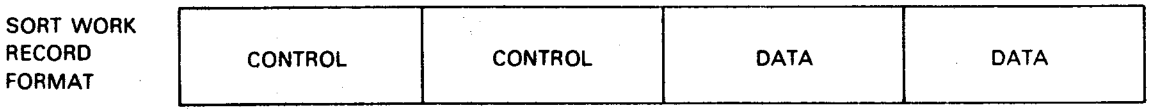
   The final merge phase merges all data strings written to the work files into one sequenced string and passes it to the sort output routine. The output routine opens your output file, writes the output data, closes the output file, terminates the sort, and returns control to the system.

   In cases where the input file is partially sequenced or is small enough so one final merge produces the required output sequence, SORT3 bypasses the preliminary merge and proceeds to the final merge, where the records are read into main storage, merged, and written to the output file.

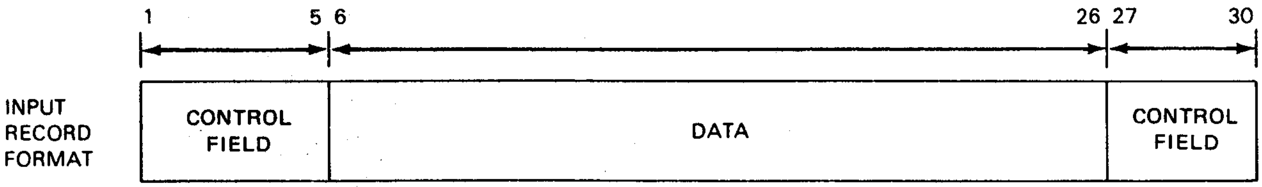
2.4. RECORD HANDLING

When your input records are read during the data input phase, SORT3 checks each record to see whether it is to be accepted or rejected on the basis of your sort specifications. SORT3 builds a sort work record for each input record accepted into the sort. The sort work record is reformatted to increase the efficiency of the sort process. (SORT3 does not change the format of the actual input record.) In the reconstructed format, control fields are placed ahead of data fields unless, of course, the control fields are to be dropped during the sort.

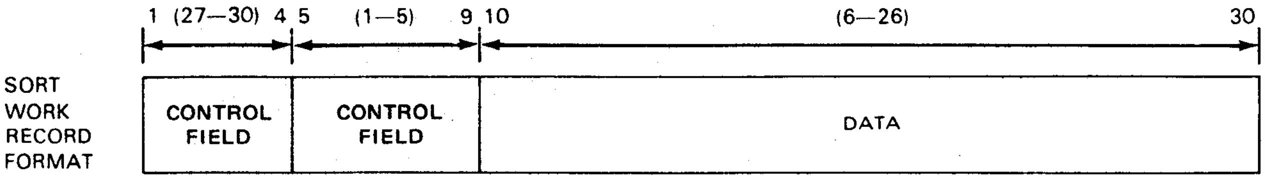




The placement of specific control fields and data fields within the sort work record is defined by the parameters of your record type and field description specifications. For example, assume you have identified positions 27 through 30 of your input records as a primary control field, and positions 1 through 5 as a secondary control field. Positions 6 through 26 contain data. Your input record would appear as:



When SORT3 accepts the input record for sorting, it repositions the record fields according to the sequence you have specified. The primary control field appears first, the secondary field next, and so on until all the control fields are properly positioned and followed by the data fields. The sort work record would appear as:



The sort work record is sent for initial (internal) sorting after it is constructed. SORT3 uses the control fields to sequence the records in either ascending or descending order according to your sort specifications. When all the records are properly sequenced, SORT3 writes the records to your output file. The output record format is the same as the sort work record format unless SORT3 is instructed to drop control fields during the sort.

2.5. TYPES OF SORTS PERFORMED BY SORT3

There are three types of sort jobs performed by SORT3: addrout (address out), tag-along (data fields can *tag along* with control fields in the sorted records), and summary tag-along (data is summarized in the sorted records).

The output from an addrout sort job (Figure 2-3) consists of 10-byte binary relative record numbers of the records in the input file.

NOTE:

An addrout sort can be used to process disk files only.

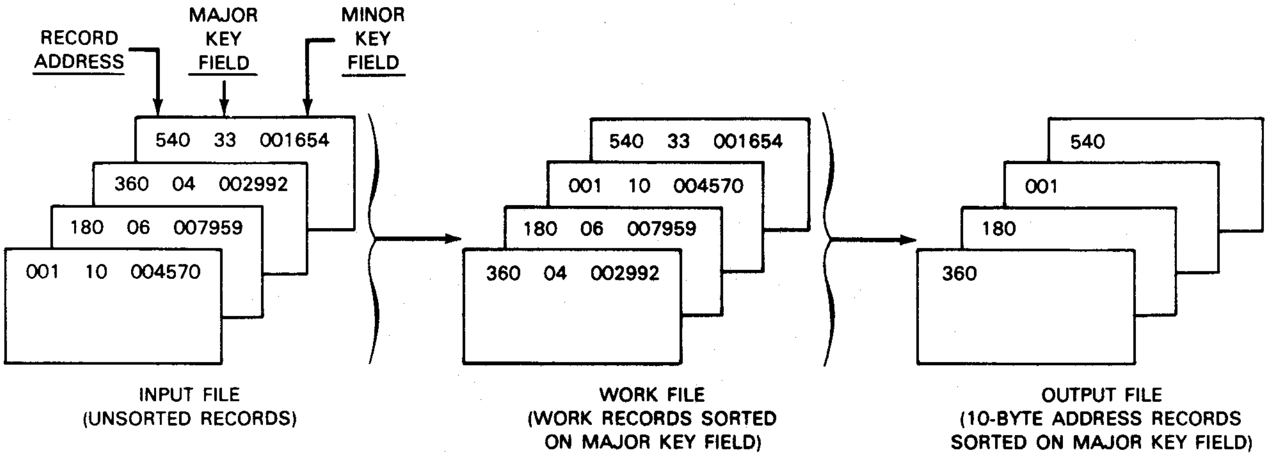


Figure 2-3. Example of Address-Out (Addrout) Sort

The output for a tag-along sort (Figure 2-4) is a file of sorted records containing the following:

- Control fields and data
- Control fields only
- Data only

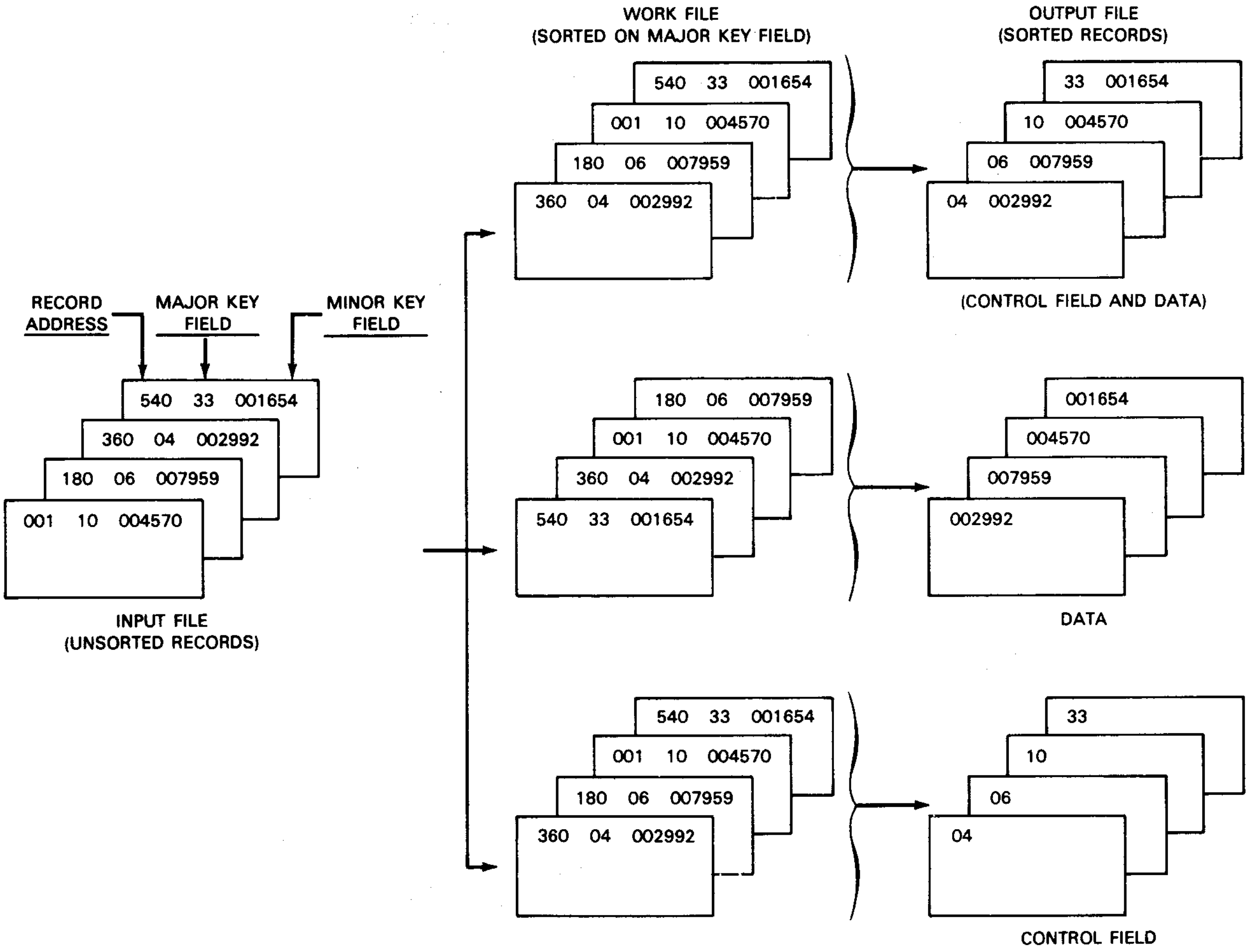


Figure 2–4. Example of Tag-Along Sort

The output for a summary tag-along is a file of sorted records containing:

- Control fields, data fields, and summary data
- Control fields only
- Data fields and summary data
- Data fields only
- Summary data fields only
- Control fields and summary data fields

3. Sort Requirements You Supply

3.1. GENERAL

To run a SORT3 program, you are responsible for:

- Identifying your job to the system
- Assigning the devices needed for the sort
- Initiating execution of the sort program
- Defining the criteria for the sort

The first three items in our list of responsibilities are achieved through the use of control statements in the job stream for SORT3. The last item is accomplished by a set of sort specifications also included in the job stream. The detail involved in the preparation of the control statements and sort specifications depends upon the complexity of the sort, the system configuration in which you run the job, and the size and format of your input files, to name a few.

Preparation of the sort specifications is probably the largest task in setting up your job. As you know, these specifications define the criteria governing performance of the sort. However, SORT3 simplifies this requirement by accepting the same sort specifications that you used in the System/3, 32, or 34 environment – namely, the header, record type, and field description sequence specifications. Your responsibility is to prepare the sort specifications (as described in 3.3) and include them as part of your control stream input.

The same does not hold true for the control statements needed to run your job. Using OS/3 job control, the control statements in your control stream must conform to the OS/3 JCL conventions as defined in the job control user guide, UP-9986 (current version). When using OS/3 job control, prepare your control statements and submit your job as you would for any other OS/3 job.

A typical job control stream for executing SORT3 under OS/3 JCL is shown in Figure 3–1. This job control stream can also be created and executed from a workstation. (See 1.7.)

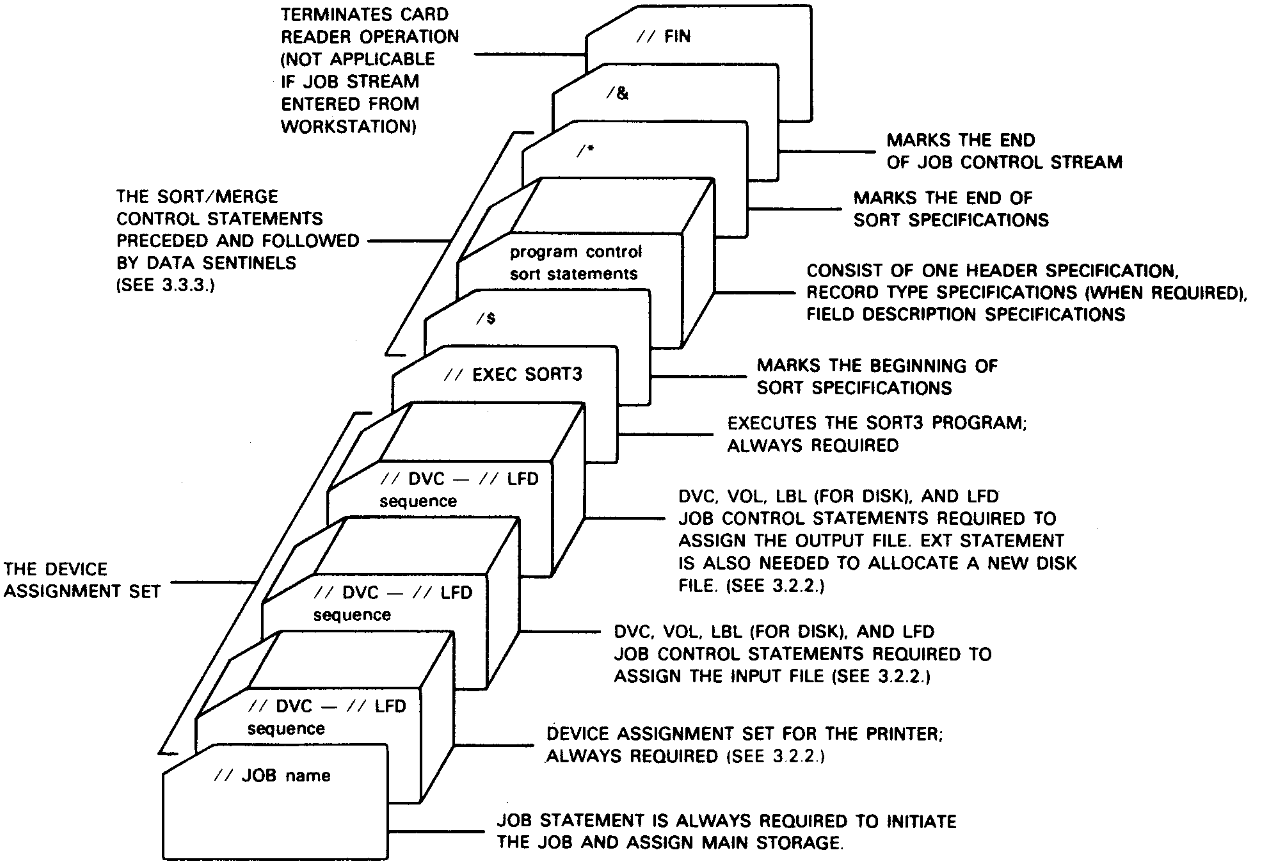

TERMINATES CARD READER OPERATION (NOT APPLICABLE IF JOB STREAM ENTERED FROM WORKSTATION)
// FIN
/&
MARKS THE END OF JOB CONTROL STREAM
/*
MARKS THE END OF SORT SPECIFICATIONS
THE SORT/MERGE CONTROL STATEMENTS PRECEDED AND FOLLOWED BY DATA SENTINELS (SEE 3.3.3.)
program control sort statements
CONSIST OF ONE HEADER SPECIFICATION, RECORD TYPE SPECIFICATIONS (WHEN REQUIRED), FIELD DESCRIPTION SPECIFICATIONS
/$
MARKS THE BEGINNING OF SORT SPECIFICATIONS
// EXEC SORT3
EXECUTES THE SORT3 PROGRAM; ALWAYS REQUIRED
// DVC — // LFD sequence
DVC, VOL, LBL (FOR DISK), AND LFD JOB CONTROL STATEMENTS REQUIRED TO ASSIGN THE OUTPUT FILE. EXT STATEMENT IS ALSO NEEDED TO ALLOCATE A NEW DISK FILE. (SEE 3.2.2.)
THE DEVICE ASSIGNMENT SET
// DVC — // LFD sequence
DVC, VOL, LBL (FOR DISK), AND LFD JOB CONTROL STATEMENTS REQUIRED TO ASSIGN THE INPUT FILE (SEE 3.2.2.)
// DVC — // LFD sequence
DEVICE ASSIGNMENT SET FOR THE PRINTER; ALWAYS REQUIRED (SEE 3.2.2.)
// JOB name
JOB STATEMENT IS ALWAYS REQUIRED TO INITIATE THE JOB AND ASSIGN MAIN STORAGE.

Figure 3—1. Typical Control Stream for Executing SORT3 under OS/3 Job Control

3.2. PREPARING JOB CONTROL STATEMENTS FOR YOUR SORT

The job control statements described in this section are used to direct the system in handling your SORT3 job in an OS/3 environment. They are responsible for:

- identifying and scheduling your job;
- assigning system resources for your job;
- defining your input, work, and output files;
- initiating the SORT3 program; and
- ending the job after the sort is completed.

3.2.1. Identifying and Scheduling Your Job

The first statement in the job stream is the JOB statement. It assigns a unique name to your job so that the system can distinguish it from other jobs being processed concurrently with your job. The JOB statement also specifies the minimum and maximum main storage requirements (in bytes) for the job and the priority of the job. The system will not schedule your job or allocate the required system resources to it if the JOB statement is omitted. The coding of a typical JOB statement may appear as:

```
// JOB PYRLSORT,,7000,9000
```

In this example, PYRLSORT is the 8-character alphanumeric name assigned to your job. The double comma indicates that you have elected not to assign a priority level to the job. The system in this case assumes a normal priority. The hexadecimal values 7000 and 9000 represent the minimum number of main storage bytes needed to execute the largest job step of your job and the maximum number of main storage bytes requested (not required) to execute the largest job step of this job.

3.2.2. Assigning Devices to Your Job

The next series of job control statements that appear in the job stream are the device assignment sets. Each device assignment set consists of as few as two job control statements (DVC and LFD), or as many as five job control statements (DVC, VOL, EXT, LBL, and LFD). The device assignment sets are used for the allocation of peripheral devices needed for printing messages, inputting data, handling data during processing, and collecting output data. They also identify the device type used, disk or tape volume mounted, and the files to be processed. Each device assignment set begins with a DVC statement that specifies the logical unit number for the device type upon which a particular file is mounted and ends with an LFD statement that associates a logical file name with that device. Detailed information about the device assignment statements and a list of specific I/O device numbers are provided in the job control user guide, UP-9986 (current version).

The first device that must be assigned for the sort job is the printer. SORT3 requires this device to print messages for operator action or information. The coding used to assign the printer may appear as:

```
// DVC 20 // LFD PRNTR
```

In this example, the printer to be assigned to your job is logical device 20. It must be assigned the system standard name PRNTR in the LFD statement.

Following the printer assignment set are the assignment sets for the input, work, and output files. The pattern of each set is similar. That is, the specifications for each file identify a device, a file on a volume, and a logical file name.

For example:

```
// DVC 65 // VOL SYS200 // LBL PAYROLL // LFD INPUT
// DVC 66 // VOL SCR200 // LBL $SCR1 // LFD DM01
// DVC 65 // VOL SYS200 // EXT SQ,C,CYL,5 // LBL EXEMPT // LFD OUTPUT
```

The statements used in this example are explained as follows:

- DVC

  The first DVC statement assigns device number 65 to your input file named PAYROLL. The second DVC statement assigns device 66 to a temporary work or sort scratch file named $SCR1. The third DVC statement also assigns device number 65 to your output file EXEMPT. Unless your input files are very low volume, it is advisable to assign one device for each sort work file and another device for your input and output files. The sort operates more efficiently when one work file is assigned per device.

- VOL

  The VOL statements uniquely identify the volumes mounted on the devices you have assigned. The input and output files mounted on device 65 are on volume SYS200 and the work file is on volume SCR200.

