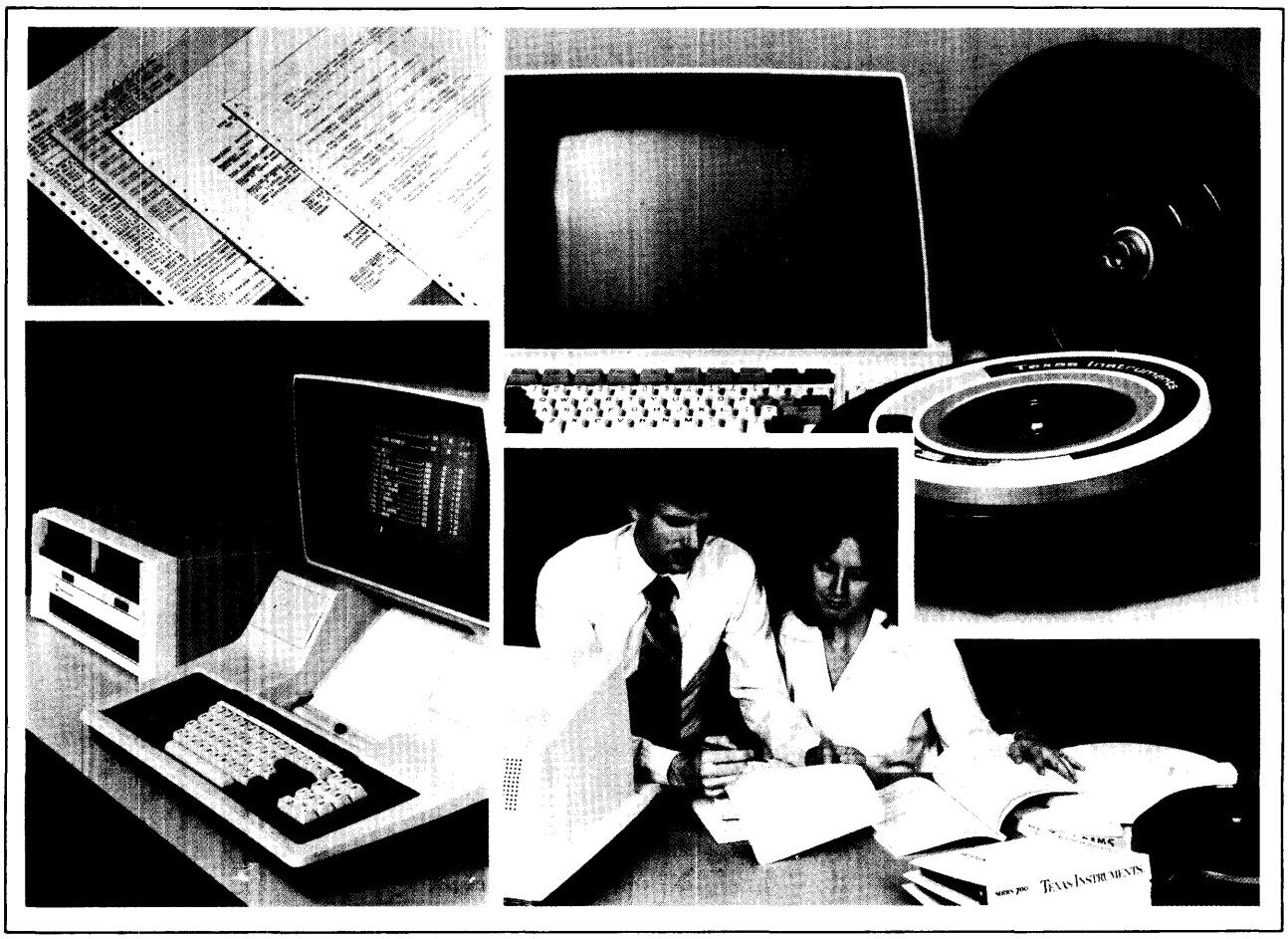
Model 990 Computer
DX Sort/Merge
User's Guide



Part No. 946252-9701 *B
15 December 1980

TEXAS INSTRUMENTS

MANUAL REVISION HISTORY

Model 990 Computer DX Sort/Merge User's Guide (946252-9701)

Original Issue ..... 1 August 1977
Change 1 ..... 15 March 1978 (ECN 419591)
Revision ..... 15 August 1979 (MCR 000134)
Revision ..... 15 December 1980 (MCR 000325)

The total number of pages in this publication is 244.

Preface

This manual contains information on the use and operation of the Texas Instruments Sort/Merge utility. The manual describes the following:

- Building specification records that define the type of sort or merge
- Executing Sort/Merge using subroutine calls from FORTRAN, COBOL, Pascal, BASIC, or assembly language programs
- Communicating with Sort/Merge interactively and in batch mode
- Using Sort/Merge specifications (includes examples of specification entry)

Anyone familiar with the operating system file structure and the interactive use of the System Command Interpreter (SCI) can use the Sort/Merge utility. Sections 1 through 3 of this manual provide an overview of the uses of the utility as well as examples of simple sorting and merging processes. The remaining sections describe the utility's full range of capabilities. Those sections that pertain to applications programming and using language programs with Sort/Merge assume that you have programming experience.

Sort/Merge supports FORTRAN, COBOL, Pascal, TI 990 BASIC, and assembly language programs. Sort/Merge no longer supports existing interfaces for Scientific and Business BASIC.

The manual contains the following sections and appendixes:

Section

1 Introduction — Describes the capabilities of the Sort/Merge utility and explains the organization of this manual.

2 Usage Overview — Defines the types of sorts and merges and gives an example of each. This section also explains the required sequence of the control statements.

3 Simple Sort and Merge Examples — Presents specific examples of two simple sort operations and one merge operation using a batch stream format.

4 Header Specification — Describes in detail the specification required to perform the various sort and merge operations and the options provided.

5 File Description Specification — Describes the attributes that can be designated for the output file, work file, and input files used by Sort/Merge.

6 Record Selection Specification — Describes the options provided to select particular records from the input file according to a specific type of record or by comparing specific data fields within each record.

7 Reformatting Specification — Describes the criteria by which a file can be sorted or by which several files can be merged in ascending or descending order using multiple control parameters. This section also explains how to change the order of the various data fields within each record.

8 Sort/Merge SCI Interface and Execution — Describes the use of the SCI to interactively specify to Sort/Merge all information required for creating the required control statements and files.

9 External Interface Specifications — Describes each of the subroutines that run the Sort/Merge utility from a language program.

10 Examples — Presents examples of the use of Sort/Merge by a parts distributing company to update a master inventory file, to record and sort daily entries, and to produce summary listings.

11 Sort/Merge Structure — Describes the various internal Sort/Merge modules and how they interact.

Appendixes

A Sort/Merge Messages

B Alternate Collating Sequence Specifications

C ASCII Codes

D EBCDIC Codes

E Data Types for FORTRAN, COBOL, Pascal, BASIC, and RPG II

F Estimating Sorting Time

G Sorting with Many Records or Limited Workspace

The following TI Model 990 Computer documents contain information related to the Sort/Merge utility:

Title	Part Number
DX10 Operating System Release 3 Reference Manual:	
Volume I: Concepts and Facilities	946250-9701
Volume II: Production Operation	946250-9702
Volume III: Application Programming Guide	946250-9703
Volume IV: Developmental Operation	946250-9704
Volume V: System Programming Guide	946250-9705
Volume VI: Error Reporting and Recovery	946250-9706
Assembly Language Programmer's Guide	943441-9701
FORTRAN Programmer's Guide	945411-9701
COBOL Programmer's Guide	945412-9701
Pascal Programmer's Guide	946290-9701
BASIC Programmer's Guide	945413-9701
Link Editor Reference Manual	949617-9701

Contents

4 — Header Specification

5 — File Description Specification

6 — Record Selection Specification

7 — Reformatting Specification

8 — Sort/Merge SCI Interface and Execution

9 — External Interface Specifications

Illustrations

Tables

1

Introduction

1.1 GENERAL

The Texas Instruments Sort/Merge utility operates under the DX operating system. This utility can sort one file or merge from two to five sorted files into a single file according to criteria specified by the user. The Sort/Merge utility can perform the following functions:

- Sort input file records into a specified sequence.
- Merge records of sorted input files.
- Select particular input file records to be included in the output file.
- Rearrange the order of the data fields within the records that are in the output file.
- Place specified characters, such as a dollar sign or a blank space, between data fields.
- Select particular data fields according to arithmetic values or constants.
- Sum numeric data fields and summarize alphabetic data fields.

Sort/Merge does not modify the contents of the input files. Instead, sequenced records are written to a separate output file. In the performance of a sort, Sort/Merge uses an input file, a temporary work file, and an output file. In a merge process, two to five input files, no work file, and one output file are involved.

1.2 CODING REQUIREMENTS

The Sort/Merge utility uses column-oriented specifications contained in a control file or an application program to process sort and merge operations. You can prepare the required code in one of the following ways:

- Prepare a control file that contains all of the necessary specifications arranged in the required order and aligned in the proper columns.
- Use the System Command Interpreter (SCI) to interactively specify to Sort/Merge the pathnames, values, and options in response to field prompts. Sort/Merge builds the control file from these responses.
- Prepare a batch stream containing the control file commands and keyword parameters that will create the control file.
- Prepare an application program containing control statements that are passed to Sort/Merge by the executing task using predefined subroutines.

You must specify the Sort/Merge commands in the control file in a particular sequence. To aid you in organizing the information that must be specified, a Sort/Merge Specification worksheet is used throughout this manual. The worksheet is divided into four major categories of specifications, each of which is described in a separate section of the manual.

1.3 ACCESSING SORT/MERGE

You can execute Sort/Merge as a utility or from a user program written in assembly language, FORTRAN, COBOL, Pascal, or BASIC. When executed as a utility, Sort/Merge processes user-entered control statements and handles its own input/output (I/O). When Sort/Merge is executed from a language program, the CALL statement construct is used, and control statements and/or data records are passed between task and Sort/Merge.

1.4. HOW TO USE THIS MANUAL

The first three sections of the manual provide a beginning user with the information necessary to perform simple sort and merge operations. Section 3 contains batch control stream examples that can be modified with pathnames and parameters to fit specific user requirements.

All users of Sort/Merge should read the paragraph in Section 2 entitled Record Selection and Reformatting. The combining of sets of file selection, record selection, and reformatting specifications is a concept that must be thoroughly understood to take full advantage of the capabilities of Sort/Merge.

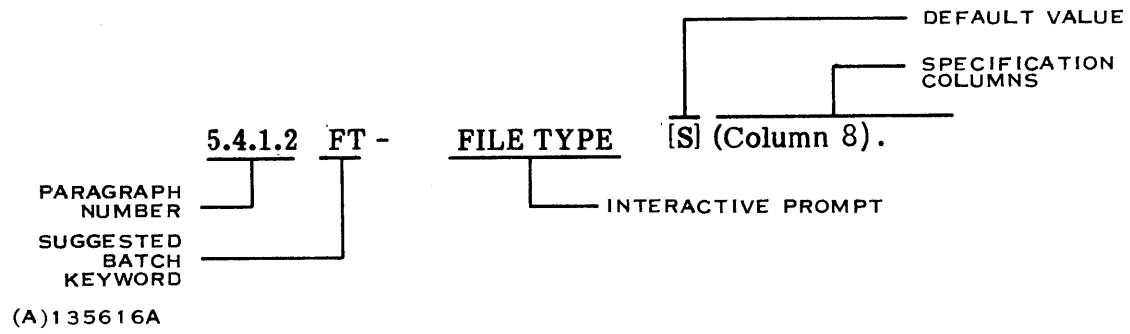
Sections 4 through 7 present all of the required and optional specifications that can be designated for a sort or merge operation. Tables in each section list the possible options, suggested keywords, and default values. Read these sections to become familiar with the options provided so that this information can be located when needed.

Section 8 explains the use of the SCI to enter the necessary specifications so that Sort/Merge can build the control file needed for processing. This section also gives details on executing Sort/Merge in batch mode.

If you intend to access Sort/Merge from a language program, read Section 9 and Appendix E, which contains a listing of the data types.

1.5 PARAGRAPH HEADING FORMAT

The format of the paragraph headings in Sections 4 through 7 is a summary of information for each specification, as shown in the following example:



The example is the heading for the paragraph describing the Input File Type specification. The suggested batch keyword is an abbreviated form of the interactive prompt. The default value, shown in brackets, is used if a carriage return is entered from a keyboard or no parameter is specified in the batch stream. The applicable column numbers for coding on the specification sheet are shown in parentheses. This formatting convention allows you to use the Table of Contents as a quick-reference summary of this information.

2

Usage Overview

2.1 GENERAL

The following paragraphs define the different types of sort and merge operations and briefly explain the ways in which fields control and manipulate data. An example of a typical application of Sort/Merge is presented, illustrating how to combine sort and merge processes. The remainder of the section describes the information specified under each of the four specification groups. The Sort/Merge Specification worksheet is helpful in organizing this information. You should examine the worksheet until you are familiar with the order in which the specifications are entered; however, you need not memorize column numbers unless you intend to code the control file instead of using a batch stream or the SCI to build it for you.

2.2 RECORD FIELDS

A record is a group of related data that is treated as one unit by Sort/Merge. The fields, or columns of data, in the input records are important in the sort and merge processes. Sort/Merge uses fields to select records that are to be included in the sort or merge operation, to reformat the records by including only particular fields or rearranging the order of the fields, and to control the sequence in which the records occur in the output listing.

Sort/Merge views an input record as a collection of fields that comprise one or more contiguous character positions of the input record. Generally, you would format records that contain the same type of information so that each contains the same kind of data in the same field positions, beginning and ending on the same column positions. However, you can place common data fields in different character positions in different sets of records and use record selection and reformatting specifications to process the records as if they had the same format.

One file can contain many different types of records that have different formats. You can use a particular field to uniquely identify each type of record. Also, you can force a character (a 1 for example) before records having one format and another character before records having a different format. You can then specify to Sort/Merge that records identified by one character be treated differently than records identified by another character (or characters).

The contents in specified fields can be read by Sort/Merge and used to process the records in various ways. Sort/Merge can select certain records for processing and omit others, depending upon the contents of a specific field. It can also rearrange selected fields for each record in the output file. The ascending or descending sequence of entire records in the output file is determined by the selection of one or more of the fields within the records.

2.2.1 Control Fields

A control field is the set of data that Sort/Merge places in numeric or alphabetic order. The control fields determine the final sequence of records in the work and output files. Two or more control

fields can give a hierarchical structure to the sorted output. The sequencing can be compounded by ordering one control field in one sequence (ascending or descending) and another control field in the opposite sequence.

2.2.2 Data Fields

A data field contains characters that are not used for sequencing. These fields comprise the tag-along portion of the input records that are included with the control fields in the work file and may be included in the output file, depending upon your specifications.

2.2.3 Forced Fields

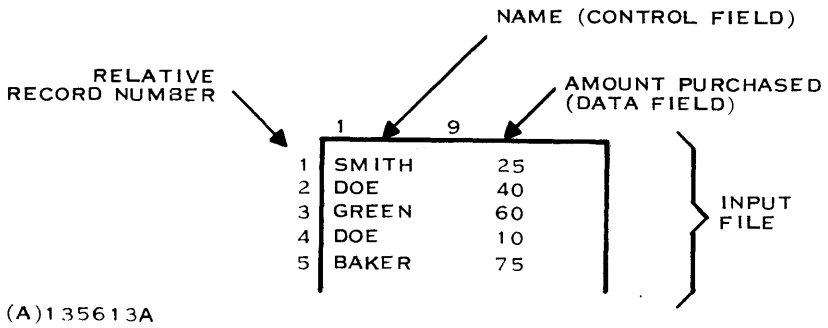
A character that is not in the input record can be forced into a particular record position or substituted for another character when the record is placed in the work file. The forced character appears in the work file as an additional character at the beginning or end of a record or between fields. More than one character may be forced, but each one is considered a separate forced field that has a field length of one character. A forced field can be used as a data field or as a control field. You can use a forced field to perform a simple function, such as inserting a dollar sign in the output file, or to perform complex functions, such as selecting various sets of different types of records according to a forced field identifier. Forced fields are described in detail in Section 7.

2.3 TYPES OF SORTS (TAG-ALONG, SUMMARY TAG-ALONG, AND ADDRESS-ONLY)

The three types of sorts are as follows:

- Tag-along sort — Data fields tag-along with the control fields in an output record.
- Summary tag-along sort — This sort is the same as a tag-along sort except that certain data fields are summed, showing only the total of the values found in that data field.
- Address-only sort — Fields from the input records are not placed into the output file; instead, the output file consists of the relative record numbers of the input records after they have been sorted according to the sequence specifications.

For the following descriptions of these three types of sorts, assume that a file contains input records that have only two fields, names of buyers and the amounts they purchased on account. The names could be used as the control field for sequencing and the amounts purchased could be the tag-along data field. The file has the following entries:



2.3.1 Tag-Along Sort

Suppose the following criteria is entered on the specification sheets:

- Sort alphabetically by name. (Essentially, this is a sort by the numerical sequence of the ASCII code used to make up the characters.)
- Include data only for Smith, Green, and Baker.

A tag-along sort can include only control fields, only data fields, or both in the output records, as shown below:

(A) BOTH CONTROL AND DATA FIELDS

(B) CONTROL FIELD ONLY

(C) DATA FIELD ONLY

(A)135614

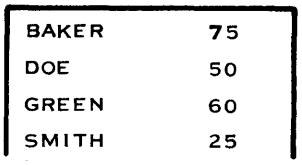
If both are included, the control fields precede the data fields (that is, control fields are on the left, and data fields are on the right). However, if you want the data fields to appear first or to be intermixed with the contents of control fields, the control fields must also be defined as data fields. The control fields are then dropped when the output file is written. Only the data fields that are to be in the output records are specified and organized in the order desired. This organization of data fields is accomplished through the control specifications.

2.3.2 Summary Tag-Along Sort

The summary tag-along sort can include the following combinations of fields:

- Control fields only
- Summary data fields only
- Summary data fields and data fields
- Summary data fields, control fields, and data fields

In a summary tag-along sort, data fields may be summed. For instance, the input file example shows two entries for the name Doe. A summary sort of this file with the dollar amounts to be summed according to each name produces an output file such as the following, which contains both control and data fields:



(A)135615

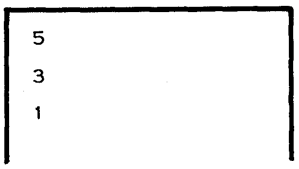
Only one Doe entry appears, showing the sum of the dollar amounts in the input records. In this case, the dollar amount fields were designated as summary data fields.

If you do not include a summation field specification, the output file consists of records with unlike control fields. Sort/Merge deletes all records with common control fields, retaining only one copy of each unique record sequence.

2.3.3 Address-Only Sort

In an address-only sort, fields in the input file are not copied into the output file. The output file contains four-byte records that specify the relative record numbers of the sorted file, according to the sequence specifications. Input records are reformatted and entered into the work file for sequencing. Part of their format includes the relative record number of the file. After the sort, these relative record numbers are placed in the output file in the sequence resulting from the sort.

For example, the tag-along sort example in this section sorts names and dollar amounts according to the alphabetical order of the names (control field). Only the data for Smith, Green, and Baker is included. If an address-only sort is designated using the same control specifications, the following output file results:



(A)135617

2.4 TYPES OF MERGES (NORMAL AND SUMMARY)

Sort/Merge can process two types of merge operations: normal and summary.

2.4.1 Normal Merge

In a normal merge, the output is composed of fields from the input records. The merged records can contain the following:

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

The sequence specifications for the merge determine the type of output produced. When the output file records contain both control fields and data fields, all of the control fields precede all of the data fields in an output file record.

2.4.2 Summary Merge

In a summary merge, the merged (output) records can contain the following:

- Control fields only
- Summary data fields only
- Summary data fields and data fields
- Summary data fields, control fields, and data fields

A summary merge produces the same results as a summary sort.

2.5 SORT/MERGE EXAMPLE

The following is an example of a store with two departments, each keeping its own records on amounts sold to each customer. Sales are entered into the computer daily. The resulting files are shown at the top of Figure 2-1. At the end of each month, a list is prepared to show the accounting department the total amount to bill each customer. Figure 2-2 shows this process in three steps:

1. Sort each file into alphabetical order; reformat the data so that the name is the first field in the records.
2. Merge the two files into one file in alphabetical order.
3. Summary sort the merged file; sum the dollar amounts and show the total purchased by each customer. Force a dollar sign before the total amount purchased.

As an alternative to steps 2 and 3, you can execute a summary merge. (The merge and summary sort in Figure 2-2 are shown as an example only.) When restocking of the store's inventory is necessary, you can execute a summary sort of the merged file in step 3 to show the total sold for each part. Similarly, you can determine buying trends by producing a report that gives a summary of the parts used by customers.

2.6 PROGRAM REQUIREMENTS

To sort a file or merge several files, Sort/Merge requires program control statements that indicate the following:

- Pathname of the file(s) to be sorted or merged
- Size and type of files for the sort or merge
- Records to be included in the sort or merge (all others are rejected)
- Input record fields to be included in the output records
- Order of the fields (left to right) in the output records

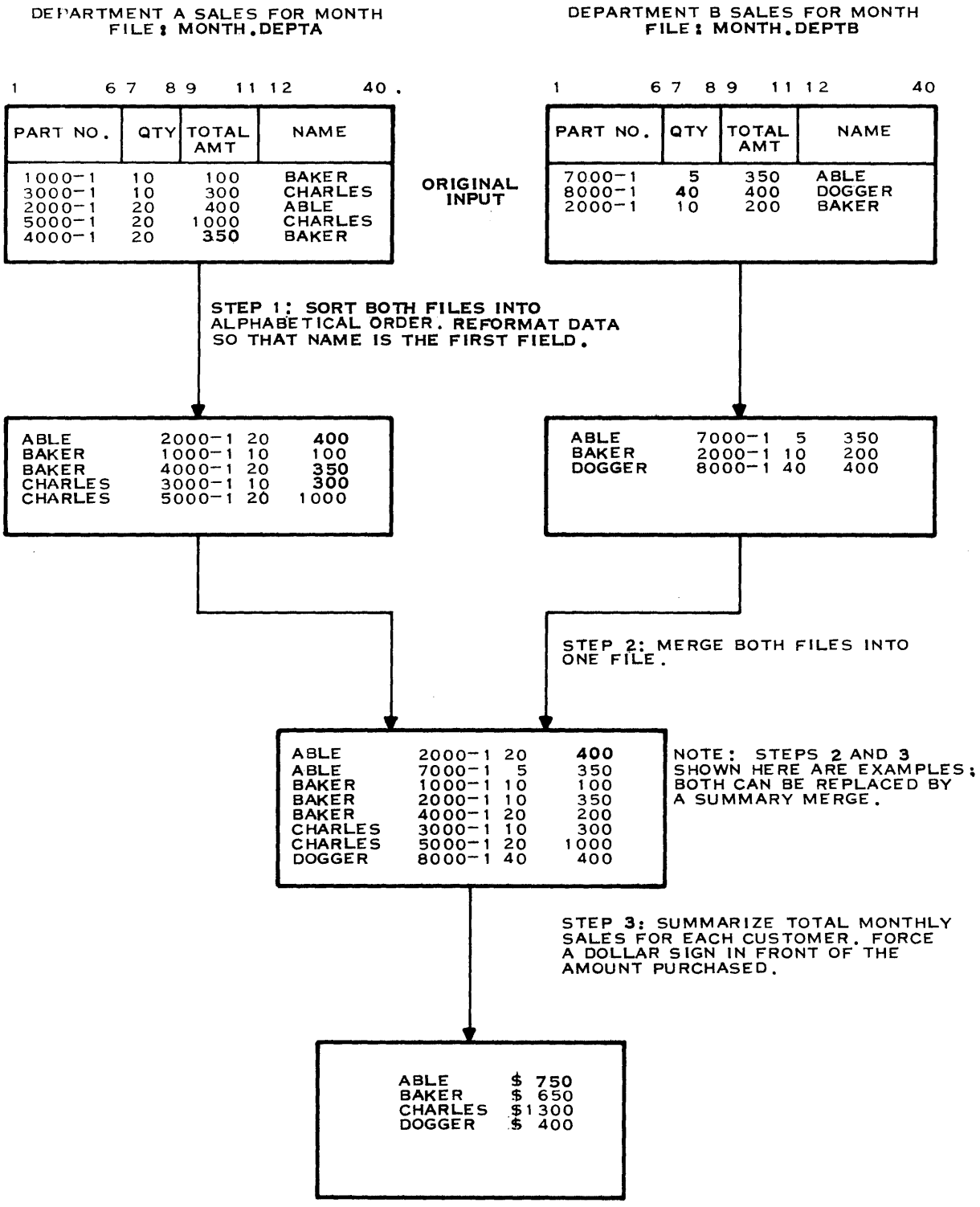


Figure 2-1. Example of Sort/Merge Process Showing Printouts of Results at Each Step

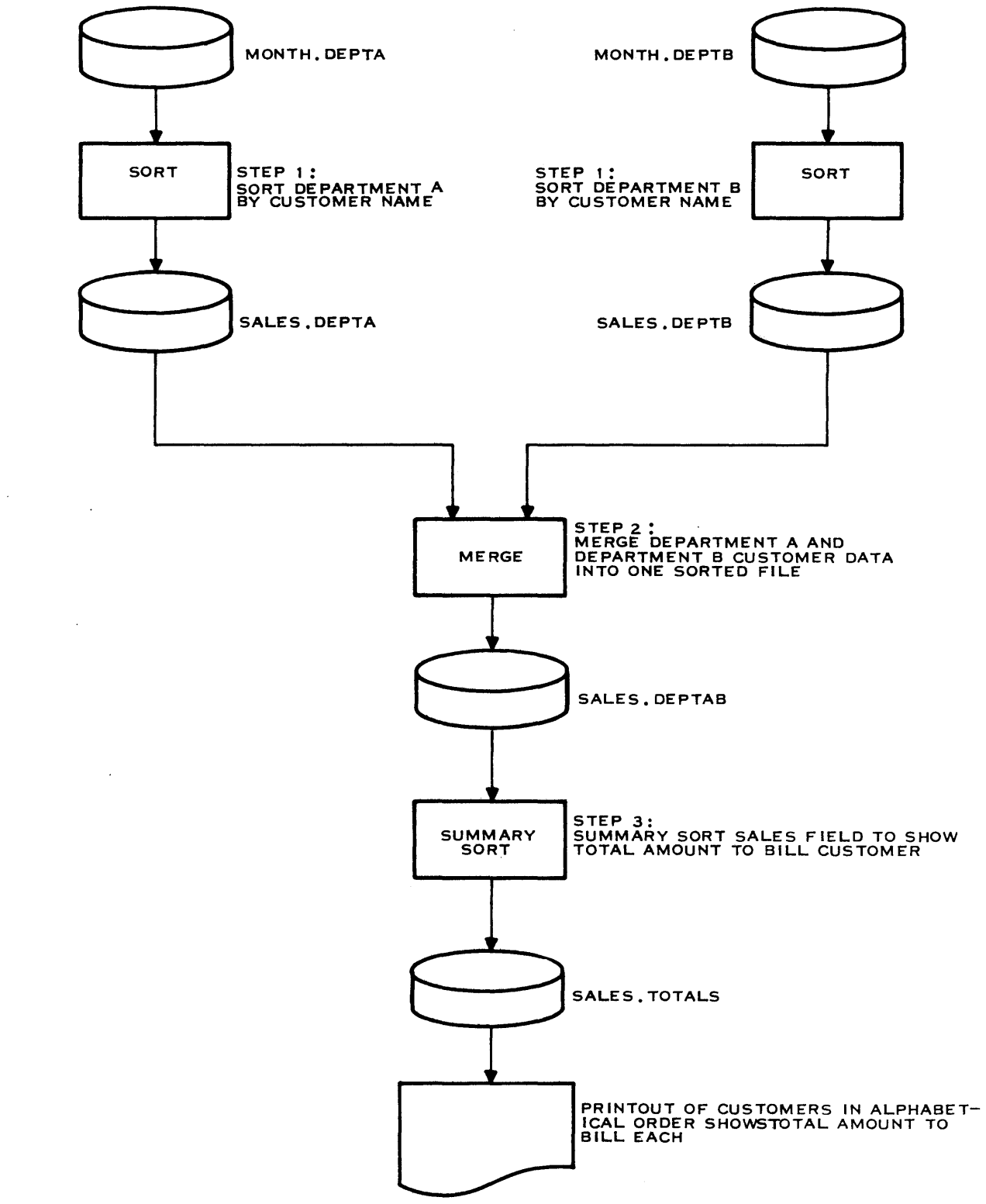


Figure 2-2. Example of Sort/Merge Process

- Fields that should be summed. (Values in a specific field are added together and a total is output.)
- Characters that are to be placed (forced) into specific locations of the output records (which characters and which positions)
- Characters to be added to the records to control the sort or merge process (but not necessarily included in the output records)
- Sequence of data fields: ascending or descending
- Data or messages to be printed
- Whether to perform a sort that does not output data records but merely gives the record number sequence in the output file (address-only sort).

Although this list does not encompass all of the control statements that you can specify for a sort or merge, it is a representative sample of such statements.