- EXT

  By specifying the EXT job control statement in a device assignment set, you can provide disk space for sort work files, designate information needed to create new files, or extend existing disk files. Each EXT statement applies to the volume specified on the immediately preceding VOL statement. In the example, the EXT statement is specified for the output file to be created.

- LBL

  The LBL statements provide data management with the file identifier used to locate your file on the specified volume. Only one LBL statement is allowed per device assignment set. In the coding example, PAYROLL is the file identifier for the input file, $SCR1 is the identifier for the work file, and the EXEMPT is the identifier for the output file.

- LFD

  To associate the file information in your job control stream with the data management file definition, you must assign a logical or internal file name to each file. Logical file names are assigned via the LFD statement. For the SORT3 program, you must use the system standard names INPUT or INPUT1 through INPUT8 for the input file and OUTPUT for the output file. (You may assign a maximum of eight input files to your job, providing they all contain the same size records.) The LFD statements for sort work files must specify the system standard names DM01 through DM03 or $SCR1 through $SCR3, in consecutive order starting with DM01 or $SCR1. Therefore, the LFD statement for the work file in the coding example is DM01.

Although the example shown uses disk devices exclusively for input, work, and output files, you are not limited to disk for these files. In addition to disk, input files may reside on card, magnetic tape, and diskette; work files can be on magnetic tape; and output files can be written to magnetic tape and diskette. For magnetic tape input, you specify the record size and block size by using the //DD job control statement.

For example:

```
    //DD ,BLCKSZ=256,RCSZ=8Ø
```

When working under consolidated data management (CDI mode), BLCKSZ is not specified. When working under basic data management (DTF mode), BLCKSZ is specified. Refer to consolidated data management macroinstructions user guide/ programmer reference, UP-9979 (current version), or basic data management user guide, UP-8068 (current version).

3.2.3. Initiating the Execution of the SORT3 Program

The // EXEC job control statement in your job control stream initiates the execution of the SORT3 program. The // EXEC statement immediately follows the device assignment sets in the job control stream. The format of the // EXEC statement for SORT3 is:

```
    // EXEC SORT3
```

3.2.4. Marking the End of Your Job

So far we have provided the system with all the control information and control data needed to execute your job. Now you must mark the end of your job so that job control does not confuse it with other jobs in the control stream. (This could occur when the system finishes executing your job and queries job control for more input.) To mark the end of job, place /& job control statement at the end of your job control stream.

If no other jobs follow your job in the control stream and your control stream is contained on punched cards, you'll want to terminate the card reader operation. This is accomplished by including the // FIN control statement as the last statement in your job control stream.

3.3. SORT CONTROL SPECIFICATIONS

To determine which modules to include in the sort, SORT3 must be instructed how to conduct the sort. Directing sort execution is accomplished through the use of sort specifications. The specifications convey:

- what type of sort to perform;
- which record types to select for sorting;
- how to format the sorted records;
- how to format the output file; and
- what information (if any) is to be printed for user/operator use.

You are responsible for supplying the sort specifications to SORT3, as part of your control stream for your job. SORT3 accepts the specifications as control data and uses the information presented in their parameter fields to structure the execution of its modules to sort the records of your file.

The SORT3 specifications are:

- Header

  Defines the type of sort, the format of the sorted file, and the system information printed.

- Record type

  Defines the record types to be included in or omitted from the sort.

- Field description

  Defines the format of the output records.

These specifications always follow the // EXEC statement in your control stream. A start-of-data sentinel (/$) marks the beginning of the specifications in your control stream, and an end-of-data sentinel (/*) marks the end.

3.3.1. Determining the Sort Specifications Needed

The number of sort specifications that must appear in your job control stream is based upon the answer to two questions concerning your job.

1. Are all the records contained in your input file to be sorted?

2. Do all the records to be sorted have the same format?

If the answer to both questions is yes, you can bypass the normal specification requirement of header, record type, and field description, and provide only the header and field description specifications. SORT3 does not have to be selective in record processing because all the records are included in the sort and they are all of the same type. (SORT3 considers the job to be an implied, include-all-records type sort.) On the other hand, SORT3 must be selective in its record processing whenever the answer to either or both questions is no. Under these circumstances, you must identify the specific record types you want included in or omitted from the sort; therefore, a record type specification must be included for each record type. The general rules for determining how to include sort specifications in your job control stream are:

1. One header specification is required for every sort job, and it is always the first specification in the sequence.

2. A record type specification is required whenever the sort is not to include every record in your file or the records selected for the sort have different formats. Under these circumstances, a record type specification is required for each type of record.

   NOTE:

   When a warning message is issued, the UPSI byte is set to hexadecimal 40. Refer to the system messages programmer/operator reference, UP-8076 (current version) to determine the nature of the warning and the corrective action needed to be taken. SORT3 continues to run when a warning message is issued.

3. Record type specifications are paired with field description specifications. A field description specification must be provided for each record type specification appearing in the job control stream.

4. Each field description specification immediately follows its associated record type specification.

5. The paired record type and field description specifications follow the header specification in the control stream.

The requirements for providing sort specifications and the sequence in which they must appear in your job control stream are summarized in Table 3–1.

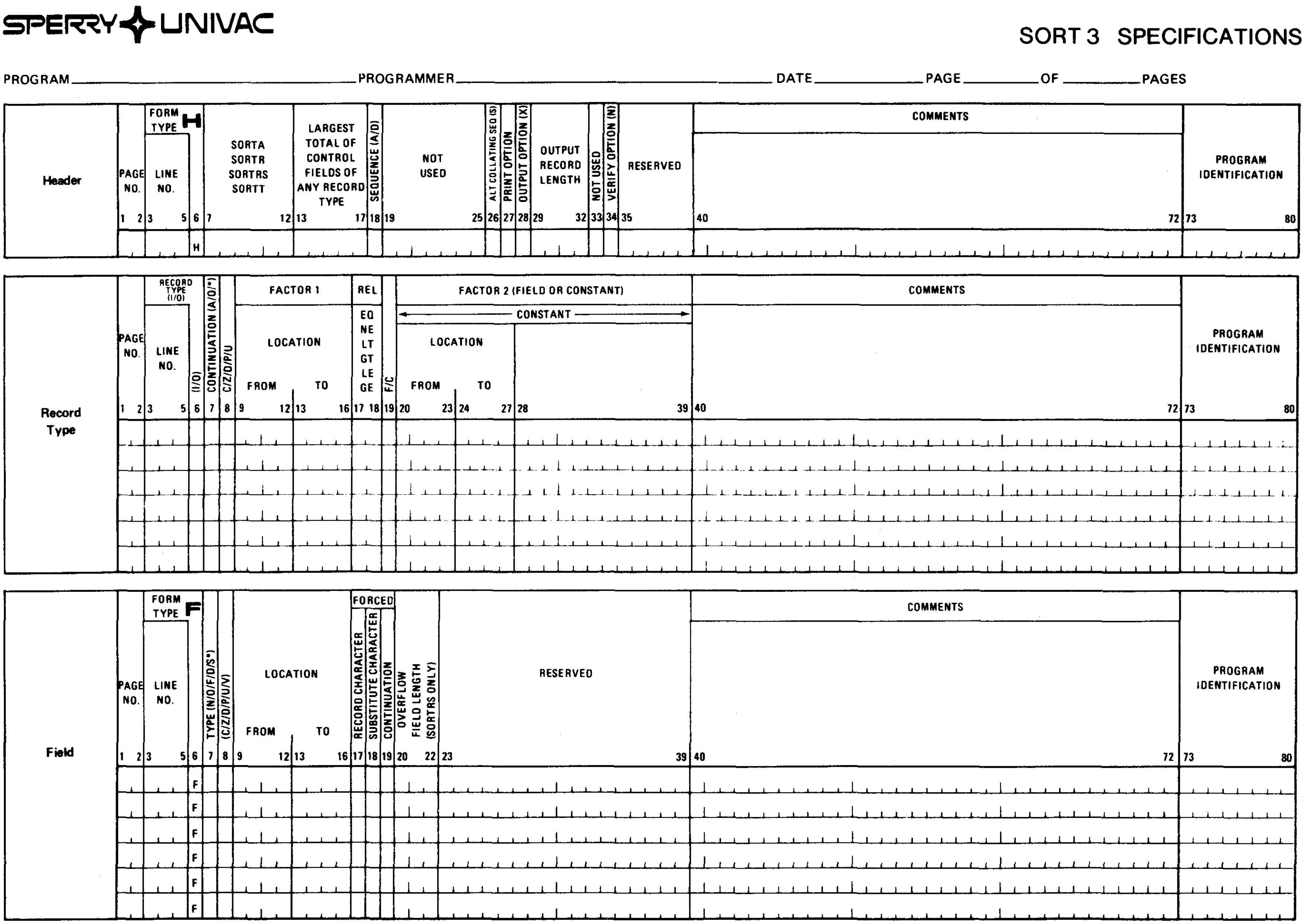
Table 3—1. Conditions Governing SORT3 Specification Requirements

Record Format	Number of Records to Be Sorted	Sequence Specifications Required (Arrange in Order Listed)
Same for all records to be sorted	All	■ Header ■ Field description
	Not all	■ Header ■ Record type ■ Field description
Several different formats for the records to be sorted	All or specified number from the file	First Record Type Format: ■ Header ■ Record type ■ Field description Second and Each Subsequent Type Format: ■ Record type ■ Field description

3.3.2. Numbering Your Sort Specifications

It is not required that you number the sort specifications appearing in your job control stream. However, numbering does avoid the possibility of getting the specifications out of sequence if you should accidentally drop or mix up the card deck containing the specifications. When used, every sort specification in your job control stream must be numbered so the SORT3 program can determine whether a specification is out of order or whether the entire specification sequence is arranged in a descending order. Because SORT3 is designed to process sort specifications numbered in ascending sequence, either of the other two conditions mentioned causes the program to issue a warning message to the operator and to halt the sort. The sort remains halted until the operator instructs SORT3 to continue processing or to terminate the job. To avoid this problem, make certain that each sort specification is properly numbered and that the entire sequence of specifications is arranged in ascending order by those numbers.

What constitutes a properly numbered sort specification? To help understand sort specification numbering, refer to the SORT3 Specifications form shown in Figure 3–2. The SORT3 Specifications form is designed so each page of the form contains the facilities for specifying one header specification, one record type specification, and one field description specification. As you can see, the field columns of the specifications correspond to the columns of the card images used in your control stream. The purpose of the form is to provide you with an easy method of organizing and defining the sort specifications applicable to your sort. When you are satisfied that the specifications are properly defined and are arranged in the order you want them processed, transferring them to the punched cards or entering them from the workstation keyboard becomes a simple matter.



SPERRY UNIVAC

SORT 3 SPECIFICATIONS

PROGRAM ______ PROGRAMMER ______ DATE ______ PAGE ______ OF ______ PAGES

			FORM TYPE H												COMMENTS	
Header	PAGE NO.	LINE NO.		SORTA SORTR SORTRS SORTT	LARGEST TOTAL OF CONTROL FIELDS OF ANY RECORD TYPE	SEQUENCE (A/D)	NOT USED	ALT COLLATING SEQ (S)	PRINT OPTION	OUTPUT OPTION (X)	OUTPUT RECORD LENGTH	NOT USED	VERIFY OPTION (N)	RESERVED		PROGRAM IDENTIFICATION
	1 2	3 5	6	7 12	13 17	18	19 25	26	27	28	29 32	33	34	35	40 72	73 80
			H													

		RECORD TYPE (I/O)				FACTOR 1		REL		FACTOR 2 (FIELD OR CONSTANT)			COMMENTS	
										CONSTANT				
Record Type	PAGE NO.	LINE NO.	(I/O)	CONTINUATION (A/O/*)	C/Z/D/P/U	LOCATION FROM	TO	EQ NE LT GT LE GE	F/C	LOCATION FROM	TO			PROGRAM IDENTIFICATION
	1 2	3 5	6	7	8	9 12	13 16	17 18	19	20 23	24 27	28 39	40 72	73 80

		FORM TYPE F						FORCED					COMMENTS	
Field	PAGE NO.	LINE NO.		TYPE (N/O/F/D/S*)	(C/Z/D/P/U/V)	LOCATION FROM	TO	RECORD CHARACTER	SUBSTITUTE CHARACTER	CONTINUATION	OVERFLOW FIELD LENGTH (SORTRS ONLY)	RESERVED		PROGRAM IDENTIFICATION
	1 2	3 5	6	7	8	9 12	13 16	17	18	19	20 22	23 39	40 72	73 80
			F											
			F											
			F											
			F											
			F											
			F											

Figure 3–2. SORT3 Specifications Form

Usually one page is sufficient to define all of the specifications needed to describe the sort to the SORT3 program. If necessary, specifications can be continued on subsequent pages. The important thing is to keep the pages arranged in ascending sequence. Therefore, every page containing specifications for your job must be assigned a 2-digit page number. Because the page number is applicable to every specification line appearing on the page, the number is also entered in the Page No. field (columns 1–2) of each specification line. For example, the first page is number 01; all the specification lines on that page start with 01 in columns 1 and 2. Subsequent pages would be numbered 02, 03, and so on.

The specification lines on each page are also numbered in ascending sequence. A 3-digit number specified in columns 3–5 (Line No. field) is used to identify each line. Make certain you define the specifications in the order in which they are to be processed by SORT3. It is suggested that you place a zero in column 5 of the Line No. field or leave this column of the field blank to allow you the capability of inserting additional or out-of-sequence specifications. This method of numbering eliminates renumbering existing specification lines whenever an insertion is required. To illustrate this, suppose you have used three lines to define a record type and have numbered the lines 010, 011, and 012. At this point you realize you omitted a line of the record type specification that should have been defined second in the sequence. Your numbering scheme leaves no room for insertion; therefore, you are forced to renumber all of the existing specification lines. Had your specification lines been numbered 010, 020, and 030, inserting the out-of-sequence specification line could have been accomplished simply by assigning it a number greater in value than 010 and less than 020. As you can see, any value from 1 to 9 assigned to column 5 will properly sequence the inserted line.

Figure 3–3 illustrates sort specification page and line numbering. Specification line and page numbering is important because SORT3 compares the 5-digit number formed by the entries in the Page No. and Line No. fields as the specifications are read. Improperly sequenced sort specifications will terminate the job.

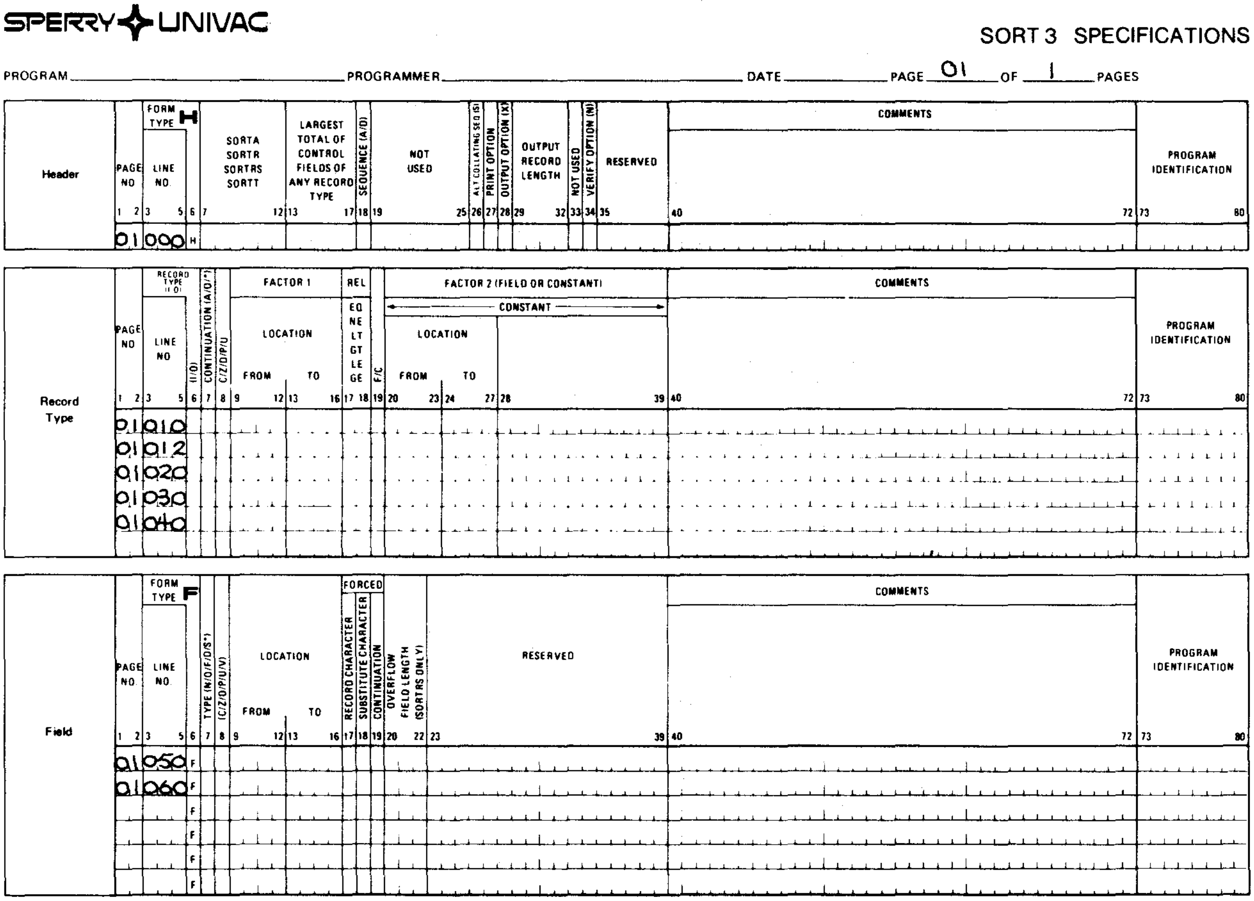
One other thing you must be concerned with when numbering sort specifications is the page and line numbers assigned to the header specification. The header specification must always be the first sort specification processed in the sequence. Therefore, it is always defined on page 01 and is given the line number 000.

3.3.3. Preparing the Sort Specifications

Now that you have determined which sort specifications are required for your job, it is time to define the specifications in a form recognizable to the SORT3 program: 80-column card image format. The sort specifications are discussed in detail in the following paragraphs; however, for quick reference purposes, they are summarized in Appendix B.

3.3.3.1. Header Specification

The first sort specification that you must prepare is the header specification. This specification allows you to identify the type of sort you want performed and to identify the criteria for formatting the output (sorted) file. Figure 3–4 shows the format of the header specification for each type of sort performed by the SORT3 program. The shaded areas identify the fields that you must consider when preparing the specifications.



SPERRY✦UNIVAC

SORT 3 SPECIFICATIONS

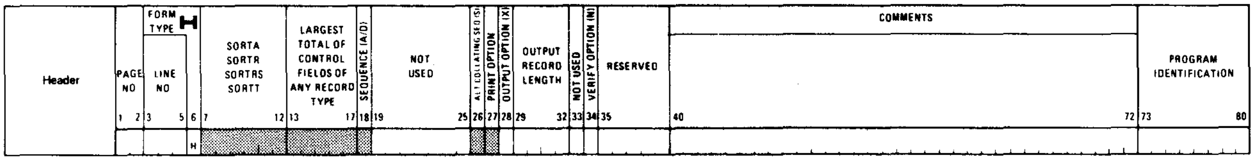
PROGRAM ______ PROGRAMMER ______ DATE ______ PAGE 01 OF 1 PAGES

Header	PAGE NO 1 2	FORM TYPE H LINE NO 3 5	6	SORTA SORTR SORTRS SORTT 7 12	LARGEST TOTAL OF CONTROL FIELDS OF ANY RECORD TYPE 13 17	SEQUENCE (A/D) 18	NOT USED 19 25	ALT COLLATING SEQ (S) 26	PRINT OPTION 27	OUTPUT OPTION (X) 28	OUTPUT RECORD LENGTH 29 32	NOT USED 33	VERIFY OPTION (N) 34	RESERVED 35	COMMENTS 40 72	PROGRAM IDENTIFICATION 73 80
	01	000	H													

Record Type	PAGE NO 1 2	RECORD TYPE (I O) LINE NO 3 5	I(O) 6	CONTINUATION (A/O/*) 7	C/Z/D/P/U 8	FACTOR 1 LOCATION FROM 9 12	 TO 13 16	REL EQ NE LT GT LE GE 17 18	F/C 19	FACTOR 2 (FIELD OR CONSTANT) ← CONSTANT → LOCATION FROM 20 23	 TO 24 27	 28 39	COMMENTS 40 72	PROGRAM IDENTIFICATION 73 80
	01	010												
	01	012												
	01	020												
	01	030												
	01	040												

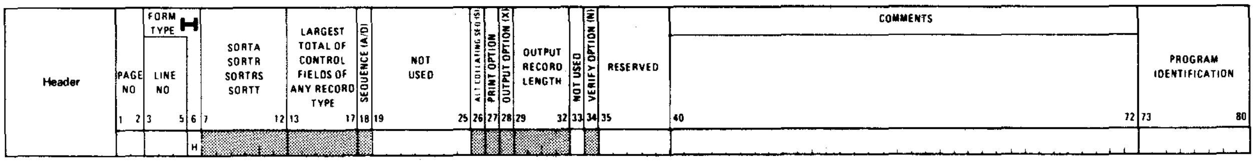
Field	PAGE NO 1 2	FORM TYPE F LINE NO 3 5	6	TYPE (N/O/F/D/S*) 7	C/Z/D/P/U/V 8	LOCATION FROM 9 12	 TO 13 16	FORCED RECORD CHARACTER 17	 SUBSTITUTE CHARACTER 18	CONTINUATION 19	OVERFLOW FIELD LENGTH (SORTRS ONLY) 20 22	RESERVED 23 39	COMMENTS 40 72	PROGRAM IDENTIFICATION 73 80
	01	050	F											
	01	060	F											
			F											
			F											
			F											
			F											

Figure 3-3. Numbering Sort Specifications



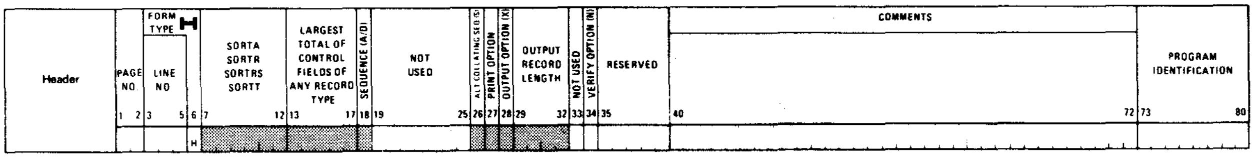
Header	PAGE NO 1 2	FORM TYPE H LINE NO 3 5	6	SORTA SORTR SORTRS SORTT 7 12	LARGEST TOTAL OF CONTROL FIELDS OF ANY RECORD TYPE 13 17	SEQUENCE (A/D) 18	NOT USED 19 25	ALT COLLATING SEQ (S) 26	PRINT OPTION 27	OUTPUT OPTION (X) 28	OUTPUT RECORD LENGTH 29 32	NOT USED 33	VERIFY OPTION (N) 34	RESERVED 35	COMMENTS 40 72	PROGRAM IDENTIFICATION 73 80
			H	▨	▨	▨		▨	▨							

a. Address-out sort (SORTA)



Header	PAGE NO 1 2	FORM TYPE H LINE NO 3 5	6	SORTA SORTR SORTRS SORTT 7 12	LARGEST TOTAL OF CONTROL FIELDS OF ANY RECORD TYPE 13 17	SEQUENCE (A/D) 18	NOT USED 19 25	ALT COLLATING SEQ (S) 26	PRINT OPTION 27	OUTPUT OPTION (X) 28	OUTPUT RECORD LENGTH 29 32	NOT USED 33	VERIFY OPTION (N) 34	RESERVED 35	COMMENTS 40 72	PROGRAM IDENTIFICATION 73 80
			H	▨	▨	▨		▨	▨	▨	▨		▨			

b. Tag-along sort (SORTR)



Header	PAGE NO 1 2	FORM TYPE H LINE NO 3 5	6	SORTA SORTR SORTRS SORTT 7 12	LARGEST TOTAL OF CONTROL FIELDS OF ANY RECORD TYPE 13 17	SEQUENCE (A/D) 18	NOT USED 19 25	ALT COLLATING SEQ (S) 26	PRINT OPTION 27	OUTPUT OPTION (X) 28	OUTPUT RECORD LENGTH 29 32	NOT USED 33	VERIFY OPTION (N) 34	RESERVED 35	COMMENTS 40 72	PROGRAM IDENTIFICATION 73 80
			H	▨	▨	▨		▨	▨	▨	▨					

c. Summary sort (SORTRS)

Figure 3-4. Header Specification Formats

- Defining Sort Type

  Selecting the type of sort is accomplished by entering one of four sort names into the field formed by columns 7 through 12 of the specification. Although four sort names are provided for your selection, only three type sorts are possible: address-out (SORTA), tag-along (SORTR), and tag-along summary (SORTRS). The SORTT entry is provided strictly for System/3 compatibility and, when specified, produces a tag-along sort the same as the SORTR entry. SORT3 will not execute if you omit this field from the specification. The sort type entry must be left-justified in the field.

- Defining Control Fields and Record Sequencing

  In addition to telling SORT3 which sort to perform, you must provide control field and record sequencing information to the program. The control field information (specified in columns 13–17) tells SORT3 how large a buffer it must provide to accommodate the control fields used in sorting your input records. You can specify anything up to 256 bytes. But rather than arbitrarily assigning a value to this field, you can compute the number of bytes needed by totaling the lengths of the control fields for each record type involved in the sort. (Control field types are discussed under column 7 of the field description specification (3.3.3.3). Normal (N) control fields, opposite (O) control fields, and forced (F) control fields are included in this calculation.) The largest total is the value entered in the field. For example, three record types are described for your sort. The total length of the control fields for the first record type is 10 bytes; the total length for the second record type is 12, and the total for the third record type is 8. A buffer size of 12 bytes can accommodate all three, so this is the value you would specify in columns 13–17. The entry must be right-justified in the field.

  Record sequencing refers to how the SORT3 program sequences the records in the sorted output file. You can specify either ascending or descending by entering an A or D, respectively, in column 18.

- Defining Standard/Alternate Collating Sequences

  With the exception of the Output Record Length field (columns 29–32), the remaining fields of the header specification need only be specified if you do not want the program's standard default options. For example, SORT3 uses a standard collating sequence for the sort. If you want to specify an alternate sequence, you must specify the character S in column 26. Of course, you must define characteristics of the alternate collation. This requires you to prepare ALTSEQ statements and place them immediately after the header specification in your job control stream. (ALTSEQ specifications are described in 3.3.4.) When you specify an alternate collating sequence, make certain you do not use a packed (P) or unpacked (U) Factor 1 in the record type specification. Otherwise, the proper records may not be included in or omitted from your output file. The reason is that the ALTSEQ specifications used to define your alternate collating sequence change the Factor 1 fields. This change may affect the unit position and sign of an unpacked decimal number or any one position of a packed decimal number. If it does, the basis of selecting records for the sort is unpredictable.

- Specifying Print Option

  The Print Option field (column 27) is also optional. SORT3 prints (on the system printer) and displays (on the system console) sort specifications, diagnostic messages, program status messages, action messages, and other system messages. This is the default case. You can limit or inhibit this service by specifying a 1, 2, or 3 in column 27. A 1 causes only program status messages, action messages, and other system messages to be printed and displayed. A 2 causes only action messages and other system messages to be printed and displayed. A 3 causes only other system messages to be printed and displayed. Note that if a job is initiated from a workstation, messages will be displayed on the workstation rather than on the system console.

- Dropping Control Fields

  If you requested one of the tag-along sorts for your job (SORTR, SORTRS, or SORTT), you can have SORT3 drop control fields from the sorted output records by entering an X in column 28. Control fields are normally dropped when opposite fields or an alternating collating sequence is specified. In both cases, SORT3 changes the control information during the sort so it is meaningless as data. If, under these circumstances, you want to retain the control information in the output record and keep it in a meaningful form, you must define the control field twice: once as control fields and once as data fields.

- Stating Output Record Length

  Previously, it was stated that the Output Record Length field (columns 29-32) was a required field. This is true whenever one of the tag-along sorts is to be performed. The entry in this field depends on whether control fields are dropped from the sorted output records (X in column 28). If the control fields are dropped, the length specified includes only data fields. The calculation is similar to that for columns 13-17; total the length of the data fields in each record type in the sort. Enter the largest value right-justified in the field. If control fields are retained in the output records, total the length of the data fields for each record type in the sort and add the largest value to the value entered in columns 13-17. Enter the sum right-justified into columns 29-32. Under both conditions, the value entered must not exceed 4096 bytes.

- Specifying Data Verification

  The Verify Option field (column 34) can be used to improve program performance (throughput) by requesting SORT3 not to verify the data written on the work files during the sort. If this field is blank, data verification is performed automatically. To inhibit this feature, enter an N in the field.

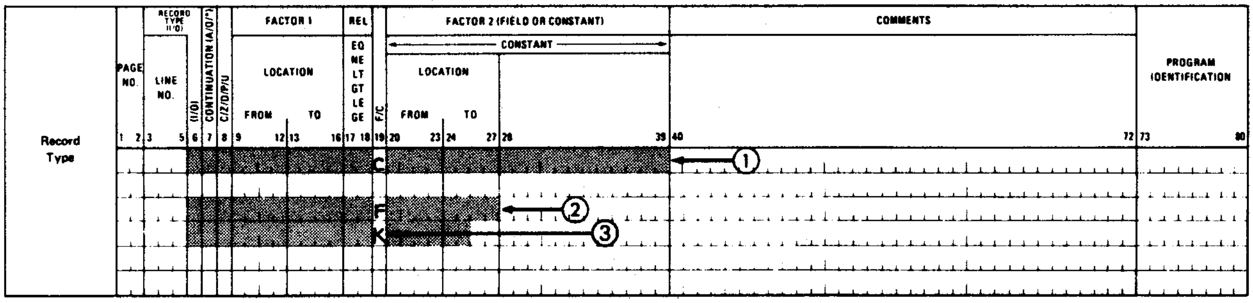
- Coding Comments

  Columns 40 through 80 have no effect on program function. They are provided for your comments and program identification. They can be printed out whenever the Print Option field (column 27) is specified as 0 or left blank.

A summary of the field entries for the header specification is provided in Appendix B.

3.3.3.2. Record Type Specification

The next sort specification you must prepare for your job is the record type specification. Record type specifications are used for defining the types of input records SORT3 is to include in or omit from the sort. Of course, there is no need to prepare record type specifications if SORT3 does not have to be selective in sorting the records of your input file. For example, if every record in your input files is to be sorted and the format of each record is the same, SORT3 does not have to decide which records to include or omit. You can, therefore, omit the record type specification from your job control stream. To SORT3, the omitted specifications imply an include-all record condition for your sort. On the other hand, a sort that includes or excludes specific record types requires you to identify these record types to the SORT3 program. In this case, you must include a record type specification for each record type to be sorted. Figure 3–5 shows the format of the record type specification and the field entries you must consider for each type of sort performed by SORT3.



LEGEND:

① Format when comparison involves input record field to a constant.

② Format when comparison involves two input record fields.

③ Format when comparision involves input record field to a keyword.

Figure 3–5. Record Type Specification Format

Before getting into the specifics of preparing the specification, let's briefly discuss how SORT3 identifies records. Records are selected or omitted on the basis of a test or comparison. That is, SORT3 looks at a particular key field or fields (control and/or data) in each record of your input file and compares the data in that field to a constant, keyword, or the data in another field of the same record. (The data you are comparing is the Factor 1 field, and the data you are comparing it against is the Factor 2 field.) The results of the comparison determine whether the record is selected or omitted from the sort.

What role do you play in this procedure for sorting records? You establish the criteria upon which SORT3 makes its decisions. For example, the information coded in your record type specifications:

- defines the length and location of the Factor 1 and Factor 2 fields used in the comparison;
- provides the constant if it is used;
- defines how the data contained in the factor fields is to be interpreted during the comparison;
- defines what the results of the comparison must be; and
- decides whether the record type based on the results of this test is to be included in or omitted from the sort.

The record type specification entries are described in the following paragraphs.

- Selecting Records for the Sort

  The decision to include or omit the record type defined in the specification is based on your entry in column 6 (Record Type field). An I in this field tells SORT3 to include in the sort only those records that meet the comparison requirements set forth in the specification. An O in column 6 instructs SORT3 to omit those input records that meet the comparison requirements defined in the specification.

  Why have an include-and-omit capability? Consider a file that contains many different types of records and you want to include only a few types in the sort. You automatically exclude all records not wanted by defining only those few types you do want sorted. Now, consider a time when you want to include all but a few record types in the sort. Rather than providing a description for each record included in the sort, you can simply describe the few records you want omitted. The remaining record types are automatically included in the sort. From this example, you can see the advantages of having both options available.

  In normal practice, an omit record description is always followed by a special version on the include record description referred to as an include-all-record description. The include-all description is defined by entering the character I in column 6 and leaving blank the fields (columns 7–39) of the specification related to describing the record type. This tells SORT3 to include all record types in the sort not previously defined to be omitted or included. If you use the include-all version, only one can be specified per job and it must be the last record type defined for that job.

- Defining the Relationship between Lines of Coding

  Because a record type description can extend beyond one line of code on the form, you must define the relationship of one line to another. Column 7 is used to define this relationship. A blank field tells SORT3 that this is the first line of code for the record description. An A entry states that this line of code is a continuation of the preceding line. (The A represents an AND function.) An O entry defines that the line of code applies to a different record type than the one described on the preceding line, but the field descriptions are common to both record types. (The O represents an OR function.) Comment lines are defined by an asterisk (*) in column 7. Comment lines have no effect on sort other than being printed if you have specified the Print option on the header specification.

- Defining Factor 1/Factor 2 Compare Operations

  How should SORT3 interpret the data in the Factor 1 and Factor 2 fields during compare operations? When these fields contain alphanumeric data, an entry of C, Z, or D must be entered in column 8 of the specification. The specific entry made depends upon what portion of the data you want included in the comparison. That is, alphanumeric data comprises two portions: a zone portion and a digit portion. The Z entry instructs SORT3 to use only the zone portion of the data. The D entry specifies only the digit portion and the C entry instructs SORT3 to use both portions. If the Factor 1 and Factor 2 fields contain numeric data, a P or U character must be specified in column 8. The P entry indicates the data is packed and the U entry indicates the data is unpacked. Packed numeric data always contains a sign (positive or negative) and only the digit portion of the data. Unpacked numeric data also contains a sign but includes both the zone and the digit portions of the data. As you can see, the data type and the method of comparison have some influence on the length of the Factor 1 and Factor 2 fields. Table 3-2 lists the available entries for column 8 and the restrictions each places on the length of the Factor 1 and Factor 2 fields.

Table 3—2. Column 8 Entries and Their Effect on Factor 1 and 2 Field Lengths

Column Entry	Compare Operation Method	Maximum Allowable Length for Factor 1 and 2 Fields
C	Use both zone and digit portions of the bytes	256 bytes
Z	Use only the zone portion of the byte	1 byte
D	Use only the digit portion of the bytes	16 bytes
P	Packed numeric data	8 bytes or 15 digits and a sign
U	Unpacked numeric data	16 digits

The remaining fields of the specifications pertain to defining the Factor 1 and Factor 2 fields for the sort, defining the conditions for field comparisons, etc.

First, let's discuss setting up the test conditions for the comparison. You have six test conditions available to use in the comparison between the Factor 1 and Factor 2 fields. Each condition instructs SORT3 to look for a specific test result from the comparison of the two fields. The record is selected or rejected based on the results of the comparison. Columns 17 and 18 are used for defining the results of the comparison. The entries for this field and the restrictions in their use (where applicable) are given in Table 3-3.

Table 3—3. Test Relationships for Factor 1 and 2 Comparisons

Column 17-18 Entry	Test Conducted
EQ	Factor 1 field equal to Factor 2 field
NE	Factor 1 field not equal to Factor 2 field
LT	Factor 1 field less than Factor 2 field*
GT	Factor 1 field greater than Factor 2 field*
LE	Factor 1 field less than Factor 2 field*
GE	Factor 1 field greater than or equal to Factor 2 field*

*These entries are not permitted when the comparison made involves only the zone portions of the data (Z specified in column 8).

The time has arrived to discuss the Factor fields. First, let's approach the Factor 1 field (columns 9–16). SORT3 does not interpret the entry in this field as actual data, but as the location of the data within your input records. As you can see, the Factor 1 field is composed of two parts. The first part (columns 9–12) defines the position at which the data begins in the record. The second part (columns 13–16) defines where the data ends. The number of positions from one point to the other also represents the length of data. Technically, the length of the data defined in the Factor 1 field can be any number of bytes from 1 to 256. In practice, however, this length cannot exceed the length of the records in the file. In addition, the length specified in the Factor 1 field is restricted by how the data is interpreted (column 8 entry) and whenever the Factor 2 field defines a constant or keyword. The allowable field lengths and the restrictions other field specifications place on this length are defined in Table 3-4.

Table 3—4. Factor 1 Field Length Requirements

Column 8 Entry	Maximum Factor 1 Field Length (in bytes)	
C	256	When Factor 2 defines a constant, the length defined must not exceed 20. When Factor 2 defines a keyword, the length must not exceed 6.
Z	1	
D	16	
P*	8	Because the field is packed, it can actually represent 15 decimal digits and a sign.
U*	16	

*Do not use a packed or unpacked Factor 1 field if an alternate collating sequence is specified in the header specification (S in column 26).

Two rules to keep in mind when coding the Factor 1 field are:

1. All entries must be right-justified (the From Location must end in column 12 and the To Location must end in column 16).

2. You need not enter anything in the From Location when a Factor 1 field length of one byte is defined.

In the brief description of how SORT3 compares the Factor fields, it was stated that the data defined by the Factor 1 field was compared against a constant, keyword, or the data in another field of the record. This constant, keyword, or field location is identified in the Factor 2 field (columns 20–39). Because the Factor 2 field can specify this type of information, you must tell the SORT3 program how to interpret the entry in the Factor 2 field. If a C is entered in column 19, the Factor 2 field contains a constant or keyword. If it contains a field position, enter an F in the column.

When the Factor 2 entry defines a field, only columns 20–27 are used. The length of the field defined must be the same as that specified for Factor 1. It must also be in the same record type as Factor 1. The purpose of the From Location and the To Location and the rules for coding are the same as for the Factor 1 field.

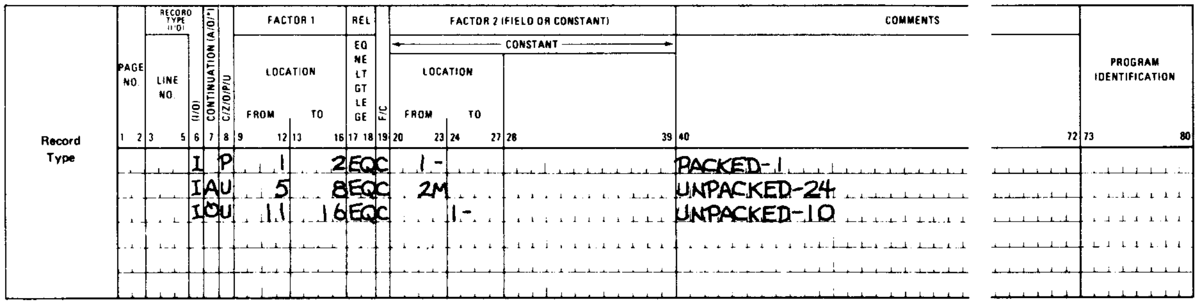
All of the columns (20–39) are used when Factor 2 defines a constant. However, the rules for coding your entry depend on whether the constant represents a packed or unpacked number, an alphanumeric constant, a numeric constant, or a signed constant. In general, the constant must be the same length as the Factor 1 field.