2.7 OPERATOR/PROGRAM COMMUNICATION

You can provide Sort/Merge with the required control statements in any of the following ways (as shown in Figure 2-3):

- Through the SCI, which interprets interactive and batch commands, bids (executes) Sort/Merge, and passes control statements to Sort/Merge
- Through separate tasks that bid Sort/Merge and pass control statements to it. (The calling task can also send records to Sort/Merge and can receive the processed records returned from Sort/Merge. These tasks can be written in assembly language, FORTRAN, COBOL, Pascal, or BASIC.)
- By executing the Sort/Merge task and specifying the pathname of a file or physical device that contains the control statement specifications

Control statements define the Sort/Merge process; they are provided to Sort/Merge in the order in which they appear on the specification sheets shown in Figures 2-4 and 2-5. The order is as follows:

- Header Specification — Section 4
- File Description Specification — Section 5
- Record Selection Specification — Section 6
- Reformatting Specification — Section 7

2.7.1 Header Specification

This specification describes the type of sort or merge, the length of the output records, and other program requirements (such as sequencing in ascending or descending order and whether to output messages during the sort or merge process).

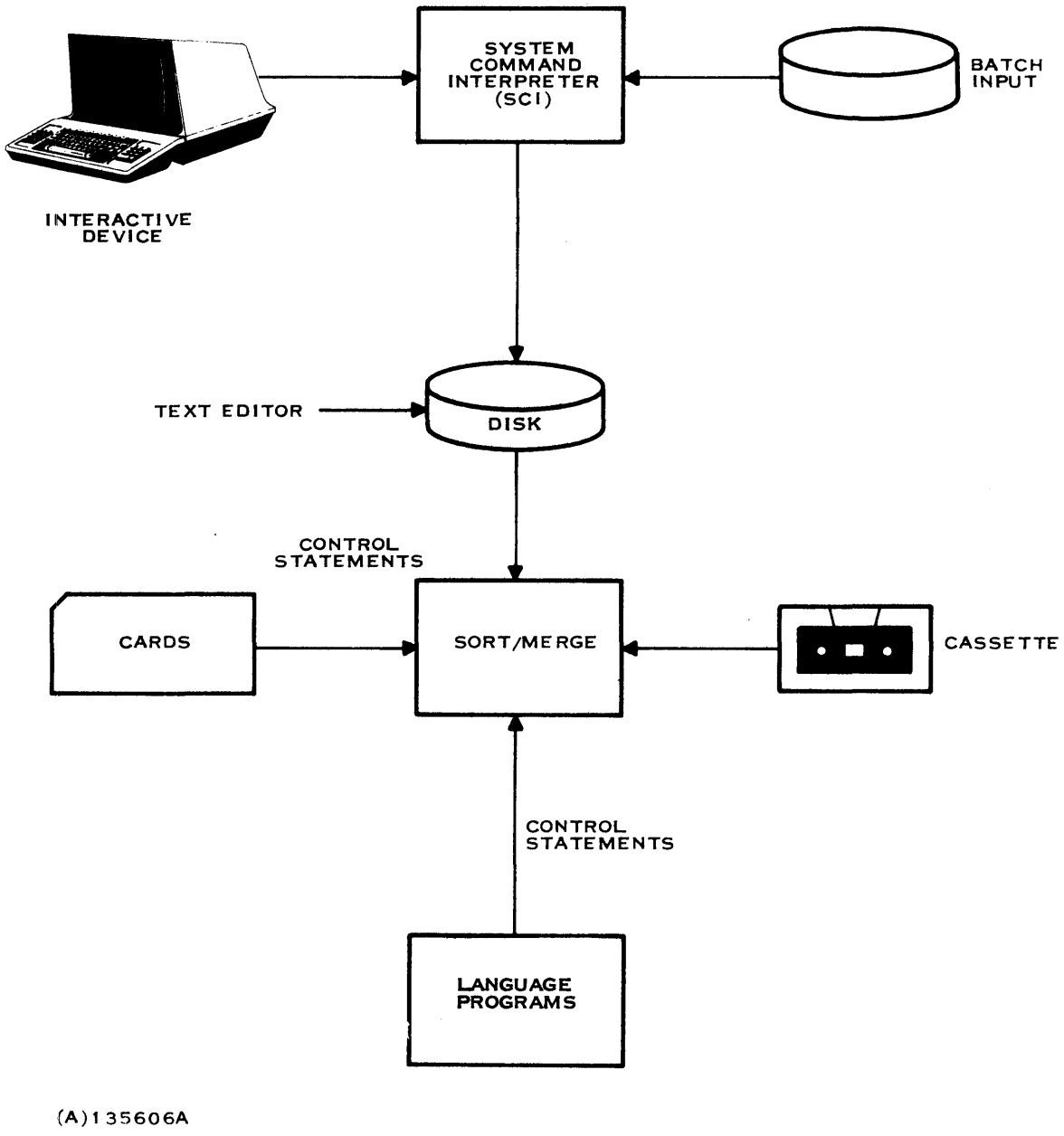


Figure 2-3. Methods of Specifying Sort/Merge Control Statements

SORT/MERGE SPECIFICATIONS, SHEET A

Page 0 0

Header Specification

Line Number (3 4 5)	Spec Type (6)	Merge: MERGE, MERGES / Sort: SORTA, SORTR, SORTRS (7 8 9 10 11 12)	Largest Total of Control Fields of Any Record Type (Bytes) (13 14 15 16 17)	Sequence (A D) (18)	Reserved (19 20 21 22 23 24 25)	Alternate Coll. Seq. (S) (26)	Print Option (0/1/3) (27)	Drop Ctrl. Flds. (X) (28)	Output Record Length (29 30 31 32)	Variable Lngth. Recs. (V) (33)	Verify Option (V) (34)	Memory Work Space (35 36 37 38 39)	Comments (Columns 40 to 72)
0 0 0	H												

File Description Specifications

OUTPUT FILE

Line Number (3 4 5)	Spec Type (6)	File Use (7)	F.T. (S/R/I) (8)	File Pathname (9 10 11 12 13 14 15 16 17 18 19 20 21 22 23)	Logical Record Length (24 25 26 27 28)	Physical Record Length (29 30 31 32 33)	Number of Logical Records (34 35 36 37 38 39 40 41 42 43 44)	Comments (Columns 45 to 72)
0 1	D	O						
0 2	D	A						

WORK FILE

Line Number (3 4 5)	Spec Type (6)	File Use (7)	F.T. (E/N) (8)	Volume Name (9 10 11 12 13 14 15 16)	Comments (Columns 17 to 80)
0 3	D	W			

INPUT FILE

Line Number (3 4 5)	Spec Type (6)	File Use (7)	F.T. (S/R/I) (8)	File Pathname (9 10 11 12 13 14 15 16 17 18 19 20 21 22 23)	L.R. Length (24 25 26 27 28)	No. of Records (29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44)	Comments (Columns 45 to 72)
0 4	D	I					
0 5	D	A					
0 6	D	I					
0 7	D	A					
0 8	D	I					
0 9	D	A					
1 0	D	I					
1 1	D	A					
1 2	D	I					
1 3	D	A					

(A) 135607

Figure 2-4. Sort/Merge Specification, Sheet A

SORT/MERGE SPECIFICATION, SHEET B

Page [1 2]

Record Selection Specification

Line Number 3 4 5	TYPE (I/O) 6	Continuation (A/0) 7	(C/Z/D/S/P/U) 8	Operand 1 Location From 9 10 11 12	Operand 1 Location To 13 14 15 16	Rel. EQ NE LT GT LE GE 17 18	F/C 19	Operand 2 (Field or Constant) Location From 20 21 22 23	Operand 2 Location To 24 25 26 27	Operand 2 Constant 28 29 30 31 32 33 34 35 36 37 38 39	Comments (Columns 40 to 72)
14											
15											
16											
17											
18											
19											
20											
21											
22											
23											
24											
25											
26											

Reformatting Specification

Line Number 3 4 5	Line Type 6	TYPE (N/O/F/D/S) 7	(C/Z/D/S/I/F/V/P/U) 8	Location From 9 10 11 12	Location To 13 14 15 16	Forced Record Character 17	Forced Substitute Character 18	Forced Continuation 19	Overflow 20 21 22	Reserved 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39	Comments (Columns 40 to 72)
27	F										
28	F										
29	F										
30	F										
31	F										
32	F										
33	F										
34	F										
35	F										
36	F										
37	F										
38	F										
39	F										
40	F										

135608

Figure 2-5. Sort/Merge Specification, Sheet B

2.7.2 File Description Specification

This specification describes the three different files used in a sort or merge operation:

- The output file is the file that contains the records after they have been processed into the specified sequence and format.
- The work file is the file on which strings of sorted records are temporarily stored until the record selection process is complete. The work file is then used to combine the strings of sorted records.
- The input file(s) is the file to be sorted or merged. A merge is performed on from two to five sorted input files.

Input and output files may be sequential, relative record, or key indexed.

NOTE

Relative record files sometimes contain unused records set aside for future expansion. You should omit these unused records during the Sort/Merge process to avoid wasting disk space.

2.7.3 Record Selection Specification

The entries on this sheet describe which records in the input file(s) are to be included or omitted during the Sort/Merge process. Specific ASCII characters found in specific positions (fields) of the input file records determine the record selection. You can check fields against other fields in the same record to determine record selection, or you can compare a field to a string of constants. Besides providing the identifying characteristics of the desired records, this specification further specifies whether the records identified are to be included in the output file.

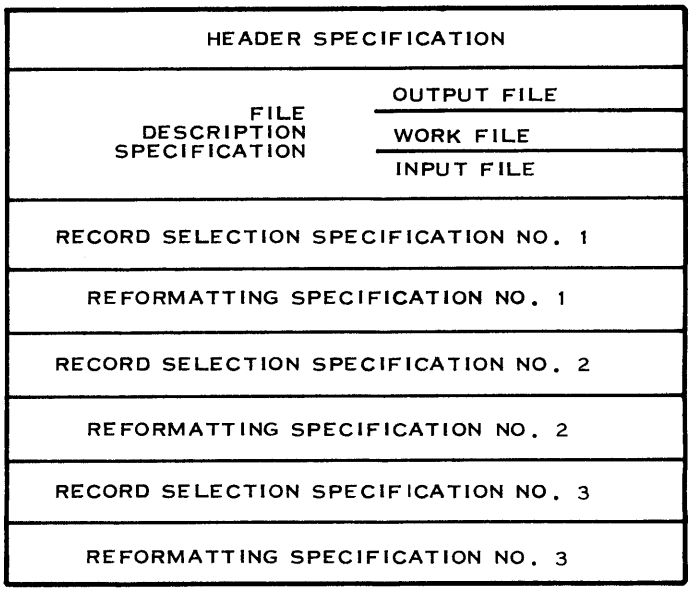
2.7.4 Reformatting Specification

The Reformatting Specification designates which record fields are to be summed, how the input record fields are to be ordered in the output records, the sequencing of individual fields, and the characters to be forced into output record positions or substituted for other characters. It also states which fields are to be treated as signed ASCII, integer, or floating point.

2.7.5 Order of Specifications

You must input the four specifications to Sort/Merge in a specific order. Input the Header Specification first, followed by the File Description, Record Selection (if used), and Reformatting Specifications.

Figure 2-6 shows the typical specification sequencing for a sort. Three sets of Record Selection and Reformatting Specifications are used because each of the record types selected by the Record Selection Specifications requires different formatting for the fields within the records. If all of the records in the file are to be included in the sort, the Record Selection Specification is not necessary and you can describe the formatting using only one group of Reformatting Specifications. Note that in a sort operation, only one input file is described in the File Description Specification (that is, only one file can be sorted at a time); however, from two to five files can be merged in a merge operation.

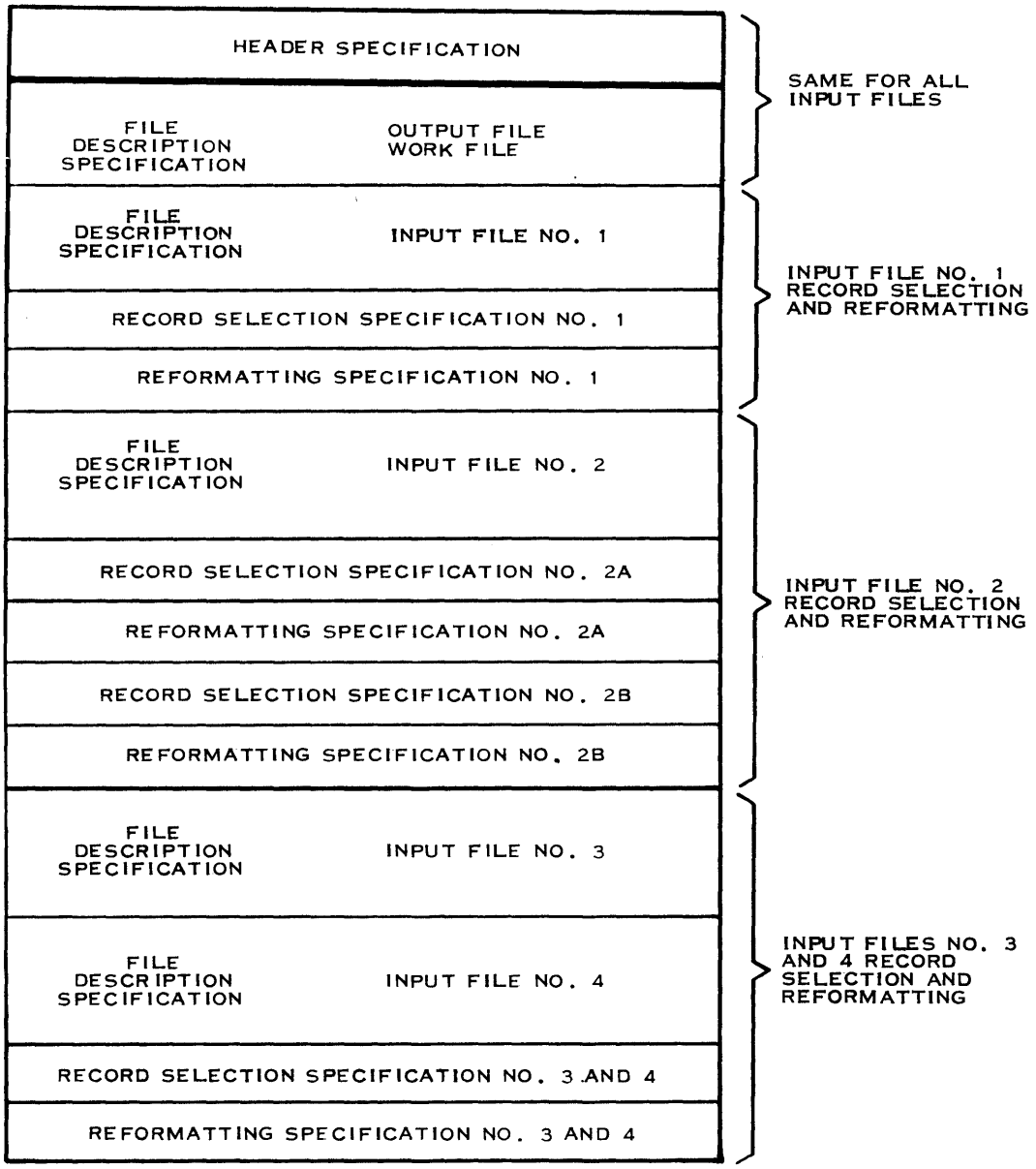


(A)135609

Figure 2-6. Typical Specification Sequence for a Sort

Although a sort or merge operation has only one Header Specification and one File Description Specification, the number of Record Selection Specifications and Reformatting Specifications can vary. Record Selection Specifications and Reformatting Specifications are specified in groups unless all of the records in the file are to be sorted or merged. In the latter case, only a Reformatting Specification is required; this is called an implied include-all sort or merge.

Figure 2-7 shows the specifications required for a typical merge procedure. Note at the bottom of the figure that different input files have their records selected by the same Record Selection Specification and corresponding Reformatting Specification. However, Input File 2 (in the center of Figure 2-7) is merged according to a selection of records specified in two different Record Selection Specifications and their accompanying Reformatting Specifications.



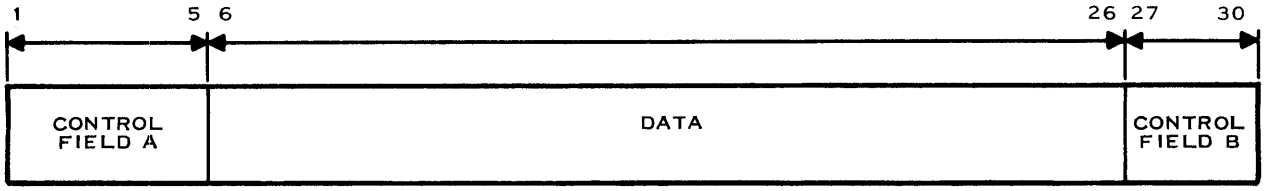
(A)135610A

Figure 2-7. Typical Specification Sequence for a Merge

2.8 RECORD SELECTION AND REFORMATTING

The following procedure is an example of how the Sort/Merge program processes the records in a file.

1. Sort/Merge reads a record from the input file:



(A)135661

2. Sort/Merge checks the Record Selection Specifications to see if this record is to be included in the sort or merge. (You might not want to sort or merge all of the records in the file.)

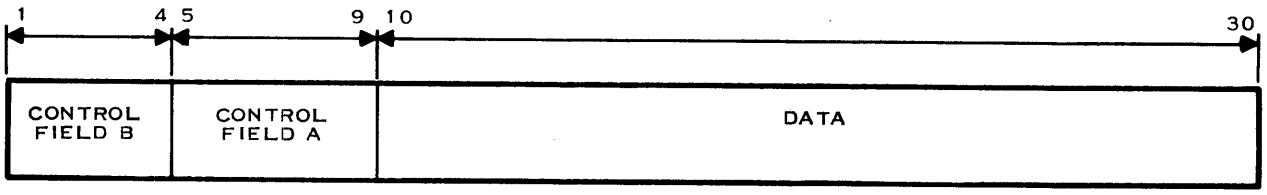
3. If the record is to participate in the sort or merge, Sort/Merge builds a work record, formatting it according to the Reformatting Sequence Specifications. (The format of the work record is important because it controls the format of the output record.) The Reformatting Specifications might specify the following:

   a. Put the contents of positions 27-30 of the input record into the first available positions (1-4) of the work record.

   b. Put the contents of positions 1-5 in the input record into the next available positions (5-9) of the work record.

   c. Put the contents of positions 6-26 in the input record into the next available positions (10-30) of the work record.

   The resulting work record would look like this:



(A)135612

   Control fields that are not dropped always precede data fields in the work and output records. To specify dropping control fields from output records, enter an X in column 28 of the Header Specification.

4. Sort/Merge checks the sequence specifications to see how to arrange the records in the output file (ascending or descending order by control fields) and writes the records to the work file in sorted strings.

5. Sort/Merge writes the records to the output file in the order specified. The only exception occurs when control fields are dropped from the output record.

3

Simple Sort and Merge Examples

3.1 GENERAL

The Sort/Merge program is a powerful, general-purpose utility that provides many different options. Optimum use of the utility requires familiarity with these options.

Until you have become familiar with the specifications required to perform more complex operations, you can use the examples presented in this section to quickly sort or merge data file records. These examples show two simple sort operations and a simple merge operation and describe how to modify them for general use.

If you are not an experienced programmer, read through the material presented in this section, then perform the exercise near the end of the section.

3.2 CODING THE SORT OR MERGE

The following sort and merge examples are in batch control stream format. You can modify the values, column locations, and pathnames (file names and device names) to specify the parameters and pathnames appropriate for your application and system. In all examples, you can replace the device names (such as DS01 or DS02) with volume names. Use the Text Editor to create a sequential file to contain the batch control stream. Enter the batch control stream statements in the exact order given in the examples. The pathname that you assign to this file is the pathname you will use with the Execute Batch (XB) command to execute the batch control stream. The procedure for executing the batch control stream is presented in a paragraph following the examples.

Table 3-1 lists and explains the command statements used in the examples. Some of the command statements are optional and are excluded, when appropriate, from the batch streams in the examples. Table 3-1 contains all of the Sort/Merge command statements in the order in which they must be entered, if used, when creating a batch control stream.

If a command statement is included in the batch stream, it may be followed by one or more keywords that specify parameters. The keywords used in the examples in this section are briefly defined to help you understand their use. Keywords for other options are listed in Sections 4 through 7. The ordering of the keywords within each command statement line is not critical.

Several of the command statements are optional because Sort/Merge automatically creates an unblocked sequential output file and assigns the temporary work file to disk if you do not specify another destination.

Table 3-1. Batch Stream Commands and Use

Command	Remarks
BATCH	Indicates the beginning of a batch stream. This command is required.
SM$SMC	Creates the control file that will contain the specifications created by the subsequent commands. This statement is required and must immediately follow the BATCH statement.
SM$HD	Builds the header control statement. This statement is required and must immediately follow the SM$SMC command. The type of sort or merge can be specified on this line; if the type is not specified, a tag-along sort (SORTR) is the default value assumed.
SM$OUT	Builds the output file control statement. It must immediately follow the SM$HD command.
SM$WKF	Builds the work file control statement. This statement is optional for a sort operation; when used, it must immediately follow the SM$OUT command. If a merge is specified, this command is not necessary and will be ignored if specified.
SM$IN	Builds the input file control statement(s). The first SM$IN must follow SM$WKF. For a sort, only one SM$IN may be specified. For a merge, at least two and no more than five SM$IN commands may be used; also, from two to five sets of the SM$IN, SM$SLC, and SM$REF sequence may be specified to create more complex selection and reformatting patterns.
SM$SLC	Builds the record selection control statement(s). This command is optional. If no SM$SLC command is used, the default includes all records in the input file. The SM$SLC command may be used in combination with the SM$IN and SM$REF commands to create different selections patterns for different record types.
SM$REF	Builds the reformatting control statement(s). SM$REF commands must be present since they define the sort control fields (keys) and data fields. SM$REF commands may be used in combination with SM$IN and SM$SLC commands to create different selection patterns for different record types or to rearrange the order of the data fields within specified records.
SM$CLS	Terminates after the file creation procedure. This must be the last command of the file creation section of a batch stream.
XBSM	Executes Sort/Merge using the specified batch file name and listing device.
EBATCH	Indicates the end of the batch control stream. This statement is required.

3.3 FIRST SORT EXAMPLE

This example sorts the 80-character records from the file DS01.INFIL in the control field's ascending order (columns 5 through 10) and places the output in file DS01.OUTFIL. The batch control stream is as follows:

```
1.  BATCH
2.  SM$SMC CFN = DS01.CONTRL
3.  SM$HD TCL = 6,ORL = 80,MS = 16000
4.  SM$OUT FP = DS01.OUTFIL,LRL = 80,PRL = 576
5.  SM$IN FP = DS01.INFIL
6.  SM$REF FT = N,BL = 5,EL = 10
7.  SM$REF FT = D,BL = 1,EL = 80
8.  SM$CLS
9.  XBSM CFN = DS01.CONTRL,LDN = LP01
10. EBATCH
```

The following paragraphs describe each of the command statements in detail.

3.3.1 SM$SMC Statement

```
SM$SMC CFN = DS01.CONTRL
```

The second line specifies that a Sort/Merge control file is to be created with the pathname DS01.CONTRL. If a file with this pathname already exists, Sort/Merge replaces the contents of that file with the newly created control file. The execution of the batch control stream generates seven Sort/Merge control statements and places them in this file. This file is not the Sort/Merge input or output file. Sort/Merge dynamically allocates the control file if it has not already been created. Sort/Merge does not delete this file at the end of the sort. For additional Sort/Merge runs, use this file with the Execute Batch Sort/Merge (XBSM) command.

This line uses the following keyword:

- CFN (Control File Name) — Specifies the name of the file that will contain the command statements for this sort. It may be a maximum of 36 characters.

3.3.2 SM$HD Statement

```
SM$HD TCL = 6,ORL = 80,MS = 16000
```

The third line uses many of the default options for the header command statement. A full sort (tag-along sort) in ascending order is the default.

This line uses the following keywords:

- TCL (Total Control Length) — Specifies the total length, in characters, of all control fields. In this example, the control field is in columns 5 through 10, which is a length of 6 characters. (This parameter has no default.)

- ORL (Output Record Length) — Specifies the total length, in characters, of the output record. Since 80-character records are being sorted, the length of the output records is 80 characters. (This parameter has no default.)

- MS (Memory Size) — Specifies the work memory available for the sort, in bytes. Specify a memory size in the range of 3,000 to 45,000 bytes, or use the default. A size of 16,000 bytes is only a suggested size to allow the sort to run faster than when the default value (2,800 bytes) is used. If 32,000 bytes are available, specify that amount.

Additional information on the Header Specification is in Section 4.

3.3.3 SM$OUT Statement

```
SM$OUT FP = DS01.OUTFIL,LRL = 80,PRL = 576
```

The fourth line specifies the output file pathname and attributes. If the file has been previously created, only the file pathname is required. This example assumes that the file has not been created and gives attributes to allow it to be created. If file attributes are given and the file was previously created, the attributes assigned at creation are used, and attributes specified in the SM$OUT statement become commentary.

This line uses the following keywords:

- FP (File Pathname) — Specifies the pathname of the file (or the name of the device) that is to contain the output data. The maximum is 36 characters.

- LRL (Logical Record Length) — Specifies a length of 80 characters for each logical record output.

- PRL (Physical Record Length) — Specifies that the physical records written to the output file are to be 576 characters in length. Each physical record contains more than one logical record.

Additional information on the Output File Specification is in Section 5.

3.3.4 SM$IN Statement

```
SM$IN FP = DS01.INFIL
```

The fifth line specifies the pathname for the input file (or device that contains the input records). Since the attributes of the input file are available to Sort/Merge, none are specified. Also, since this is a sort operation, only one file is specified.

This line uses the following keyword:

- FP (File Pathname) — Specifies the pathname of the file (or the name of the device) that contains the input file. The maximum is 36 characters.

Additional information on the Input File Specification is in Section 5.

3.3.5 SM$REF Statements

This example includes two reformatting command statements. The first one specifies the control field by which the records are to be ordered. The second one defines the data field. The use of more than one control field and more than one data field is explained in Section 7.

3.3.5.1 Control Field Specification.

```
SM$REF FT=N,BL=5,EL=10
```

The sixth line specifies the location within the record of the field to be used as the control field that determines the order of the records during the sort. The default character selection is a full, ASCII-character sort.

NOTE

All control fields must be specified before data fields are specified.

This line uses the following keywords:

- FT (Field Type) — Specifies the type of control field: N specifies normal sequence, and O specifies an opposite sequence. A parameter in the header control statement specifies the order of the sort operation as either ascending or descending. In this example, since no parameter is specified, the default (ascending sequence) is assumed. In this case, N means ascending order.

- BL (Beginning Location) — Specifies the high-order column of the control field. Since the control field is columns 5 through 10, the BL is 5.

- EL (Ending Location) — Specifies the low-order column of the control field. Since the control field is 5 through 10, the EL is 10.

You can specify additional control fields before specifying data fields. The control field specified first is the primary control field, and all records are sorted in order according to the primary control field. If a second control field is specified, all records that have identical primary control fields are sorted according to the secondary control field, and so on. Refer to Section 7 for additional information on using multiple control fields.

3.3.5.2 Data Field Specification.

```
SM$REF FT=D,BL=1,EL=80
```

The seventh line specifies the location of the data field. Since the data is 80 characters long, the field specifies that 80 characters of data are to be written to the output file for each record in the input file.

This line uses the following keywords:

- FT (Field Type) — Specifies that this is a data field (D).

- BL (Beginning Location) — Specifies that the field begins in column 1.
- EL (Ending Location) — Specifies that the field ends in column 80.

Multiple data fields can also be used to reformat records (Section 7).

3.3.6 SM$CLS Statement

```
SM$CLS
```

The eighth line terminates generation of the control file.

3.3.7 XBSM Statement

```
XBSM CFN = DS01.CONTRL,LDN = LP01
```

The ninth line executes the Sort/Merge program by using the control file that the previous statements built.

This line uses the following keywords:

- Control File Name) — Specifies the pathname of the file (or device that contains the control file. In this example, the control file DS01.CONTRL has been built by the previous seven statements.
- LDN (Listing Device Name) — Specifies the pathname of the file (or device) onto which Sort/Merge information, error, and warning messages are to be written. The line printer (LP01) is used as an example. (To avoid interrupting the printing operation of another user on your system, you can write this listing to a file.)

When the sort is complete, the EBATCH statement terminates the batch stream. Note that no work file was specified in this batch stream. The work file for this sort was defaulted to the system disk DS01 and was automatically released at program completion. You can use multiple XBSM statements if the control files have been previously built.

3.4 SECOND SORT EXAMPLE

This example shows a sort with record selection. It sorts (tag-along sort) 80-character records from the input file DS01.INFILE and places the output in DS03.OUTFIX. Assume that DS03.OUTFIX has been previously created; it is not necessary to specify logical or physical record length. Also, only records with a value greater than 100 in columns 5 through 10 are to be included in the sort. These records will be selected and will appear in the output file. The primary control field (columns 5 through 10) specifies ascending order. A second control field (columns 12 through 15) is specified with descending order. The total length of the combined control fields is 10.

This sort demonstrates the optimal positioning of the output file, work file, and input file for maximum sort speed. The input and output files can be on the same disk volume, but the work file should be on a different volume.