For example, if Factor 1 is a 4-position field, the constant field must take up four positions. If the constant is the number 6, enter the 6 in column 23, and either leave columns 20–22 blank or fill them with zeros.

If the Factor 1 field contains a packed number, the length of the constant (including the sign) must be twice the length of the Factor 1 field. The reason is that Factor 1 data is in packed form, and the constant is in unpacked form. When alphanumeric constants are specified (column 8 entry is C, Z, or D), the constant must be the same length as the factor 1 field and must always begin in column 20. When numeric constants are specified (column 8 entry is P, U, or D), they must be right-justified within the field length defined in Factor 1 (within twice the field length if Factor 1 is a packed number). For example, assume that Factor 1 defines a 6-position field in the input record and that Factor 2 is the numeric constant 456. To right-justify the constant within six positions, put the constant in columns 23-25. Because leading zeros are not required (to SORT3, blanks and zeros look the same), columns 20-25 could contain either 000456 or 456 with three leading blanks.

For signed constants, the last character in the constant must be its sign (+ or -) when Factor 1 is a packed number. If Factor 1 is an unpacked number, and the constant is a negative number, the last digit in the constant must be a character that indicates both the numeric value of the last digit and the negative sign for the entire constant.

The following example shows the entries you make for records that have a packed -1 in positions 1 and 2 of the record, an unpacked -24 in positions 5 through 8 of the record, and an unpacked -10 in positions 11 through 16 of the record.

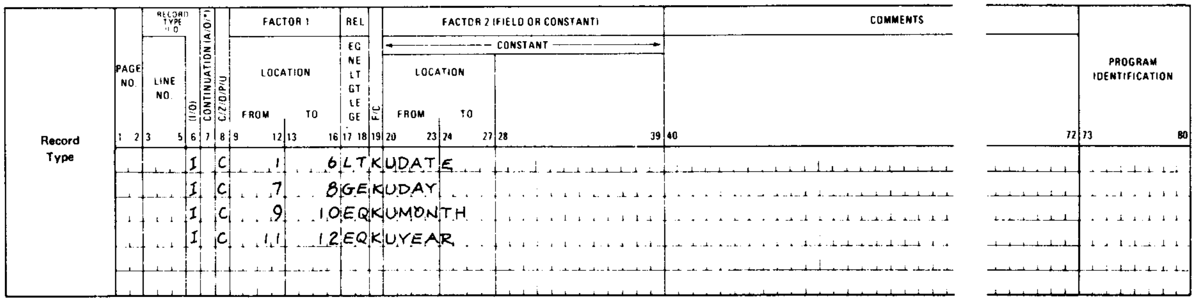


	PAGE NO. 1–2	RECORD TYPE (I/O) LINE NO. 3–5	(I/O) 6	CONTINUATION (A/O/*) 7	C/Z/O/P/U 8	FACTOR 1 LOCATION FROM 9–12	FACTOR 1 LOCATION TO 13–16	REL EQ NE LT GT LE GE 17–18	F/C 19	FACTOR 2 (FIELD OR CONSTANT) CONSTANT LOCATION FROM 20–23	LOCATION TO 24–27	28–39	COMMENTS 40–72	PROGRAM IDENTIFICATION 73–80
Record Type			I		P	1	2	EQ	C	1-			PACKED-1	
			I	A	U	5	8	EQ	C	2M			UNPACKED-24	
			I	O	U	11	16	EQ	C		1-		UNPACKED-10	

As in the header specification, comments can be written in columns 40 through 80. They have no effect on program functions, and are printed only if the Print Option is specified in the header specification.

When the Factor 2 entry defines a keyword, the column 8 entry must be C, and the column 19 entry must be K. The permissible entries for a keyword are UDATE, UDAY, UMONTH, and UYEAR. The Factor 1 field length must be 6 if UDATE is specified, and the format of the date in the record field must be the same as UDATE. If UDAY, UMONTH, or UYEAR is specified, the Factor 1 field length must be 2.

The following example shows the entries you make when you are comparing fields with keywords.

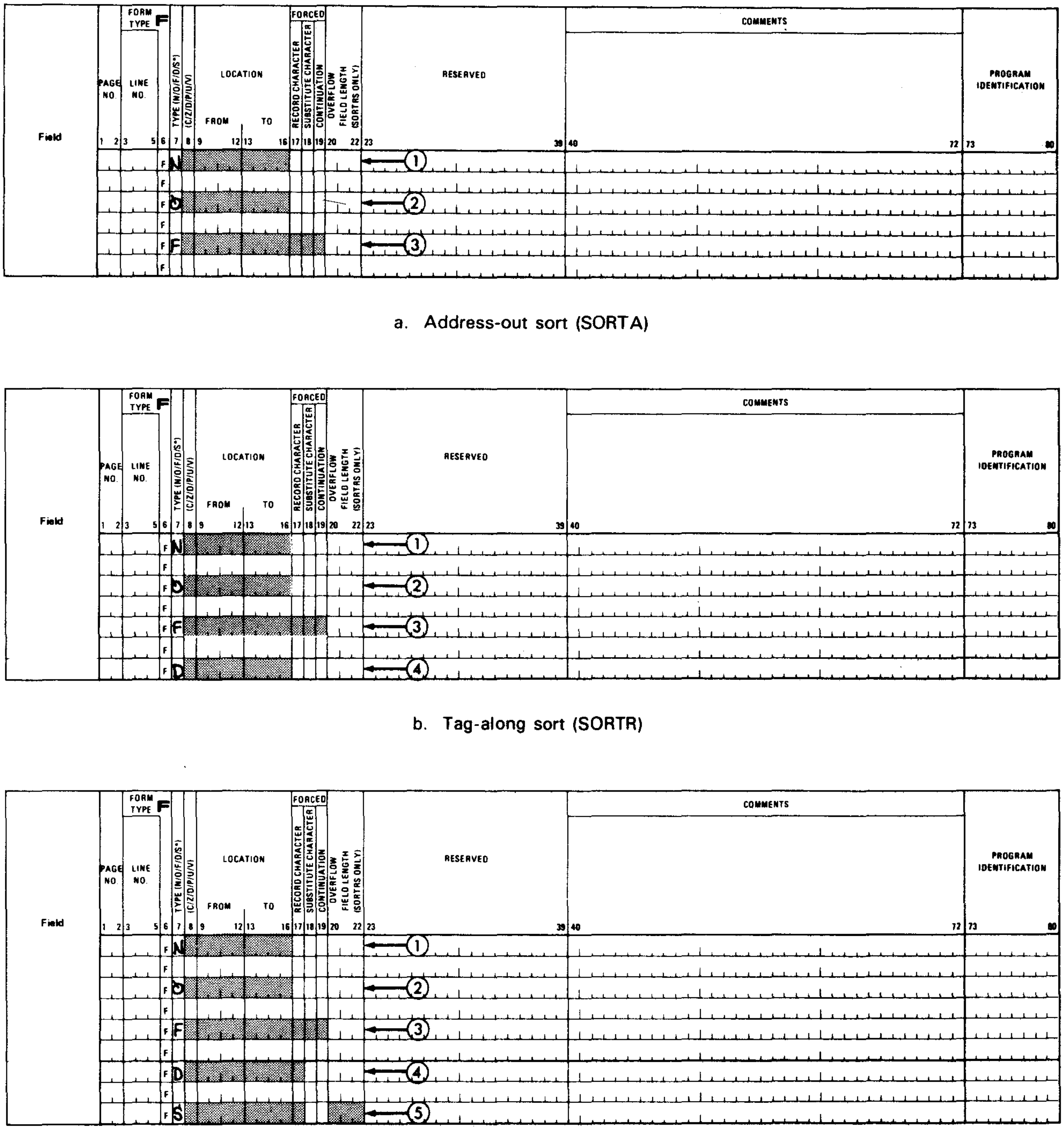


Record Type	PAGE NO.	LINE NO.	RECORD TYPE I-O (I/O)	CONTINUATION (A/O/*)	C/Z/D/P/U	FACTOR 1 LOCATION FROM	FACTOR 1 LOCATION TO	REL EQ NE LT GT LE GE	F/C	FACTOR 2 (FIELD OR CONSTANT) CONSTANT LOCATION FROM	TO		COMMENTS		PROGRAM IDENTIFICATION
	1 2	3 5	6	7	8	9 12	13 16	17 18	19	20 23	24 27	28 39	40	72	73 80
			I		C	1	6	LT	K	UDAT	E				
			I		C	7	8	GE	K	UDAY					
			I		C	9	10	EQ	K	UMON	TH				
			I		C	11	12	EQ	K	UYEA	R				

A summary of the field entries for the record type specification is provided in Appendix B.

3.3.3.3. Field Description Specification

The last sort specification to be prepared is the field description specification. This specification instructs SORT3 how to format the records in the output file. For address-out sorts (SORTA), the field parameters in each line describe the control fields used to sort record addresses. For tag-along sorts (SORTR and SORTT) and summary sorts (SORTRS), the field parameters in each line define the fields SORT3 uses to create the output records. Figure 3–6 shows the format of the field description specification and the fields that must be considered for each type of sort performed by SORT3.



a. Address-out sort (SORTA)

b. Tag-along sort (SORTR)

c. Summary sort (SORTRS)

LEGEND:

① Defining normal control fields

② Defining opposite control fields

③ Defining forced control fields

④ Defining data control fields

⑤ Defining summary data fields

Figure 3–6. Field Description Specification Formats

When records are written to a disk file, SORT3 normally blocks the file by a factor of 8 unless this block size exceeds 1024 bytes; in which case, the block size is made as close as possible to 1024 without exceeding this value. If any record size exceeds 512 bytes, then the output file is unblocked. The output block size of a non-IRAM/MIRAM file can be selected by using the // DD control statement. Records written to diskette files are always unblocked, and the block size of records written to tape files is specified by tape.

In reviewing the field description specification format, note that columns 7–16 must be specified for all sort jobs. Columns 17–19 are applicable only when forced control fields are used, and columns 20–22 apply only to summary sorts.

To make certain SORT3 properly interprets the contents of the field description specification, you must define whether a control field, data field, or comment is being described in each specification line. Defining the field type is a function of the entry in column 7 of the specification. Data field descriptions are indicated by a D entered in the column, comments are indicated by an asterisk (*) in the column, and summary data is indicated by an S in the column. Control field descriptions are indicated by column entries of N, O, or F, depending on the type of control field described (normal, opposite, or forced).

- Data Field Descriptions (D)

  Data field descriptions are applicable only to the tag-along sorts (SORTR, SORTRS) performed by SORT3. They define the fields you want SORT3 to include in your sorted output records. They are not used for defining the fields used in sorting the records. When your input file contains more than one type of record, it is not necessary that the total length of data fields defined be the same for all record types, or that all record types contain the same number of data fields.

  The SORT3 program blank-fills short data fields to maintain a uniform length for all data fields. It is necessary that, within each set of included record type and field description lines, the data field description lines follow the control field description lines in your control stream. (SORT3, when building work records, requires control records to appear ahead of data records.)

  Do not include data field descriptions in your control stream for address-out sorts (SORTA) because SORT3 will process the description line as a comment.

- Comments (*)

  Comment lines serve no function other than helping you document the sort. When properly identified (* in column 7), they can be coded anywhere in the specification. However, it is preferred that they be coded in columns 40–80 to avoid confusion. Comments are printed only when the Print Option (0 or blank in column 27) is specified in the header specification.

- Summary Data Field Descriptions (S)

Summary data field descriptions (S in column 7) can be defined for all three sort jobs performed by SORT3. However, the data fields are summarized (added together) only in the summary tag-along sort (SORTRS). For tag-along sorts (SORTR and SORTT), summary data fields are processed as normal data fields. For address-out sorts (SORTA), the summary data fields are processed as comments.

It is important, when performing a summary sort, that the summary data fields in the work and output records of the individual record types be located in the same position on each record. This is not a consideration for the input records. No more than 24 data fields can be summarized for each record type included in the sort.

The format for summary fields is defined by the first summary field description specification processed for an included record type. It is suggested that a summary specification be specified for each record type included in the sort. The advantage is that SORT3 issues a warning whenever summary fields are not aligned; you can then make the changes necessary. If summary field specifications are not provided, the data field specifications should align the data for summarization. If summary specifications are not provided when a summary tag-along sort (SORTRS) is specified, the output of the job produced a file consisting of records with unlike control fields. This is due to the fact that SORT3 eliminates all but one copy of each record having common control fields.

Another consideration when defining summary data fields is the possibility of overflow. To allow for the possibility of an anticipated overflow condition, you should complete the overflow field length entry in columns 20-22. These columns are used only by a summary tag sort to eliminate the possibility of an overflow condition. (An entry in the overflow field length columns is ignored for forced fields because they are only one byte in length.) The entry made in columns 20-22 effectively increases the length of the summary data field and should reflect the sum of the summary data field length and the anticipated overflow length. Entries in columns 20-22 must also be right-justified and must not exceed the maximum field length determined by your entry in column 8.

To illustrate the coding for columns 20-22, assume you want to summarize an unpacked field in positions 8-11 of the input record. You know that the output will exceed the 4-position summary field by one position. To allow for the expected overflow, specify an entry of 5 in column 22.

If packed fields are summarized, columns 20-22 should specify the number of bytes of packed data. For example, to summarize a packed field in positions 3-6 of the input record, knowing that the output will exceed the 4-position packed summary field (seven numbers plus sign) by 1 position, specify a 5 in column 22 (nine numbers plus sign).

- Control Field Descriptions (N, O, F)

  Normal control field descriptions (N in column 7) are used to sort records in the normal sequence as specified by the sequence field (column 18) in the header specification for the job.

  Opposite control field descriptions (O in column 7) are also used to sort records. However, they instruct SORT3 to sequence the records opposite to that specified by the sequence field in the header specification.

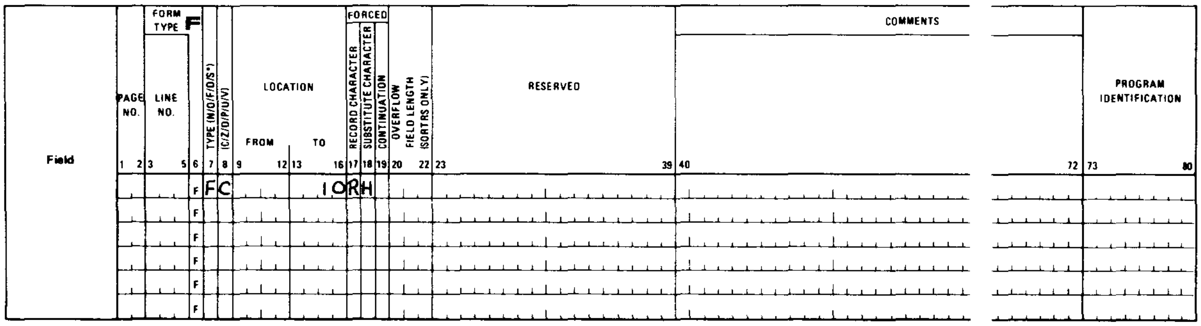
  By defining normal and opposite control fields for your job, records can be sorted so some control fields are in ascending order and other control fields are in descending order.

  Force control field descriptions (F in column 7) are used to modify the contents of control fields in the work records constructed for the sort. Forced control field descriptions affect work records and output records, but not your input records.

When a line of the specification is identified as a forced control field line (F in column 7), SORT3 looks at columns 9-16 to identify the position of the control field in the input record, then it checks columns 17-19 to see how the control field is affected. That is, the entries in columns 17-19 tell SORT3 how to modify the control field when it is placed into the work record. The column 17 entry specifies which character in the control field (identified in columns 13-16) SORT3 is to replace. The entry in column 18 gives the replacement character for the field or, when column 17 is not used, adds a new character to the control field identified. The column 19 entry shows a continuation in the force field description (relates 'the specification to the preceding specification line). To review the various conditions under which you may use forced control fields, three examples are provided as follows:

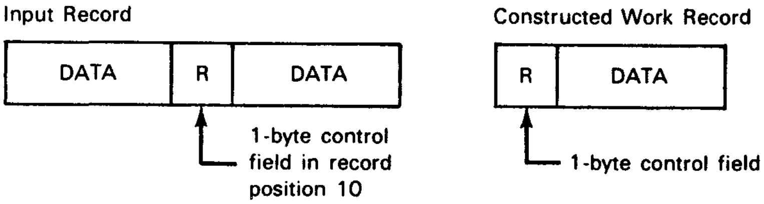
Example 1:

Example 1 illustrates a conditional force. Assume that each record in the file to be sorted has a 1-byte control field in record position 10. If the byte contains the character R, SORT3 replaces it with the character H before sorting the records. The field description specification is coded as follows:

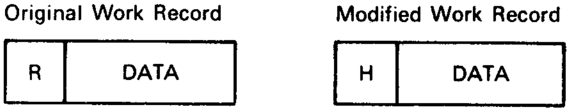


		FORM TYPE F						FORCED					COMMENTS	PROGRAM IDENTIFICATION
	PAGE NO.	LINE NO.		TYPE (N/O/F/D/S*)	(C/Z/D/P/U/V)	LOCATION FROM	LOCATION TO	RECORD CHARACTER	SUBSTITUTE CHARACTER	CONTINUATION	OVERFLOW FIELD LENGTH (SORTRS ONLY)	RESERVED		
Field	1 2	3 5	6	7	8	9 12	13 16	17	18	19	20 22	23 39	40 72	73 80
			F	F	C		10	R	H					
			F											
			F											
			F											
			F											
			F											

The SORT3 program constructs a work record for the selected input record.



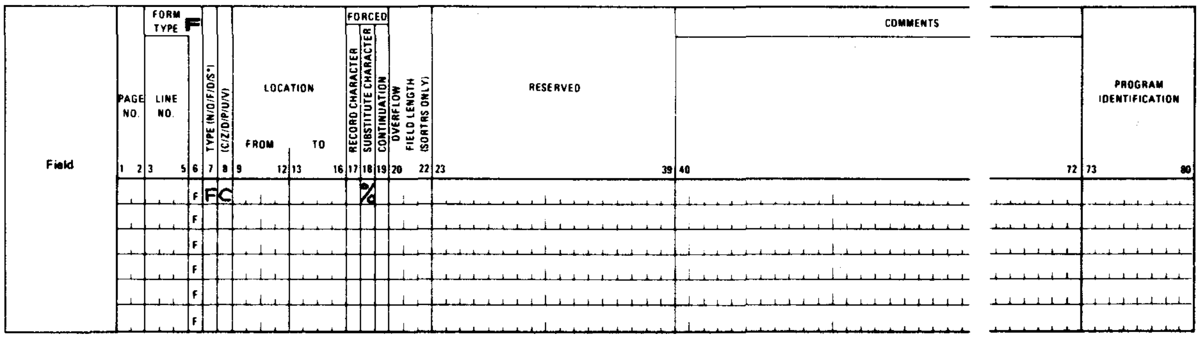
The content of the control field is checked for the character R. If it contains an R, the control field content is changed to H in the work record.



Example 2:

Example 2 illustrates an unconditional force. An unconditional force allows you to add (force) a new character into your output records. This is done without basing the force on a specific control field of the input record as you would for a conditional force. Therefore, you do not code columns 9-16 of the specification. However, you must define the character that is being forced. You define this character in column 18.

The position that the forced character occupies in the output record is determined by the sequence in which it appears in your control stream. That is, if it is the first control field specification defined in your control stream, it will be the first control field in the output record. This is shown in the following coding:



Field	PAGE NO. 1 2	LINE NO. 3 5	FORM TYPE F 6	TYPE (N/O/F/D/S*) 7	(C/Z/D/P/U/V) 8	LOCATION FROM 9 12	LOCATION TO 13 16	FORCED RECORD CHARACTER 17	FORCED SUBSTITUTE CHARACTER 18	CONTINUATION 19	OVERFLOW FIELD LENGTH (SORTRS ONLY) 20 22	RESERVED 23 39	COMMENTS 40 72	PROGRAM IDENTIFICATION 73 80
			F	F	C				%					
			F											
			F											
			F											
			F											
			F											

When processed, this specification instructs SORT3 to place a percent sign in the first control field position of the work and output records. All other control fields are positioned after the percent sign. For example, if the input record format appears as:

DATA	CONTROL FIELD A	DATA	CONTROL FIELD B	DATA

the work record constructed from the field specification appears as follows:

%	CONTROL FIELD A	CONTROL FIELD B	DATA

If the forced control field is defined after the field descriptions for A and B, then the percent sign (%) occupies the third control field position (first undefined control field available) in the record. The work record constructed from the specification would appear as:

CONTROL FIELD A	CONTROL FIELD B	%	DATA

Example 3:

Example 3 illustrates a force-all condition. Force-all is a special form of conditional force (Example 1) that can occur only when a control field in the input record does not contain a particular field entry. If, for example, a specific control field in the input record does not contain a particular character, you can direct SORT3 to change the contents of the control field in the work record. Force-all specifications usually follow a series of conditional force specifications. In the specifications shown, SORT3 checks the 1-byte control field in position 1 to see whether it contains the characters A, B, or #. If it does, A is replaced with the character 2, B is replaced with a 4, and # with a $. If the control field does not contain an A, B, or # entry, SORT3 places + in the control field.

Field	PAGE NO. 1–2	LINE NO. 3–5	FORM TYPE F 6	TYPE (N/O/F/D/S*) 7	(C/Z/D/P/U/V) 8	LOCATION FROM 9–12	LOCATION TO 13–16	FORCED RECORD CHARACTER 17	FORCED SUBSTITUTE CHARACTER 18	FORCED CONTINUATION 19	OVERFLOW FIELD LENGTH (SORTRS ONLY) 20–22	RESERVED 23–39	COMMENTS 40–72	PROGRAM IDENTIFICATION 73–80
			F	F	C		1	A	2					
			F	F	C		1	B	4	X				
			F	F	C		1	#	$	X				
			F	F	C				*	X				
			F											
			F											

Note the use of the X in column 19. This shows that each specification is a continuation of the preceding line.

SORT3 replaces the specified control fields with hexadecimal FF or 00 if (depending on whether ascending or descending sequence is specified in the header specification):

- the force-all specification did not follow the conditional force specifications; and
- SORT3 cannot locate any of the characters specified in the conditional force specification.

The following list summarizes the rules for coding the field description specification for forced control fields:

- Defining a Conditional Force Character
  - Fill in columns 1-6 as for any control field.
  - Put an F in column 7.
  - Define the position of the control field in the input record in columns 13-16.
  - Enter the character to be replaced in column 17.
  - Enter the character it is to be replaced with in column 18.
- Defining a Force-All Character
  - Fill in columns 1-6 as for any control field.
  - Put an F in column 7.
  - Enter the character that replaces the control field in column 18.
  - Put any character in column 19. (The character in column 19 tells SORT3 that the line is a continuation of the preceding line.)
  - Leave columns 9-17 blank.

  NOTE:

  If a force-all line is not placed after conditional force and SORT3 does not find the specified characters in the control field of the input record, then SORT3:

  1. *replaces the control field character with hexadecimal FF (if ascending sequence is specified in the header); or*
  2. *replaces the control field character with hexadecimal 00 (if descending sequence is specified in the header).*

■ Defining an Unconditional Force Character

- Fill in columns 1-6 as for any control field.
- Put an F in column 7.
- Put the character to be forced in column 18.
- Leave columns 9-17 blank.

After defining the type of field being described, you must indicate to SORT3 what portion of the input record it must use to build and sort work records. This definition is the function of the entry made in column 8. If the data to be used is alphanumeric, you can enter the characters C, Z, or D in column 8. An entry of C tells SORT3 to use both zone and digit portions of the data bytes in the fields defined. If you want only SORT3 to use the zoned portion of each byte, enter a Z in column 8. The D entry limits SORT3 to use of the digit portion of each byte. When numeric data is used for building and sorting work files, you must define to SORT3 whether the data being used are signed packed decimal numbers or signed unpacked decimal numbers. This is accomplished by entering a P or U, respectively, in column 8.

In a situation where SORT3 is to force characters into a data field of the work record, enter a V in column 8 and define the character to be forced in column 18.

To illustrate how the column 8 entry functions, assume that a 1-byte control field in the input record can contain any one of the following characters:

Character	Zone	Digit
$	0101	1011
A	1100	0001
B	1100	0010
C	1100	0011

If you wanted the records sorted into ascending order using the digit portion of the control field characters, put a D in column 8. The characters will appear in the following order in the output record:

A

B

C

$

If the records are to be sorted into ascending order using both the zone and digit portions, enter a C in column 8. The order of the characters in the output records will be:

$

A

B

C

If you had a Z entered in column 8 and specified ascending sequence in the header specification, the records with a $ control field precede records with an A, B, or C control field. Because A, B, and C have identical zone portions, records with any of these characters as a control field will not be any special order after the sort.

If you want to sort records so that some control fields are in ascending order and other control fields are in descending order, opposite control fields should be used. An opposite field is sorted in ascending order if descending order is specified on the header specification, or in descending order if ascending order is specified on the header specification. If the file contains different record types, all of which have an opposite control field in the same record position, the column 8 entries for all these control fields must be D, C, Z, or any combination of C and Z. With any other combination, the results of the sort will be unpredictable.

NOTE:

When using opposite control fields, SORT3 changes them into meaningless control field information when building the work record. Therefore, information is usually dropped by coding an X in column 28 of the header specification for tag-along or summary sorts. To retain the original control field data in the output record, repeat the description for the information as a data field. The same holds true when using packed or unpacked control fields.

If you specified packed or unpacked control fields (normal or opposite), SORT3 changes the control fields while building the work record. Therefore, the control field information must be dropped by coding an X in column 28 of the header specification. To retain the original control field data in the output record, redefine the information as a data field.

When using control fields to sequence information in the sorted records, the following rules must be followed:

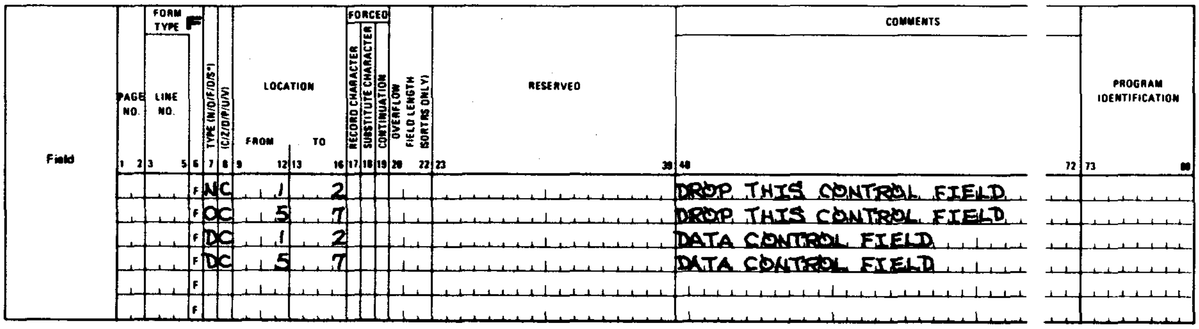
- Only one character is allowed in a forced control field.
- Either a conditional or an unconditional force can be indicated.
- A force-all must be preceded by a conditional force.
- A forced control field can be defined by placing an F in column 7 of the field specifications.

The order in which control fields are described in the field specification lines determines the sequence of the records (tag-along sort) or the record addresses (addrout sort) in the sorted file.

Suppose a file is to be sorted in ascending order (A in column 18 of the Header line) and each record in that file has a normal control field in positions 1-2 and an opposite control field in positions 5-7. Each record represents one customer's order for a separate item. The part number is in position 1-2; the number of parts ordered is in positions 5-7. The unsorted file appears as:

Input Record Name	Input Record Position						
	1	2	3	4	5	6	7
0	2	0			3	0	0
1	4	0			2	5	0
2	3	0			6	0	0
3	1	4			4	5	0
4	6	0				7	0
5	3	0				5	9
6	6	0			1	3	0
	Part Number				Number Ordered		

The first control field can be used to sort the records in ascending order according to the part number. The second control field is then used to sort the number of parts ordered in descending order within each group of parts. The field specification would be coded as follows:



Field	PAGE NO. 1 2	LINE NO. 3 5	FORM TYPE F 6	TYPE (N/O/F/D/S*) 7	(C/Z/D/P/U/V) 8	LOCATION FROM 9 12	LOCATION TO 13 16	FORCED RECORD CHARACTER 17	FORCED SUBSTITUTE CHARACTER 18	CONTINUATION 19	OVERFLOW FIELD LENGTH (SORTRS ONLY) 20 22	RESERVED 23 39	COMMENTS 40 72	PROGRAM IDENTIFICATION 73 80
			F	N	C	1	2						DROP THIS CONTROL FIELD	
			F	O	C	5	7						DROP THIS CONTROL FIELD	
			F	D	C	1	2						DATA CONTROL FIELD	
			F	D	C	5	7						DATA CONTROL FIELD	
			F											
			F											

The sorted output file is formatted as follows:

Output Record Number	Output Record Position				
	1	2	3	4	5
0	1	4	4	5	0
1	2	0	3	0	0
2	3	0	6	0	0
3	3	0		5	9
4	4	0	2	5	0
5	6	0	1	3	0
6	6	0		7	0

Part Number (positions 1–2) Number Ordered (positions 3–5)

After completing the column 7 and 8 entries, you must identify where the record field being used starts and ends in the input record. The position at which the field begins in the record is entered (right-justified) in columns 9–12 (From columns). The position at which the field ends in the record is entered (right-justified) in columns 13–16 (To columns). The order in which the fields are described in the specification determines the order they appear in the sort output records. For example, suppose you have an input record that looks as follows:

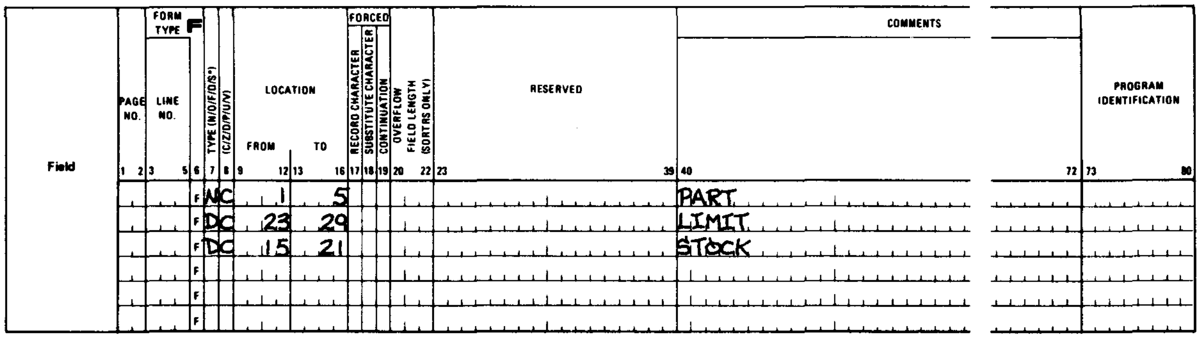
Record Field Positions

PART (part number)		COST (price per item)		STOCK (balance in stock)		LIMIT (reorder limit)
1 5	6	7 12	13 14	15 21	22	23 29

But you want your output (sorted) record to look like:

PART	LIMIT	STOCK
1 5	6 12	13 19

To format the output record, columns 9-16 would have to be coded as follows:



Field	PAGE NO.	LINE NO.	FORM TYPE F	TYPE (N/O/F/D/S*)	(C/Z/D/P/U/V)	LOCATION FROM	LOCATION TO	FORCED RECORD CHARACTER	FORCED SUBSTITUTE CHARACTER	CONTINUATION	OVERFLOW FIELD LENGTH (SORT RS ONLY)	RESERVED	COMMENTS	PROGRAM IDENTIFICATION
	1 2	3 5	6	7	8	9 12	13 16	17	18	19	20 22	23 39	40 72	73 80
			F	N	C	1	5						PART	
			F	D	C	23	29						LIMIT	
			F	D	C	15	21						STOCK	
			F											
			F											
			F											

The limitations to the maximum length of the field described in columns 9-16 are determined by the entry in column 8.

Column 8 Entry	Maximum Allowable Field Length (bytes)
C	256
Z	1
D	16
P	8
U	16
V	1

When fields one byte in length are being described, leave columns 9-12 (From) blank and enter the record position of that byte right-justified in columns 13-16.

Comments pertaining to field description specification are coded in columns 40-80 and have no effect on the sort function. Comments are printed only if specified in the Print option of the header specification.

A summary of the field entries for the field description specification is provided in Appendix B.

3.3.4. Defining an Alternate Collating Sequence

If you elect to use a collating sequence other than the standard collating sequence provided by the SORT3 program, you are required to define the alternate collating sequence to be used in its place. To do this, you must prepare alternate collation (ALTSEQ) statements and include them in your job control stream. ALTSEQ statements are prepared in 80-column punched-card format and are positioned immediately after the header specification in the control stream. A punched card with double asterisks (** in columns 1 and 2) immediately follows the ALTSEQ statements to mark their ending in the job stream. When inserted into your job stream, ALTSEQ statements should appear as follows:

```
Job control statements
     .
     .
     .
// EXEC SORT3
/$
Header specification
ALTSEQ statements
**
Record type/field description specifications as required
/*
.
.
.
Job control statements
```

You may include as many ALTSEQ statements as needed to define the alternate collating sequence. Each new statement, however, must begin in column 1 and must begin with ALTSEQ.

The rules for preparing ALTSEQ statements are as follows:

1. Enter ALTSEQ in columns 1 through 6.

2. Leave columns 7 and 8 blank.

3. Enter, into columns 9 and 10, the hexadecimal equivalent of the character being moved from its normal position in the collating sequence.

4. Enter, into columns 11 and 12, the hexadecimal equivalent of the character whose position in the collating sequence is to be assumed by the character specified in step 3.

5. Repeat steps 3 and 4 for as many pairs as required to define the characters that must be taken out of the normal sequence. Do *not* leave spaces between sets of hexadecimal entries.

6. End the series of statements by placing a card with double asterisks (**) in columns 1 and 2 after the last ALTSEQ statement.

Although ALTSEQ statements do not affect data fields or forced control field characters, they do affect Factor 1 and Factor 2 fields, normal and opposite control fields, and control field characters before they are replaced or added to by forced fields.

You should consider what effect an alternate collating sequence will have on these fields for your particular job. In addition, packed and unpacked Factor 1 and 2 fields must not be specified when an alternate collating sequence is used.

Another consideration when using an alternate collating sequence is whether the characters moved in the sequence are considered equal or unequal. That is, when a character is moved into the sequence position normally assigned to another character, both the new and the original character occupy the same position. They are considered equal. If they are not to be considered equal, the character that originally occupied the position must be moved to another position. To illustrate this point, two examples are provided. The first example shows the coding required to change one character in the sequence (characters are considered equal). The second example applies to changing several characters where they are unequal.

Example 1:

```
ALTSEQ    5Ø5B
**
```

The character defined by hexadecimal 50 (&) is moved to the position defined by hexadecimal 5B ($). The ampersand and the dollar sign both occupy the same position and are therefore considered equal.

Example 2:

```
ALTSEQ    4EF3F3F4F4F5
**
```

The characters represented by the hexadecimal values shown in the ALTSEQ format are as follows:

≠	4E
3	F3
4	F4
5	F5

The format shown moves the character ≠ into the position occupied by the character 3. Because you do not want them to be considered equal, you must move the character 3 to another position. To maintain the proper sequence in the collation, the character 3 is moved into the character 4 position, 4 is moved into character 5 position, and so on. Basically, you have altered the collating sequence so that ≠ is inserted between 2 and 3.

4. Program and Control Stream Examples

4.1. GENERAL

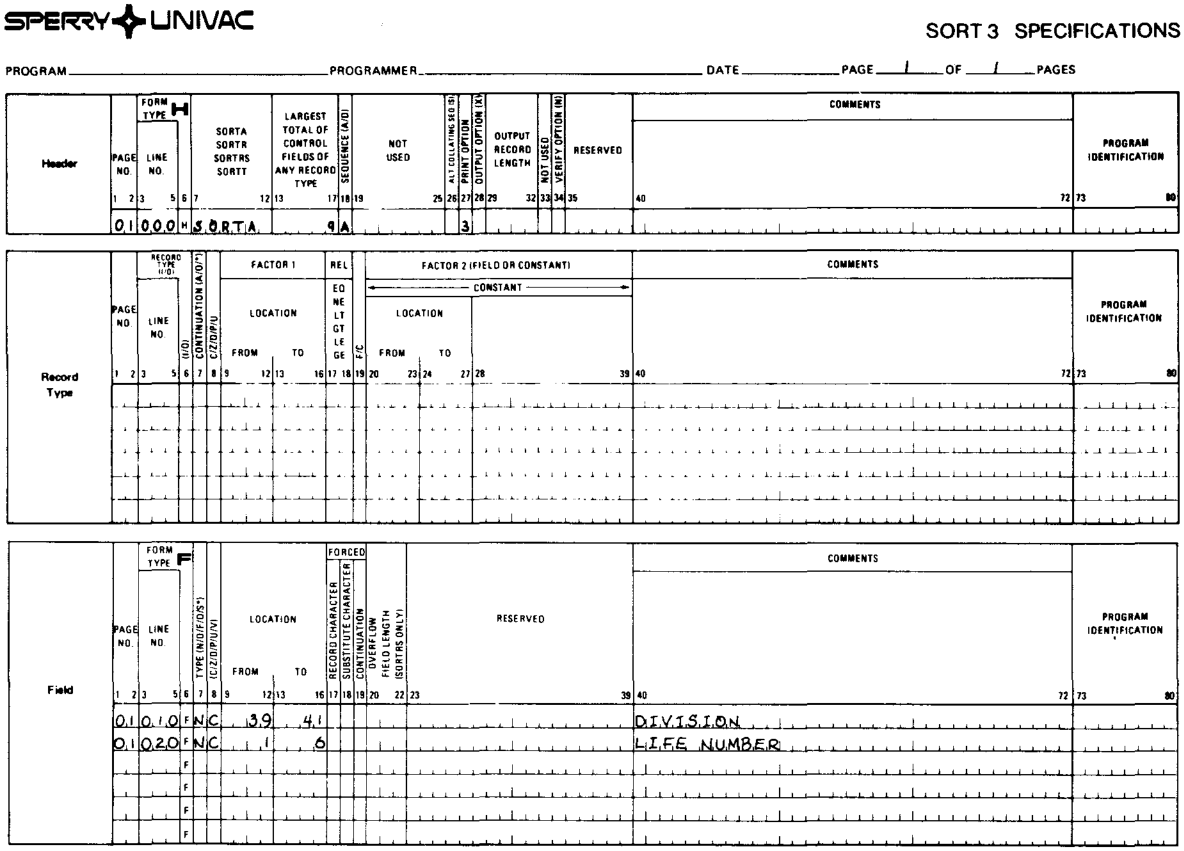
This section contains examples that illustrate program coding and job control streams for the SORT3 operation.

4.2. SORT PROGRAM CONTROL SPECIFICATION EXAMPLES AND CONTROL STREAMS

The following three examples illustrate an address-out sort, a tag-along sort, and a summary sort.