The batch control stream statements are as follows:

```
1.  BATCH
2.  SM$SMC CFN = DS01.CONTRLA
3.  SM$HD TCL = 10,PO = 0,ORL = 80,MS = 16000
4.  SM$OUT FP = DS03.OUTFIX
5.  SM$WKF WFV = DS02
6.  SM$IN FP = DS01.INFILE
7.  SM$SLC CP = C,O1B = 5,O1E = 10,REL = GT,FOC = C,O2C = 000100
8.  SM$REF FT = N,BL = 5,EL = 10
9.  SM$REF FT = O,BL = 12,EL = 15
10. SM$REF FT = D,BL = 1,EL = 80
11. SM$CLS
12. XBSM CFN = DS01.CONTRLA,LDN = DS01.LIST
13. EBATCH
```

3.4.1 SM$SMC Statement

```
SM$SMC CFN = DS01.CONTRLA
```

The second line specifies that the Sort/Merge control file be created and placed in a file named DS01.CONTRLA.

3.4.2 SM$HD Statement

```
SM$HD TCL = 10,PO = 0,ORL = 80,MS = 16000
```

The third line specifies by default a full sort in ascending order.

This line uses the following keywords:

- TCL (Total Control Length) — Specifies the total length, in characters, of all control fields. In this example, an ascending field is in columns 5 through 10 and a descending field is in columns 12 through 15. This gives a total length of 10 characters. (This parameter has no default.)

- PO (Print Option) — Specifies that the generated control statements be listed along with information, error, and warning messages.

- ORL (Output Record Length) — Specifies the total length, in characters, of the output record. Since 80-character records are being sorted, the length of the output records is 80 characters. (This parameter has no default.)

- MS (Memory Size) — Specifies the work memory available for the sort, in bytes. A size of 16,000 bytes is only a suggested size to allow the sort to run faster than when the default value (2,800 bytes) is used. If 32,000 bytes are available, you should specify that amount.

3.4.3 SM$OUT Statement

```
SM$OUT FP = DS03.OUTFIX
```

The fourth line specifies the output file pathname. No file attributes are specified since, in this example, the file has been previously created on DS03. If not previously created, a sequential file will be created by Sort/Merge with 80-character records unblocked.

This line uses the following keyword:

- FP (File Pathname) — Specifies the pathname of the file (or device) that is to contain the output listing. The maximum is 36 characters.

3.4.4 SM$WKF Statement

```
SM$WKF WFV = DS02
```

The fifth line specifies that the work file be allocated on disk DS02. The work file is allocated as a temporary file. For maximum efficiency, always allocate the work file on a volume that contains neither the input nor the output file.

This line uses the following keyword:

- WFV (Work File Volume) — Defines the device or volume to which the work file is to be allocated.

3.4.5 SM$IN Statement

```
SM$IN FP = DS01.INFILE
```

The sixth line defines the pathname for the input file (or device). Since the attributes assigned to the file when created are available to Sort/Merge, no attributes are specified.

This line uses the following keyword:

- FP (File Pathname) — Specifies the pathname of the file (or device) that contains the input file. The maximum is 36 characters.

3.4.6 SM$SLC Statement

```
SM$SLC CP = C,O1B = 5,O1E = 10,REL = GT,FOC = C,O2C = 000100
```

The seventh line specifies selection of certain records to be included in the sort. Records that meet the criteria are sorted and appear in the output file. All records that do not meet the criteria are omitted. Omitted records are not sorted and do not appear in the output file. In this example, all records with a value greater than 100 in columns 5 through 10 are included.

This line uses the following keywords:

- CP (Character Portion) — Specifies an ASCII character comparison.

- O1B (Operand 1 Beginning) — Specifies the high-order, leftmost column of the comparison field. In this example, columns 5 through 10 have an O1B of 5.

- O1E (Operand 1 Ending) — Specifies the low-order, rightmost column of the comparison field. In this example, columns 5 through 10 have an O1E of 10.

- REL (Relationship) — Specifies the type of comparison, as follows:

  G T — Greater than
  L T — Less than
  E Q — Equal
  N E — Not equal
  L E — Less than or equal
  G E — Greater than or equal

- FOC (Field or Constant) — Specifies if the comparison is to another field within the record or to a constant. The C specifies constant. (See Section 6 for additional information.)

- O2C (Operand 2 Constant) — Specifies the constant to be used in the comparison. Since a number must be specified for each position within the field, leading zeros are required. In this example, the constant 000100 is to be compared with the value in columns 5 through 10.

3.4.7 SM$REF Statements

The following paragraphs explain the reformatting specifications in this example.

3.4.7.1 Control Field Specification.

```
SM$REF FT = N,BL = 5,EL = 10
SM$REF FT = O,BL = 12,EL = 15
```

The eighth and ninth lines specify the control fields. Since columns 5 through 10 are specified first, they constitute the primary control field; columns 12 through 15 are the secondary field.

This line uses the following keywords:

- FT (Field Type) — An N specifies normal control field or ascending sequence in this example. An O specifies opposite control field or descending sequence.

- BL (Beginning Location) — Specifies the beginning column position of the control field in the input file.

- EL (Ending Location) — Specifies the ending column position of the control field in the input file.

3.4.7.2 Data Field Specification.

```
SM$REF FT = D,BL = 1,EL = 80
```

The tenth line specifies the data field.

This line uses the following keywords:

- FT (Field Type) — The D specifies that this is the data field to be written to the output file for each sorted record. Since the order of the fields within the record remains the same as in the input file, the entire data field (columns 1 through 80) is specified.

- BL (Beginning Location) — Specifies the beginning column position of the data field in the input file.

- EL (Ending Location) — Specifies the ending column position of the data field in the input file.

3.4.8 SM$CLS Statement

```
SM$CLS
```

The eleventh line terminates generation of the control statement file.

3.4.9 XBSM Statement

```
XBSM CFN = DS01.CONTRLA,LDN = DS01.LIST
```

The XBSM statement executes the Sort/Merge program using the control statement file that the previous statements built.

This line uses the following keywords:

- CFN (Control File Name) — Assigns the pathname DS01.CONTROLA to the control file created by this batch stream.

- LDN (Listing Device Name) — Specifies the pathname of the device or file to which Sort/Merge is to list the information, error, and warning messages produced during the sort process. In this example, the file name is DS01.LIST. If this file already exists, Sort/Merge replaces the contents of the current file with the messages listing. If the file does not exist, Sort/Merge creates a sequential file to contain the messages listing. Delete this file after checking it for errors.

3.5 MERGE EXAMPLE

The following control statements specify that three input files are to be merged into one output file in descending order, using a control field contained in columns 5 through 10 of the input files. All three input files have been previously sorted in descending order, using the same control field.

```
1. BATCH
2. SM$SMC CFN = DS01.CONTRLM
3. SM$HD SMT = MERGE,SMS = D,TCL = 10,ORL = 80,MS = 16000
4. SM$OUT FP = DS01.MRGOUT
5. SM$IN FP = DS01.MRG1
6. SM$IN FP = DS01.MRG2
```

```
7.  SM$IN FP = DS01.MRG3
8.  SM$REF FT = N,BL = 5,EL = 10
9.  SM$REF FT = D,BL = 1,EL = 80
10. SM$CLS
11. XBSM CFN = DS01.CONTRLM,LDN = LP01
12. EBATCH
```

3.5.1 SM$SMC Statement

```
SM$SMC CFN = DS01.CONTRLM
```

The second line specifies that the Sort/Merge control statement file be placed in a file with the pathname DS01.CONTRLM. Sort/Merge creates a sequential file with this pathname if the file does not already exist. If the file does exist, Sort/Merge replaces the existing contents of the file with the newly created control statement file.

3.5.2 SM$HD Statement

```
SM$HD SMT = MERGE,SMS = D,TCL = 6,ORL = 80,MS = 16000
```

The third line specifies a merge operation in descending order.

This line uses the following keywords:

- SMT (Sort/Merge Type) — Specifies that a merge process is to be performed.
- SMS (Sort/Merge Sequence) — The D specifies that the normal ordering sequence is to be descending. (An A would specify ascending.)
- TCL (Total Control Length) — Specifies the total length, in characters, of all control fields. In this example, only one control field, columns 5 through 10, is used. This field has a length of 6 characters. (This parameter has no default.)
- ORL (Output Record Length) — Specifies the total length, in characters, of the output record. Since 80-character records are being merged, the length of the output records is 80 characters. (This parameter has no default.)
- MS (Memory Size) — Specifies the work memory available for the merge, in bytes. A size of 16,000 bytes is only a suggested size to allow the merge to run faster than when the default value (2,800 bytes) is used. If 32,000 bytes are available, you should specify that amount.

3.5.3 SM$OUT Statement

```
SM$OUT FP = DS01.MRGOUT
```

The fourth line is the pathname of the file (or device) that is to contain the merged output created by Sort/Merge. If the file does not exist, Sort/Merge creates a sequential file with this pathname.

This line uses the following keyword:

- FP (File Pathname) — Specifies the pathname of the file (or the name of the device) that is to contain the output listing. The maximum is 36 characters.

Note that no work file is specified in this command stream since a merge operation does not use a work file.

3.5.4 SM$IN Statement

```
SM$IN FP = DS01.MRG1
SM$IN FP = DS01.MRG2
SM$IN FP = DS01.MRG3
```

The fifth through seventh lines specify the pathnames of the three input files. The files must be in the same sequence and must use the same control field. Any out-of-sequence condition may terminate the merge process or cause unpredictable results.

These lines use the following keyword:

- FP (File Pathname) — Specifies the pathname of the file (or the name of the device) that contains the input file. The maximum is 36 characters.

3.5.5 SM$REF Statements

The following paragraphs explain the reformatting specification statements.

3.5.5.1 Control Field Specification.

```
SM$REF FT = N,BL = 5,EL = 10
```

The eighth line specifies the order and location of the control field.

This line uses the following keywords:

- FT (Field Type) — Specifies that this is a control field ordered in normal (N) order. Since the header specification designated the sequence as descending, normal means descending.

- BL (Beginning Location) — Specifies that column 5 in the input files is the beginning location of the control field.

- EL (Ending Location) — Specifies that column 10 in the input files is the ending location of the control field.

3.5.5.2 Data Field Specification.

```
SM$REF FT = D,BL = 1,EL = 80
```

The ninth line specifies that this is a data field.

This line uses the following keywords:

- FT (Field Type) — The D specifies that this is a data field.
- BL (Beginning Location) — Specifies column 1 in the input files as the beginning column of the data field.
- EL (Ending Location) — Specifies column 80 in the input files as the ending column of the data field.

3.5.6 SM$CLS Statement

```
SM$CLS
```

The tenth line terminates generation of the control statement file.

3.5.7 XBSM Statement

```
XBSM CFN = DS01.CONTRLM,LDN = LP01
```

The eleventh line executes the Sort/Merge program by using the control statement file that the previous statements built.

This line uses the following keywords:

- CFN (Control File Name) — The name of the file (or device) that is to contain the control file built by the previous command statements.
- LDN (Listing Device Name) — The pathname of the file (or device) to which Sort/Merge information, error, and warning messages are to be written. The line printer (LP01) is used as an example.

3.5.8 Executing Sort/Merge in Batch Mode

When you have prepared the batch control stream file by copying the format of the examples in this section, use the Execute Batch (XB) command to execute the batch control statements that will code the control file which will run the Sort/Merge utility. Both the input file (or files) and the batch control stream file must have been previously prepared before you execute this command.

The XB command has the following field prompts:

```
EXECUTE BATCH
    INPUT ACCESS NAME:  <pathname>
  LISTING ACCESS NAME:  <pathname>
```

In response to the INPUT ACCESS NAME prompt, enter the pathname you assigned to the file (or device) that contains the batch control stream. This is not the pathname used in the batch control stream to specify the control file pathname (SM$SMC).

In response to the LISTING ACCESS NAME prompt, enter the pathname of a file (or device) to which the listing made by the SCI during the batch processing may be written. This is not the same as the listing of Sort/Merge messages specified for the XBSM command statement in your batch control stream.

To reuse the control file built with the batch control stream, execute Sort/Merge in batch by using the Execute Batch Sort/Merge (XBSM) command.

The XBSM command has the following field prompts:

```
EXECUTE BATCH SORT/MERGE   <VERSION X.X.X>
        CONTROL FILE NAME: <pathname>
   LISTING DEVICE NAME:    <pathname>
```

In response to the CONTROL FILE NAME prompt, enter the pathname of the control file, that is, the pathname specified for SM$SMC.

In response to the LISTING DEVICE NAME prompt, enter the pathname of a file (or device) to which the listing made by the SCI during the batch processing may be written. This is not the same as the listing of Sort/Merge messages specified for the XBSM command statement in your batch control stream.

You can also specify the XBSM command when in the SCI command mode by entering the following:

```
XBSM  CFN = <pathname>,LDN = <pathname>
```

The file name following CFN is the pathname you specified in the batch stream as the control file name.

3.6 BEGINNING EXERCISE

Use the steps described in the following paragraph to create an input file named DS01.TESTIN and a batch control stream file named DS01.TEST. Check with the person in charge of your system to confirm that these pathnames are acceptable. If not, obtain another character string to replace the DS01 component of the pathname in all occurrences for the following exercise.

The input file contains the records to be sorted or merged. The batch control stream file, DS01.TEST, contains all of the control statements and keywords needed to perform any of the sort or merge examples presented in this section. Some of the control statements are not required; however, they are included so that you can use this file to experiment with different types of sorts and merges by modifying the keyword parameters.

NOTE

The following procedures assume you have a 911 VDT. If you are using any other type of input device, use the comparable function keys for that terminal.

3.6.1 Create an Input File

The input file consists of records (in this case, 80-character lines) containing data fields. To facilitate moving from one field to another, use the Modify Tabs (MT) command to set tab stops. Then, use the Execute Editor with Scaling (XES) command to create the file. Use the following procedure:

1. Enter SCI command mode by pressing the CMD key.
2. Specify the Modify Tabs command by entering MT, and press the RETURN key.
3. In response to the TAB COLUMNS prompt, specify tab stops at 1, 10, 20, and 30 as follows and then press RETURN:

```
MODIFY TABS
        TAB COLUMNS:  1,10,20,30
```

4. Press the CMD key again, enter XES, and press RETURN.
5. In response to the FILE ACCESS NAME prompt, press RETURN.
6. When the end-of-file (EOF) marker appears, enter the data, using the New Line (blank gray) key or Compose Mode (F7) key as necessary.
7. Type the following, using the TAB key and pressing the RETURN key at the end of each line. To be sorted properly, the input in each data field should be typed in the same manner. As an experiment, type the name Don Baker with both uppercase and lowercase characters (as shown in line 5). Place last names in columns 10 through 19; first names in columns 20 through 29; and numbers in columns 30 through 39, right justified. Although this particular column designation is arbitrary, it is easy to remember and gives you extra space to experiment by entering data fields of your own.

```
BAKER          DON                200
SMITH          SAM                 50
WOOD           DOTTY              100
BAKER          HANK                 2
Baker          Don                100
SMITH          SAM                100
....1 .....2 .....3 .....4 .....5 .....6...
```

8. Align the number column and proofread the input, correcting any errors.
9. To quit the text edit mode, press the CMD key, type QE, and press RETURN.
10. Enter a response of NO to the ABORT? prompt by pressing RETURN.
11. Specify the pathname DS01.IN in response to the OUTPUT FILE ACCESS NAME prompt.
12. Respond NO to the REPLACE? prompt.
13. Enter a null response to the MOD LIST ACCESS NAME prompt by pressing RETURN.

3.6.2 Create a Batch Control Stream File

Creating a batch control stream file also requires the use of the Text Editor. The control file that this batch control stream builds is named DS01.TESTCTR. This is to be a tag-along sort with three control fields specified. The primary field contains the last names in the input file, and the secondary field contains the first names in the file. Both fields are sorted in ascending order. The third control field, the numbers column in the input file, is sorted in descending order (highest number listed first) if the person has more than one record.

To create a batch control stream file, use the following procedure:

1. Enter SCI command mode by pressing the CMD key.
2. Specify the Execute Editor with Scaling command by typing XES and pressing RETURN.
3. In response to the FILE ACCESS NAME prompt, press the RETURN key.
4. When the EOF marker appears, type the following batch control stream, using the New Line (blank gray) key or Compose Mode (F7) key as necessary:

```
BATCH
SM$SMC CFN = DS01.TESTCTR
SM$HD SMT = SORTR,SMS = A,TCL = 30,ORL = 80,MS = 16000
SM$OUT FP = DS01.TESTOUT,LRL = 80,PRL = 576
SM$WKF WFV = DS01
SM$IN FP = DS01.TESTIN
SM$REF FT = N,BL = 10,EL = 19
SM$REF FT = N,BL = 20,EL = 29
SM$REF FT = O,BL = 30,EL = 39
SM$REF FT = D,BL = 1,EL = 80
SM$CLS
XBSM CFN = DS01.TESTCTR,LDN = DS01.TESTLIST
EBATCH
```

5. Proofread your input and correct any errors.
6. To quit the text edit mode, press the CMD key, type QE, and press RETURN.
7. Enter a response of NO to the ABORT? prompt by pressing RETURN.
8. Specify the pathname DS01.TEST in response to the OUTPUT FILE ACCESS name prompt.
9. Respond NO to the REPLACE? prompt.
10. Enter a null response to the MOD LIST ACCESS NAME prompt by pressing RETURN.

3.6.3 Execute the Batch Control Stream

Using the following procedure, enter the Execute Batch (XB) command to execute the batch control statements that will run the Sort/Merge utility.

1. Enter the SCI command mode by pressing the CMD key.
2. Enter XB and press RETURN.
3. Respond to the XB command prompts as follows and press RETURN.

```
EXECUTE BATCH
INPUT ACCESS NAME:    DS01.TEST
LISTING ACCESS NAME:  DS01.TEMPLIST
```

4. Since this is a quick sort operation, press the CMD key, type WAIT, press RETURN, and wait for the background task to complete.

3.6.4 Examine the Output

When Sort/Merge has completed processing, use the Show File (SF) command or Print File (PF) command to examine the following files:

- DS01.TEST — Contains the reusable batch control stream. To use this file for all of your sort or merge operations, modify the input parameters and the pathnames to fit each example.
- DS01.TESTCTR — Contains the Sort/Merge control file that the batch control stream built. This file is reusable.
- DS01.TESTIN — Contains the data input. This file is reusable.
- DS01.TESTOUT — Contains the sorted records from the input file. This file is reusable.
- DS01.TESTLIST — Contains the information, error, and warning messages listed by Sort/Merge. Delete this file after confirming the output file information, or continue to use this same pathname for the Sort/Merge message listings each time you perform a sort or merge operation.
- DS01.TEMPLIST — Contains the complete listing of the batch process and should be examined to confirm that the sort operation completed successfully. This file contains information about any errors made in preparing the batch control stream file. Delete this file after examining the information, or continue to use this same pathname.

The DS01.TEST and DS01.TESTIN files should match the input given in the preceding paragraphs.

3.6.4.1 Sort/Merge Control Statement File. The DS01.TESTCTR file should contain the following:

```
00000HSORTR     30A          1X   80   16000
00010DDSDS01. TESTOUT
00020DA   80 576
00030DISDS01. TESTIN
00040DA
00050FNC   10   19
00060FNC   20   29
00070FDC   30   39
00080FDC    1   80
```

3.6.4.2 Sorted Output File. The DS01.TESTOUT file should contain the sorted records, as follows:

```
BAKER       DON        200
BAKER       HANK         2
Baker       Don        100
SMITH       SAM        100
SMITH       SAM         50
WOOD        DOTTY      100
```

Notice that the line typed in initial capital letters was sorted at the end of the records beginning with B. Also, notice that HANK BAKER was inserted after DON BAKER, as specified by the secondary control field, and that the figures following SAM SMITH are in descending order, as specified by the third control field.

3.6.4.3 Sort/Merge Messages Listing. The DS01.TESTLIST file contains the Sort/Merge messages and should be similar to the following:

```
INFO    4:    END OF GENERATION PHASE

INFO    3:    NUMBER OF SPECIFICATIONS=      9

INFO    0:    NO ERRORS

INFO   23:    BYTES OF WORKING STORAGE AVAILABLE= 16000

INFO   24:    BYTES REQUIRED FOR SPECIFICATIONS =   352

INFO   14:    INPUT FILE RECORD SIZE=     80

INFO   15:    INPUT FILE BLOCK SIZE =    864
```

```
INFO  16:   OUTPUT FILE RECORD SIZE =     80

INFO  17:   OUTPUT FILE BLOCK SIZE=    576

INFO   5:   SORT EXECUTION HAS BEGUN

INFO  18:   WORK FILE RECORD SIZE =    112

INFO   9:   NUMBER OF INPUT FILE RECORDS READ =     10

INFO  10:   NUMBER OF INPUT FILE RECORDS SELECTED =     10

INFO  30:   NUMBER OF WORK FILE UNITS  USED =      1

INFO  29:   NUMBER OF PASSES REMAINING=      0

INFO  25:   SORT FINAL PASS HAS BEGUN

INFO  26:   SORT FINAL PASS COMPLETED

INFO  13:   NUMBER OF OUTPUT RECORDS WRITTEN=     10
```

3.6.4.4 SCI Batch Listing. The DS01.TEMPLIST created during the batch operation should contain the following. If an error occurred, examine both DS01.TESTLIST and DS01.TEMPLIST to determine the cause. Correct the error in the batch control stream file (DS01.TEST), and execute it again.

```
** SCI990 ** SCI990 ** SCI990 ** SCI990 **BATCH
 <0001> BATCH
LS (LIST SYNONYMS) ?            NO
BATCH LISTING ACCESS NAME       DS01.TEMPLIST
BATCH INPUT ACCESS NAME         DS01.TEST
STATION ID                      ST15
USER ID                         CK0011
2:13:15 FRIDAY, SEP 19, 1980.
 <0002> SM$SMC CFN=DS01.TESTCTR
CONTROL FILE NAME               DS01.TESTCTR
 <0003> SM$HD  SMT=SORTR,SMS=A,TCL=30,ORL=80,MS=16000
SORT MERGE TYPE                 SORTR
TOTAL CONTROL LENGTH            30
SORT MERGE SEQUENCE             A
OUTPUT RECORD LENGTH            80
MEMORY SIZE                     16000
VERIFY OPTION                   ** NULL **
VARIABLE LENGTH RECORDS         ** NULL **
DROP CONTROL FIELDS             X
PRINT OPTION                    1
ALTERNATE SEQUENCE              ** NULL **
 <0004> SM$OUT FP=DS01.TESTOUT,LRL=80,PRL=576
FILE PATHNAME                   DS01.TESTOUT
LOGICAL RECORD LENGTH           80
PHYSICAL RECORD LENGTH          576
```

```
NUMBER OF LOGICAL RECORDS     ** NULL **
FILE TYPE                     S
 <0005> SM$IN FP=DS01.TESTIN
FILE PATHNAME                 DS01.TESTIN
NUMBER OF LOGICAL RECORDS     ** NULL **
LOGICAL RECORD LENGTH         ** NULL **
FILE TYPE                     S
 <0006> SM$REF FT=N,BL=10,EL=19
FIELD TYPE                    N
BEGINNING LOCATION            10
ENDING LOCATION               19
OVERFLOW                      ** NULL **
CONTINUATION                  ** NULL **
SUBSTITUTE CHARACTER          ** NULL **
RECORD CHARACTER              ** NULL **
CHARACTER PORTION             C
 <0007> SM$REF FT=N,BL=20,EL=29
FIELD TYPE                    N
BEGINNING LOCATION            20
ENDING LOCATION               29
OVERFLOW                      ** NULL **
CONTINUATION                  ** NULL **
SUBSTITUTE CHARACTER          ** NULL **
RECORD CHARACTER              ** NULL **
CHARACTER PORTION             C
 <0008> SM$REF FT=D,BL=30,EL=39
FIELD TYPE                    D
BEGINNING LOCATION            30
ENDING LOCATION               39
OVERFLOW                      ** NULL **
CONTINUATION                  ** NULL **
SUBSTITUTE CHARACTER          ** NULL **
RECORD CHARACTER              ** NULL **
CHARACTER PORTION             C
 <0009> SM$REF FT=D,BL=1,EL=80
FIELD TYPE                    D
** SCI990 ** SCI990 ** SCI990 ** SCI990 **BATCH
BEGINNING LOCATION            1
ENDING LOCATION               80
OVERFLOW                      ** NULL **
CONTINUATION                  ** NULL **
SUBSTITUTE CHARACTER          ** NULL **
RECORD CHARACTER              ** NULL **
CHARACTER PORTION             C
 <0010> SM$CLS
 <0011> XBSM CFN=DS01.TESTCTR,LDN=DS01.TESTLIST
CONTROL FILE NAME             DS01.TESTCTR
LISTING DEVICE NAME           DS01.TESTLIST
 SORT/MERGE COMPLETED NORMALLY
 <0012> EBATCH
CODE                          ** NULL **
TEXT                          ** NULL **
LS (LIST SYNONYMS) ?          NO
```

3.6.5 Perform a Merge Operation

Create a second input file with pathname DS01.TESTIN2 and sort the file using the same control fields as in the previous exercise. In the batch control stream file, change the input file and output file pathnames. When the sort process is completed, merge the two sorted output files. Use the following input for DS01.TESTIN2 or create your own list. Since all of the numbers in this list end in 2, you can easily identify them when the lists are merged; consequently, you can determine the priority assigned to the first input file.

BAKER	DON	52
MURPHY	JAN	52
ACKER	DIXIE	72
STEWART	KIM	102
PALMER	DOUGLAS	12
JONES	EVAN	122
PACE	ELAINE	62
BAKER	DON	202

3.6.6 Delete the Exercise Files

When you have finished the exercises, delete the created files by using the Delete File (DF) command.

3.7 SUMMARY

The control statements in this section are models that can be adapted by the one-time user to fit specific applications. As a result, many of the features of Sort/Merge have not been shown in any of the examples. The remaining sections of this manual provide details about the additional features of Sort/Merge.

4

Header Specification

4.1 GENERAL

The Header Specification provides Sort/Merge with the following information:

- Type of sort (address only, tag-along, or summary tag-along)
- Type of merge (normal or summary)
- Primary collating sequence (ascending or descending)
- Sequence of the sorted file (standard ASCII or alternate)
- Type of execution information to be printed, if any (for example, diagnostic messages and status data)
- Whether to keep or drop the control fields in the output file
- Whether to verify the data sent to the work file

NOTE

Use only one Header Specification for each sort or merge.

Figure 4-1 shows the column headings on the Header Specification sheet; Table 4-1 lists the options that can be entered for these parameters.

SORT/MERGE SPECIFICATIONS, SHEET A

Page 1 2: 0 0

Header Specification

Line Number	Spec. Type	Merge: MERGE, MERGES / Sort: SORTA, SORTR, SORTRS	Largest Total of Control Fields of Any Record Type (Bytes)	Sequence (A/D)	Reserved	Alternate Coll. Seq. (S)	Print Option	Drop Ctrl. Flds. (X)	Output Record Length	Variable Lngth. Recs.	Verify Option (V)	Memory Work-Space	Comments
3 4 5	6	7 8 9 10 11 12	13 14 15 16 17	18	19 20 21 22 23 24 25	26	27	28	29 30 31 32	33	34	35 36 37 38 39	(Columns 40 to 72)
0 0 0	H												

135623

Figure 4-1. Header Specification Format

Table 4-1. Header Specification Column Summary

Suggested Batch Keywords	Columns	Entries	Explanation (Interactive Prompts Capitalized)	Paragraph Number
—	1-2[1]	00	Page number	4.2.1
—	3-5[1]	000	Header line number	4.2.1
—	6[1]	H	Header Specification line	4.2.2
SMT	7-12[1]		Sort/Merge type:	4.2.3
		SORTR[2]	— Tag-along sort	
		SORTA	— Address-only sort	
		SORTRS	— Summary tag-along sort	
		MERGE	— Normal merge	
		MERGES	— Summary merge	
TCL	13-17[1]	1-256	Total control length: length (in characters) of the longest control field used to sort the characters; numeric, right-justified.	4.2.4
SMS	18[1]		Sort/Merge sequence:	4.2.5
		A[2]	— Sorted records in output file to be in ascending sequence	
		D	— Sorted records in output file to be in descending sequence	
—	19-25	Blank	Reserved	
AS	26		Alternate sequence:	4.2.6
		Blank[2]	— Use standard ASCII code collating sequence	
		S	— Use an alternate collating sequence. ALTSEQ statement defines the sequence	
PO	27		Print option:	4.2.7
		0 or blank	— Specification lines, diagnostic messages, and program-status messages	
		1[2]	— Program-status messages	
		2	— Reserved	
		3	— No messages	
		4	— Special use as debug aid (see Section 9)	
DCF	28		Drop control fields:	4.2.8
		Blank	— Keep control fields in tag-along output records	
		X[2]	— Drop control fields in tag-along output records	
ORL	29-32[3]	1-4096	Output record length: length (in characters) of the longest output record in tag-along sort	4.2.9

Table 4-1. Header Specification Column Summary (Continued)

Suggested Batch Keywords	Columns	Entries	Explanation (Interactive Prompts Capitalized)	Paragraph Number
VAR	33		Variable length records:	4.2.10
		Blank[2]	— Fixed length output records	
		V	— Variable length output records	
VO	34		Verify option:	4.2.11
		Blank[2]	— Do not verify data sent to work file	
		V	— Verify data sent to work file	
MS	35-39		Memory size (workspace)	4.2.12
		Blank[2]	— Reserve 2800 bytes for workspace	
		3000-45000	— Number of bytes to reserve for workspace	
—	40-80		Comment field; enter any ASCII characters.	4.2.13

Notes:

[1] These columns must be filled.

[2] Default value.

[3] These columns must be filled for a tag-along sort or merge.

4.2 COLUMN DESCRIPTION

The following paragraphs explain each of the column entries for the Header Specification.

4.2.1 Page Number and Line Number (Columns 1 through 5)

Columns 1 through 5 comprise a five-digit number that ensures that Sort/Merge receives specification lines in the correct order. This order follows two rules:

- Specification Sheet A lines are entered before Sheet B lines
- Lines are entered as shown on the sheets, top to bottom

This means that the Header Specification will be the first line entered and will be on page 00 and line 000. All File Description Specifications follow the Header Specification with line numbers in ascending order as printed on Sheet A. Note that the Record Selection and Reformatting Specifications have higher line numbers preprinted in columns 3 and 4 so that Sheet B lines will follow Sheet A in the prescribed ascending order. If an additional Sheet B is needed, its page number must be a higher number to maintain the proper sequence. Reformatting Specifications always follow the Record Selection Specifications, even though the latter may require more than one sheet.

Column 5 is blank to allow you to insert data between two lines that are already filled in. For example, to include two lines of additional data between lines 28x and 29x (by default these are 280 and 290), write the data on two unused lines with numbers between 280 and 290 (such as, 281 and 282). These lines should be entered after line 280, as shown in Figure 4-2.

NOTE

You should fill in column 5 with a zero when using the default value.

4.2.2 Specification Type (Column 6)

Column 6 contains a preprinted H to identify this as the header line.

4.2.3 SMT — Sort/Merge Type [SORTR] (Columns 7 through 12)

Columns 7 through 12 indicate the type of Sort/Merge, as follows:

Columns 7 through 12 Entry	Explanation
SORTA	Address-only sort
SORTA	Tag-along sort
SORTRS	Summary tag-along sort
MERGE	Normal merge
MERGES	Summary merge

Entries are left justified (that is, they must begin in column 7).

4.2.4 TCL — Total Control Length (Columns 13 through 17)

Use the following to determine the appropriate entry for columns 13 through 17:

1. Sum the lengths (in characters) of the control fields (denoted by an N, O, or F entry in column 7 of the Reformatting Specification) for each record type.

2. Enter the largest sum (that is, the largest sum of the control field lengths) in these columns. The sum can range from 1 to 256 characters. (For more information, see paragraph 7.3.3.)

Reformatting Specification

Line Number (3 4 5)	Type (6)	TYPE (N/O/F/D/S) (7)	(C/Z/D/SI/F/V/P/U) (8)	Location From (9 10 11 12)	Location To (13 14 15 16)	Forced Record Character (17)	Forced Substitute Character (18)	Forced Continuation (19)	Overflow (20 21 22)	Reserved (23 24 25 26 28 29 30 31 32 33 34 35 36 37 38 39)	Comments (Columns 40 to 72)
270	F	NC		12	20						
280	F	NC		60	64						
290	F	NC		8	10						
281	F	FC					$				
282	F	OC		1	4						
32	F										
33	F										
34	F										

135624

Figure 4-2. Inserting Additional Lines

4.2.5 SMS — Sort/Merge Sequence [A] (Column 18)

Input records can be sorted in an ascending or descending order, depending upon the entry in column 18 of the Header Specification. Control fields in the Reformatting Specification can further specify that individual control fields be sorted in an opposite manner or in the sequence specified in column 18 of the Header Specification. An A is placed in column 18 of the Header Specification to specify ascending order; a D specifies descending order.

Column 18 Entry	Explanation
A	Normal sequence is ascending
D	Normal sequence is decending

4.2.6 AS — Alternate Sequence [Blank] (Column 26)

Collation is accomplished according to the logical relationships of the ASCII character set or according to the values of an alternate character set. Appendix C defines the values of the ASCII character set.

Column 26 Entry	Explanation
Blank	Use ASCII collating sequence
S	Use alternate collating sequence

Most users leave column 26 blank. Generally, the only users of the alternate collating sequences are European firms that use special alphabetic characters (such as the German ä, ö, ü, and the Spanish ñ) in their collating sequence or users that must interface with computers that do not accept the standard ASCII character set. The collating sequences are defined as follows:

- ASCII Collating Sequence — The ASCII collating sequence is the ASCII character set as supported by the TI 990 computer system. A blank in column 26 tells the system to use the ASCII collating sequence. Slight variations in the ASCII collating sequence occur depending on whether it is desired to use both the zone and digit portions of the characters in the records, the zone portions only, or the digit portions only. Appendix C explains the ASCII code character set and identifies the zone and digit portions of the code.

- Alternate Collating Sequence — An S in column 26 tells the program to use an alternate collating sequence. You must supply Alternate Sequence (ALTSEQ) statements immediately following the Header Specification. (See Appendix B.)

4.2.7 PO — Print Option [1] (Column 27)

The Sort/Merge program can print the following:

- A list of the control specifications (Header, File Description, Record Selection, and Reformatting)

- Diagnostic messages appropriate for control specifications

- Program-status messages indicating the progress of the sort or merge

- Information, error, and warning messages when debugging an external subroutine that accesses Sort/Merge. This debug aid is described in Section 9.

Column 27 indicates which information is to be printed:

Column 27 Entry	Explanation
0 or blank	Print control specifications, diagnostic messages, and program-status messages
1	Print program-status messages
2	Reserved
3	Print none of these
4	Debug messages

4.2.8 DCF — Drop Control Fields [X] (Column 28)

Column 28 applies only to a tag-along sort (SORTR or SORTRS) operation. Column 28 indicates whether the control fields are to be included in the output records after the records are sorted or merged.

Column 28 Entry	Explanation
Blank	Keep control fields in output
X	Drop control fields in output

Control fields should be dropped whenever integer, real, floating point, opposite control fields, or an alternate collating sequence is used. In these situations, control field characters may be altered by the Sort/Merge program in a way that makes them meaningless to the user. For example, ALTSEQ changes the control fields to the characters specified in the alternate sequence specification.

To retain a meaningful version of the control field information, define the same input character positions twice: once as a control field and once as a data field. Data fields are not disturbed by the Sort/Merge program.

4.2.9 ORL — Output Record Length (Columns 29 through 32)

Columns 29 through 32 apply to all jobs and must be nonzero. The entry in these columns tells the program the length of the records in the final sorted file.

When control fields are not dropped (a blank in column 28), the output record length is the sum of all control field and data field lengths.

When control fields are dropped (an X in column 28), the entry in columns 29 through 32 of the Header Specification is the sum of the lengths of all data fields only. Because data field lengths may vary on different input record types, the output record length is the sum of data field lengths for any one record in the file.

For an address-only sort (SORTA), the output record length must always be specified as 4. SORTA outputs a four-byte binary record number in the format required by the I/O supervisor call (SVC) block, bytes 12 through 15 (record number). Refer to the *Model 990 Computer DX10 Operating System Release 3 Reference Manual, Volume III, Application Programming Guide*, for more information.

If the results of a SORTA are placed in a relative record output file, you cannot use COBOL to input that file because the COBOL delete indicator is in byte 1 of each record. When using COBOL, place the output from SORTA in a sequential file only. Refer to the *Model 990 Computer COBOL Programmer's Guide* for more information on the delete indicator with relative record files.

4.2.10 VAR — Variable Length Records [Blank] (Column 33)

Input files to Sort/Merge may contain variable-length records. If column 33 is blank, the length of the output records is fixed at the size specified in the output record length. If a V is specified, the record size is determined by the smaller value of the specified output record size and the input record size.

Column 33 Entry	Explanation
Blank	Fixed length output records
V	Variable length output records

4.2.11 VO — Verify Option [Blank] (Column 34)

To verify that data was not changed during the write from memory to the work file on disk, use the verify option. The verify option compares the disk image to that in memory. Using this option ensures accuracy but requires additional time, thus decreasing operation speed.

Column 34 Entry	Explanation
Blank	Do not verify data in work file
V	Verify data in work file

4.2.12 MS — Memory Size [Blank] (Columns 35 through 39)

In these columns, specify the decimal number of bytes to be used for the Sort/Merge memory work area. If you leave this field blank, 2800 bytes will be allocated automatically for the Sort/Merge work area. You can specify any value between 3,000 and 45,000 bytes. In general, the more memory available for the work area, the faster the sort will run. (See Appendix F.) This parameter is not required for a merge.

4.2.13 Comments (Columns 40 through 80)

These columns are used to make comments using any characters in the computer's character set. If listing of control specifications is selected for this task (the column 27 entry is a zero or blank), the comments are printed with the specifications. Comments have no effect on Sort/Merge.

5

File Description Specification

5.1 GENERAL

The File Description Specification (Figure 5-1) provides Sort/Merge with information about the files to be used in the operation. The order in which the files are specified is critical. The output file is described first; the work file, when designated, is described next; and the input file(s) is described last, as follows:

- Output File — Mandatory specification that provides the pathname of the output file (or device) that is to contain the results of the sort or merge process. The second line on the form is optional and is used to specify the physical record length, the logical record length, and the number of logical records.

- Work File — Optional specification that defines the disk volume that will contain a temporary file to be used as a scratch file during a sort process. If this one-line specification is not used, Sort/Merge will define this file using the logical record length entry from the Input File Specification, and this file will be allocated on the system disk. The work file is not used for the merge operation.

- Input File — Mandatory entry that provides the pathname(s) of the file or files (or device) containing the records to be processed. While a sort operation has only one input file, a merge operation can merge from two to five input files. The second line is used to specify the logical record length, the physical record length, and the number of logical records and is optional; however, you must specify the approximate number of records if the work file is to be created nonexpandable.

Tables 5-1 through 5-3 list the options that can be specified for the output file, work file, and input file. Subsequent paragraphs describe these options in detail.

5.1.1 Synonyms

You can use synonyms in file pathnames. Only 36 column positions are available for specification of a pathname to Sort/Merge. However, by using synonyms, you can specify a pathname that contains 48 characters. For the work file, the synonym is assigned only to the disk name.

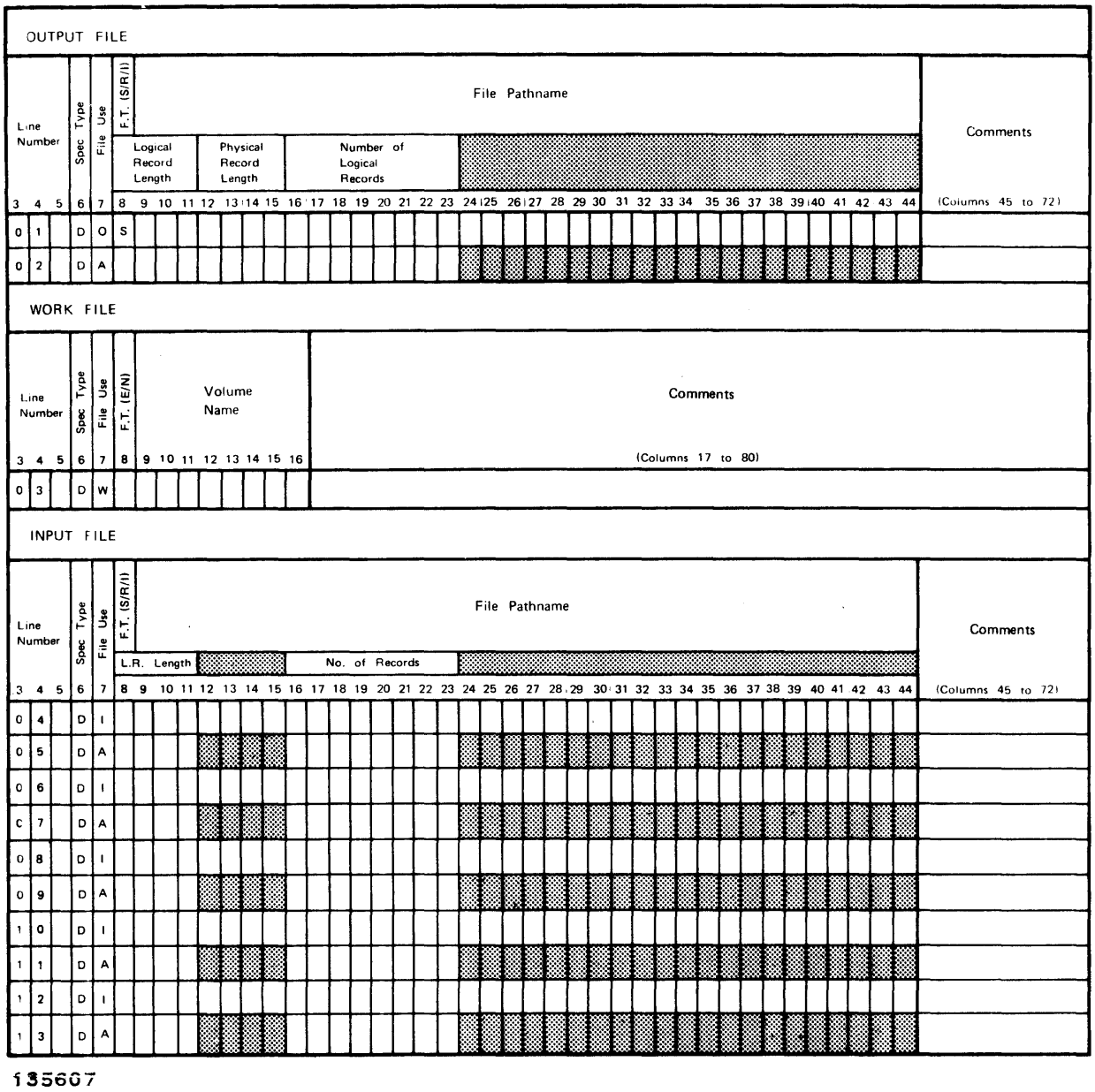
5.1.2 Page and Line Numbers (Columns 1 through 5)

The page numbers and line numbers preprinted on the worksheet are explained in Section 4.

5.1.3 Specification Type (Column 6)

A preprinted D in column 6 of each line on the worksheet identifies these lines as file description specifications.

File Description Specifications



135607

Figure 5-1. File Description Specification

5.2 OUTPUT FILE SPECIFICATION

The output file specified will contain the output of the sort or merge process. Table 5-1 lists the parameters that you may specify for the output file. These parameters describe the type of disk file, and its pathname, physical and logical record length, and number of physical records.

Table 5-1. Output File Specification Column Summary

Suggested Batch Keywords	Columns	Entries	Explanation (Interactive Prompts Capitalized)	Paragraph Number
Line 1:				
1	1-2[1]	00	Preprinted; identifies page 00	4.2.1
—	3-5[1]	010	Preprinted; identifies line 010	4.2.1
—	6[1]	D	Preprinted; designates this as a line in the File Description Specification	5.1.4
—	7[1]	O	Preprinted; identifies first line of Output File Specification	
FT	8[1]		File type:	
		S[2]	— Specifies sequential file	5.2.1.2
		R	— Specifies relative record file	5.2.1.2
		I	— Specifies key-indexed file	5.2.1.2
FP	9-44[2]		File pathname; output file name in the format required by the operating system; maximum of 36 characters for Sort/Merge	5.2.1.3
	45-80		Comments; enter up to 36 ASCII characters	
Line 2:				
—	3-5[1]	020	Preprinted; identifies line 020	4.2.1
—	6[1]	D	Preprinted; designates this as a line in the File Description Specification	5.1.4
—	7[1]	A	Preprinted; identifies a second line of Output File Specification	5.2.2.1
LRL	8-11	Integer	Logical record length, specified in bytes; default is explained in paragraph 5.2.2.2	5.2.2.2
PRL	12-15	Integer	Physical record length, length is specified in bytes; default is explained in paragraph 5.2.2.3	5.2.2.3
NOLR	16-23	Integer	Number of logical records; this is the record capacity of the output file; default is explained in paragraph 5.2.2.4	5.2.2.4
—	45-80		Comments; enter up to 36 ASCII characters	

Notes:
[1] These columns must be filled.
[2] Default value.

5.2.1 First Line of Output File Specification

Figure 5-2 shows the entries to this line.

5.2.1.1 File Use (Column 7). A preprinted O in column 7 identifies the first line of the Output File Specification.

5.2.1.2 FT — File Type (Column 8). This column specifies the file type for the output records.

Column Entry	Explanation
S	Sequential file organization
R	Relative record file organization
I	Key-indexed file organization

The Sort/Merge task can only create a sequential file. If you want the output to be a relative record or key-indexed file, you must create the file with the proper specifications prior to executing Sort/Merge. See the *Model 990 Computer DX10 Operating System Release 3 Reference Manual, Volume II, Production Operation*, for additional information on creating key-indexed files.

5.2.1.3 FP — File Pathname (Columns 9 through 44). You must use a file (or device) for the output destination. The pathname syntax is defined in the reference manual for the DX10 operating system. The maximum pathname length that can be specified is 36 characters. The output file pathname may be the same name as the input file pathname. In this case, the output is written over the input file and the previous contents of the input file are destroyed.

5.2.2 Second Line of Output File Specification

Figure 5-3 shows the entries for this line. This line is optional.

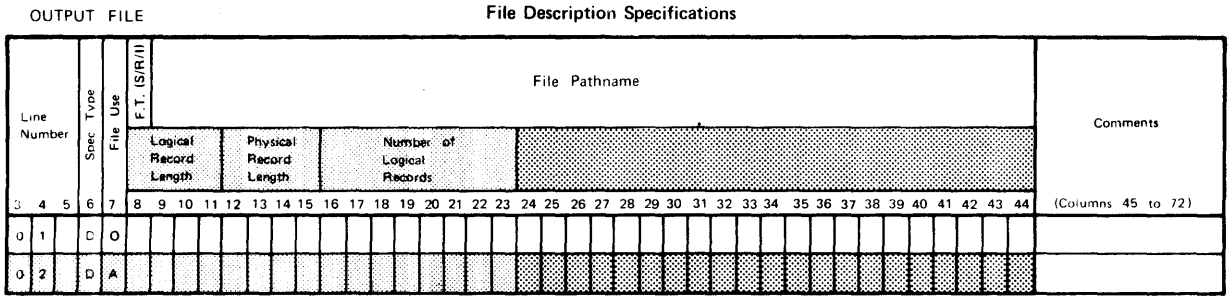


Figure 5-2. First Line of Output File Specification

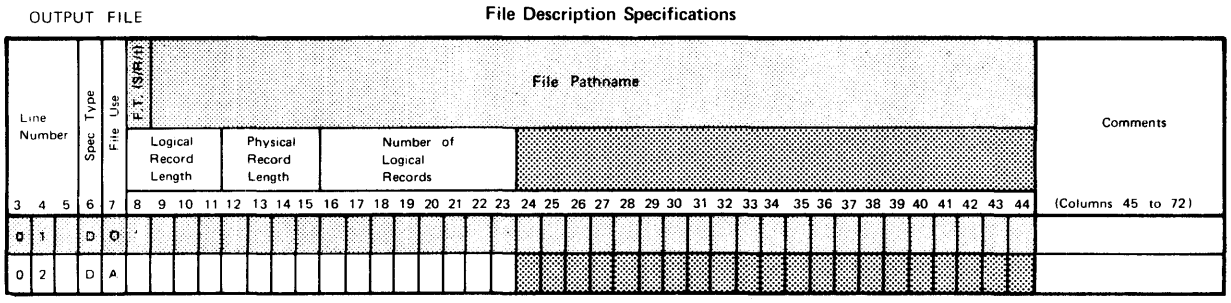


Figure 5-3. Second Line of Output File Specification

The following rules apply:

- If the file was previously created, the record sizes specified at creation are used and this line becomes commentary.
- If the second line is not used and the file was not previously created, Sort/Merge creates an unblocked, sequential file using the output record length from the header statement.
- If both the output logical record length and the output physical record length are specified and the file was not previously created, an expandable sequential file is created.
- If the output logical record length, the output physical record length, and the number of logical records are specified and the file was not previously created, a nonexpandable, sequential file is created.

5.2.2.1 File Use (Column 7). A preprinted A in column 7 identifies this as a continuation of the entries for the previous line.

5.2.2.2 LRL — Logical Record Length [Blank] (Columns 8 through 11). The logical record length for the output file is specified in these columns. If Sort/Merge is to create the file, the logical record length specified in the Header Specification is used. If the file already exists and the logical record length is not specified, the logical record length specified at file creation is used.

5.2.2.3 PRL — Physical Record Length [Blank] (Columns 12 through 15). The physical record length may be specified for output files. If it is not specified and the file already exists, the physical record length specified at file creation is used. If the length is not specified in the Output File Specification and Sort/Merge is to create the file, the file is created unblocked.

5.2.2.4 NOLR — Number of Logical Records [Blank] (Columns 16 through 23). The number of logical records may be specified for an output file. If this number is not specified for a file that must be created by Sort/Merge, an expandable output file is created.

5.3 WORK FILE SPECIFICATION

The Work File Specification (Figure 5-4) defines an area on the disk to be used as a work area during the sort process. The work file is not used for a merge operation; if specified, it is ignored.

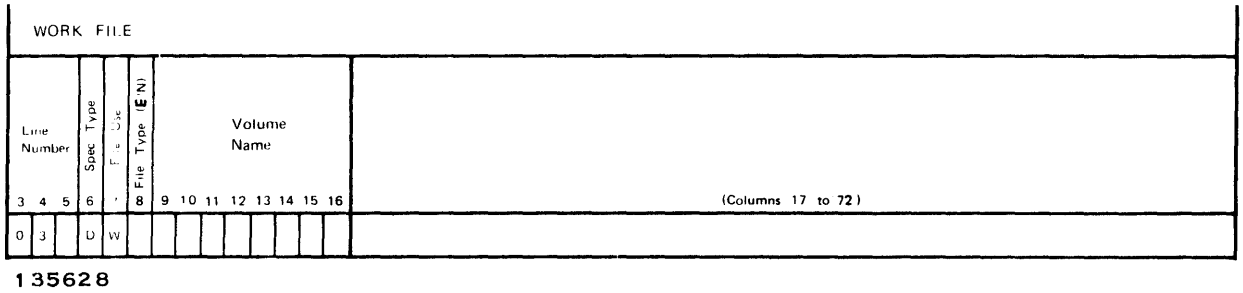


Figure 5-4. Work File Specification

You can specify only one work file per sort operation; however, specification of the work file is optional. If not specified, the work file is automatically created on the system disk and is released upon completion of Sort/Merge processing. When possible, the work file should be specified on a disk unit that is not being used for the Sort/Merge input or output files. This technique results in faster sorts.

It is not necessary to specify variables such as the logical and physical block size and the number of physical blocks in the file for the work file. The Sort/Merge optimization algorithm determines optimal physical block size based on the available memory, the record (or control field) size, and the number of input records. Thus, any attempt to specify block size could result in slower sort times. (See paragraph 5.4.2.3.)

Table 5-2 lists the entries for the Work File Specification.

5.3.1 File Use (Column 7)

A preprinted W in this column identifies this as the Work File Specification.

5.3.2 WFT — Work File Type [E] (Column 8)

If the Input File Specification (paragraph 5.4) does not specify the number of records to be sorted, this field is ignored and the work file is allocated expandable.

Column 8 Entry	Explanation
N or blank	File is allocated nonexpandable
E	File is allocated expandable

Table 5-2. Work File Specification Column Summary

Suggested Batch Keywords	Columns	Entries	Explanation (Interactive Prompts Capitalized)	Paragraph Number
—	3-5[1]	030	Preprinted; identifies line 030	4.2.1
—	6[1]	D	Preprinted; designates a line in File Description Specification	5.1.4
—	7[1]	W	Preprinted; identifies this as the Work File Specification	5.3.1
WFT	8		Work file type:	5.3.2
		N or blank	— File allocated nonexpandable	
		E[2]	— File allocated expandable	
WFV	9-16		Work file volume:	5.3.3
		DS01[2] or blank	— System disk	
		DSnn or volume	— Disk nn; 8-character volume name	

Notes:

[1] These columns must be filled in if Work File Specification is used.

[2] Default value.

5.3.3 WFV — Work File Volume [DS01] (Columns 9 through 16)

This is the disk name or volume on which the file is to be created. The volume name under which the disk is installed is a maximum of eight characters long, with the first character alphabetic.

Column 9 through 12 Entry	Explanation
DS01 or blank	File is allocated on system disk
DSnn	File is allocated on disk nn
Volume	One- to eight-character name is specified when disk is installed

5.4 INPUT FILE SPECIFICATION

The Input File Specification is located at the bottom of the File Description Specification (Figure 5-5). This specification has two-line entries, similar to those of the Output File Specification at the top of the worksheet. Space is provided to describe up to five input files, the maximum that can be used in a merge program. (Only one file is used for a sort operation.) Table 5-3 describes the entries available for specifying the output file.

Sequential, relative record, or key-indexed files created by the file management routines or devices (such as card readers), can be used as an input file for Sort/Merge (for example, a COBOL file or a file from a previous Sort/Merge run).

NOTE

Relative record files sometimes contain unused records set aside for future expansion. You should delete these unused records during the Sort/Merge process to avoid wasting disk space.

5.4.1 First Line of Input File Specification

Figure 5-5 shows the entries for this line.

5.4.1.1 File Use (Column 7). A preprinted I in this column identifies the first line of a two-line entry describing the input file.

5.4.1.2 FT — File Type [S] (Column 8). This specification defines the file organization as either sequential, relative record, or key-indexed.

Column 8 Entry	Explanation
S	Sequential file organization
R	Relative record file organization
I	Key-indexed file organization

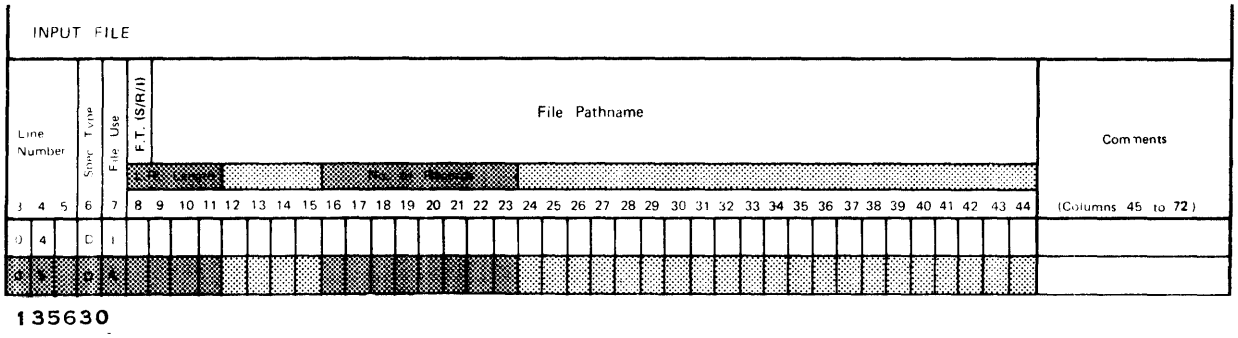


Figure 5-5. First Line of Input File Specification

Table 5-3. Input File Specification Column Summary

Suggested Batch Keywords	Columns	Entries	Explanation (Interactive Prompts Capitalized)	Paragraph Number
Line 1:				
—	3-5[1]	04x-12x	Preprinted; numbers specify sequence of specification lines	4.2.1
—	6[1]	D	Preprinted; designates a line in File Description Specification	5.1.4
—	7[1]	I	Preprinted, every other line; designates first line of Input File Specification	5.4.1.1
FT	8		File type:	5.4.1.2
		S[2]	— Sequential file	
		R	— Relative file	
		I	— Key-indexed file	
FP	9-44[1]		File pathname; the input file pathname in the format required by the operating system (for example, DS05.HDWRDIR.TOOLIB.HMRFIL); maximum of 36 characters	5.4.1.3
	45-80		Comments; enter up to 36 ASCII characters	
Line 2:				
—	3-5	05x-13x	Preprinted; numbers specify sequence of specification lines	4.2.1
—	6	D	Preprinted; designates a line in File Description Specification	5.1.4
—	7	A	Preprinted, every other line; identifies second line of Input File Specification	5.4.2.1
LRL	8-11	Integer	Logical record length; required by the language interfaces or when using devices that do not have a default logical record length	5.4.2.2
NOLR	16-23	Integer	Number of logical records; required if the work file is created nonexpandable	5.4.2.3

Notes:

[1] These columns must be filled.

[2] Default value.

5.4.1.3 FP — File Pathname (Columns 9 through 44). You must specify a file or device for an input file pathname. The pathname syntax is defined in the reference manual for the DX10 operating system. The maximum pathname length allowed is 36 characters.

5.4.2 Second Line of Input File Specification

Figure 5-6 shows the entries for this line. If expandable is specified in the Work File Specification, this line is usually optional (see paragraph 5.4.2.2); otherwise, Sort/Merge uses the data on this line to allocate the work file area.

5.4.2.1 File Use (Column 7). A preprinted A in this column signifies that this is the second entry of a two-line entry describing an input file. It always follows a first-line entry marked by an I in column 7.

5.4.2.2 LRL — Logical Record Length [Blank] (Columns 8 through 11). This is the logical record length, right-justified (ends in column 11), of the input file. It is required when Sort/Merge is accessed from a language program and is required for intertask communication. It is also required when using an input device that does not have a defined logical record length (for example, magnetic tape). If a value is specified and is larger than the defined logical record length of the file, the specified length is used. If this field is specified and is smaller than the defined logical record length, the defined logical record length is used.