Example 1 shows the sort specifications for an address-out sort. The purpose of the job is to produce an output file containing the 10-byte relative addresses of all the records in the input file. The records are sorted in ascending order by company division number (control field positions 39–41 of the input record) and then by employee life number (control field positions 1–6) within each division. You would code the sort specifications for this job as follows:

Example 1:



SPERRY✦UNIVAC

SORT 3 SPECIFICATIONS

PROGRAM ____________ PROGRAMMER ____________ DATE ________ PAGE 1 OF 1 PAGES

Header	PAGE NO. 1 2	LINE NO. 3 5	FORM TYPE H 6	SORTA SORTR SORTRS SORTT 7 12	LARGEST TOTAL OF CONTROL FIELDS OF ANY RECORD TYPE 13 17	SEQUENCE (A/D) 18	NOT USED 19 25	ALT COLLATING SEQ (S) 26	PRINT OPTION 27	OUTPUT OPTION (X) 28	OUTPUT RECORD LENGTH 29 32	NOT USED 33	VERIFY OPTION (N) 34	RESERVED 35	COMMENTS 40 72	PROGRAM IDENTIFICATION 73 80
	01	000	H	SORTA	9	A			3							

Record Type	PAGE NO. 1 2	LINE NO. 3 5	RECORD TYPE (I/O) 6	CONTINUATION (A/O/*) 7	C/Z/D/P/U 8	FACTOR 1 LOCATION FROM 9 12	FACTOR 1 LOCATION TO 13 16	REL EQ NE LT GT LE GE 17 18	F/C 19	FACTOR 2 (FIELD OR CONSTANT) LOCATION FROM 20 23	FACTOR 2 LOCATION TO 24 27	CONSTANT 28 39	COMMENTS 40 72	PROGRAM IDENTIFICATION 73 80

Field	PAGE NO. 1 2	LINE NO. 3 5	FORM TYPE F 6	TYPE (N/O/F/D/S*) 7	(C/Z/D/P/U/V) 8	LOCATION FROM 9 12	LOCATION TO 13 16	FORCED RECORD CHARACTER 17	FORCED SUBSTITUTE CHARACTER 18	CONTINUATION 19	OVERFLOW FIELD LENGTH (SORTRS ONLY) 20 22	RESERVED 23 39	COMMENTS 40 72	PROGRAM IDENTIFICATION 73 80
	01	010	F	N	C	39	41						DIVISION	
	01	020	F	N	C	1	6						LIFE NUMBER	
			F											
			F											
			F											
			F											

In this example, the header statement defines the job as an address-out sort by the entry SORTA in columns 7–12 and specifies the longest total length of the control fields used for the sort as 9. (The two control fields, division and life number, are 3 and 6, respectively, and are contained on the same record type.) The entry A in column 18 indicates ascending order for the sort. Because no alternate collating sequence is specified, SORT3 will use the standard collating sequence. The printing of all messages is inhibited by the 3 entry in column 27.

Because all of the input records are involved in the sort, and they all have the same format, it is not necessary to prepare a record type specification for this job. SORT3 assumes an include-all situation.

Both control fields used for sorting the records are identified as normal control fields by the N entry in column 7 of the field description specification. The entry C in column 8 indicates that both the zone and the digit portion of the characters in the two control fields are used for the sort. Because the division number is the first field used in the sort, its position in the input record is defined first. It occupies three positions in the record, beginning at position 39 and ending at position 41. The next field used for sorting (life number) occupies six positions beginning at position 1 and ending at position 6.

The control stream to perform this example is as follows:

```
// JOB EXAMPLE,,8ØØØ,8ØØØ
// DVC 2Ø // LFD PRNTR
// DVC 65 // VOL SYS2ØØ // LBL DISKOUT // LFD INPUT
// DVC 66 // VOL SCR2ØØ // LBL$SCR1 // LFD DMØ1
// DVC 65 // VOL SYS2ØØ
// EXT MI,C,,CYL,2
// LBL CARDIN // LFD OUTPUT
// EXEC SORT3
/$
     HSORTA     9A      3                          DIVISION
     FNC  39  41                                   LIFE NUMBER
     FNC   1   6
/*
/&
// FIN
```

You can also use the general editor to create these control streams from a workstation. To do this, proceed as follows:

1. Turn on your workstation and log onto the system. Refer to interactive services commands and facilities user guide/programmer reference, UP-9972 (current version).

2. After you have successfully logged on, press the FUNCTION key and (while holding it down) press the SYSMODE key.

3. Key in the following job control stream:

```
Activates     EDT
EDT
                1.0000 // JOB EXAMPLE,,8000,8000
                2.0000 // DVC 20
                3.0000 // LFD PRNTR
Job             4.0000 // DVC 65
control         5.0000 // VOL SYS200
stream          6.0000 // LBL DISKOUT
for             7.0000 // LFD INPUT
execution       8.0000 // DVC 66
under           9.0000 // VOL SCR200
OS/3           10.0000 // LBL $SCR1
               11.0000 // LFD DM01
               12.0000 // DVC 65
               13.0000 // VOL SYS200
               14.0000 // EXT MI,C,,CYL,2
               15.0000 // LBL CARDIN
               16.0000 // LFD OUTPUT
               17.0000 // EXEC SORT3
               18.0000 /$
               19.0000       HSORTA       9A          3
               20.0000       FNC   39   41                            DIVISION
               21.0000       FNC    1    6                            LIFE NUMBER
               22.0000 /*
               23.0000 /&
Stores job     24.0000 @WRITE△MO=EXAMPLE,FIL=$Y$JCS
control
stream

Terminates     25.0000 @HALT
EDT
```

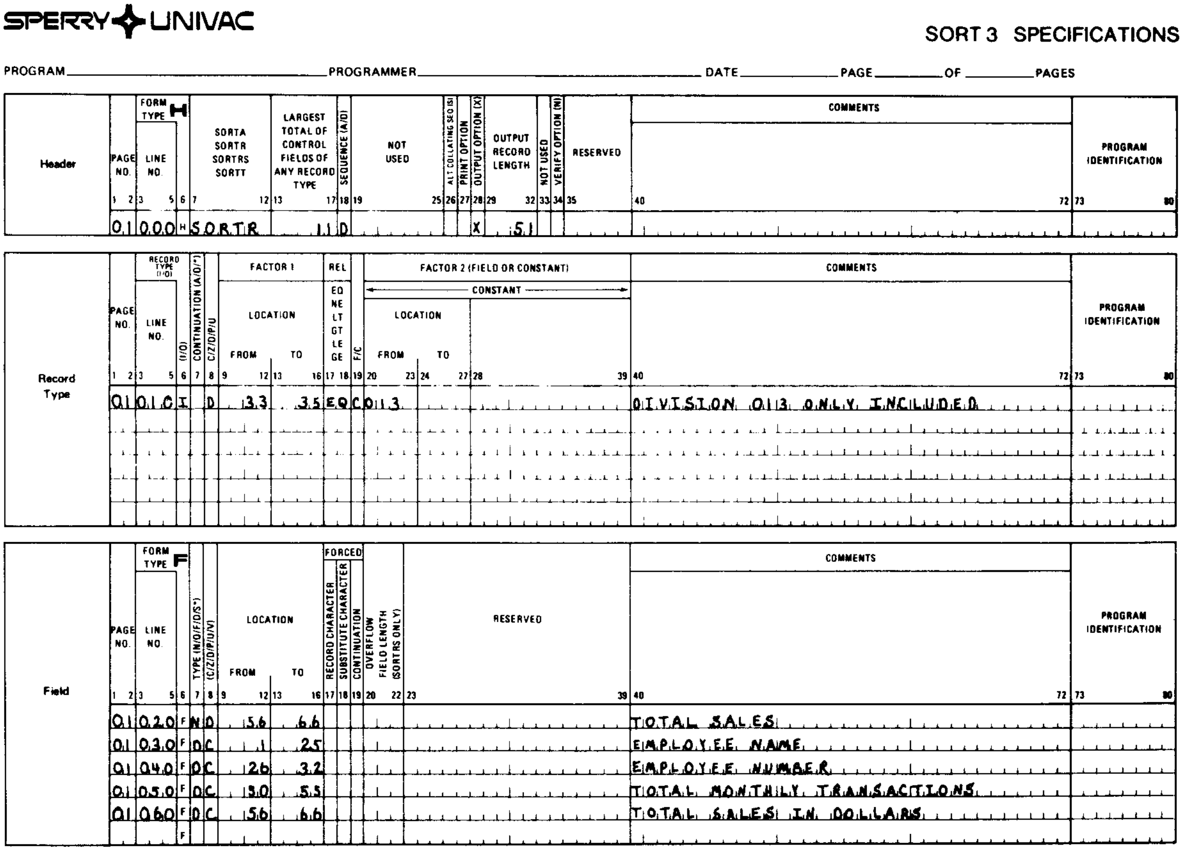
NOTE:

A '△' indicates a space.

At this point, the job control stream has been stored on $Y$JCS. You can log off the system by entering the LOGOFF command, or you can execute the program by entering RV EXAMPLE. If you log off the system, you can execute your program later by first logging on and then entering RV EXAMPLE.

Example 2 shows the sort specifications for a tag-along sort. The output file produced is to contain the records of only those salespersons working for division 013 of the company. Records are to be sorted in descending sequence by total sales. Each output record is to contain the employee name and number, total monthly transactions, and total sales in dollars. The division number is to be dropped from the output. You would code the sort specifications for this job as follows:

Example 2:



SPERRY UNIVAC

SORT 3 SPECIFICATIONS

PROGRAM_____ PROGRAMMER_____ DATE_____ PAGE_____ OF _____ PAGES

Header	PAGE NO.	FORM TYPE H LINE NO.		SORTA SORTR SORTRS SORTT	LARGEST TOTAL OF CONTROL FIELDS OF ANY RECORD TYPE	SEQUENCE (A/D)	NOT USED	ALT COLLATING SEQ (S)	PRINT OPTION	OUTPUT OPTION (X)	OUTPUT RECORD LENGTH	NOT USED	VERIFY OPTION (N)	RESERVED	COMMENTS	PROGRAM IDENTIFICATION
	1 2	3 5	6	7 12	13 17	18	19 25	26	27	28	29 32	33	34	35	40 72	73 80
	01	000	H	SORTR	11	D				X	51					

Record Type	PAGE NO.	RECORD TYPE (I/O) LINE NO.	(I/O)	CONTINUATION (A/O/*)	C/Z/D/P/U	FACTOR 1 LOCATION FROM	FACTOR 1 LOCATION TO	REL EQ NE LT GT LE GE	F/C	FACTOR 2 (FIELD OR CONSTANT) CONSTANT LOCATION FROM	TO		COMMENTS	PROGRAM IDENTIFICATION
	1 2	3 5	6	7	8	9 12	13 16	17 18	19	20 23	24 27	28 39	40 72	73 80
	01	01C	I		D	33	35	EQ	C	013			DIVISION 013 ONLY INCLUDED	

Field	PAGE NO.	FORM TYPE F LINE NO.		TYPE (N/O/F/D/S*)	(C/Z/D/P/U/V)	LOCATION FROM	LOCATION TO	FORCED RECORD CHARACTER	FORCED SUBSTITUTE CHARACTER	CONTINUATION	OVERFLOW FIELD LENGTH (SORTRS ONLY)	RESERVED	COMMENTS	PROGRAM IDENTIFICATION
	1 2	3 5	6	7	8	9 12	13 16	17	18	19	20 22	23 39	40 72	73 80
	01	020	F	N	D	56	66						TOTAL SALES	
	01	030	F	O	C	1	25						EMPLOYEE NAME	
	01	040	F	O	C	26	32						EMPLOYEE NUMBER	
	01	050	F	O	C	50	55						TOTAL MONTHLY TRANSACTIONS	
	01	060	F	O	C	56	66						TOTAL SALES IN DOLLARS	
			F											

The header statement defines the job as a tag-along sort (SORTR) and defines the total length of the control field used for the sort as 11. (Total sales field extends from position 56 to 66 in the record – a length of 11.) The entry D in column 18 specifies descending order for the sort and the X in column 28 instructs SORT3 to drop control fields from the output records. The total length of the output record, as determined by the data fields described in the field description specification, is 51. (Control field lengths are not included when they are dropped from the output records.) The standard collation sequence is used and all messages are printed.

To select only those records applicable to the salespersons employed at division 13, the record statement sets up a comparison (EQ in columns 17–18) between the division number field of the input record (33–35) and the constant 013 (columns 20–23). If the comparison proves equal, the record is included (I in column 6) in the sort.

Because the selected records are sorted according to the value in the total sales field (56–66), this field must be the first described in the field description specification. The entry N defines the field as a normal control field. Only the data portion of the characters in this field are to be used for the sort as indicated by the entry D in column 8. The control field begins at position 56 and ends at position 66. The remaining field description specifications define the data fields (D in column 7) that are to be included in the output records. The position (entries in columns 9–16) of each field is also identified for the SORT3 program. Take note that the total sales field is described twice, once as a control field and once as a data field. This was purposely done so this field would appear in the output record. Remember, SORT3 was instructed to drop control fields from the output records.

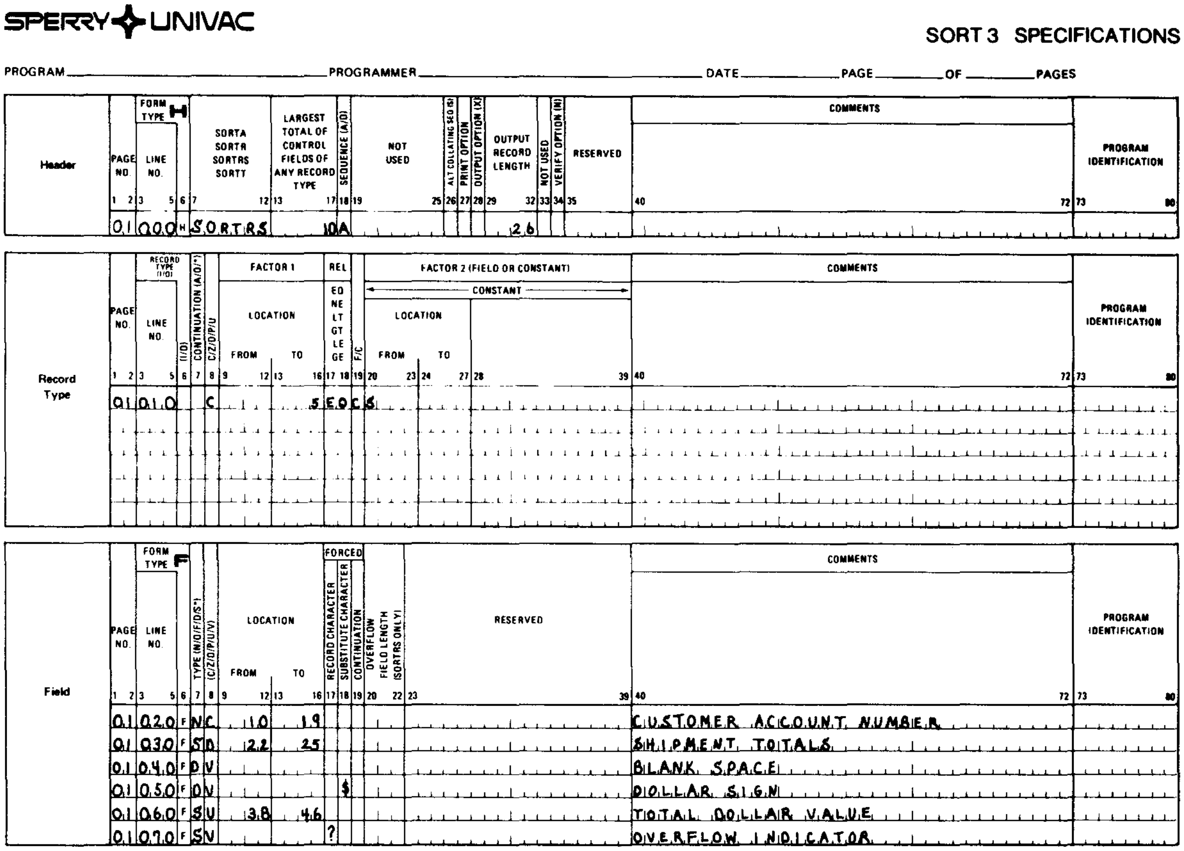
The control stream to perform this example is as follows:

```
// JOB EXAMP6,,8ØØØ,8ØØØ
// DVC 2Ø // LFD PRNTR
// DVC 65 // VOL SYS2ØØ // LBL HONPROC // LFD INPUT
// DVC 65 // VOL QC1111
// EXT MI,C,,CYL,5
// LBL NAME // LFD OUTPUT
// DVC 66 // VOL DØØ921 // LBL $SCR1 // LFD DMØ1
// EXEC SORT3
/$
     HSORTR    11D        X  49
     I D  33  35EQC13                             TOTAL SALES
     FND  56  66                                  EMPLOYEE NAME
     FNC   1  25                                  EMPLOYEE NUMBER
     FDC  26  32                                  TOTAL MONTHLY TRANSACTIONS
     FDC  5Ø  55                                  TOTAL SALES IN DOLLARS
     FDC  56  66                                  TOTAL SALES IN DOLLARS
/*
/&
// FIN
```

You can also use the general editor to create this control stream from a workstation. To do this, follow the procedure that is described for example 1.

Example 3 shows the sort specifications for a summary sort. The purpose of this job is to produce an output file that lists, by customer account, the total number of shipments made to each customer and the total dollar value of those shipments. The program, therefore, must be capable of selecting only those records of customers to whom shipments were made, and to summarize the data in the shipment and dollar value fields of each record to produce the totals required for the output. During a summary sort, duplicate records are deleted.

Example 3:



SPERRY UNIVAC

SORT3 SPECIFICATIONS

PROGRAM ____ PROGRAMMER ____ DATE ____ PAGE ____ OF ____ PAGES

Header	PAGE NO. 1–2	LINE NO. 3–5	FORM TYPE H 6	SORTA SORTR SORTRS SORTT 7–12	LARGEST TOTAL OF CONTROL FIELDS OF ANY RECORD TYPE 13–17	SEQUENCE (A/D) 18	NOT USED 19–25	ALT COLLATING SEQ (S) 26	PRINT OPTION 27	OUTPUT OPTION (X) 28	OUTPUT RECORD LENGTH 29–32	NOT USED 33	VERIFY OPTION (N) 34	RESERVED 35	COMMENTS 40–72	PROGRAM IDENTIFICATION 73–80
	01	000	H	SORTRS	10	A					26					

Record Type	PAGE NO. 1–2	RECORD TYPE (I/O) LINE NO. 3–5	(I/O) 6	CONTINUATION (A/O/*) 7	C/Z/D/P/U 8	FACTOR 1 LOCATION FROM 9–12	FACTOR 1 LOCATION TO 13–16	REL EQ NE LT GT LE GE 17–18	F/C 19	FACTOR 2 (FIELD OR CONSTANT) CONSTANT LOCATION FROM 20–23	LOCATION TO 24–27	28–39	COMMENTS 40–72	PROGRAM IDENTIFICATION 73–80
	01	010			C		5	EQ	C	S				

Field	PAGE NO. 1–2	LINE NO. 3–5	FORM TYPE F 6	TYPE (N/O/F/D/S*) 7	(C/Z/D/P/U/V) 8	LOCATION FROM 9–12	LOCATION TO 13–16	FORCED RECORD CHARACTER 17	FORCED SUBSTITUTE CHARACTER 18	CONTINUATION 19	OVERFLOW FIELD LENGTH (SORTRS ONLY) 20–22	RESERVED 23–39	COMMENTS 40–72	PROGRAM IDENTIFICATION 73–80
	01	020	F	N	C	10	19						CUSTOMER ACCOUNT NUMBER	
	01	030	F	S	D	22	25						SHIPMENT TOTALS	
	01	040	F	D	V								BLANK SPACE	
	01	050	F	O	V				$				DOLLAR SIGN	
	01	060	F	S	U	38	46						TOTAL DOLLAR VALUE	
	01	070	F	S	V			?					OVERFLOW INDICATOR	

The SORTRS entry in columns 7–12 of the header specification defines the job as a summary sort. The largest control field for any record is 10, and the records are to be sorted into ascending order (A in column 18) by customer account number. Standard collation is used, all messages are printed, and control fields are not to be dropped from the output records (columns 26–28 blank). The output record length (26) specified in columns 29–32 is a total of the control field lengths and data field lengths specified in the field description specification.

The record type specification identifies the records to be included (I in column 6) in the sort as those with the character S in position 5 of the record. The records are sorted by customer account number as defined by columns 9-16 of coding line 020 of the field description. As the individual records are sorted, the two data fields identified in coding lines 030 and 060 of the field description are summarized for identical customer account numbers. The output records, therefore, reflect the total number of shipments made to each customer and the total dollar value of those shipments. The output record format will also contain a dollar sign preceding the total dollar value field as specified by the use of the force field entry in coding line 050. An occurrence of an overflow condition while summarizing the summary data fields is indicated by a question mark (?) in the last field of the output record.

Appendix A. Standard EBCDIC and ASCII Collating Sequences

A.1. GENERAL

Appendix A provides two useful tables containing collating sequences. The first (Table A-1) presents a cross-reference table that enables you to compare the following standard codes commonly used in data processing and in OS/3:

- Hollerith punched-card code
- EBCDIC (Extended Binary Coded Decimal Interchange Code)
- ASCII (American National Standard Code for Information Interchange)
- Binary bit-pattern (bit-configuration) representation for an 8-bit system
- Hexadecimal representation

Table A-2 provides a convenient chart of OS/3 EBCDIC graphics only.

A.2. EBCDIC/ASCII/HOLLERITH CORRESPONDENCE

Table A-1 is a cross-reference table depicting the correspondences among the Hollerith punched-card code, ASCII, and EBCDIC. The table is arranged in the sorting (or collating) sequence of the binary bit patterns that have been assigned to the codes, with 0000 0000 being the lowest value in the sequence and 1111 1111 the highest. These binary bit patterns are sorted in a left-to-right sequence (most significant to least significant bit).

Note that the column headed *Decimal* uses decimal numbers to represent the positions of the codes and bit patterns in this sequence, but counts the position of the lowest value as the zero position rather than the first. Thus, the position of the highest value bit pattern, 1111 1111, is represented in the decimal column by 255, whereas it is actually the 256th in the sequence. This scheme, corresponding to the common convention for numbering bytes in which the first byte of a group is byte 0, is convenient when you are constructing a 256-byte translation table.

The column headed *Decimal* also represents the collating sequence for the EBCDIC graphic characters shown in the fourth column of the table; the fifth column, *Hollerith Punched-Card Code*, contains the hole patterns assigned to these EBCDIC graphics. Empty space in the fourth column represents the positions of the EBCDIC control characters; the EBCDIC space character is represented in the fourth column by the conventional notation SP at decimal position 64, and the corresponding card code is *no punches.*

The ASCII graphic characters, listed in the sixth column of Table A–1, are also in their collating sequence, and the hole patterns in the seventh column correspond to the ASCII graphics. The ASCII space character is represented by the notation *SP* in the sixth column at decimal position 32; the corresponding card code is, again, *no punches.* The empty space in the sixth column represents the positions of the ASCII control characters. The shading in the ASCII graphic character column indicates where the 128-character ASCII code leaves off: there are no ASCII graphic or control characters that correspond to the bit patterns higher in collating sequence than 0111 1111 (the 128th in Table A–1).

A.2.1. Hollerith Punched-Card Code

The Standard Hollerith punched-card code specifies 256 hole patterns in 12-row punched cards. Hole patterns are assigned to the 128 characters of ASCII and to 128 additional characters for use in 8-bit coded systems. These include the EBCDIC set. Note that no sorting sequence is implied by the Hollerith code itself.

A.2.2. EBCDIC

EBCDIC is an extension of Hollerith coding practices. It comprises 256 characters, each of which is represented by an 8-bit pattern. Table A–1 shows the EBCDIC graphic characters only; the EBCDIC control characters are not indicated.

A.2.3. ASCII

ASCII, which comprises 128 coded characters, each represented by an 8-bit pattern, includes both control characters and graphic characters. Only the latter are shown in Table A–1.

Table A—1. Cross-Reference Table: EBCDIC/ASCII/Hollerith (Part 1 of 5)

EBCDIC					ASCII	
Decimal	Hexadecimal	Binary	EBCDIC Graphic Character	Hollerith Punched-Card Code	ASCII Graphic Character	Hollerith Punched-Card Code
0	00	0000 0000		12-0-9-8-1		12-0-9-8-1
1	01	0000 0001		12-9-1		12-9-1
2	02	0000 0010		12-9-2		12-9-2
3	03	0000 0011		12-9-3		12-9-3
4	04	0000 0100		12-9-4		9-7
5	05	0000 0101		12-9-5		0-9-8-5
6	06	0000 0110		12-9-6		0-9-8-6
7	07	0000 0111		12-9-7		0-9-8-7
8	08	0000 1000		12-9-8		11-9-6
9	09	0000 1001		12-9-8-1		12-9-5
10	0A	0000 1010		12-9-8-2		0-9-5
11	0B	0000 1011		12-9-8-3		12-9-8-3
12	0C	0000 1100		12-9-8-4		12-9-8-4
13	0D	0000 1101		12-9-8-5		12-9-8-5
14	0E	0000 1110		12-9-8-6		12-9-8-6
15	0F	0000 1111		12-9-8-7		12-9-8-7
16	10	0001 0000		12-11-9-8-1		12-11-9-8-1
17	11	0001 0001		11-9-1		11-9-1
18	12	0001 0010		11-9-2		11-9-2
19	13	0001 0011		11-9-3		11-9-3
20	14	0001 0100		11-9-4		9-8-4
21	15	0001 0101		11-9-5		9-8-5
22	16	0001 0110		11-9-6		9-2
23	17	0001 0111		11-9-7		0-9-6
24	18	0001 1000		11-9-8		11-9-8
25	19	0001 1001		11-9-8-1		11-9-8-1
26	1A	0001 1010		11-9-8-2		9-8-7
27	1B	0001 1011		11-9-8-3		0-9-7
28	1C	0001 1100		11-9-8-4		11-9-8-4
29	1D	0001 1101		11-9-8-5		11-9-8-5
30	1E	0001 1110		11-9-8-6		11-9-8-6
31	1F	0001 1111		11-9-8-7		11-9-8-7
32	20	0010 0000		11-0-9-8-1	SP	No punches
33	21	0010 0001		0-9-1	!	12-8-7
34	22	0010 0010		0-9-2	"	8-7
35	23	0010 0011		0-9-3	#	8-3
36	24	0010 0100		0-9-4	$	11-8-3
37	25	0010 0101		0-9-5	%	0-8-4
38	26	0010 0110		0-9-6	&	12
39	27	0010 0111		0-9-7	'	8-5
40	28	0010 1000		0-9-8	(	12-8-5
41	29	0010 1001		0-9-8-1	)	11-8-5
42	2A	0010 1010		0-9-8-2	*	11-8-4
43	2B	0010 1011		0-9-8-3	+	12-8-6
44	2C	0010 1100		0-9-8-4	,	0-8-3
45	2D	0010 1101		0-9-8-5	—	11
46	2E	0010 1110		0-9-8-6	.	12-8-3
47	2F	0010 1111		0-9-8-7	/	0-1
48	30	0011 0000		12-11-0-9-8-1	0	0
49	31	0011 0001		9-1	1	1
50	32	0011 0010		9-2	2	2
51	33	0011 0011		9-3	3	3
52	34	0011 0100		9-4	4	4
53	35	0011 0101		9-5	5	5
54	36	0011 0110		9-6	6	6

Table A—1. Cross-Reference Table: EBCDIC/ASCII/Hollerith (Part 2 of 5)

EBCDIC					ASCII	
Decimal	Hexa-decimal	Binary	EBCDIC Graphic Character	Hollerith Punched-Card Code	ASCII Graphic Character	Hollerith Punched-Card Code
55	37	0011 0111		9-7	7	7
56	38	0011 1000		9-8	8	8
57	39	0011 1001		9-8-1	9	9
58	3A	0011 1010		9-8-2	:	8-2
59	3B	0011 1011		9-8-3	;	11-8-6
60	3C	0011 1100		9-8-4	<	12-8-4
61	3D	0011 1101		9-8-5	=	8-6
62	3E	0011 1110		9-8-6	>	0-8-6
63	3F	0011 1111		9-8-7	?	0-8-7
64	40	0100 0000	SP	No punches	@	8-4
65	41	0100 0001		12-0-9-1	A	12-1
66	42	0100 0010		12-0-9-2	B	12-2
67	43	0100 0011		12-0-9-3	C	12-3
68	44	0100 0100		12-0-9-4	D	12-4
69	45	0100 0101		12-0-9-5	E	12-5
70	46	0100 0110		12-0-9-6	F	12-6
71	47	0100 0111		12-0-9-7	G	12-7
72	48	0100 1000		12-0-9-8	H	12-8
73	49	0100 1001		12-8-1	I	12-9
74	4A	0100 1010	[	12-8-2	J	11-1
75	4B	0100 1011	.	12-8-3	K	11-2
76	4C	0100 1100	<	12-8-4	L	11-3
77	4D	0100 1101	(	12-8-5	M	11-4
78	4E	0100 1110	+	12-8-6	N	11-5
79	4F	0100 1111	!	12-8-7	O	11-6
80	50	0101 0000	&	12	P	11-7
81	51	0101 0001		12-11-9-1	Q	11-8
82	52	0101 0010		12-11-9-2	R	11-9
83	53	0101 0011		12-11-9-3	S	0-2
84	54	0101 0100		12-11-9-4	T	0-3
85	55	0101 0101		12-11-9-5	U	0-4
86	56	0101 0110		12-11-9-6	V	0-5
87	57	0101 0111		12-11-9-7	W	0-6
88	58	0101 1000		12-11-9-8	X	0-7
89	59	0101 1001		11-8-1	Y	0-8
90	5A	0101 1010	]	11-8-2	Z	0-9
91	5B	0101 1011	$	11-8-3	[	12-8-2
92	5C	0101 1100	*	11-8-4	\	0-8-2
93	5D	0101 1101	)	11-8-5	]	11-8-2
94	5E	0101 1110	;	11-8-6	∧	11-8-7
95	5F	0101 1111	∧	11-8-7	–	0-8-5
96	60	0110 0000	–	11	`	8-1
97	61	0110 0001	/	0-1	a	12-0-1
98	62	0110 0010		11-0-9-2	b	12-0-2
99	63	0110 0011		11-0-9-3	c	12-0-3
100	64	0110 0100		11-0-9-4	d	12-0-4
101	65	0110 0101		11-0-9-5	e	12-0-5
102	66	0110 0110		11-0-9-6	f	12-0-6
103	67	0110 0111		11-0-9-7	g	12-0-7
104	68	0110 1000		11-0-9-8	h	12-0-8
105	69	0110 1001		0-8-1	i	12-0-9
106	6A	0110 1010	¦	12-11	j	12-11-1
107	6B	0110 1011	,	0-8-3	k	12-11-2
108	6C	0110 1100	%	0-8-4	l	12-11-3
109	6D	0110 1101	—	0-8-5	m	12-11-4

Table A—1. Cross-Reference Table: EBCDIC/ASCII/Hollerith (Part 3 of 5)

EBCDIC					ASCII	
Decimal	Hexadecimal	Binary	EBCDIC Graphic Character	Hollerith Punched-Card Code	ASCII Graphic Character	Hollerith Punched-Card Code
110	6E	0110 1110	>	0-8-6	n	12-11-5
111	6F	0110 1111	?	0-8-7	o	12-11-6
112	70	0111 0000		12-11-0	p	12-11-7
113	71	0111 0001		12-11-0-9-1	q	12-11-8
114	72	0111 0010		12-11-0-9-2	r	12-11-9
115	73	0111 0011		12-11-0-9-3	s	11-0-2
116	74	0111 0100		12-11-0-9-4	t	11-0-3
117	75	0111 0101		12-11-0-9-5	u	11-0-4
118	76	0111 0110		12-11-0-9-6	v	11-0-5
119	77	0111 0111		12-11-0-9-7	w	11-0-6
120	78	0111 1000		12-11-0-9-8	x	11-0-7
121	79	0111 1001	`	8-1	y	11-0-8
122	7A	0111 1010	:	8-2	z	11-0-9
123	7B	0111 1011	#	8-3	{	12-0
124	7C	0111 1100	@	8-4	¦	12-11
125	7D	0111 1101	'	8-5	}	11-0
126	7E	0111 1110	=	8-6	~	11-0-1
127	7F	0111 1111	"	8-7		12-9-7
128	80	1000 0000		12-0-8-1	▨	11-0-9-8-1
129	81	1000 0001	a	12-0-1	▨	0-9-1
130	82	1000 0010	b	12-0-2	▨	0-9-2
131	83	1000 0011	c	12-0-3	▨	0-9-3
132	84	1000 0100	d	12-0-4	▨	0-9-4
133	85	1000 0101	e	12-0-5	▨	11-9-5
134	86	1000 0110	f	12-0-6	▨	12-9-6
135	87	1000 0111	g	12-0-7	▨	11-9-7
136	88	1000 1000	h	12-0-8	▨	0-9-8
137	89	1000 1001	i	12-0-9	▨	0-9-8-1
138	8A	1000 1010		12-0-8-2	▨	0-9-8-2
139	8B	1000 1011		12-0-8-3	▨	0-9-8-3
140	8C	1000 1100		12-0-8-4	▨	0-9-8-4
141	8D	1000 1101		12-0-8-5	▨	12-9-8-1
142	8E	1000 1110		12-0-8-6	▨	12-9-8-2
143	8F	1000 1111		12-0-8-7	▨	11-9-8-3
144	90	1001 0000		12-11-8-1	▨	12-11-0-9-8-1
145	91	1001 0001	j	12-11-1	▨	9-1
146	92	1001 0010	k	12-11-2	▨	11-9-8-2
147	93	1001 0011	l	12-11-3	▨	9-3
148	94	1001 0100	m	12-11-4	▨	9-4
149	95	1001 0101	n	12-11-5	▨	9-5
150	96	1001 0110	o	12-11-6	▨	9-6
151	97	1001 0111	p	12-11-7	▨	12-9-8
152	98	1001 1000	q	12-11-8	▨	9-8
153	99	1001 1001	r	12-11-9	▨	9-8-1
154	9A	1001 1010		12-11-8-2	▨	9-8-2
155	9B	1001 1011		12-11-8-3	▨	9-8-3
156	9C	1001 1100		12-11-8-4	▨	12-9-4
157	9D	1001 1101		12-11-8-5	▨	11-9-4
158	9E	1001 1110		12-11-8-6	▨	9-8-6
159	9F	1001 1111		12-11-8-7	▨	11-0-9-1

Table A—1. Cross-Reference Table: EBCDIC/ASCII/Hollerith (Part 4 of 5)

EBCDIC					ASCII	
Decimal	Hexa-decimal	Binary	EBCDIC Graphic Character	Hollerith Punched-Card Code	ASCII Graphic Character	Hollerith Punched-Card Code
160	A0	1010 0000		11-0-8-1		12-0-9-1
161	A1	1010 0001	~	11-0-1		12-0-9-2
162	A2	1010 0010	s	11-0-2		12-0-9-3
163	A3	1010 0011	t	11-0-3		12-0-9-4
164	A4	1010 0100	u	11-0-4		12-0-9-5
165	A5	1010 0101	v	11-0-5		12-0-9-6
166	A6	1010 0110	w	11-0-6		12-0-9-7
167	A7	1010 0111	x	11-0-7		12-0-9-8
168	A8	1010 1000	y	11-0-8		12-8-1
169	A9	1010 1001	z	11-0-9		12-11-9-1
170	AA	1010 1010		11-0-8-2		12-11-9-2
171	AB	1010 1011		11-0-8-3		12-11-9-3
172	AC	1010 1100		11-0-8-4		12-11-9-4
173	AD	1010 1101		11-0-8-5		12-11-9-5
174	AE	1010 1110		11-0-8-6		12-11-9-6
175	AF	1010 1111		11-0-8-7		12-11-9-7
176	B0	1011 0000		12-11-0-8-1		12-11-9-8
177	B1	1011 0001		12-11-0-1		11-8-1
178	B2	1011 0010		12-11-0-2		11-0-9-2
179	B3	1011 0011		12-11-0-3		11-0-9-3
180	B4	1011 0100		12-11-0-4		11-0-9-4
181	B5	1011 0101		12-11-0-5		11-0-9-5
182	B6	1011 0110		12-11-0-6		11-0-9-6
183	B7	1011 0111		12-11-0-7		11-0-9-7
184	B8	1011 1000		12-11-0-8		11-0-9-8
185	B9	1011 1001		12-11-0-9		0-8-1
186	BA	1011 1010		12-11-0-8-2		12-11-0
187	BB	1011 1011		12-11-0-8-3		12-11-0-9-1
188	BC	1011 1100		12-11-0-8-4		12-11-0-9-2
189	BD	1011 1101		12-11-0-8-5		12-11-0-9-3
190	BE	1011 1110		12-11-0-8-6		12-11-0-9-4
191	BF	1011 1111		12-11-0-8-7		12-11-0-9-5
192	C0	1100 0000	{	12-0		12-11-0-9-6
193	C1	1100 0001	A	12-1		12-11-0-9-7
194	C2	1100 0010	B	12-2		12-11-0-9-8
195	C3	1100 0011	C	12-3		12-0-8-1
196	C4	1100 0100	D	12-4		12-0-8-2
197	C5	1100 0101	E	12-5		12-0-8-3
198	C6	1100 0110	F	12-6		12-0-8-4
199	C7	1100 0111	G	12-7		12-0-8-5
200	C8	1100 1000	H	12-8		12-0-8-6
201	C9	1100 1001	I	12-9		12-0-8-7
202	CA	1100 1010		12-0-9-8-2		12-11-8-1
203	CB	1100 1011		12-0-9-8-3		12-11-8-2
204	CC	1100 1100		12-0-9-8-4		12-11-8-3
205	CD	1100 1101		12-0-9-8-5		12-11-8-4
206	CE	1100 1110		12-0-9-8-6		12-11-8-5
207	CF	1100 1111		12-0-9-8-7		12-11-8-6
208	D0	1101 0000	}	11-0		12-11-8-7
209	D1	1101 0001	J	11-1		11-0-8-1

Table A—1. Cross-Reference Table: EBCDIC/ASCII/Hollerith (Part 5 of 5)

EBCDIC					ASCII	
Decimal	Hexa-deci-mal	Binary	EBCDIC Graphic Character	Hollerith Punched-Card Code	ASCII Graphic Character	Hollerith Punched-Card Code
210	D2	1101 0010	K	11-2		11-0-8-2
211	D3	1101 0011	L	11-3		11-0-8-3
212	D4	1101 0100	M	11-4		11-0-8-4
213	D5	1101 0101	N	11-5		11-0-8-5
214	D6	1101 0110	O	11-6		11-0-8-6
215	D7	1101 0111	P	11-7		11-0-8-7
216	D8	1101 1000	Q	11-8		12-11-0-8-1
217	D9	1101 1001	R	11-9		12-11-0-1
218	DA	1101 1010		12-11-9-8-2		12-11-0-2
219	DB	1101 1011		12-11-9-8-3		12-11-0-3
220	DC	1101 1100		12-11-9-8-4		12-11-0-4
221	DD	1101 1101		12-11-9-8-5		12-11-0-5
222	DE	1101 1110		12-11-9-8-6		12-11-0-6
223	DF	1101 1111		12-11-9-8-7		12-11-0-7
224	E0	1110 0000	\	0-8-2		12-11-0-8
225	E1	1110 0001		11-0-9-1		12-11-0-9
226	E2	1110 0010	S	0-2		12-11-0-8-2
227	E3	1110 0011	T	0-3		12-11-0-8-3
228	E4	1110 0100	U	0-4		12-11-0-8-4
229	E5	1110 0101	V	0-5		12-11-0-8-5
230	E6	1110 0110	W	0-6		12-11-0-8-6
231	E7	1110 0111	X	0-7		12-11-0-8-7
232	E8	1110 1000	Y	0-8		12-0-9-8-2
233	E9	1110 1001	Z	0-9		12-0-9-8-3
234	EA	1110 1010		11-0-9-8-2		12-0-9-8-4
235	EB	1110 1011		11-0-9-8-3		12-0-9-8-5
236	EC	1110 1100		11-0-9-8-4		12-0-9-8-6
237	ED	1110 1101		11-0-9-8-5		12-0-9-8-7
238	EE	1110 1110		11-0-9-8-6		12-11-9-8-2
239	EF	1110 1111		11-0-9-8-7		12-11-9-8-3
240	F0	1111 0000	0	0		12-11-9-8-4
241	F1	1111 0001	1	1		12-11-9-8-5
242	F2	1111 0010	2	2		12-11-9-8-6
243	F3	1111 0011	3	3		12-11-9-8-7
244	F4	1111 0100	4	4		11-0-9-8-2
245	F5	1111 0101	5	5		11-0-9-8-3
246	F6	1111 0110	6	6		11-0-9-8-4
247	F7	1111 0111	7	7		11-0-9-8-5
248	F8	1111 1000	8	8		11-0-9-8-6
249	F9	1111 1001	9	9		11-0-9-8-7
250	FA	1111 1010		12-11-0-9-8-2		12-11-0-9-8-2
251	FB	1111 1011		12-11-0-9-8-3		12-11-0-9-8-3
252	FC	1111 1100		12-11-0-9-8-4		12-11-0-9-8-4
253	FD	1111 1101		12-11-0-9-8-5		12-11-0-9-8-5
254	FE	1111 1110		12-11-0-9-8-6		12-11-0-9-8-6
255	FF	1111 1111		12-11-0-9-8-7		12-11-0-9-8-7

A.3. OS/3 COLLATING SEQUENCE FOR EBCDIC GRAPHIC CHARACTERS

Table A–2 shows the OS/3 collating sequence for EBCDIC characters and unsigned decimal data. The collating sequence ranges from low (0000 0000) to high (1111 1111). The bit configurations that do not correspond to symbols (e.g., 0–73, 81–89) are not shown. Some of these correspond to control commands for printers and other devices.