5.4.2.3 NOLR — Number of Logical Records [Blank] (Columns 16 through 23). This field must be specified if the work file is to be allocated nonexpandable. If NOLR is specified and the work file is expandable, NOLR/8 is used for both primary and secondary disk file allocation parameters when the work file is created. If NOLR is not specified, the work file is allocated using default primary and secondary disk allocations.

NOTE

No increase in performance is gained by using nonexpandable files. In addition, Sort/Merge will not produce the desired results if the nonexpandable files are not created large enough to handle the desired sort.

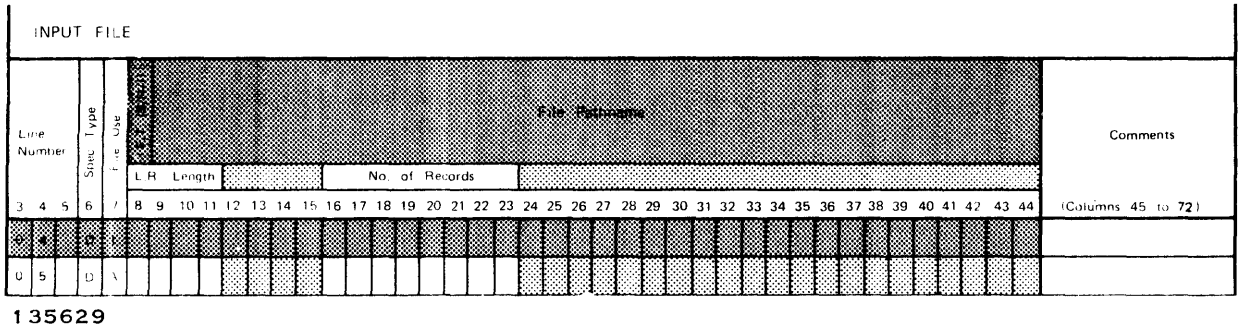


Figure 5-6. Second Line of Input File Specification

6

Record Selection Specification

6.1 GENERAL

The Record Selection Specification designates which records Sort/Merge is to select for the work and output files. Criteria for record selection include the following:

- One or more specified characters in positions of the input record
- An input record field that contains a value with a specific relationship to a value in another field within that same record; that relationship is one of the following: equal to, not equal to, less than, greater than, less than or equal to, or greater than or equal to.

The following are examples of records that Sort/Merge can select from an input file:

- Records that have an A or a B in the first position of the input record
- Records that have a value between 120 and 140 in positions 6 through 10
- Records that have one of the letters A through M (inclusive) in position 44
- Records in which the value in positions 66 through 67 is greater than or equal to the value in positions 68 through 69
- Records that meet all of the above criteria

Figure 6-1 shows the entries to the Record Selection Specification. Table 6-1 is a summary of the entries to the Record Selection Specification.

NOTE

If all records in a file have the same format and all are to be sorted or merged, a Record Selection Specification is not necessary.

6.2 COLUMN DESCRIPTION

The following paragraphs explain the column entries for the Record Selection Specification.

6.2.1 Page Numbers and Line Numbers (Columns 1 through 5)

The page numbers and line numbers are described in paragraph 4.2.1.

SORT/MERGE SPECIFICATION, SHEET B

Record Selection Specification

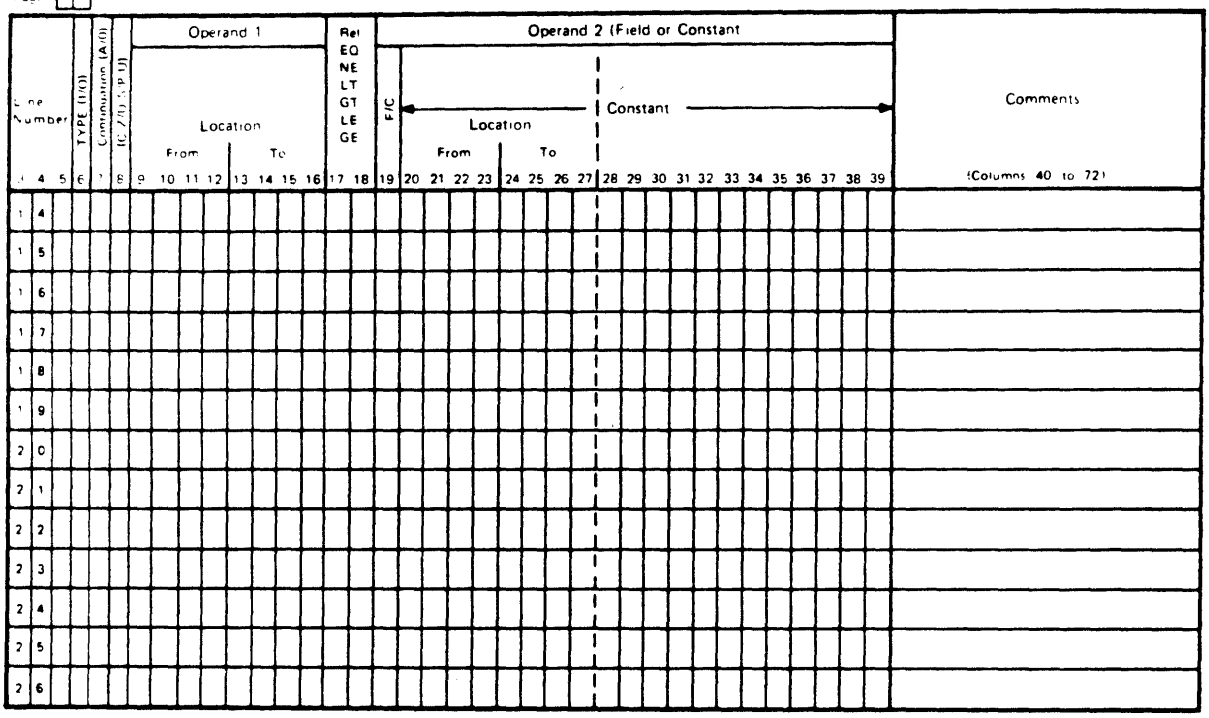


Figure 6-1. Record Selection Specification

6.2.2 RST — Record Selection Type [I] (Column 6)

Use column 6 to specify the type of line. An I in this column indicates either an include or an include-all line; an O indicates an omit line.

Column 6 Entry	Explanation
I	Include or include-all line
O	Omit line

6.2.2.1 Include Line. Include lines identify records that you want sorted or merged with your output file. A record is commonly identified by the presence (or absence) of a certain character in a specific record position; this identification method distinguishes one input record type from the other types in the file. The record identification is entered on an include line. Records not described in include lines will not be sorted.

Table 6-1. Record Selection Specification Column Summary

Suggested Batch Keywords	Columns	Entries	Explanation (Interactive Prompts Capitalized)	Paragraph Number
—	1-2[1]	01-99	Page number	4.2.1
—	3-5[1]	14x-26x	Preprinted; numbers specify sequence of specification lines	
RST	6[1]		Record selection type:	6.2.2
		I[2]	— Include records described on this line	
		O	— Omit records described on this line	
CON	7[1]		Continuation:	6.2.3
		Blank[2]	— A blank indicates that this is the only line or the first line describing the record type; if more than one line, continuation lines with A or O in column 7 follow this entry	
		A	— An AND continuation line	
		O	— An OR continuation line	
CP	8[1]		Character portion:	6.2.4
		C[2]	— Compare both zone and digit portions	
		Z	— Compare only zone portion	
		D	— Compare only digit portion	
		S	— Compare signed ASCII	
		P	— Compare packed fields	
		U	— Compare unpacked fields	
O1B	9-12[1]		Operand 1 beginning:	6.2.5
		Blank[2]	— Field is one position; see columns 13-16	
		1-4096	— Position on the input record in which operand 1 begins	
O1E			Operand 1 ending:	6.2.5
	13-16[1]	1-4096	column position in input record of one character field or position in which operand 1 ends	
REL	17-18[1]		Relational operator:	6.2.6
			Shows the relation of operand 1 to operand 2:	
		EQ	— Equal	
		NE	— Not equal	
		LT	— Less than	
		GT	— Greater than	
		LE	— Less than or equal	
		GE	— Greater than or equal	

Table 6-1. Record Selection Specification Column Summary (Continued)

Suggested Batch Keywords	Columns	Entries	Explanation (Interactive Prompts Capitalized)	Paragraph Number
FOC	19[1]		Field or constant:	6.2.7
		F	— Operand 2 is a field	
		C	— Operand 2 is a constant	
O2B	20-23[1]		Operand 2 beginning:	6.2.8.1
		Blank[2]	— Field is one position; see columns 24-27	
		1-4096	— Position on the input record in which operand 2 begins	
O2E	24-27	1-4096	Operand 2 ending; position in input record where operand 2 ends	6.2.8.1
O2C	20-39		Operand 2 constant; these columns contain the character string to which the operand 1 field is being compared; can be several lines long (256 characters maximum)	6.2.8.2
—	40-80		Comments; enter up to 41 ASCII characters	6.2.9

Notes:

[1] These columns must be filled.

[2] Default values.

6.2.2.2 Include-All Line. An include-all line designates that all input record types that have not been described by preceding include and/or omit lines will be sorted. Because of the inclusive nature of this line, record identifications are not supplied in columns 7 through 39. However, record types included by an include-all line must have identical Reformatting Specifications.

NOTE

Only one include-all line is allowed per input file. When used, the line must be the last line of the set of Record Selection Specifications.

Records not described by either include or include-all lines do not take part in the sort or merge.

6.2.2.3 Omit Line. Omit lines identify those records that you do not want sorted. Omit lines are not required but can be helpful when you are using many types of records but are omitting only a

few. An include-all line usually follows an omit line to ensure that all records not described by omit lines will be sorted.

6.2.2.4 Include and Omit Sets. Reformatting Specification lines complete the identification of input records involved in the sort or merge. Section 7 describes these Reformatting Specifications in detail. The combination of Record Selection Specifications and Reformatting Specification lines for one or more record types is called an *include set*.

A record type is identified by at least one characteristic for all records in that type. For example, records may be recognized by the characters ABC in positions 4, 5, and 6 of the input records.

Besides the include set, there is an *omit set*. An omit set identifies record types to be omitted (or excluded) in a sort or merge. An omit set never requires Reformatting Specification lines because excluded fields need not be described.

Sort/Merge processes both the include and omit sets in the order in which they appear in the sequence specifications.

Using Include or Omit Sets. Include sets are used when sorting or merging only a few records in a file or files. An include set is used for each type of record to be sorted or merged.

Omit sets are used when sorting or merging all but a few records in a file. All omit sets must be followed by either an include set for each type of record to be sorted or an include-all set. The include-all states that the sort or merge should include all records not omitted or included by previous omit or include sets.

Mixing Include and Omit Sets. You can mix include and omit sets. However, you should take care in doing so since Sort/Merge processes the sets in the order in which they are coded. For example, to omit all records with a 3 in position 20, while including all records with a 3 in positions 20 and 25, you must specify the include set before the omit set. If the omit set were specified first, all of the records to be sorted would be omitted.

Include/Omit Set Summary. The following summarizes the criteria for record selection:

- An include set must have at least one Reformatting Specification line.
- An omit set does not need a Reformatting Specification line.
- An omit set must be followed by an include set (usually an include-all); the only records to be included will be those in the include set.
- A typical omit set is followed by an include set that states that the sort or merge should include all records not specifically omitted by the previous omit lines.
- The last entry in the Record Selection Specification must specify an include set.
- Only one include-all line is allowed for an input file. Tables 6-2 through 6-5 show examples of include and omit sets. Note that the combinations of include and omit sets depend on the entry in column 7 (explained in paragraph 6.2.3).

Table 6-2. Columns 6 and 7 Entries for Include Sets

Column 6	Column 7	Explanation
I	Blank	This is the first line of an include set on the Record Selection Specification. This line describes a record type to be included in a sort or merge. Additional specifications for record selection in this include set can follow with an A or O in column 7.
I	A	This is an additional line to an include set description. This line describes additional specifications that a record must meet in order to be included in the sort or merge. (The record must meet both the previous criteria **and** these additional criteria to be included.)
I	O	This is an additional line to an include set description. This line states that a record meeting the criteria on this line will be selected. (The record must meet either previous criteria **or** these criteria to be included.)

Table 6-3. Columns 6 and 7 Entries for Omit Sets

Column 6	Column 7	Explanation
0	Blank	This is the first line of an omit set on the Record Selection Specification. This line describes a record type to be omitted in a sort or merge. Additional specifications for record selection in this omit set can follow with an A or O in column 7.
O	A	This is an additional line to an omit set description. This line describes additional specifications that a record must meet in order to be omitted in the sort or merge. (The record must meet both the previous criteria **and** these additional criteria to be omitted.)
O	O	This is an additional line to an omit set description. This line states that a record meeting the criteria on this line will be omitted. (The record must meet either the previous criteria **or** these criteria to be omitted.)

Table 6-4. Include Set Combinations of Specification Lines

Column 6	Column 7	Explanation
I I	Blank A	One or more lines with A in column 7 following the first line signifies include AND lines.
I I	Blank O	One or more lines with O in column 7 following the first line signifies include OR lines.
I I I	Blank A O	One or more A and O lines, in any order, following the first line signifies include AND lines and OR lines describing one record type.
I	Blank	One include line without A or O lines following signifies an include-all entry. This tells Sort/Merge to include all records not previously defined by preceding include or omit lines.
Blank	Blank	An implied include-all line. If no Record Selection Specification line is used, all records in the input file will be sorted.

Table 6-5. Omit Set Combinations of Specification Lines

Column 6	Column 7	Explanation
O O	Blank A	One or more lines with A in column 7 following the first line signifies omit AND lines.
O O	Blank O	One or more lines with O in column 7 following the first line signifies omit OR lines.
O O O	Blank A O	One or more A and O lines, in any order, following the first line signifies omit AND lines and OR lines describing one record type.

6.2.3 CON — Continuation [Blank] (Column 7)

Column 7 indicates the relationship between this line and the preceding Record Type Specification line.

An AND line is a continuation of the set of identifiers for a single Record Type Specification line. This type of line is used if more than one condition is required to identify a record. Record descriptions in an AND relationship use a single set of Field Specifications, since only one record is being defined by this include set.

An OR line provides an alternate record type (or types) to share a single set of Field Specifications. Again, all of the Record Type Specification lines in an OR relationship share a single set of Field Specifications. Together, these lines form an include set.

Define a new include set (blank in column 7) when a preceding field description is not appropriate for the record type about to be defined. The following summarizes the column 7 entries.

Column 7 Entries	Explanation
Blank	The first line of a set of include or omit lines. (The type of set is indicated by the column 6 entry.)
A	This line is a continuation of the preceding line (AND line).
O	This defines a different record type than the preceding line, but the Field Specifications are the same for both (OR line).

6.2.4 CP — Character Portion [Blank] (Column 8)

Column 8 tells Sort/Merge how to interpret data in the operand 1 and operand 2 fields during compare operations. The only allowable entries are alphanumeric and signed ASCII. When the fields contain alphanumeric data, a C, Z, or D entry tells Sort/Merge what portions of the characters to use.

The letter S in this column specifies that the field contains a signed ASCII value. This means the characters contain a sign character signifying plus or minus; this character is the least significant character of the field. Thus, Sort/Merge will use the signed value of the numerical string to determine relationships of that field to other fields or absolute values.

The letter P in this column specifies that the field contains numeric data in packed format. The letter U in this column specifies that the field contains numeric data in unpacked format.

The following specifies the possible values for Column 8 entries.

Column 8 Entry	Explanation	Maximum Field Length (in characters)
C	Compare both zone and digit portions of characters	256
Z	Compare only zone portion of the character	1
D	Compare only the digit portion of the characters	16
S	Compare using the least significant byte of the field as a sign byte	16
P[1]	Compare numeric data in packed format	8 (16 digits)
U[1]	Compare numeric data in unpacked format	16

Note:

[1] Do not specify P or U when an alternate collating sequence is being used (S in column 26 of header line).

6.2.4.1 Character Portion Indicators (C, Z, or D in Column 8). As explained in paragraph 6.1, Sort/Merge can select only those records having specific values in certain portions of the record. For example, each record being searched in the input file might be in an 80-position format, with each position corresponding to a character in a column on a card. The values for each character on the card are stored in memory or on the disk in a binary format, the same format that comprises the ASCII character set.

Figure 6-2 shows several of the ASCII characters. This figure also shows how the eight-bit ASCII binary code is divided into zone and digit portions.

On inspection of the ASCII number set, notice that all numbers have the same zone portion (that is, all eight-bit representations of ASCII numbers have the binary representation of 0011_2 for the first four bits). However, the digit portions of ASCII numbers are different, ranging from 0001_2 (a binary 1) for decimal 1 to 1001_2 (a binary 9) for decimal 9.

Letters in ASCII code use different zone portions as well as different digit portions. Note that the codes range from a 41_{16} for an A (the zone portion is 4, the digit portion is 1) to a $5A_{16}$ for a letter Z.

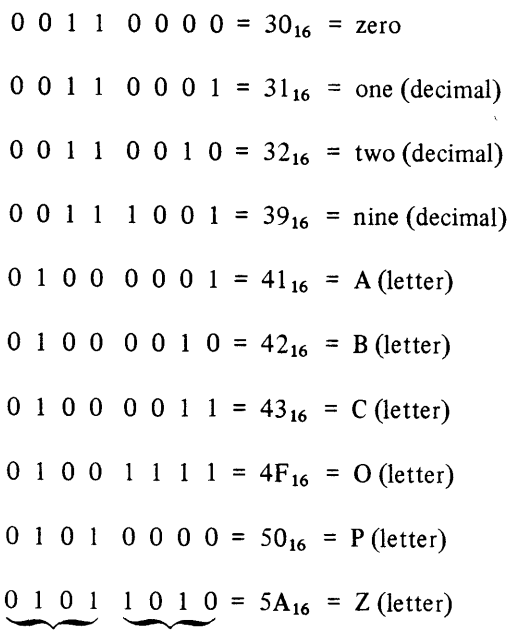
Depending on the particular application, you may want either to compare only the digit portions of the characters in the input record or to use the entire binary representation (zone and digit portions) in the comparison. Designating this is the purpose of the column 8 entry.

The comparison of zone and digit portions of characters may not be as important to the ASCII user as it is to the user who employs an alternate sequence to sort or merge files created in EBCDIC or in another character coding scheme. Appendix D shows the EBCDIC collating sequence.

Column 8 is involved in the identification of a record type. The record item whose location is entered into operand 1 (columns 9 through 16) is compared against the constant or item specified in operand 2 (columns 20 through 27). Column 8 specifies the type of comparison to make. When operand 1 is an alphanumeric item, column 8 specifies whether a comparison should be made on the basis of the high-order four bits (the zone), the low-order four bits (the digit), or the entire eight bits of each character of the operand 1 field.

With a zone comparison or a digit comparison, several characters share an equal comparison. A full character or zone comparison on the characters 1 and A yields an unequal result; however, 1 and A yield an equal result in a digit comparison.

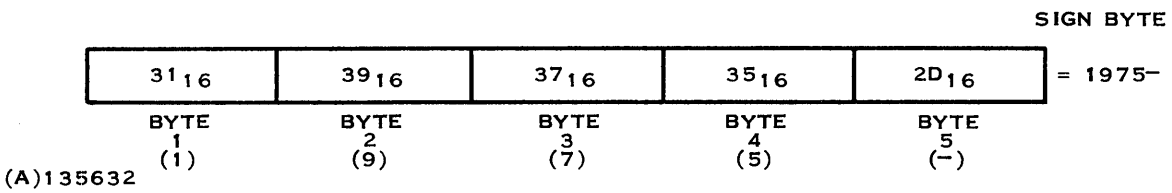
If, in another example, a D is entered in column 8 when only those records with a 3 in column 20 and a 3 in column 46 are needed, the entry might produce not only the records requested but also a lot of unwanted records (for example, records with C in those positions, because an ASCII C and an ASCII 3 both have 001_2 in their zone portion). To obtain only the records with a 3 in column 20 and a 3 in column 46, enter a C in column 8. The C tells Sort/Merge to use both the zone and digit portions of characters in its comparison; no other ASCII character has both the same zone and digit portions as a 3.



Note: A complete ASCII character set is shown in Appendix C.

Figure 6-2. Zone and Digit Portions of ASCII Characters

6.2.4.2 Signed ASCII Indicator (S in Column 8). If signed ASCII is used, the sign is placed in the least significant byte of the numerical character string, as shown below, for the number -1975:

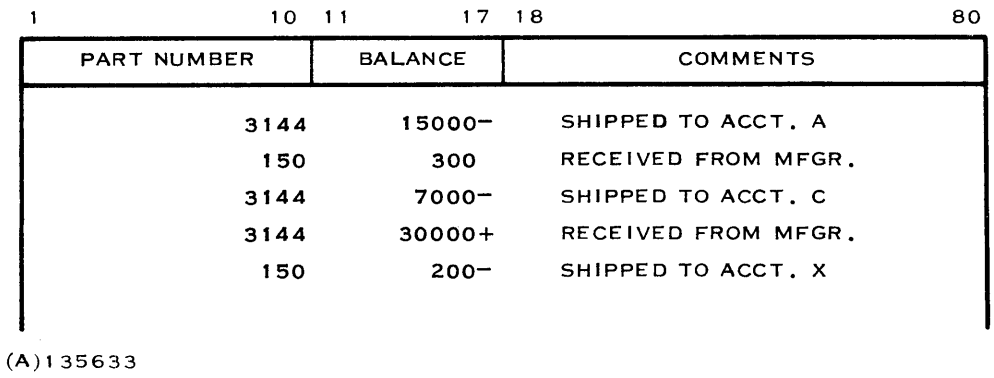


The sign in the least-significant byte is indicated by the ASCII code for a minus sign ($2D_{16}$)or a positive sign ($2B_{16}$). A positive value is also indicated by the ASCII code for a space (20_{16}).

NOTE

When specifying a signed ASCII constant, specify all digits, with the exception that you can use a blank space to denote the positive sign.

Since the sign must be in the least significant byte, you can enter data in the input file in the following format:



Thus, Sort/Merge uses the sign value to make the record selection. In the above example, the sign byte is in position 17 of the input file, and the field containing the balance values begins in position 11 (and ends in position 17). To specify this in the operand 1 columns, place the first position of the field in the From columns, right-justified, and the last position of the field in the To columns, right-justified. The To column is the column of the sign. Figure 6-3 shows how to include all records when the BALANCE entries in the above example contain values greater than zero.

SORT/MERGE SPECIFICATION, SHEET B

Page 1 2

Record Selection Specification

Line Number	TYPE (I/O)	Continuation (A/O)	(C/Z/D/S/P/U)	Operand 1 Location From	Operand 1 Location To	Rel. EQ NE LT GT LE GE	F/C	Operand 2 (Field or Constant) Location From	Operand 2 Location To	Operand 2 Constant	Comments
3 4 5	6	7	8	9 10 11 12	13 14 15 16	17 18	19	20 21 22 23	24 25 26 27	28 29 30 31 32 33 34 35 36 37 38 39	(Columns 40 to 72)
1 4	O I	S		1 1	1 7	G T	C	0 0 0 0 0 0 0		+	
1 5											

SPECIFIED: INCLUDE RECORDS IN WHICH THE SIGNED ASCII VALUES IN INPUT RECORD POSITIONS 11 TO 17 ARE GREATER THAN (GT) ZERO.

135634

Figure 6-3. Example of Specifying Signed ASCII Value

6.2.5 O1B and O1E — Operand 1 Beginning and Ending [Blank] (Columns 9 through 16)

The operand 1 field identifies the positions on the input records that are to be compared (included or omitted from the output records). Sort/Merge obtains an image of the data at these specified positions in the input records and compares it to the data identified by the operand 2 entry (paragraph 6.2.8) according to the comparison criteria (for example, equal to or greater than) in columns 17 and 18 (paragraph 6.2.6).

The operand 1 field cannot describe a field length larger than the length of the records being used. For example, if the records are in an 80-column format (such as records contained on 80-column cards), the value of the operand 1 field cannot exceed 80.

Column Number	Column Entry	Explanation
9-12	Blank or 1-4096	Identifies first position of input record field (blank if field is one character long).
13-16	1-4096	Identifies last position on record field.

The maximum total characters that can be specified in an operand 1 or 2 field is 256. The operand 1 entry is further restricted by the column 8 entry, as follows:

Column 8 Entry	Maximum Characters in Factor 1 Field
C	256
Z	1
D	16

Column 8 Entry (Continued)	Maximum Characters in Factor 1 Field
S	16
P	8 (16 digits)
U	16

Sometimes it is necessary to identify a record type by characters at several record positions, requiring several sets of operand 1 and operand 2 fields. To define additional fields, use additional lines on the Record Selection Specification; however, you must place the letter A or O in column 7 for additional lines.

The operand 1 field contains two four-column entries. The first, marked From, identifies the first record position; the second, marked To, identifies the last record position. For example, to select zip codes specified in record positions 61 through 67, specify record position values, right-justified, in columns 11/12 and 15/16, respectively. However, if you check for one character in record position 25, this value is marked in the To field, right-justified, without an entry in the From field.

In Figure 6-4, the user specified a search for those records with an X in position 3 and a number between 100 and 199 in positions 21 through 23. Note that several operand 1 and 2 fields are required to describe the desired records.

6.2.6 REL — Relational Operator [Blank] (Columns 17 and 18)

The operand 1 and 2 entries establish the position of certain ASCII characters on the record. These characters must be compared to other characters or values to determine whether to send the record to the output file. Columns 17 and 18 specify the relationship between operand 1 and operand 2, as follows:

Columns 17 and 18 Entry	Explanation
EQ	Operand 1 is equal to operand 2
NE	Operand 1 is unequal to operand 2
LT	Operand 1 is less than operand 2
GT	Operand 1 is greater than operand 2
LE	Operand 1 is less than or equal to operand 2
GE	Operand 1 is greater than or equal to operand 2

Sort/Merge selects records for the input file on the basis of these relationships being true.

SORT/MERGE SPECIFICATION, SHEET B

Page 1 2

Record Selection Specification

Line Number	TYPE (I/O)	Continuation (A/O)	(C/Z/D/S/P/U)	Operand 1 Location From	Operand 1 Location To	Rel. EQ NE LT GT LE GE	F/C	Operand 2 (Field or Constant) Location From	Operand 2 Location To	Operand 2 Constant	Comments
3 4 5	6	7	8	9 10 11 12	13 14 15 16	17 18	19	20 21 22 23	24 25 26 27	28 29 30 31 32 33 34 35 36 37 38 39	(Columns 40 to 72)
1 4	O	I	C			3E	Q	C	X		
1 5	O	I	A	C	21	23	G	E	C	100	
1 6	O	I	A	C	21	23	L	E	C	199	

SPECIFIED: INCLUDE RECORDS WITH X IN POSITION 3 AND NUMBERS 100 TO 199 IN POSITIONS 21 TO 23.

135635

Figure 6-4. Example of Operand 1 and Operand 2 Entries

6.2.7 FOC — Field or Constant [Blank] (Column 19)

The operand 1 field can be one or more characters or a string of numerical values. To compare this field to another field (of equal length), specify the appropriate record positions in the operand 2 field. You can also compare the contents in the operand 1 field to one or more constants, such as a specific number or one or more characters.

The following specifies the possible values for column 19 entries:

Column 19 Entry	Explanation
C	Operand 2 is a constant; value in operand 1 position(s) is compared to this constant.
F	Operand 2 is a field; value in operand 1 position(s) is compared to the values in the field in operand 2.

To define the contents of the operand 2 field, place either C for constant or F for field in column 19, as shown in Figure 6-5.

6.2.8 Operand 2 Location or Constant [Blank] (Columns 20 through 39)

The following paragraphs explain the entries for the operand 2 location or constant.

6.2.8.1 O2B and O2E — Operand 2 Beginning and Ending [Blank] (Columns 20 through 27). These columns identify the record positions that will be compared to the positions specified in the operand 1 location field. Since this identifies a field on the input record, the letter F must be in column 19. The values in the From column identify the record position that begins the field. The values in the To column identify the last position in the field on the input record. The operand 1 field and the operand 2 field must be on the same record, and the number of characters in both fields must be the same.

SORT/MERGE SPECIFICATION, SHEET B

Page 1 2

Record Selection Specification

Line Number	TYPE (I/O)	Continuation (A/O)	(C/Z/D/S/P/U)	Operand 1 Location From	Operand 1 Location To	Rel. EQ NE LT GT LE GE	F/C	Operand 2 (Field or Constant) Location From	Operand 2 Location To	Operand 2 Constant	Comments
3 4 5	6	7	8	9 10 11 12	13 14 15 16	17 18	19	20 21 22 23	24 25 26 27	28 29 30 31 32 33 34 35 36 37 38 39	(Columns 40 to 72)
1 4	O	I	C	20	29	GT	F	30	39		
1 5	O	IA	C	41	48	EQ	C	PART NO.			

SPECIFIED: INCLUDE RECORDS IN WHICH POSITIONS 20 TO 29 ARE GREATER THAN POSITIONS 30 TO 39 AND POSITIONS 41 TO 48 CONTAIN THE CHARACTERS PART NO.

135636

Figure 6-5. Example of Field and Constant Entries

NOTE

If operand 2 is a field, the field position values in the From and To columns are right-justified to columns 23 and 27, respectively.

If operand 2 is a constant or a constant string, the constants are left-justified; that is, the character in the first position in operand 1 is compared to the character in column 20.

Figure 6-6 shows how operand 1 and operand 2 describe fields in the input record.

If the constant in operand 2 is a character string requiring more than the 20 positions in columns 20 through 39 on the Record Selection Specification sheet, you can use additional lines, as shown in Figure 6-8. The operand 2 constant field, in columns 20 through 39, contains the character string with the input record columns defined in the operand 1 field. You can use up to 256 characters as constants if both zone and digit fields are checked. (See paragraph 6.2.4 for the use of other fields.)

6.2.8.2 O2 — Operand 2 Constant [Blank] (Columns 20 through 39). These 20 columns contain the constant or a constant string to which operand 1 is to be compared. When this field contains a constant, place the letter C in column 19.

When operand 2 is a constant, it must occupy the same number of input record positions as specified in operand 1. For example, assume that the operand 1 field specifies three positions and the relational operator and operand 2 specify a number greater than 5 in that three-position field; in this case, the operand 2 entry should also be three positions long, with the 5 in the least-significant position (a 005 entry in columns 20 through 22). Unused character positions to the left of the operand 2 character(s) can be blanks or zero filled. Figure 6-7 shows an example of zero- and blank-filled fields; in this example, operand 1 specified more positions than the constant contains.

SORT/MERGE SPECIFICATION, SHEET B

Page 1 2

Record Selection Specification

Line Number	TYPE (I/O)	Continuation (A/O)	(C/Z/D/S/P/U)	Operand 1 Location From	Operand 1 Location To	Rel. EQ NE LT GT LE GE	F/C	Operand 2 (Field or Constant) Location From	Operand 2 Location To	Operand 2 Constant	Comments
3 4 5	6	7	8	9 10 11 12	13 14 15 16	17 18	19	20 21 22 23	24 25 26 27	28 29 30 31 32 33 34 35 36 37 38 39	(Columns 40 to 72)
1 4	I	C		6	12	GT	F	16	22		

SPECIFIED: INCLUDE RECORDS IN WHICH POSITIONS 6 TO 12 ARE GREATER THAN POSITIONS 16 TO 22.

135637

Figure 6-6. Example of Operand 2 Field and Constant

SORT/MERGE SPECIFICATION, SHEET B

Page 1 2

Record Selection Specification

Line Number	TYPE (I/O)	Continuation (A/O)	(C/Z/D/S/P/U)	Operand 1 Location From	Operand 1 Location To	Rel. EQ NE LT GT LE GE	F/C	Operand 2 (Field or Constant) Location From	Operand 2 Location To	Operand 2 Constant	Comments
3 4 5	6	7	8	9 10 11 12	13 14 15 16	17 18	19	20 21 22 23	24 25 26 27	28 29 30 31 32 33 34 35 36 37 38 39	(Columns 40 to 72)
1 4	O I	C		3	5	GT	C	0 0 5			
1 5	O I	A	C		8	EQ	C	A			
1 6	O I	A	C	11	14	GT	C	10			

SPECIFIED: INCLUDE RECORDS WITH A VALUE GREATER THAN 5 IN POSITIONS 3 TO 5, THE LETTER A IN POSITION 8 AND A VALUE GREATER THAN 10 IN POSITIONS 11 TO 14.

135638

Figure 6-7. Example of Operand 2 Constants

SORT/MERGE SPECIFICATION, SHEET B

Page 1 2

Record Selection Specification

Line Number	TYPE (I/O)	Continuation (A/O)	(C/Z/D/S/P/U)	Operand 1 Location From	Operand 1 Location To	Rel. EQ NE LT GT LE GE	F/C	Operand 2 (Field or Constant) Location From	Operand 2 Location To	Operand 2 Constant	Comments
3 4 5	6	7	8	9 10 11 12	13 14 15 16	17 18	19	20 21 22 23	24 25 26 27	28 29 30 31 32 33 34 35 36 37 38 39	(Columns 40 to 72)
1 4	I	C		10	29	EQ	C	AJAX STORE'S SPECIAL			FIND
1 5	I	A	C	30	49	EQ	C	CUSTOMER CODE A2444			CHARACTER
1 6	I	A	C	50	51	EQ	C	01			STRING

SPECIFIED: INCLUDE RECORDS WITH A SPECIFIC CHARACTER STRING IN INPUT RECORD POSITIONS 10 THROUGH 51.

135639

Figure 6-8. Example of Multiline Constant Field

If operand 1 is packed, the last character in the constant must be its sign (+ or −). If operand 1 is unpacked and the constant is a negative number, the last character in the constant must indicate both the numeric value of the last digit and the negative sign for the entire constant (Figure 6-9). For a signed ASCII constant, the sign (+ or −) is always the last character of the constant.

In Figure 6-10, the entries specify sorting records that have a packed negative 2 (−2) in positions 2 and 3, an unpacked negative 27 in positions 6 through 9, an unpacked negative 10 in positions 12 through 17, and a signed ASCII negative 14 in positions 19 through 23.

6.2.9 Comment Lines (Columns 40 through 80)

To include annotated comments about a Record Selection Specification line, place the comments in columns 40 through 80. These comments will be printed with a specification summary if a zero or blank is placed in column 27 of the Header Specification.

6.2.10 Examples

Figures 6-11 through 6-15 show different forms of the completed Record Selection Specification.

Last Digit	Character to Code
0	}or] (minus zero code)
1	J
2	K
3	L
4	M
5	N
6	O
7	P
8	Q
9	R

Figure 6-9. Unpacked Constant Coding for Negative Values

SORT/MERGE SPECIFICATION, SHEET B

Page 1 2

Record Selection Specification

Line Number	TYPE (I/O)	Continuation (A/O)	(C/Z/D/S/P/U)	Operand 1 Location From	Operand 1 Location To	Rel. EQ NE LT GT LE GE	F/C	Operand 2 (Field or Constant) Location From	Operand 2 Location To	Operand 2 Constant	Comments (Columns 40 to 72)
14	I		P	2	3	NE	C	2	-		PACKED -2
15	I	A	U	6	9	NE	C	2	P		UNPACKED -27
16	I	A	U	12	17	EQ	C		1	1	UNPACKED -10
17	I	A	S	19	23	EQ	C	14	-		SIGNED ASCII -14
18											
19											
20											
21											
22											
23											
24											
25											
26											

Figure 6-10. Packed, Unpacked, and Signed ASCII Sort Specifications

SORT/MERGE SPECIFICATION, SHEET B

Page 1 2

Record Selection Specification

Line Number	TYPE (I/O)	Continuation (A/O)	(C/Z/D/S/P/U)	Operand 1 Location From	Operand 1 Location To	Rel. EQ NE LT GT LE GE	F/C	Operand 2 (Field or Constant) Location From	Operand 2 Location To	Operand 2 Constant	Comments (Columns 40 to 72)
14	I		C	7	9	EQ	F	17		19	
15											

SPECIFIED: INCLUDE ALL RECORDS IN WHICH THE CHARACTERS IN INPUT RECORD POSITIONS 7 TO 9 ARE THE SAME AS THOSE IN POSITIONS 17 TO 19. CHECK BOTH ZONE AND DIGIT PORTIONS OF CHARACTERS.

135640

Figure 6-11. Include Set Example

SORT/MERGE SPECIFICATION, SHEET B

Page 1 2

Record Selection Specification

Line Number	TYPE (I/O)	Continuation (A/O)	(C/Z/D/S/P/U)	Operand 1 Location From	Operand 1 Location To	Rel. EQ NE LT GT LE GE	F/C	Operand 2 (Field or Constant) Location From	Operand 2 Location To	Operand 2 Constant	Comments (Columns 40 to 72)
3 4 5	6	7	8	9 10 11 12	13 14 15 16	17 18	19	20 21 22 23	24 25 26 27	28 29 30 31 32 33 34 35 36 37 38 39	
1 4	O	C		30	33	EQ	C	DATA			OMIT IF DATA
1 5	O	AC		50	53	EQ	C	DATA			IN POSITIONS
1 6	O	OC		70	73	GT	C		10		OMIT IF > 10
1 7	I										INCLUDE-ALL

SPECIFIED: OMIT ALL RECORDS WHICH HAVE THE WORD DATA IN COLUMNS 30 THROUGH 33 AND 50 THROUGH 53 OR WHICH HAVE A VALUE GREATER THAN 10 IN COLUMNS 70 THROUGH 73.

135641

Figure 6-12. Omit Set Example

SORT/MERGE SPECIFICATION, SHEET B

Page 1 2

Record Selection Specification

Line Number	TYPE (I/O)	Continuation (A/O)	(C/Z/D/S/P/U)	Operand 1 Location From	Operand 1 Location To	Rel. EQ NE LT GT LE GE	F/C	Operand 2 (Field or Constant) Location From	Operand 2 Location To	Operand 2 Constant	Comments (Columns 40 to 72)
3 4 5	6	7	8	9 10 11 12	13 14 15 16	17 18	19	20 21 22 23	24 25 26 27	28 29 30 31 32 33 34 35 36 37 38 39	
1 4	O	C		5	9	GT	C	0 0 5 0	0		> 500
1 5	I	C			3	EQ	C	X			FIND X

SPECIFIED: INCLUDE ALL RECORDS WITH AN X IN POSITION 3 BUT OMIT ALL RECORDS WITH A VALUE GREATER THAN 500 IN POSITIONS 5 TO 9. CHECK BOTH ZONE AND DIGIT PORTIONS OF CHARACTERS.

135642

Figure 6-13. Example of Include and Omit Set Mixed

SORT/MERGE SPECIFICATION, SHEET B

Page 1 2

Record Selection Specification

Line Number	TYPE (I/O)	Continuation (A/O)	(C/Z/D/S/P/U)	Operand 1 Location From	Operand 1 Location To	Rel EQ NE LT GT LE GE	F/C	Operand 2 (Field or Constant) Location From	Operand 2 Location To	Operand 2 Constant	Comments (Columns 40 to 72)
3 4 5	6	7	8	9 10 11 12	13 14 15 16	17 18	19	20 21 22 23	24 25 26 27	28 29 30 31 32 33 34 35 36 37 38 39	
1 4	O	C		3	6	EQ	F	13	16		OMIT IF EQUAL
1 5	I										INCLUDE-ALL

SPECIFIED: INCLUDE EVERY RECORD EXCEPT THOSE IN WHICH THE VALUE IN POSITIONS 3 TO 6 IS EQUAL TO THE VALUE IN POSITIONS 13 TO 16. CHECK BOTH ZONE AND DIGIT PORTIONS OF CHARACTERS.

135643

Figure 6-14. Example of Omit and Include-All Set

SORT/MERGE SPECIFICATION, SHEET B

Page 1 2

Record Selection Specification

Line Number	TYPE (I/O)	Continuation (A/O)	(C/Z/D/S/P/U)	Operand 1 Location From	Operand 1 Location To	Rel EQ NE LT GT LE GE	F/C	Operand 2 (Field or Constant) Location From	Operand 2 Location To	Operand 2 Constant	Comments (Columns 40 to 72)
3 4 5	6	7	8	9 10 11 12	13 14 15 16	17 18	19	20 21 22 23	24 25 26 27	28 29 30 31 32 33 34 35 36 37 38 39	
1 4	I				7	EQ	C	Z			FIND Z OR
1 5	IO				8	EQ	C	Z			FIND Z
1 6	IA			10	15	GT	C		2000		> 2000

SPECIFIED: INCLUDE RECORDS WHICH HAVE THE LETTER Z IN EITHER POSITION 7 OR 8 AND WHICH HAVE A VALUE GREATER THAN 2000 IN POSITIONS 10 TO 15. CHECK BOTH ZONE AND DIGIT PORTIONS OF CHARACTERS.

135644

Figure 6-15. Example of Include Set with AND/OR Combination

7

Reformatting Specification

7.1 GENERAL

This section describes the Reformatting Specification and how to use this specification to designate the following:

- Arrangement of records in the output file
- Fields of the selected input records that will be used in the sort process
- Order of the fields of individual records in the work and output files (for example, customer name, part number, cost, and date)
- Sequence of individual fields (ascending order or descending order)
- ASCII character that will replace another character or be placed between fields in the work and output files (Sort/Merge does not change input records)
- Method of summing (adding) a selected field, the end result being the numeric total or the list of common data for the selected field

7.2 RECORDS AND FIELDS

Records that Sort/Merge inputs or outputs are divided into fields. No individual field can have more than 256 characters. However, you can combine individual fields to produce larger records. Thus, if a record contains 500 characters, the following field representations are valid:

- Two fields of 250 characters each
- One field of 256 characters and one field of 244 characters
- Any number of fields that have a combined length of 500 characters

Since the total length of the fields combined to produce the control (key) field is limited to 256 characters, the maximum length of the sort or merge is 256 characters. You can combine data fields to produce fields larger than 256 characters in length.

7.2.1 Control and Data Fields

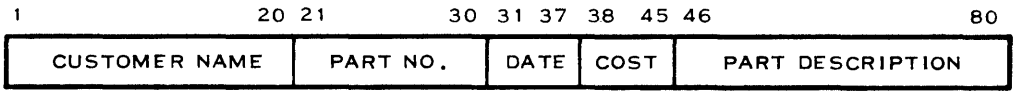
The Reformatting Specification designates one or more control fields that Sort/Merge uses as key fields for sequencing records or fields within records. A control field can be one character or a sequence of characters identified by position in the input record.

You can specify another type of control field to force a particular character into a specified position in the work file, which places the character in the output file. This character can replace the character currently in that position, or it can be placed into a work file position unconditionally.

Fields other than control fields are called *data fields*. As with control fields, you can force and sum data fields; however, you cannot use data fields as control fields for sequencing. Data fields tag along after their control fields.

In many cases, you will not want to include all of the characters of an input file record in the work file and output file records. You can view the input records as a collection of fields, each field being one or more characters identified by its character positions. For example, a customer name in positions 1 through 20 can be one field, a part number in positions 21 through 30 can be another field, and so on. By breaking the input record into a collection of fields, you can then identify the fields to be included in the Sort/Merge process.

For example, suppose that some of the records of the input file are divided into the following fields:

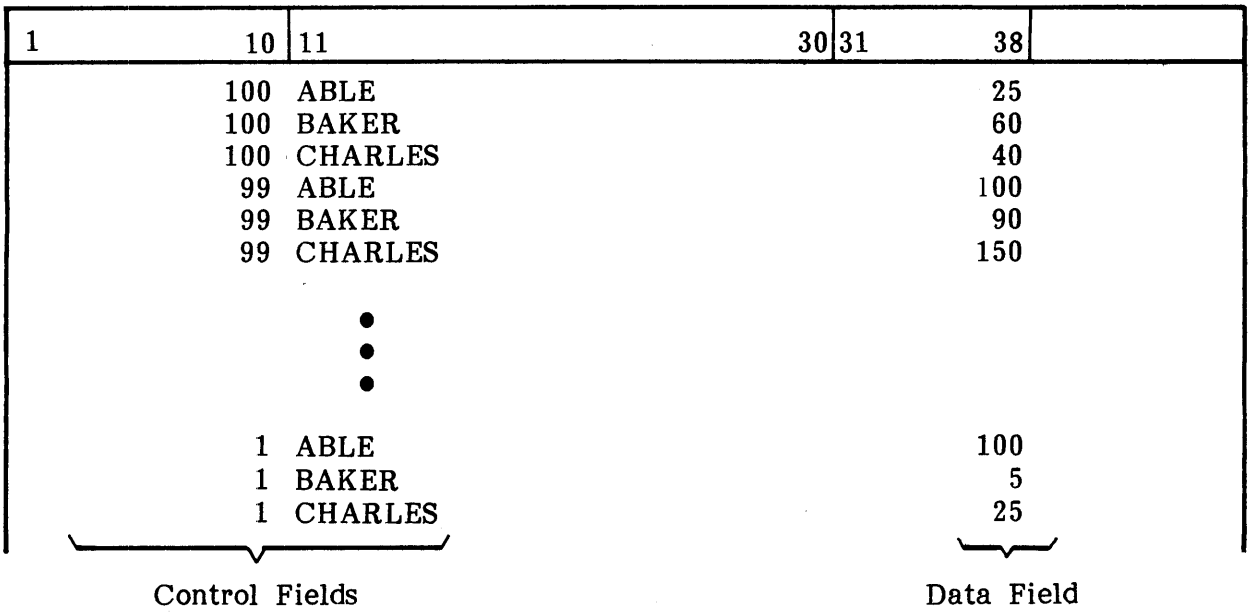


(A)135649

Through the Reformatting Specification, you can change the data structure in various ways. You can rearrange the fields by specifying that PART NO. be listed first and by deleting DATE and PART DESCRIPTION, as follows:

(A)135650

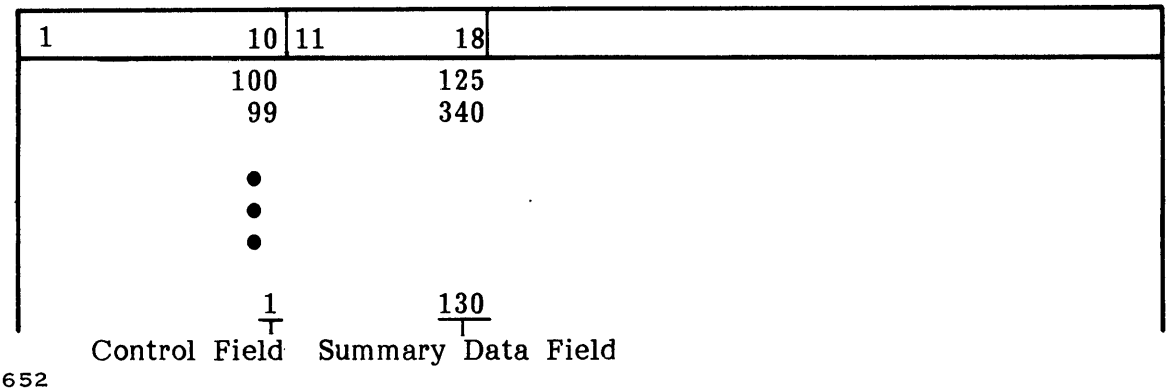
Also, you can sequence one field in ascending order (for example, Able, Baker, Charles in the name field) and another field in descending order (for example, 100, 99, 98 in the PART NO. column). Thus, the Reformatting Specification can specify exactly how the output records will appear. In this example, they appear as follows:



135651

Note that this file is sequenced by PART NO. and CUSTOMER NAME, while COST tags along with the first two fields. Since the part number and customer name fields are sequenced, they are the control fields; COST is a data field. In the work file, control fields will be first (left or high-order side) and data fields will follow (right or low-order side). Only data fields can be summary fields.

If the example file is formatted with PART NO. as a descending-order control field and COST as a summary data field, the following file results:

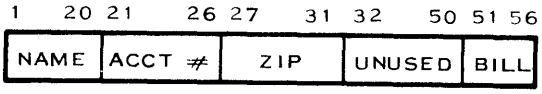


135652

Both control fields and data fields can be forced fields, allowing a character to be forced between fields or to be substituted for another character in a particular position. For example, you can force a dollar sign into the position preceding the COST data field.

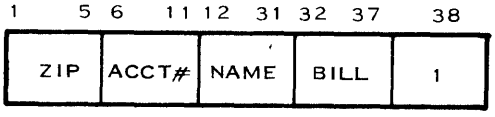
7.2.2 Field Placement within Reformatting Specification

The order in which you make entries on the Reformatting Specification indicates the order in which the fields are to appear on the output records. For example, suppose the following is the input record format:



(A)135645

Also, suppose the desired output format is as follows:



(A)135646

If the sort control field is the zip code and the sort control fields are not dropped, the following Reformatting Specification is used. (This section explains each specification in detail.)

Reformatting Specification

Line Number	Type	TYPE (N/O/F/D/S)	(C/Z/D/S/I/F/V/P/U)	Location From	Location To	Forced Record Character	Forced Substitute Character	Forced Continuation	Overflow	Reserved	Comments
				9 10 11 12	13 14 15 16	17	18	19	20 21 22	23 24 25 26 28 29 30 31 32 33 34 35 36 37 38 39	(Columns 40 to 72)
270	F	NC		27	31						
280	F	DC		21	26						
290	F	DC		1	20						
300	F	DC		51	56						
310	F	DV					1				

135647

If the sort control field is ACCT # and the same output format is required, drop the control fields and use the following Reformatting Specification:

Reformatting Specification

Line Number	Type	TYPE (N/O/F/D/S)	(C/Z/D/S/I/F/V/P/U)	Location From	Location To	Forced Record Character	Forced Substitute Character	Forced Continuation	Overflow	Reserved	Comments
3 4 5	6	7	8	9 10 11 12	13 14 15 16	17	18	19	20 21 22	23 24 25 26 28 29 30 31 32 33 34 35 36 37 38 39	(Columns 40 to 72)
270	F	NC		21	26						
280	F	DC		27	31						
290	F	DC		21	26						
300	F	DC		1	20						
310	F	DC		51	56						
320	F	DV					1				

135648

Note in the second Reformatting Specification that the ACCT # field (columns 21 through 26) is specified twice—once as a control field and once as a data field. When the control fields are dropped, this control field does not appear in the output file and the output format is as previously defined.

Use the following data field guidelines:

- Describe all control field lines before data field lines for each record type.
- The number of data fields need not be the same for all record types.
- The total lengths of data fields for one record type need not equal the lengths of data fields in other types. Shorter fields will be blank-filled on the low-order end.

For control fields, follow these guidelines:

- The number of control fields need not be the same for all record types.
- The total lengths of control fields in one record type need not equal the lengths of control fields in other record types.

NOTE

When two records with identical control field contents are sorted, the actual order of the records in the output file is unpredictable.

7.3 COLUMN DESCRIPTIONS

Table 7-1 summarizes the entries on the Reformatting Specification.

Table 7-1. Reformatting Specification Column Summary

Prompt	Columns	Entries	Explanation	Paragraph
—	1-2[1]	01-99	Page number.	4.2.1
—	3-5[1]	27x-40x	Preprinted; numbers specify sequence of specification lines.	4.2.1
—	6[1]	F	Preprinted; designates a line in the Reformatting Specification.	7.3.2
FT	7[1]		Field type:	7.3.3
		N	— Normal sequence.	7.3.3.1
		O	— Opposite sequence.	7.3.3.1
		F	— Forced control field.	7.3.3.2
		D	— Data field.	7.3.3.3
		S	— Summary data field.	7.3.3.4
CP	8[1]		Character portion:	7.3.4
		C[2]	— Use both zone and digit portions.	7.3.4
		Z	— Use only the zone portion (1-character only).	
		D	— Use only the digit portion.	7.3.4.1
		V	— Force a character into a data field.	7.3.4.1
		S	— Field is signed ASCII format.	7.3.4.2
		I	— Field is signed integer format.	7.3.4.3
		F	— Field is floating point format.	7.3.4.4
		P	— Field is a signed packed decimal number.	7.3.4.5
		U	— Field is a signed unpacked decimal number.	7.3.4.5
		L	— Field is unpacked format with a leading sign.	7.3.4.6
		B	— Field is signed ASCII with a beginning sign.	7.3.4.7
BL	9-12[1]		Beginning location:	7.3.5
		Blank[2]	— Field contains only one character.	
		1-4096	— Starting position of field in record.	
EL	13-16[1]		Ending location:	7.3.5
		1-4096	Last position of field in record.	
RC	17[3]		Record character:	7.3.6
		ASCII character	Character to be conditionally forced into position specified in columns 13-16.	

Table 7-1. Reformatting Specification Column Summary (Continued)

Prompt	Columns	Entries	Explanation	Paragraph
SC	18[3]	ASCII character	Substitute character: Force this character between fields.	7.3.7
CON	19[3]	Blank[2] Nonblank ASCII character	Continuation: — Not a continuation of a conditional force. — Continuation of the preceding conditional force.	7.3.8
OVF	20-22	Integer	Overflow: For summary sorts and merges; specify total positions required to contain resulting sum.	7.3.9
—	40-80		Comments; enter up to 41 ASCII characters.	

Notes:

[1] Columns that must be filled for all sorts.

[2] Default values.

[3] Columns that must be filled when using forced control fields.

7.3.1 Page Numbers and Line Numbers (Columns 1 through 5)

The page numbers and line numbers are described in paragraph 4.2.1.

7.3.2 Line Type (Column 6)

This column contains the preprinted letter F, identifying these lines as Reformatting Specifications.

7.3.3 FT — Field Type (Column 7)

The following summarizes the column 7 entries:

Column 7 Entry	Explanation	Paragraph Number
N	Normal sequencing, as specified in column 18 of the Header Specification. This is a control field.	7.3.3.1

Column 7 Entry	Explanation	Paragraph Number
O	Opposite sequencing, as specified in column 18 of the Header Specification. This is a control field.	7.3.3.1
F	Forcing of a character into a particular position. The character can also substitute for a character already in a position. This is a control field.	7.3.3.2
D	Data field.	7.3.3.3
S	Summary data field. Data in this field is to be summed (added).	7.3.3.4

7.3.3.1 Use of Normal and Opposite Control Fields (N or O in Column 7). An N in this column indicates that fields are to be sequenced in either ascending or descending order as specified in column 18 of the Header Specification. The letter O in this column specifies that the sequence is to be opposite the order specified in column 18.

Figure 7-1 shows a Reformatting Specification that specifies that the output file is to be sequenced according to the data in input record positions 1 through 4, 5 through 9, and 11 through 15. Data in positions 1 through 4 and 5 through 9 will be sequenced in the order specified in column 18 of the Header Specification; data in input record positions 11 through 15 will be sequenced in the order opposite that specified in column 18 of the Header Specification.

7.3.3.2 Forced Control Field (F in Column 7). You might need to force a character into a record position and then use that character in the sequencing process. For example, in selecting different record types, for three different include sets, you can force the first character of each record type to a 1, 2, or 3. Then, a sort controlled by these first numbers results in the output file containing the types grouped in the order 1, 2, 3.

Reformatting Specification

Line Number	Type	TYPE (N/O/F/D/S)	(C/Z/D/S/I/F/V/P/U)	Location From	Location To	Forced Record Character	Forced Substitute Character	Forced Continuation	Overflow	Reserved	Comments
3 4 5	6	7	8	9 10 11 12	13 14 15 16	17	18	19	20 21 22	23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39	(Columns 40 to 72)
2 7	F	N	C	1	4						
2 8	F	N	C	5	9						
2 9	F	O	C	11	15						

SPECIFIED: SEQUENCE DATA IN INPUT RECORD POSITIONS 1 TO 4 AND 5 TO 9 IN NORMAL ORDER ACCORDING TO COLUMN 18 OF THE HEADER SPECIFICATION. SEQUENCE DATA IN INPUT RECORD POSITIONS 11 TO 15 IN OPPOSITE ORDER.

135653

Figure 7-1. Example of Specifying Normal and Opposite Order Sequencing

You can also specify a data field force (paragraph 7.3.3.3); however, you cannot use data field force to sequence data.

The three types of forces are as follows:

- Conditional force — A character is forced into a specific position in the work file only if a certain character presently occupies that position. For example, an X is placed in position 4 only if the number 1 is now in that position. If the character is not found and ascending order was specified, a character with the code FF_{16} is substituted. If the character is not found and descending order was specified, a null character (hexadecimal zero) is substituted. One position can have more than one conditional force. For example, an X in position 7 is changed to a 1; a Y in position 7 is changed to a 2; a Z in position 7 is changed to a 3; and so on.

- Force-all — If the character specified in a conditional force is not found, an alternate character is forced into that position. For example, a conditional force looks for the character Z in input record position 23; if found, the Z is changed to number 1 in the work file records. If a Z is not found, all such records will have a null character in that position if the order is ascending. The force-all prevents the position from being forced to a zero if Z is not found. A force-all, if used, must follow a conditional force.

- Unconditional force — A specific character is forced to a position regardless of the character currently in that position.

For a conditional force, use the following steps:

1. Enter an F in column 7 to indicate a force.

2. Enter the location of the character to be changed in columns 13 through 16, right-justified. (You can place blanks or zeros in the unused columns.)

3. Enter the character to be changed (conditional character) in column 17. If this character is not found and a force-all force does not follow this conditional force, the position indicated is changed either to a null character if descending order was specified or to FF_{16} (binary ones) if ascending order was specified.

4. Enter the substitute character in column 18.

5. If the same position has more than one conditional force, enter the additional conditional forces on the lines following the first; also, enter any nonblank ASCII character in column 19 to indicate a continuation. You can specify only one character in one specification line. Figure 7-2 is an example of a continuing conditional force operation.

For a force-all force, use the following steps:

1. Enter the force-all line after its accompanying conditional force line.

2. Enter an F in column 7 to indicate a force.

3. Enter the substitute character in column 18.

Reformatting Specification

Line Number	Type	TYPE (N/O/F/D/S)	(C/Z/D/S/I/F/V/P/U)	Location From	Location To	Forced Record Character	Forced Substitute Character	Forced Continuation	Overflow	Reserved	Comments (Columns 40 to 72)
27	F	F	C		7	X	1				COND. FORCE
28	F	F	C		7	Y	2	X			COND. FORCE
29	F	F	C		7	Z	3	X			COND. FORCE

SPECIFIED: CHANGE THE CHARACTER IN POSITION 7 AS FOLLOWS:
IF AN X, CHANGE TO A 1.
IF A Y, CHANGE TO A 2.
IF A Z, CHANGE TO A 3.