Packed-decimal, zoned-decimal, fixed-point, and normalized floating-point data is collating algebraically; i.e., each quantity is interpreted as having a sign.

Table A—2. OS/3 Collating Sequence: EBCDIC Graphics (Part 1 of 2)

Collating Sequence	Bit Configuration	Symbol	Meaning
0	0000 0000		
64	0010 0000	SP	Space
⋮			
74	0100 1010	[	Opening bracket
75	0100 1011	.	Period, decimal point
76	0100 1100	<	Less than sign
77	0100 1101	(	Left parenthesis
78	0100 1110	+	Plus sign
79	0100 1111	!	Exclamation point
80	0101 0000	&	Ampersand
⋮			
90	0101 1010	]	Closing bracket
91	0101 1011	$	Dollar sign
92	0101 1100	*	Asterisk
93	0101 1101	)	Right parenthesis
94	0101 1110	;	Semicolon
95	0101 1111	¬	Logical NOT
96	0110 0000	-	Minus sign, hyphen
97	0110 0001	/	Slant
⋮			
106	0110 1010	|	Vertical bar
107	0110 1011	,	Comma
108	0110 1100	%	Percent sign
109	0110 1101	—	Underscore
110	0110 1110	>	Greater than sign
111	0110 1111	?	Question mark
⋮			
122	0111 1010	:	Colon
123	0111 1011	#	Number sign
124	0111 1100	@	At sign
125	0111 1101	'	Apostrophe, prime
126	0111 1110	=	Equals sign
127	0111 1111	"	Quotation marks
⋮			
129	1000 0001	a	
130	1000 0010	b	
131	1000 0011	c	
132	1000 0100	d	
133	1000 0101	e	
⋮			
134	1000 0110	f	
135	1000 0111	g	
136	1000 1000	h	
137	1000 1001	i	
⋮			
145	1001 0001	j	
146	1001 0010	k	
147	1001 0011	l	
148	1001 0100	m	

Table A—2. OS/3 Collating Sequence: EBCDIC Graphics (Part 2 of 2)

Collating Sequence	Bit Configuration	Symbol	Meaning
149	1001 0101	n	
150	1001 0110	o	
151	1001 0111	p	
152	1001 1000	q	
153	1001 1001	r	
:			
161	1010 0001	~	Tilde
162	1010 0010	s	
163	1010 0011	t	
164	1010 0100	u	
165	1010 0101	v	
166	1010 0110	w	
167	1010 0111	x	
168	1010 1000	y	
169	1010 1001	z	
:			
192	1100 0000	{	Opening brace
193	1100 0001	A	
194	1100 0010	B	
195	1100 0011	C	
196	1100 0100	D	
197	1100 0101	E	
198	1100 0110	F	
199	1100 0111	G	
200	1100 1000	H	
:			
201	1100 1001	I	
:			
208	1101 0000	}	Closing brace
209	1101 0001	J	
210	1101 0010	K	
211	1101 0011	L	
212	1101 0100	M	
213	1101 0101	N	
214	1101 0110	O	
215	1101 0111	P	
216	1101 1000	Q	
217	1101 1001	R	
:			
224	1110 0000	\	Reverse slant
226	1110 0010	S	
227	1110 0011	T	
228	1110 0100	U	
229	1110 0101	V	
230	1110 0110	W	
231	1110 0111	X	
232	1110 1000	Y	
233	1110 1001	Z	
:			
240	1111 0000	0	
241	1111 0001	1	
242	1111 0010	2	
243	1111 0011	3	
244	1111 0100	4	
245	1111 0101	5	
246	1111 0110	6	
247	1111 0111	7	
248	1111 1000	8	
249	1111 1001	9	

Appendix B. SORT3 Specifications Summary

B.1. GENERAL

This appendix summarizes the SORT3 specifications and is provided as a quick reference aid. The SORT3 specifications are described in detail in Section 3.

B.2. HEADER SPECIFICATION

Function:

> Defines the type of sort operation you want SORT3 to perform. It also defines the criteria for formatting the sorted output file. Only one header specification is permitted for each sort job.

Format:

Header	PAGE NO.	FORM TYPE LINE NO.	H	SORTA SORTR SORTRS SORTT	LARGEST TOTAL OF CONTROL FIELDS OF ANY RECORD TYPE	SEQUENCE (A/D)	NOT USED	ALT COLLATING SEQ (S)	PRINT OPTION	OUTPUT OPTION (X)	OUTPUT RECORD LENGTH	NOT USED	VERIFY OPTION (N)	RESERVED	COMMENTS	PROGRAM IDENTIFICATION
	1 2	3 5	6	7 12	13 17	18	19 25	26	27	28	29 32	33	34	35	40 72	73 80
			H													

Table B-1 summarizes the header specification field entries.

Table B—1. Column Summary for Header Specification (Part 1 of 2)

Column Number	Allowable Entries	Explanation
1-2	00	Page number
3-5	000	Line number of specification
6	H	Identifies specification as the header specification. This is a mandatory entry.
7-12	SORTA	Identifies the job as an address-out sort (disk files only)
	SORTR	Identifies the job as a tag-along sort
	SORTRS	Identifies the job as a summary tag-along sort
	SORTT	Provided for System/3 compatibility. If specified, SORT3 performs a tag-along sort.
	An entry in this field is mandatory. If omitted, SORT3 will not execute.	
13-17	1-256	Specifies the longest control field used in sorting your input records. An entry in this field is mandatory and must be right-justified within the field.
18	A	Tells SORT3 to arrange records in ascending order in the output file
	D	Tells SORT3 to arrange records in descending order in the output file
19-25	Blank	Not used
26	Blank	Tells SORT3 to use the standard OS/3 collating sequence in compare operations
	S	Tells SORT3 to use an alternate collating sequence in compare operations. You are responsible for providing the ALTSEQ statements needed for defining the collating sequence to be used.
27	0 or blank	Tells SORT3 to print and display: Sort specifications Diagnostic messages Program status messages Action messages Other system messages
	1	Tells SORT3 to print and display: Program status messages Action messages Other system messages
	2	Tells SORT3 to print and display action messages and other system messages only
	3	Tells SORT3 to print and display other system messages only
28	Blank	Tells SORT3 to retain control fields in the output records for tag-along sort jobs
	X	Tells SORT3 not to retain control fields in the output record for tag-along sort jobs
29-32	1-4096	Specifies the length of the output records in a tag-along sort job. An entry in this field is mandatory for tag-along sorts. The entry must be right-justified.
33	Blank	Not used

Table B—1. Column Summary for Header Specification (Part 2 of 2)

Column Number	Allowable Entries	Explanation
34	Blank	Tells SORT3 to verify the data written on the work file
	N	Tells SORT3 not to verify the data written on the work file
35-39	Blank	Reserved
40-80	Blank or any OS/3 characters	Not used by SORT3. May be used for comments or program identification

B.3. RECORD TYPE SPECIFICATION

Function:

Defines the criteria that SORT3 must use in determining which records in your input file are to be included or omitted from the sort. It is not necessary to prepare record type specifications when your sort job includes all the records in your file and they all have the same format.

Format:

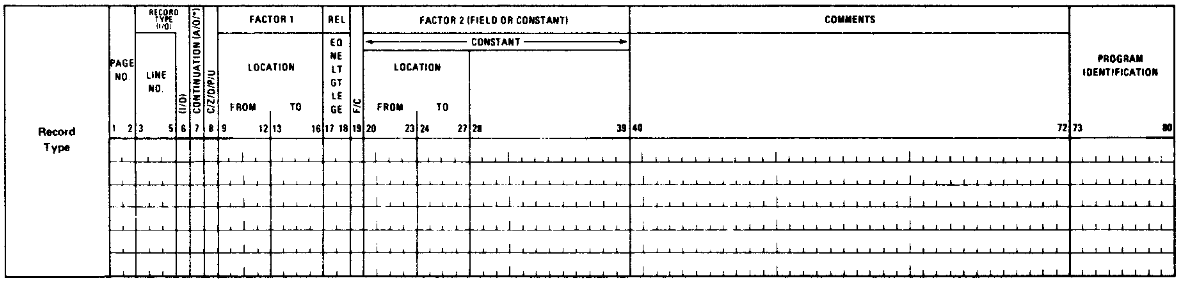


Table B-2 summarizes the record specification field entries.

Table B—2. Column Summary for Record Type Specification (Part 1 of 2)

Column Number	Allowable Entries	Explanation
1-2	00–99	Page number
3-5	01n–06n	Line number of specification. (Leave column 5 blank or enter any value to keep each line of the specification in ascending sequence.)
6	I	Tells SORT3 that the record defined in this specification is to be included in the sort
	O	Tells SORT3 that the record defined in this specification is to be omitted from the sort
7	Blank	Tells SORT3 that this is the first line of an include (I) or omit (O) record type specification
	A	Tells SORT3 that this specification line is a continuation of the record definition described in the previous specification line (AND function)
	O	Tells SORT3 that this specification line defines a record type different from that described in the previous specification line (OR function)
	*	Tells SORT3 that this is a comment line
8	C	Tells SORT3 to use both zone and digit portions of characters during compare operations
	Z	Tells SORT3 to use only the zone portion of 1-character fields during compare operations
	D	Tells SORT3 to use only the digit portion of characters during compare operations
	P	Tells SORT3 that data is signed packed decimal
	U	Tells SORT3 that data is signed unpacked decimal
9-12	1-4096	Specifies the position at which the Factor 1 field begins in the input record. Entry must be right-justified.
	Blank	Factor 1 field is one character long.
13-16	1-4096	Specifies the position at which the Factor 1 field ends in the input record. Entry must be right-justified.
17-18	EQ	Tells SORT3 that the results of the comparison between the Factor 1 and Factor 2 fields must be equal
	NE	Tells SORT3 that the results of the comparison between the Factor 1 and Factor 2 fields must not be equal
	LT	Tells SORT3 that the Factor 1 field must be less than Factor 2 field
	GT	Tells SORT3 that the Factor 1 field must be greater than the Factor 2 field
	LE	Tells SORT 3 that the Factor 1 field must be less than or equal to the Factor 2 field
	GE	Tells SORT 3 that the Factor 1 field must be greater than or equal to the Factor 2 field
19	C	Defines Factor 2 as a constant
	F	Defines Factor 2 as another field in the same input record
	K	Defines Factor 2 as a keyword: UDATE, UDAY, UMONTH, or UYEAR

Table B—2. Column Summary for Record Type Specification (Part 2 of 2)

Column Number	Allowable Entries	Explanation
20-23	1-4096	Specifies the position at which the Factor 2 field begins in the input record. Entry must be right-justified.
	Blank	Factor 2 field is one character long.
24-27	1-4096	Specifies the position at which the Factor 2 field ends in the input record. Entry must be right-justified.
20-39	Any characters	Defines the constant for Factor 2 field. Entry must be right-justified when Factor 2 is alphanumeric constant, right-justified when numeric constant. The length of the Factor 2 field must be the same as the length specified for the Factor 1 field. If Factor 1 field is a packed number, the Factor 2 field length must be twice the length of the Factor 1 field.
40-80	Blank or any OS/3 characters	Not used by SORT3. May be used for comments or program identification

B.4. FIELD DESCRIPTION SPECIFICATION

Function:

Defines how you want the records formatted in the sorted output file. For tag-along and summary sort jobs, the field descriptions appearing in your job stream define the fields used to create the output records. For address-out sort jobs, the field descriptions define the control fields used to sort the record addresses for the output file. Field descriptions are required for every sort job.

Format:

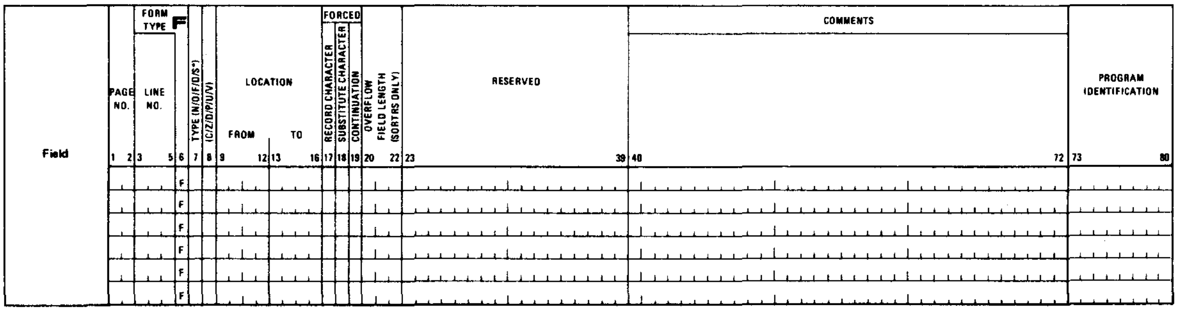


Table B-3 summarizes the field description specification field entries.

Table B—3. Column Summary for Field Description Specification (Part 1 of 2)

Column Number	Allowable Entries	Explanation
1-2	00-99	Page number
3-5	01n-06n	Line number of the specification. (Leave column 5 blank or enter any value to keep each line of the specification in ascending sequence.)
6	F	Identifies this specification as a field description
7	N	Tells SORT3 that this is a normal control field description and that is to be used to sort records in the normal sequence as specified by the entry in column 18 of the header specification
	O	Tells SORT3 that this is an opposite control field description and that it is to be used to sort records in a sequence opposite to that specified by the entry in column 18 of the header specification
	F	Tells SORT3 that this is a forced control field description and that is to be used to force the addition of a character or the modification of a given field in the work records constructed for the sort
	D	Tells SORT3 that this is a data field description and that is to be included in the output record (Applicable only to tag-along and summary sorts. Data field descriptions for address-out sorts are processed as comments.)
	S	Tells SORT3 that this is a summary data field description and that is to be summarized when used in a summary type sort (Summary data fields are processed as normal data fields when specified for tag-along sorts and as comments when specified for address-out sorts.)
	*	Tells SORT3 that this specification line is a comment
8	P	Tells SORT3 that the data defined is numeric and consists of packed, signed decimal numbers
	U	Tells SORT3 that the data defined is numeric and consists of unpacked, signed decimal numbers
	C	Tells SORT3 to use both the zone and digit portions of each byte in the input record field defined to build sort work records
	Z	Tells SORT3 to use only the zone portion of each byte in the input record field defined to build sort work records
	D	Tells SORT3 to use only the digit portion of each byte in the input record field defined to build sort work records
	V	Tells SORT3 to force the character defined in column 18 into the data field specified in the specification
9-12	1-4096	Specifies the position at which the field begins in the output record. Entry must be right-justified.
	Blank	Specifies that the field is only one byte long.
13-16	1-4096	Specifies the position at which the field ends in the output record. Entry must be right-justified.

Table B—3. Column Summary for Field Description Specification (Part 2 of 2)

Column Number	Allowable Entries	Explanation
17	Any character	Defines the character to be replaced in the input record (replacement or forced character specified in column 18)
18	Any character	Specifies the character that replaces the character defined in column 17. Character substitution actually takes place in the work and output records and not in the input record.
19	Blank	Tells SORT3 that the forced control field described in this specification line is not a continuation of the preceding specification line
	Any character	Tells SORT3 that the forced control field described in this specification line is a continuation of the preceding specification line
20-22	1-256	Specifies overflow field length for summary sort jobs. Entry must be right-justified.
23-39	Not used	
40-80	Blank or any OS/3 characters	Not used by SORT3. May be used for comments or program identification

Index

CUT

SPERRY

USER COMMENT SHEET

We will use your comments to improve subsequent editions.

NOTE: Please do not use this form as an order blank.

____

(Document Title)

_______________ _______________ _______________

(Document No.) *(Revision No.)* *(Update No.)*

Comments:

From:

____

(Name of User)

____

(Business Address)

Fold on dotted lines, and mail. (No postage stamp is necessary if mailed in the U.S.A.)
Thank you for your cooperation

CUT

FOLD

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 21 BLUE BELL, PA.

POSTAGE WILL BE PAID BY ADDRESSEE

SPERRY CORPORATION

ATTN.: SOFTWARE SYSTEMS PUBLICATIONS

P.O. BOX 500
BLUE BELL, PENNSYLVANIA 19424

FOLD

SPERRY

USER COMMENTS

We will use your comments to improve subsequent editions.

NOTE: Please do not use this form as an order blank.

____

(Document Title)

____________ ____________ ____________

(Document No.) *(Revision No.)* *(Update Level)*

Comments:

From:

____

(Name of User)

____

(Business Address)

Fold on dotted lines, and mail. (No postage is necessary if mailed in the U.S.A.)
Thank you for your cooperation

FOLD

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 21 BLUE BELL, PA.

POSTAGE WILL BE PAID BY ADDRESSEE

SPERRY CORPORATION

ATTN: SYSTEM PUBLICATIONS

P.O. BOX 500
BLUE BELL, PENNSYLVANIA 19422-9990

FOLD

UNISYS

USER COMMENTS

We will use your comments to improve subsequent editions.

NOTE: Please do not use this form as an order blank.

____

(Document Title)

____________________ ____________________ ____________________

(Document No.) *(Revision No.)* *(Update Level)*

Comments:

From:

____

(Name of User)

____

(Business Address)

Fold on dotted lines, and mail. (No postage is necessary if mailed in the U.S.A.)
Thank you for your cooperation

CUT

FOLD

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 21 BLUE BELL, PA.

POSTAGE WILL BE PAID BY ADDRESSEE

Unisys Corporation
E/MSG Product Information Development
PO Box 500 C1-NE6
Blue Bell, PA 19422-9990

FOLD