135654

Figure 7-2. Conditional Force Example

4. Enter a nonblank ASCII character in column 19 to indicate that this line is a continuation of the preceding conditional force line.

5. Do not enter any characters in columns 9 to 17. (Notice that the record position is not specified as in the preceding conditional force line.)

Figure 7-3 is an example of a force-all following a conditional force.

Reformatting Specification

Line Number	Type	TYPE (N/O/F/D/S)	(C/Z/D/S/I/F/V/P/U)	Location From	Location To	Forced Record Character	Forced Substitute Character	Forced Continuation	Overflow	Reserved	Comments (Columns 40 to 72)
27	F	F	C		15	A	1				COND. FORCE
28	F	F	C		15	B	2	X			COND. FORCE
29	F	F	C				$	X			FORCE-ALL

SPECIFIED: CHANGE THE CHARACTER IN POSITION 15 AS FOLLOWS:
IF AN A, CHANGE TO A 1 (CONDITIONAL FORCE).
IF A B, CHANGE TO A 2 (CONDITIONAL FORCE).
IF NEITHER AN A OR B, CHANGE TO A DOLLAR SIGN
(FORCE-ALL FORCE).

135655

Figure 7-3. Example of a Force-All Force Following Conditional Forces

For an unconditional force, use the following steps:

1. Enter an F in column 7 to indicate a force.
2. Enter the character to be forced in column 18.
3. Do not enter any characters in columns 9 through 17.

For example, if an input record with dollar values is in positions 20 through 24 and 30 through 34, you can reformat the output record with data in positions 30 through 34 by specifying this field first in the Reformatting Specification. By using the unconditional force and placing a character(s) before or after specific fields, the character(s) can be unconditionally forced between fields, in front of fields, and so on. Do not specify a location for the character in columns 9 through 16; the character will be placed in the work and output files according to its positions in regard to other fields, as shown in the Reformatting Specification.

Figure 7-4 is an example of an unconditional force.

Remember the following rules for forced control fields:

- Only the work file and output file records are affected (the input records are not changed).
- If a force-all does not follow a conditional force and the conditional character is not found in the one-position field specified, the contents of that position are changed either to null characters (hexadecimal zero) for a descending format or to a character with code FF_{16} (binary ones) for an ascending format.
- You can specify only one forced character per line on the Reformatting Specification.
- To describe more than one conditional force for the same position, use additional conditional force lines with a nonblank character in column 19.
- Unconditional forces place characters in the next work record position in the order in which fields appear on the Reformatting Specification.

7.3.3.3 Data Fields (D in Column 7). A data field (D entry) is appropriate for tag-along sorts only (SORTR and SORTRS). In an address-only sort, a data field is treated as a comment. Data fields are units of information that are included in the output records but not used in the sequencing of the file.

NOTE

Within an include set, you must describe all control field lines before any data field lines.

Paragraph 7.3.3.1 describes how data fields are formatted in work files when you specify more than one record type (in more than one Record Selection Specification).

Reformatting Specification

File Number	Type	TYPE (N/O/F/D/S)	(C/Z/D/S/I/F/V/P/U)	Location From	Location To	Forced: Record Character	Forced: Substitute Character	Forced: Continuation	Overflow	Reserved	Comments
3 4 5	6	7	8	9 10 11 12	13 14 15 16	17	18	19	20 21 22	23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39	(Columns 40 to 72)
2	F	F	C				1				UNCOND. FORCE
2	F	N	C	20	24						NORMAL SEQ

SPECIFIED: FORCE THE NUMBER 1 INTO THE FIRST POSITION OF THE WORK FILE AND THEN USE THIS NUMBER IN THE SEQUENCING OF DATA. THIS SPECIFICATION AND ITS ACCOMPANYING RECORD SELECTION SPECIFICATION CAN BE USED WITH OTHER SUCH SPECIFICATION SETS (EACH WITH A DIFFERENT NUMBER FORCED INTO POSITION 1) TO SEQUENCE RECORDS OF DIFFERENT RECORD TYPES (AS SELECTED BY THE RECORD SELECTION SPECIFICATION).

(A)135656A

Figure 7-4. Example of Unconditional Force

As with control fields, you can also force data fields; however, data fields (forced or not) are not used for output file sequencing. The position of the data fields in the Reformatting Specification is the order of the fields in the output records.

To specify a data field, use the following steps:

1. Enter a D in column 7.
2. If a field consists of only one character, leave columns 9 through 12 blank and enter the column location of the one character in columns 13 through 16, right-justified. If a field is longer than one character, enter the beginning character position of the data field in columns 9 through 12, right justified, and the ending character position in columns 13 through 16, right justified. Right-justify position numbers with zeros or blanks on the left.

To specify a forced data field, the following steps are necessary:

1. Enter a D in column 7.
2. Enter a V in column 8.
3. Enter the character to be forced in column 18.
4. Leave columns 9 through 17 blank.
5. The character in column 18 is forced into the next available position on the work record.

Figure 7-5 shows examples of both forced and unforced data fields.

Reformatting Specification

Line Number (3 4 5)	Type (6)	TYPE (N/O/F/D/S) (7)	(C/Z/D/S/I/F/V/P/U) (8)	Location From (9 10 11 12)	Location To (13 14 15 16)	Forced: Record Character (17)	Forced: Substitute Character (18)	Forced: Continuation (19)	Overflow (20 21 22)	Reserved (23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39)	Comments (Columns 40 to 72)
27	F	NC		20	24						
28	F	DC		10	12						
29	F	DV					b				BLANK SPACE
30	F	DV					$				DOLLAR SIGN
31	F	DC		4	9						

SPECIFIED: SEQUENCE THE FILES BY DATA IN INPUT RECORD POSITIONS 20 TO 24. IF THE CONTROL FIELD IS TO BE INCLUDED IN THE OUTPUT RECORDS, IT IS FOLLOWED BY THE DATA IN POSITIONS 10 TO 12, A BLANK (FORCED DATA), A DOLLAR SIGN (FORCED DATA), AND THEN THE DATA IN POSITIONS 4 TO 9.

135657

Figure 7-5. Example of Forced and Unforced Data Fields

7.3.3.4 Summary Data Fields (S in Column 7). Summary fields are fields of data that are summed (added). Figure 7-6 is an example of a Reformatting Specification to summarize sales records. The fields within these records include part number, sales quantity by day, and dollar amounts of sales by day. This specification creates a summary record showing total number of items sold and total sales (the two summary data entries) according to part number (the control field entry).

NOTE

The sum might require more spaces than the size of the summary data field. Use columns 20 through 22 to allow for overflow, as explained in paragraph 7.3.9.

You can specify summary fields for any Sort/Merge type, but addition actually occurs only in a summary sort (SORTRS) or summary merge (MERGES). In a tag-along sort (SORTR) or normal merge (MERGE), summary fields are treated as data fields. In an address-only sort (SORTA), summary field descriptions are treated as comments.

You can summarize no more than 24 fields for each record type, and you can sum only the zone portion of characters.

The first include set that contains summary specifications defines the summary format for all included records. All include sets should contain the same summary specifications if possible; the data specification should align the data for summarizing.

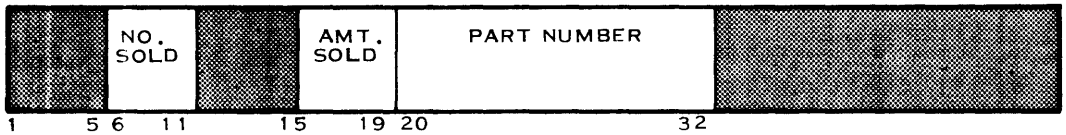
Sort/Merge sorts all records according to control fields first; then, it summarizes the summary data fields. If a SORTRS is requested but no summary data fields are defined in the Reformatting Specification, Sort/Merge outputs one record for each control field; this record will be the first record sorted. In other words, Sort/Merge retains only one record for each specific control field.

7.3.4 CP — Character Portion [C] (Column 8)

The column 8 entry further defines the fields by specifying the following:

- Whether the fields contain characters, digits, zones, signed ASCII characters, signed integers, floating point numbers, or packed or unpacked numeric data
- Whether this line is forcing a data character
- Whether the sort uses the entire ASCII character or only the zone or digit portion
- Overflow indication

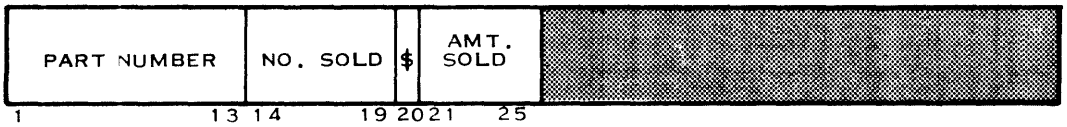
INPUT RECORDS



Reformatting Specification

Line Number	Type	TYPE (N/O/F/D/S)	IC/Z/D/S/I/F/V/P/U)	Location From	Location To	Forced Record Character	Forced Substitute Character	Forced Continuation	Overflow	Reserved	Comments (Columns 40 to 72)
27	F	N	C	20	32						PART NO.
28	F	S	C	6	11						NO. SOLD
29	F	D	U				$				DOLLAR
30	F	S	C	15	19						SOLD

OUTPUT RECORD



SPECIFIED: SORT A SERIES OF INPUT RECORDS THAT SHOWS THE NUMBER SOLD AND DOLLAR AMOUNT SOLD OF DIFFERENT PART NUMBERS. SORT THESE INTO AN OUTPUT FILE WITH EACH RECORD SHOWING THE TOTAL NUMBER SOLD AND TOTAL DOLLAR AMOUNT SOLD FOR EACH PART NUMBER. FORCE A DOLLAR SIGN BEFORE THE TOTAL AMOUNT SOLD.

NOTE: THE SUM MAY REQUIRE MORE SPACES THAN THE SIZE OF THE SUMMARY DATA FIELD. TO PROVIDE SUFFICIENT ROOM FOR THE SUM, ALLOW FOR AN OVERFLOW USING COLUMNS 20 TO 22.

(A)135658A

Figure 7-6. Example of Summary Data Fields

Note that the Record Selection Specification also uses the C, Z, D, P, U, and S entries in column 8; however, the V, I, F, L, and B entries are unique to the Reformatting Specification. The I and F entries identify fields containing signed integers and floating point numbers, respectively. I format supports both single and extended integer formats. F format supports both real and double precision numbers. You can use the I, F, L, and B fields only as control fields (with an N or O in column 7) and in record selection or summation. For example, you cannot specify that records with a −3.56 be included in or omitted from a sort or merge.

Tables 7-2 and 7-3 summarize the combinations of column 7 and column 8 entries and the maximum amount of characters that can be in the accompanying data or control field for that combination. Use the V entry in column 8 only with a D or S in column 7, as shown in Table 7-3. Forced fields can only be one character in length (an F or V in column 8).

You can designate C, Z, D, P, U, or S in column 8 for both the Record Selection Specification and the Reformatting Specification. However, their differences are apparent:

- The Record Selection Specification uses this column entry to determine which records to include in the operation. This specification compares characters in two different fields or compares one field to a character string.
- The Reformatting Specification determines the sequencing of the records selected by ordering them in ascending or descending sequence, according to the numerical values in designated fields.

NOTE

Results of the Sort/Merge program are unpredictable if you enter an operator other than C, Z, D, V, S, I, F, P, U, L, or B in column 8.

The following summarizes the column 8 entries:

Column 8 Entry	Explanation
C[1]	Use both the zone and digit portions of ASCII characters.
Z[1]	Use only the zone portion of ASCII characters.
D[1]	Use only the digit portion of ASCII characters.
V[1]	Force the data character (in column 18) or summary overflow character into the next position in the work file record.
S[1]	This field contains a signed ASCII value.
I[2]	This field contains a signed integer value.
F[2]	This field contains a floating point value.
P[1]	This field contains packed numeric data.

Column 8 Entry	Explanation
U[1]	This field contains unpacked numeric data.
L[1]	This field contains unpacked numeric data with a leading sign.
B[1]	This field contains ASCII data with a beginning sign.

Notes:

[1] Field location identified by beginning and ending column positions (when applicable).

[2] Must be used with a control field (N or O in column 7) only.

Table 7-2. Possible Column 8 Entries for Column 7 Entry

Column 7 Entry	Column 8 Entry	Maximum Field Length (In Characters)
N or O	C	256
	Z	1
	D	16
	S	16
	I	2, 4, or 8
	F	4 or 8
	P	8
	U	16
F	C	1
	Z	1
	D	1
	L	16
	B	16
D	C	256
	Z	1

Table 7-2. Possible Column 8 Entries for Column 7 Entry (Continued)

Column 7 Entry	Column 8 Entry	Maximum Field Length (in Characters)
	D	16
	V	1
	P	8
	U	16
	L	16
	B	16
S	C	256
	D	16
	S	16
	V	1
	P	8
	U	16

Table 7-3. Possible Column 7 Entries for Column 8 Entry

Column 7 Entry	Column 8 Entry	Maximum Field Length (in Characters)
N, O, D, or S	C	256
F		1
N, O, F, or D	Z	1
N, O, D, or S	D	16
F		1
D or S	V	1
N, O, or S	S	16
N or O	I	2, 4, or 8

Table 7-3. Possible Column 7 Entries for Column 8 Entry (Continued)

Column 7 Entry	Column 8 Entry	Maximum Field Length (In Characters)
N or O	F	4 or 8
N, O, D, or S	P	8
N, O, D, or S	U	16
N, F, or O	L	16
N, F, or O	B	16

7.3.4.1 Character Portion and Data Force (C, Z, D, V, S, I, F, P, U, L, or B in Column 8). The column 8 entry specifies which portion of the character (zone, digit, or both) should be used to determine the sequencing of data in the work and output files. It also designates whether a data field is a forced data field. Observe the following when using these entries:

- You can use any of the column 7 entries.
- The beginning and ending positions of input records specify the field length.

For sorting in ascending or descending order, use both the zone and digit portions of the characters since the ASCII code values use both portions to define the entire character. It may be possible to sort numerical-only values by using only the digit portion since only the digit portion of the ASCII code varies for numbers (the zone portion is the same for all numbers). Appendix C provides a complete list of ASCII codes and their binary portions.

For example, if a one-character control field for three input records contains the ASCII code for the characters A (41_{16}), B (42_{16}), and 5 (35_{16}), the records are sorted in a different ascending order, depending on the column 8 entry, as shown below:

Column 8 = C	Column 8 = Z	Column 8 = D
5	5	A
A	A[1]	B
B	B[1]	5

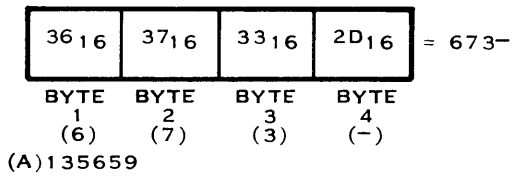
Note:

[1] Letters will follow in unpredictable order.

The letter V in column 8 signifies that this is a forced entry. (See Figure 7-5.)

7.3.4.2 Signed ASCII (S in Column 8). You can use signed ASCII fields for sorting. Note the following rules:

- The sign must be in the least-significant (rightmost) byte of the field.
- The field must be a control or summary field for a sort or merge (that is, N, O, or S entry in column 7).
- The beginning and ending input record positions specify the field length.
- Signed ASCII numbers have the sign in the least-significant byte of the character string. For example, the number −673 is constructed as follows:



The S in column 8 tells Sort/Merge to treat the rightmost byte of the field as a sign. If a blank (20_{16}) or a plus sign ($2B_{16}$) is found in the sign byte, the number is positive; if a minus sign ($2D_{16}$) is found, it is negative. Figure 7-7 shows an example of an input file containing a signed ASCII field and a Reformatting Specification to include the values in this field.

7.3.4.3 Signed Integer (I in Column 8). A signed integer contains a sign indicator in its most significant bit and is a two's complement value. The following are rules for signed integers:

- The integer must be a control field for sorting or merging only (that is, N or O entry in column 7).
- Records may contain binary data as well as ASCII characters. Values can be multiple precision, and the field normally begins on a word boundary (that is, in an odd-numbered column of the input file).

The I in column 8 tells Sort/Merge to treat the field as a signed integer with the sign bit as the most significant bit. Place the starting position of the field in columns 9 through 12. Usually, this value is an odd number. (Since integers must be on a word boundary, input record positions 1, 3, and so on are in memory on a word boundary.) Both the From and To location entries are right-justified.

Figure 7-8 shows both single and extended integer entries.

7.3.4.4 Floating Point Numbers (F in Column 8). The first byte of floating point fields contains the sign bit (most significant bit) and a seven-bit exponent expressing the values of 16^{-64} to 16^{63}. The following are several rules for floating point numbers:

- The field must be a control field for sorting or merging only (that is, N or O entry in column 7).
- Fields can be multiple precision and normally begin on a word boundary (that is, in an odd-numbered column of the input file).

INPUT RECORDS

```
1          9 10          16 17                                        80
+-----------+-------------+-------------------------------------------+
| PART NO.  |     QTY     |              PART DESCRIPTION             |
+-----------+-------------+-------------------------------------------+
|  1000        325-          WIDGET NO.  1                           |
|  1001        400           WIDGET NO.  2                           |
|  1074        1000-         WIDGET NO. 75                           |
+-----------+-------------+-------------------------------------------+
```

Reformatting Specification

Line Number	Type	TYPE (N/O/F/D/S)	(C/Z/D/S/I/F/V/P/U)	Location From	Location To	Forced Record Character	Forced Substitute Character	Forced Continuation	Overflow	Reserved	Comments
3 4 5	6	7	8	9 10 11 12	13 14 15 16	17	18	19	20 21 22	23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39	(Columns 40 to 72)
2 7	F	N	S	1 0	1 6						SIGNED ASCII

SPECIFIED: SORT IN NORMAL ORDER THE SIGNED ASCII FIELD IN POSITIONS 10 TO 16 OF THE INPUT RECORD (INCLUDING THE SIGN BYTE).

(A)135660

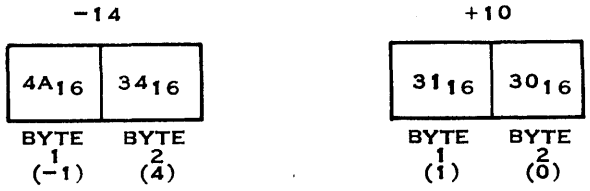
Figure 7-7. Specifying Signed ASCII Control Field

Entries in the From and To columns must be right-justified, and the From column entry may be an odd-numbered position of the input file. Figure 7-9 shows entries for a floating point field.

7.3.4.5 Packed and Unpacked Numeric Data (P or U in Column 8). If you specify packed or unpacked control fields (N or O in column 7), Sort/Merge modifies the control fields while building the work record. Therefore, it is recommended that packed and unpacked control fields be dropped (by entering X in column 28 on the header line).

7.3.4.6 Leading Signed Unpacked Numeric Data (L in Column 8). If you specify leading signed unpacked control fields (N or O in column 7), Sort/Merge modifies the control fields while building the work record. It is recommended that unpacked control fields be dropped (by entering X in column 28 on the header line). The example in Figure 7-11 shows leading signed unpacked numeric data.

For example, the numbers -14 and +10 are constructed as follows:



(A)143698

Reformatting Specification

Line Number (3 4 5)	Type (6)	TYPE (N/O/F/D/S) (7)	(C/Z/D/S/I/F/V/P/U) (8)	Location From (9 10 11 12)	Location To (13 14 15 16)	Forced: Record Character (17)	Forced: Substitute Character (18)	Forced: Continuation (19)	Overflow (20 21 22)	Reserved (23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39)	Comments (Columns 40 to 72)
27	F	N	I	5	6						SINGLE PRECISION
28	F	N	I	7	10						DOUBLE PRECISION

SPECIFIED: SORT IN NORMAL SEQUENCE SINGLE PRECISION INTEGERS (TWO BYTES IN LENGTH) BEGINNING IN INPUT RECORD POSITION 5. SORT IN NORMAL SEQUENCE DOUBLE PRECISION INTEGERS (FOUR BYTES IN LENGTH) BEGINNING IN INPUT RECORD POSITION 7.

135661

Figure 7-8. Specifying Integer Control Fields

Reformatting Specification

Line Number (3 4 5)	Type (6)	TYPE (N/O/F/D/S) (7)	(C/Z/D/S/I/F/V/P/U) (8)	Location From (9 10 11 12)	Location To (13 14 15 16)	Forced: Record Character (17)	Forced: Substitute Character (18)	Forced: Continuation (19)	Overflow (20 21 22)	Reserved (23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39)	Comments (Columns 40 to 72)
27	F	NF		11	14						
28	F	OF		15	22						

SPECIFIED: SORT IN NORMAL ORDER THE FOUR-BYTE FLOATING POINT VALUE THAT BEGINS IN POSITION 11. SORT IN OPPOSITE ORDER THE EIGHT-BYTE VALUE THAT BEGINS IN POSITION 15

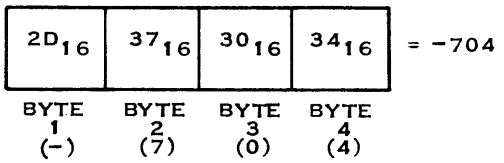
135662

Figure 7-9. Specifying Floating Point Control Fields

7.3.4.7 Beginning Signed ASCII (B in Column 8). You can use beginning signed ASCII fields for sorting. The following are several rules for using beginning signed ASCII:

- The sign must be in the most significant (leftmost) byte of the field.
- The field must be a control field for sorting or merging only (that is, N or O entry in column 7).
- The beginning and ending input record positions specify the field length.

Beginning signed ASCII numbers have the sign in the most significant byte of the character string. For example, the number −704 is constructed as follows:



(A)143699

The B in column 8 instructs Sort/Merge to treat the leftmost byte of the field as a sign. If a blank (20_{16}) or a plus sign ($2B_{16}$) is found in the sign byte, the number is positive; if a minus sign ($2D_{16}$) is found, the number is negative. Figure 7-10 shows an example of an input file containing a beginning signed ASCII field and a Reformatting Specification to include the values in this field.

7.3.5 BL and EL — Beginning and Ending Locations [Blank] (Columns 9 through 16)
These columns identify the beginning and ending positions in the input record of the field being sorted or merged.

Place the number of the beginning position in the From columns and the number of the ending position (when applicable) in the To columns. Both entries must be right-justified and blank- or zero-filled on the left. If the field contains only one character, leave the From column blank and enter the character's position in the To column (right-justified). Column entries are summarized as follows:

Columns	Entry	Explanation
9–12	Blank or 1-4096	Position in input record identifying start of field
13–16	1-4096	Position in input record identifying end of field

7.3.6 RC — Record Character [Blank] (Column 17)
To conditionally force a character (if a certain character is in a specific input record position), place the conditional character in column 17. (See paragraph 7.3.3.2 and Figure 7-2 for a complete explanation of conditional forcing and the use of this column.)

INPUT RECORDS

```
1          9 10          16 17                                        80
PART NO.   QTY             PART DESCRIPTION
1000        -325           WIDGET NO.  1
1001        -400           WIDGET NO.  2
1074       -1000           WIDGET NO. 75
```

Reformatting Specification

Line Number	Line Type	TYPE (N/O/F/D/S)	(C/Z/D/S/I/F/V/P/U)	Location From	Location To	Forced Record Character	Forced Substitute Character	Forced Continuation	Overflow	Reserved	Comments
3 4 5	6	7	8	9 10 11 12	13 14 15 16	17	18	19	20 21 22	23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39	(Columns 40 to 72)
2 7	F	N	B	1 0	1 6						BEGINNING ASCII

SPECIFIED: SORT IN NORMAL ORDER THE BEGINNING SIGNED ASCII FIELD IN POSITIONS 10 TO 16 OF THE INPUT RECORD (INCLUDING THE SIGN BYTE).

(A)143712

Figure 7-10. Specifying Beginning Signed ASCII Control Fields

7.3.7 SC — Substitute Character [Blank] (Column 18)

To force a character into a specified position, place the character in column 18. (This applies whether the character is a control field force or a data field force.) Paragraph 7.3.3.2 provides a complete explanation of control field forcing and the use of this column; paragraph 7.3.3.4 and Figure 7-5 provide a complete explanation of data field forcing and the use of column 18.

7.3.8 CON — Continuation [Blank] (Column 19)

Use this column to set two or more conditions for a conditional force of one control field. For instance, to make a conditional force on one input record position if the character in that position is X, Y, or Z, place the succeeding conditions on successive lines. To make the entries a continuation of the first condition, place a nonblank ASCII character in column 19 on the lines following the first condition. Paragraph 7.3.3.3 and Figure 7-2 provide a complete explanation of a continued conditional force.

7.3.9 OVF — Overflow (Columns 20 through 22)

To ensure against an overflow condition on a summary sort or merge, you can specify additional spaces for an overflow field in columns 20 through 22. Since the sum of values contained in a certain number of columns can take up more space than any individual number, this entry provides additional space for the overflow. This value must be the sum of the summary data field plus enough bytes for the overflow. The entry is right-justified, ending in column 22, and cannot be more than the maximum field length of the data type to be summed (as specified in Tables 7-2 and 7-3).

For example, if the summary field is three characters long and five characters are needed for the sum, enter the number 5 in column 22, as shown in Figure 7-11.

Reformatting Specification

Line Number	Type	TYPE (N/O/F/D/S) (C/Z/D/S/I/F/V/P/U)	Location From	Location To	Forced Record Character	Forced Substitute Character	Forced Continuation	Overflow	Reserved	Comments
3 4 5	6	7	9 10 11 12	13 14 15 16	17	18	19	20 21 22	23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39	(Columns 40 to 72)
2	F	NC	21	29						
2	F	SC	11	13				5		

SPECIFIED: SORT RECORDS ACCORDING TO THE DATA IN POSITIONS 21 TO 29, AND SUM THE VALUES IN POSITIONS 11 TO 13 FOR EACH CONTROL FIELD. PROVIDE AN OVERFLOW OF TWO ADDITIONAL CHARACTERS FOR THE SUM.

135663

Figure 7-11. Example of Overflow as Specified in Columns 20 through 22

7.3.10 Specifying an Overflow Indicator Field

To set an overflow indicator, perform the following steps:

1. Enter an S in column 7.
2. Enter a V in column 8.
3. If an overflow occurs in any summary sort or merge, the initial value of the overflow field is replaced by the character specified in column 17. If no character is specified, an asterisk (*) is used as a replacement character.
4. If no overflow occurs in any summary sort or merge, the overflow field contains the character specified in column 18.

8

Sort/Merge SCI Interface and Execution

8.1 GENERAL

The System Command Interpreter (SCI) serves as the single interface between the user and the operating system. The SCI accepts input from interactive devices, such as data terminals or video display terminals (VDTs), or from sequential devices, such as disk files, card readers, or cassettes. The input is interpreted through command procedure definitions. These procedures activate processors that use the information collected by the procedures. The SCI operates in two modes, interactive or batch. Also, the interactive mode may operate in either record-oriented or VDT mode. The following paragraphs describe how to use SCI to create Sort/Merge command statements in each SCI mode and how to use these statements to execute Sort/Merge. Figure 8-1 shows the interaction of Sort/Merge with the SCI.

8.2 INTERACTIVE MODE

In the interactive mode, SCI issues prompts either on a VDT screen or to an interactive printing device such as an ASR. Respond to each prompt as it appears; SCI and its associated processors then use the information to build the Sort/Merge control file. The two types of interactive mode are record-oriented mode and VDT mode.

8.2.1 Record-Oriented Mode

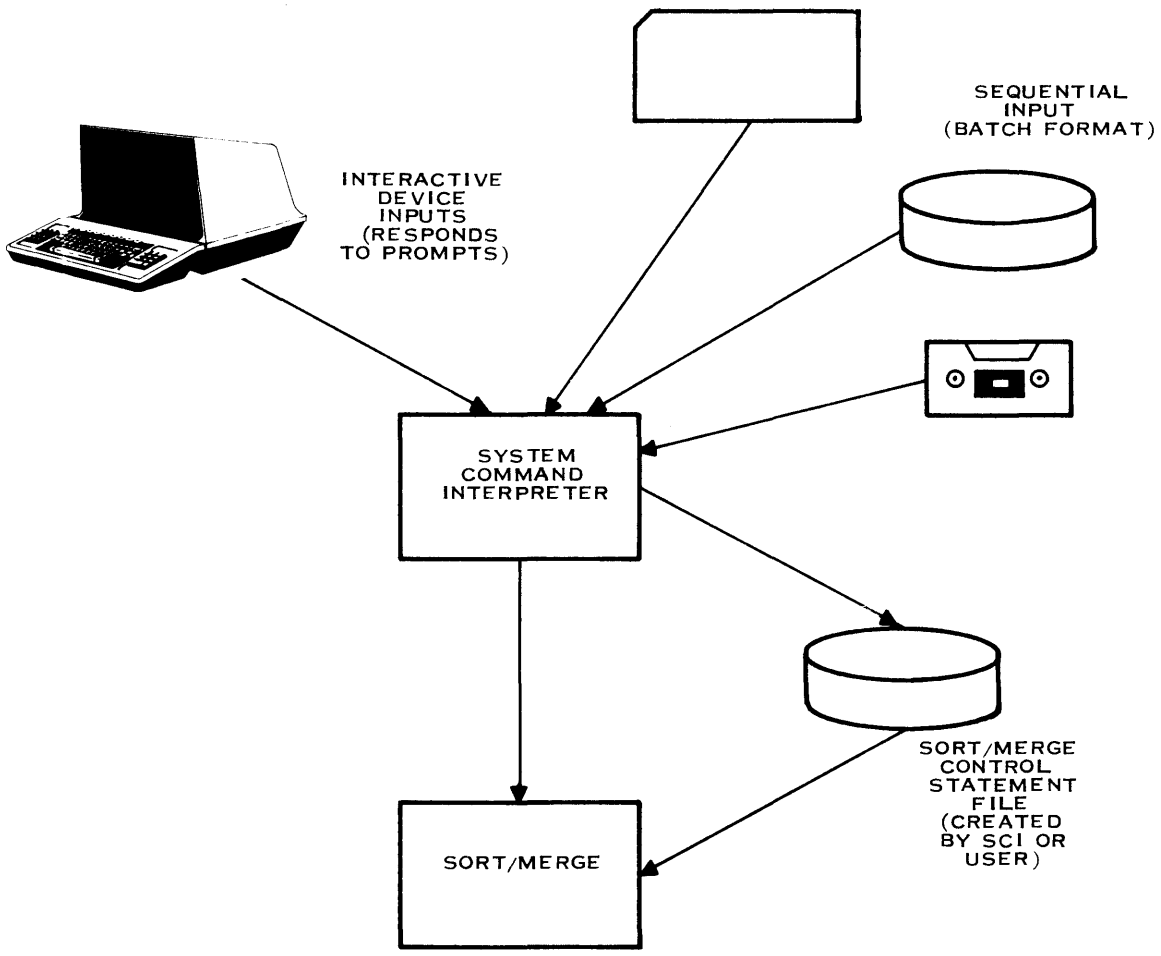
In record-oriented mode, prompts appear one at a time on the device; your response is required before the next prompt appears. Prompts are of the following form:

```
KEYWORD(s): user response
```

When you enter a response, the input is checked to see if it is an integer, a string, or a pathname. If the response does not match the required data type, the prompt reappears. If the response can be defaulted, you can choose the default by using the carriage return key. Figure 8-2 shows an example of prompts in the record-oriented mode.

8.2.2 VDT Mode

In the VDT mode, the entire set of prompts for the command procedure appears on the screen at one time. A heading (if supplied) also appears, and defaults are shown for the prompts that have defaults. Figure 8-3 shows an example of prompts in the VDT mode.



(A)135757

Figure 8-1. SCI Interaction with Sort/Merge

```
[] XSMF
EXECUTE SORT/MERGE
 NEW CONTROL FILE?:    N
 CONTROL FILE NAME:    .SPEC
SORT/MERGE EXECUTION OPTION
 RUN SORT/MERGE?: YES
SORT/MERGE LISTING DEVICE
 LISTING DEVICE NAME:    LP01
 SORT/MERGE COMPLETED NORMALLY
```

Figure 8-2. Sample Record-Oriented Mode Prompts

```
**SORT/MERGE HEADER SPECIFICATION**
          SORT MERGE TYPE: SORTR
     TOTAL CONTROL LENGTH:
      SORT MERGE SEQUENCE: A
       ALTERNATE SEQUENCE:
             PRINT OPTION: 1
      DROP CONTROL FIELDS: X
     OUTPUT RECORD LENGTH:
  VARIABLE LENGTH RECORDS:
            VERIFY OPTION:
              MEMORY SIZE:
```

Figure 8-3. Sample VDT Mode Prompts

8.2.3 Foreground and Background Processing

Use either of the following two commands to invoke Sort/Merge in the interactive mode:

- XSM (Execute Background Sort/Merge) — Issues prompts on an interactive device and then starts execution of Sort/Merge in the background mode. When you use XSM, control of the interactive device returns to the SCI when Sort/Merge execution begins; consequently, you can continue working with other activities in foreground while Sort/Merge is processing. When Sort/Merge completes, a message is queued to and displayed by the SCI. This message states how the sort or merge operation completed (such as SORT/MERGE NORMAL COMPLETION). When you use XSM, the response to the LISTING DEVICE NAME prompt cannot be the interactive device from which execution began.

- XSMF (Execute Sort/Merge Foreground) — Issues prompts on an interactive device and then runs Sort/Merge in the foreground. Control of the interactive device does not return to the SCI until completion of the sort or merge operation. You can specify your terminal in response to the LISTING DEVICE NAME prompt.

8.2.4 Interactive Mode Usage

Use the Execute Sort/Merge (XSMF) and Execute Background Sort/Merge (XSM) commands to enter command statement specifications in the interactive mode. In response to either of these commands, the following field prompt appears:

NEW CONTROL FILE?:

Enter a Y or YES to create a new Sort/Merge control file; enter N or NO to specify an existing file. If the card reader or cassette contains the control file, the answer must be N or NO.

NOTE

Sort/Merge must have a valid control file in order to execute correctly. When in doubt, create a new control file.

The next prompt is as follows:

CONTROL FILE NAME:

Respond to this prompt by entering the control file name to be created or used to run Sort/Merge. If the control file is on cassette or cards, specify CSnn or CRnn (where nn is the device number). You must enter a valid pathname.

If you are creating a new control file, the Header Specification prompts (Figures 8-2 and 8-3) appear next. Additional prompts are issued until the full control file has been created. (See Table 8-1.)

When you specify that an existing control file be used (by responding NO to the NEW CONTROL FILE? prompt) or when all prompts have been issued and a new control file has been created, the following field prompt is displayed:

RUN SORT MERGE?:

If the response to this prompt is N or NO, control returns to SCI and no further prompts appear. If the response is Y or YES, the following prompt appears:

LISTING DEVICE NAME:

Enter the name of the file or device that will receive the output of the Sort/Merge command statements and warning, error, and information messages. After you have specified the file or listing device, Sort/Merge executes.

Table 8-1. Interactive Prompts and Responses for Control File Generation

Suggested Batch Keyword	Prompt	Default	Other Values
Header Specification (SM$HD):			
SMT	SORT MERGE TYPE	SORTR	SORTRS/SORTA/ MERGE/MERGES
TCL	TOTAL CONTROL LENGTH	—	(integer)
SMS	SORT MERGE SEQUENCE	A	D
AS	ALTERNATE SEQUENCE	blank	S
PO	PRINT OPTION	1	0/1/3
DCF	DROP CONTROL FIELDS	X	
ORL	OUTPUT RECORD LENGTH	—	(integer)
VAR	VARIABLE LENGTH RECORDS	blank	V
VO	VERIFY OPTION	blank	V
MS	MEMORY SIZE	blank	(integer)
Alternate Sequence Specification (SM$ALT):[1]			
EAP	ENTER ALTERNATE PAIRS	—	(string of pairs)
Output File Specification (SM$OUT):			
FP	FILE PATHNAME	—	(pathname)
FT	FILE TYPE	S	R/I
LRL	LOGICAL RECORD LENGTH	blank	(integer)
PRL	PHYSICAL RECORD LENGTH	blank	(integer)
NOLR	NUMBER OF LOGICAL RECORDS	blank	(integer)
Work File Specification (SM$WKF):[2]			
WFV	WORK FILE VOLUME	DS01	(volume pathname)
WFT	WORK FILE TYPE	E	N
Input File Specification (SM$IN):			
FP	FILE PATHNAME	—	(pathname)
FT	FILE TYPE	S	R/I
LRL	LOGICAL RECORD LENGTH	blank	(integer)
NOLR	NUMBER OF LOGICAL RECORDS	blank	(integer)
Record Selection Specification (SM$SLC):			
RST	RECORD SELECTION TYPE	I	blank
CON	CONTINUATION	blank	A/O
CP	CHARACTER PORTION	blank	C/Z/D/S/P/U
O1B	OPERAND 1 BEGINNING	blank	(integer)
O1E	OPERAND 1 ENDING	blank	(integer)
REL	RELATIONAL OPERATOR	blank	EQ/NE/LT/GT/ LE/GE

Table 8-1. Interactive Prompts and Responses for Control File Generation (Continued)

Suggested Batch Keyword	Prompt	Default	Other Values
FOC	FIELD OR CONSTANT	blank	F/C
O2B	OPERAND 2 BEGINNING[3]	blank	(integer)
O2E	OPERAND 2 ENDING[3]	blank	(integer)
O2C	OPERAND 2 CONSTANT[4]	blank	(constant)
Reformatting Specification (SM$REF):			
FT	FIELD TYPE	—	N/O/F/D/S
CP	CHARACTER PORTION	C	Z/D/V/S/F/I/P/U/L/B
BL	BEGINNING LOCATION	blank	(integer)
EL	ENDING LOCATION	—	(integer)
RC	RECORD CHARACTER[5]	blank	(ASCII)
SC	SUBSTITUTE CHARACTER[5]	blank	(ASCII)
CON	CONTINUATION[5]	blank	(integer)
OVF	OVERFLOW[6]	blank	(integer)

Notes:

[1] Displayed only if the response to ALTERNATE SEQUENCE (AS) is S. A null response to ENTER ALTERNATE PAIRS terminates alternate sequence specifications.

[2] Displayed only if the response to SORT/MERGE TYPE (SMT) is not MERGE or MERGES.

[3] Displayed if response to FIELD or constant (FC) is not C.

[4] Displayed if response to FIELD or constant (FC) is C.

[5] Displayed if FIELD TYPE (FT) is F.

[6] Displayed if FIELD TYPE (FT) is S.

Figure 8-4 shows the various options available to the interactive user. The following paragraph discusses interactive prompting methods for running Sort/Merge.

BACKGROUND EXECUTION

```
[ ]XSM

     EXECUTE BACKGROUND SORT/MERGE
                NEW CONTROL FILE?:  N
                CONTROL FILE NAME:  .XYZ

                  RUN SORT MERGE?:  Y
               LISTING DEVICE NAME:  LP01

[ ]

     Other Tasks                              SORT/MERGE Execution

     SORT/MERGE NORMAL COMPLETION

[ ]
```

FOREGROUND EXECUTION
(From ST02)

```
[ ]XSMF
     EXECUTE SORT/MERGE
                NEW CONTROL FILE?:  NO
                CONTROL FILE NAME:  CR01

                  RUN SORT MERGE?:  YES
               LISTING DEVICE NAME:  ST02

     SORT/MERGE Executes
[ ]
```

Figure 8-4. Interactive Options — No Control File Creation

8.2.5 Interactive Prompting

To specify that SCI build a control file, respond Y or YES to the NEW CONTROL FILE? prompt, issued by the XSM or XSMF command. SCI then interactively issues all of the field prompts to obtain the necessary information. Figure 8-5 shows the categories of prompts, in the order in which they appear.

Prompting Category	Remarks
HEADER SPECIFICATION ALTERNATE SEQUENCE	If S in response to ALTERNATE SEQUENCE prompt of Header Specification
OUTPUT FILE	
WORK FILE	If SORTR, SORTA, SORTRS in Header Specification
INPUT FILE	
NEXT INPUT FILE	Only if MERGE or MERGES in Header Specification

RECORD SELECTION SPECIFICATION or REFORMATTING SPECIFICATION
NEXT:
  RECORD SELECTION SPECIFICATION
  REFORMATTING SPECIFICATION

Figure 8-5. Order of Interactive Prompting for Control File Creation

The prompts are presented in the proper order to create the control file until the NEXT prompt is encountered. The NEXT prompt allows you to specify a variety of sorts and merges using the SCI. The NEXT prompt has the following two forms:

```
ENTER NEXT SPECIFICATION — INPUT, SELECTION, REFORMATTING
  NEXT:

ENTER NEXT SPECIFICATION — SELECTION OR REFORMATTING
  NEXT:
```

The first specification is used with merges where more than one input file is specified. In a sort only one input file is used and the second form of the specification is used. Respond to the NEXT prompt with an I to specify input file, an S to specify Record Selection, or an R to specify Reformatting Specification. The NEXT prompt gives the flexibility to mix selection and reformatting specifications to produce more complex control sequences. When only the new line (carriage return) is entered in response to the NEXT prompt, specification of the control file is completed and the following prompt appears:

```
RUN SORT/MERGE?:
```

Figure 8-6 gives an example of a complete interactive session to build a control file on DS01.CONTRL. Notice that the Sort/Merge utility is executed at the end of the session, using the control file built during the session.

```
[] XSMF
EXECUTE SORT/MERGE
 NEW CONTROL FILE?     YES
 CONTROL FILE NAME:    .CONTRL
**SORT/MERGE HEADER SPECIFICATION**
 SORT MERGE TYPE: SORTR
 TOTAL CONTROL LENGTH:    6
 SORT MERGE SEQUENCE: A   D
 ALTERNATE SEQUENCE:
 PRINT OPTION: 1   3
 DROP CONTROL FIELDS: X
 OUTPUT RECORD LENGTH:    80
 VARIABLE LENGTH RECORDS:
 VERIFY OPTION:
 MEMORY SIZE:    16000
**SORT/MERGE OUTPUT FILE SPECIFICATION**
 FILE PATHNAME:    .SM.OUT1
 FILE TYPE: S
 LOGICAL RECORD LENGTH:    80
 PHYSICAL RECORD LENGTH:    560
 NUMBER OF LOGICAL RECORDS:
**SORT/MERGE WORK FILE SPECIFICATION**
 WORK FILE VOLUME: DS01
 WORK FILE TYPE: E
**SORT/MERGE INPUT FILE SPECIFICATION**
 FILE PATHNAME:    .SM.SRTDAT
 FILE TYPE: S
 LOGICAL RECORD LENGTH:
 NUMBER OF LOGICAL RECORDS:
ENTER NEXT SPECIFICATION - SELECTION OR REFORMATTING
 NEXT:    S
**SORT/MERGE RECORD SELECTION SPECIFICATION**
 RECORD SELECTION TYPE: I   O
 CONTINUATION:
 CHARACTER PORTION:    C
 OPERAND 1 BEGINNING:
 OPERAND 1 ENDING:    5
 RELATIONAL OPERATOR:    LT
 FIELD OR CONSTANT:    C
SELECTION COMPARISON CONSTANT
 OPERAND 2 CONSTANT:    3
ENTER NEXT SPECIFICATION - SELECTION OR REFORMATTING
 NEXT:    S
**SORT/MERGE RECORD SELECTION SPECIFICATION**
 RECORD SELECTION TYPE: I
 CONTINUATION:
 CHARACTER PORTION:
 OPERAND 1 BEGINNING:
 OPERAND 1 ENDING:
 RELATIONAL OPERATOR:
 FIELD OR CONSTANT:
```

Figure 8-6. Typical Interactive Prompt and Response Sequence for Building Control Specification File (Sheet 1 of 2)

```
SELECTION COMPARISON FIELD
 OPERAND 2 BEGINNING:
 OPERAND 2 ENDING:
ENTER NEXT SPECIFICATION - SELECTION OR REFORMATTING
 NEXT:    R
**SORT/MERGE REFORMATTING SPECIFICATION**
 FIELD TYPE:    F
 CHARACTER PORTION: C
 BEGINNING LOCATION:
 ENDING LOCATION:    5
FORCED CHARACTER INFORMATION
 RECORD CHARACTER:    4
 SUBSTITUTE CHARACTER:    A
 CONTINUATION:
ENTER NEXT SPECIFICATION - SELECTION OR REFORMATTING
 NEXT:    R
**SORT/MERGE REFORMATTING SPECIFICATION**
 FIELD TYPE:    F
 CHARACTER PORTION: C
 BEGINNING LOCATION:
 ENDING LOCATION:    5
FORCED CHARACTER INFORMATION
 RECORD CHARACTER:
 SUBSTITUTE CHARACTER:    Z
 CONTINUATION:    C
ENTER NEXT SPECIFICATION - SELECTION OR REFORMATTING
 NEXT:    R
**SORT/MERGE REFORMATTING SPECIFICATION**
 FIELD TYPE:    N
 CHARACTER PORTION: C
 BEGINNING LOCATION:    23
 ENDING LOCATION:    27
ENTER NEXT SPECIFICATION - SELECTION OR REFORMATTING
 NEXT:    R
**SORT/MERGE REFORMATTING SPECIFICATION**
 FIELD TYPE:    D
 CHARACTER PORTION: C
 BEGINNING LOCATION:    1
 ENDING LOCATION:    30
ENTER NEXT SPECIFICATION - SELECTION OR REFORMATTING
 NEXT:
SORT/MERGE EXECUTION OPTION
 RUN SORT/MERGE?: YES
SORT/MERGE LISTING DEVICE
 LISTING DEVICE NAME:    LP01
 SORT/MERGE COMPLETED NORMALLY
[]
```

Figure 8-6. Typical Interactive Prompt and Response Sequence for Building Control Specification File (Sheet 2 of 2)

8.3 BATCH MODE

You can use a file containing batch command statements to produce a Sort/Merge control file and to run Sort/Merge. Use the Text Editor or other means of creating records in the batch format described in the following paragraphs. Use the Execute Batch (XB) command or the Execute Batch Sort/Merge (XBSM) command to initiate a batch input. For further information on the batch commands, see the *DX10 Operating System Reference Manual, Volume II.*

8.3.1 Batch Stream Commands

Use the commands described in this paragraph to create control files and run Sort/Merge in batch mode. The first record in a batch file must be the BATCH command, and the second must be the SM$SMC command.

NOTE

If SM$SMC is not the second command of a Sort/Merge batch stream, the results of XB are unpredictable.

The format of SM$SMC is as follows:

```
SM$SMC  CFN = filename
```

The file specified will contain the completed control specifications.

Table 8-2 details the various commands and the order in which they must be given. You can repeat SM$IN once for each input file in a merge. You can repeat SM$SLC and SM$REF in sets to create detailed specifications. Execute these commands in the order required by the specification sheets (as shown in earlier sections of this manual). Although the order of keywords within a specification is not critical, you should use the same order as you would use on the specification worksheets. You need not specify keywords that are defaulted, but, if you fail to specify a required entry, an error termination of Sort/Merge results.

The last command of the control file creation sequence is SM$CLS. This command terminates the creation of the control stream in an orderly fashion. It is a good practice to always include this command, although no harm is done if it is omitted.

The EBATCH command must be the last record in a batch file.

8.3.2 Batch Record Format

Batch records must conform to the following rules:

- The maximum length of a record is 80 characters. If you need more than one record to specify the parameters for a command statement, end the first record with a comma or an equal sign.
- Each record in the file begins with a particular procedure name. If any keyword-response sets are required, the first set must be on the same record as the procedure name.
- Keyword-response sets are separated by commas, as follows:

```
COMMAND KEY1 = RESPONSE1,KEY2 = RESPONSE2
```

The order of the keywords is not critical. Table 8-1 gives suggested batch keywords. If you use keywords other than those recommended, the keyword selected might not be unique, producing unpredictable results.

Table 8-2 describes each of the Sort/Merge commands. The table presents the commands in the order in which they must occur if used.

Table 8-2. Batch Stream Commands and Use

Command	Remarks
SM$SMC	Creates the control file that will contain the specifications created by the subsequent commands. This statement is required and must be the second command in the batch stream (that is, following the BATCH command).
SM$HD	Builds the header control statement. Must be specified and must immediately follow the SM$SMC command.
SM$OUT	Builds the output file control statement. It must be specified immediately after SM$HD.
SM$WKF	Builds the work file control statement. It may be specified for a sort; when used, it must follow the SM$OUT command. This command is not necessary for a merge and will be ignored if specified.
SM$IN	Builds the input file control statement(s). The first SM$IN must follow SM$WKF. For a sort, only one SM$IN may be specified. For a merge, at least two and no more than five SM$IN commands may be used. For a merge, these commands may be intermixed with SM$SLC and SM$REF commands to create more complex selection and reformatting patterns.
SM$SLC	Builds the record selection control statement(s). SM$SLC commands may be mixed with SM$IN and SM$REF commands to create different selection patterns for different record types. The SM$SLC command is not required. If no SM$SLC command is used, the default value includes all records.
SM$REF	Builds the reformatting control statement(s). SM$REF commands may be mixed with SM$IN and SM$SLC commands to create different reformatting for different selected record sets or different input files. SM$REF commands must be present to define the sort control fields or keys.
SM$CLS	Terminates after the file creation procedure. This should be the last command of the file creation section of a batch stream.
XBSM	Executes Sort/Merge by using the specified batch file name and listing device.

8.3.3 Executing Sort/Merge in Batch Mode

The command to execute Sort/Merge in batch is as follows:

XBSM CFN = filename,LDN = device(or filename)

If the control file has previously been created, only the XBSM command is required. If the control file is being built in the batch stream, the file name on the XBSM statement must match the file name on the SM$SMC statement. Figure 8-7 shows a file created by the Text Editor and then used as a batch stream with the XB command.

```
BATCH
SM$SMC  CFN=DS03. FILNAM
SM$HD  SMT=SORTR, TCL=6, ORL=80, MS=8000
SM$OUT  FP=DS03. TEST, LRL=80, PRL=560
SM$WKF  WFV=DS02, WFT=E
SM$IN  FP=DS03. INFIL
SM$SLC  RST=I
SM$REF  FT=N, CP=C, BL=32, EL=37
SM$REF  FT=D, CP=C, BL=1, EL=80
SM$CLS
XBSM  CFN=DS03. FILNAM, LDN=LP01
EBATCH
```

Figure 8-7. Example of Entries to Create a Batch File and Execute Sort/Merge

9

External Interface Specifications

9.1 GENERAL

You can access Sort/Merge from COBOL, FORTRAN, Pascal, BASIC, or assembly language programs as either a synchronous task or an asynchronous task, as follows:

- As an asynchronous task, executing concurrently with the calling task to process records from input files and to send processed records to an output file.
- As a synchronous task, to receive records from the calling task for processing and/or to send records to the calling task following processing; in this case, a certain degree of asynchronous processing occurs when the String Merge Module of Sort/Merge is operating on the work file.

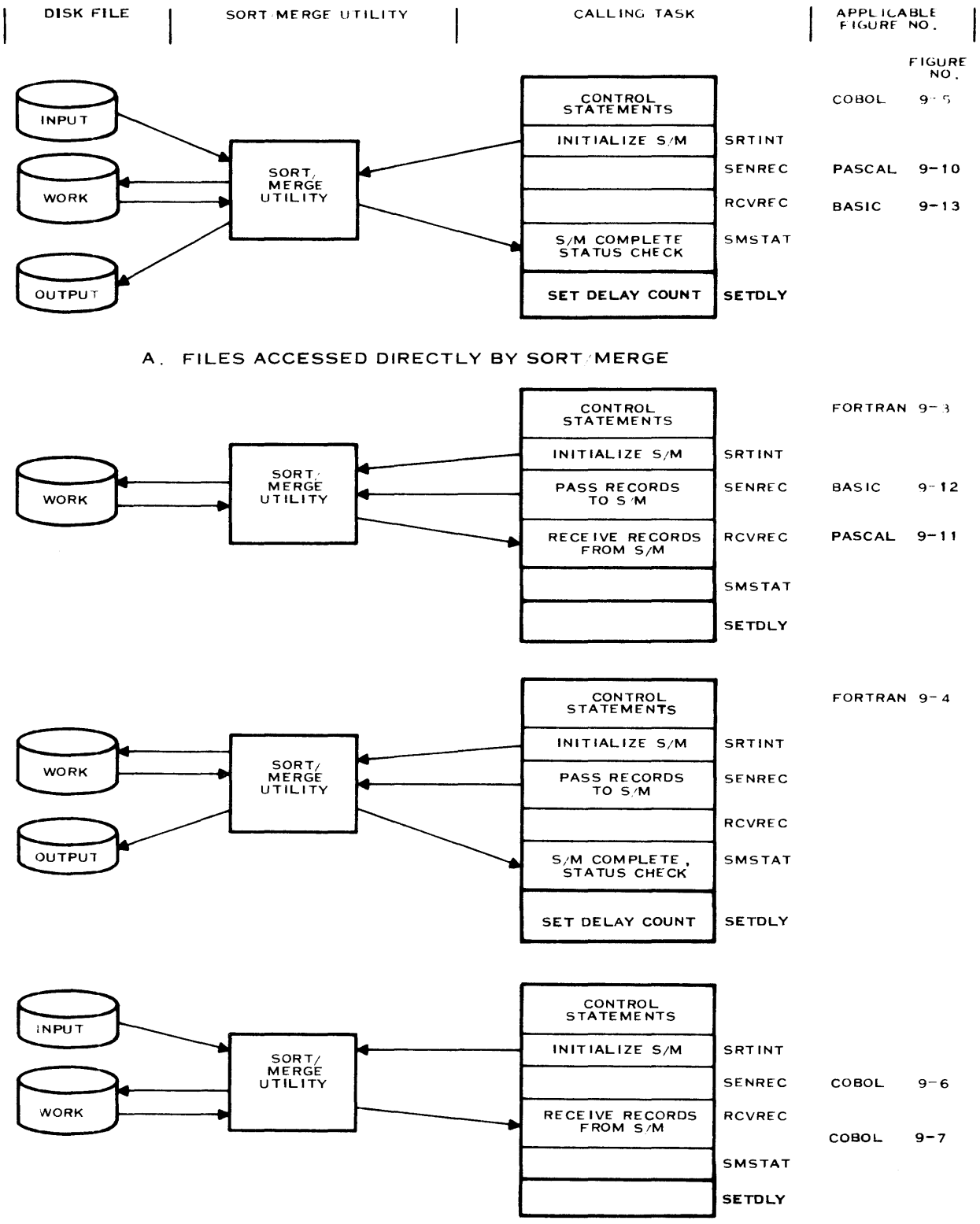
9.1.1 Intertask Subprograms

Data is transferred between the calling program and Sort/Merge via intertask communication using GETDATA and PUTDATA supervisor calls (SVC). The following subroutines are linked to the calling task to provide this intertask communication. All of these subroutines are optional except SRTINT.

SRTINT	Initializes the Sort/Merge task by bidding Sort/Merge and passing the control specifications to Sort/Merge. This is a mandatory subroutine.
SENREC	Sends records to Sort/Merge from the calling program.
RCVREC	Transfers records that the calling program receives from Sort/Merge.
SMSTAT	Suspends the calling program until the entire Sort/Merge task is completed.
SETDLY	Modifies the delay count for the Time Delay SVC used by the SMSTAT subroutine.

9.1.2 Transmission of Records to and from Sort/Merge

After the calling program initializes Sort/Merge, the Control Statement Compiler Module interprets the File Description Specification to determine the input and output files. If the pathname describes a file, Sort/Merge directly accesses the records from the input disk file and returns them to the output disk file (part A of Figure 9-1). You cannot use synonyms for file names within the sort control block definition of a calling program; however, you may use logical names. If the input or output file pathname is specified by the characters @PROC@, the respective input record source or the output record destination will be the calling program. Part B of Figure 9-1 shows the three modes of this type of access.



(A)135766A

Figure 9-1. External Communication to Sort/Merge

The rules for using @PROC@ as the pathname are as follows:

- An @PROC@ for the output file pathname requires no continuation line.
- An @PROC@ for the input file pathname requires a continuation line with the logical record length specified.
- In a merge, @PROC@ can be used for only one of the two to five input files. If used, it requires a continuation line specifying the logical record length.

9.1.3 Control Statement Criteria

Control statements in assembly language, FORTRAN, COBOL, Pascal, or BASIC must be contained in a contiguous array; each statement must be 44 characters long. This corresponds to the first 44 characters of the control statement specifications and means that comments are not supported.

The last control statement must be a slash-asterisk (/*), which is the end-of-file (EOF) mark. This mark must be included in the count when specifying the number of control statements.

9.1.4 Use of Intertask Communication

Sort/Merge uses the GETDATA and PUTDATA SVCs to transmit and receive records and control statements between the calling task and the Sort/Merge task. The run-time task ID number of the calling task and the Sort/Merge task is used as the communication channel identifier with the GETDATA and PUTDATA calls.

9.2 SUBPROGRAM USAGE

Paragraphs 9.2.1 through 9.2.6 contain information on linkable subprograms that communicate with Sort/Merge. These examples use a general CALL format; however, the formats for the FORTRAN, COBOL, Pascal, and BASIC CALL statements are analogous; assembly language also follows a similar format. A comparison of the statement used to call the initialization subprogram for each of these languages is as follows:

FORTRAN:

```
CALL SRTINT(CNTRL,NUMSTS,STATIS)
```

COBOL:

```
CALL "SRTINT" USING CNTRL,NUMSTS,STATIS.
```

BASIC:

```
CALL ".SMLIB.BASCSM"(SRTINT,A$(1),N,S)
```

Pascal:

```
SRTINT(CNTRL,NUMSTS,STATIS);
```

Assembly Language:

```
          REF   SRTINT
            .
            .
            .
          LI    RO,ARGLST       Address of argument list
          BLWP  @SRTINT         Call subroutine
            .
            .
            .
ARGLST    DATA  6               Byte count of argument list
                                (twice the number of arguments)
          DATA  CNTRL           Control statement area address
          DATA  NUMST           Number of control statements
          DATA  STATIS          Return status area
```

9.2.1 Initialization

Sort/Merge must be initialized before you can send records to or receive records from the Sort/Merge module. The general format for calling initialization is as follows:

```
CALL SRTINT(CONTROL-STATEMENTS,NUMBER-STATEMENTS,STATIS)
```

where:

CONTROL-STATEMENTS is the name of the array containing the control statements in the calling program. Each statement must be 44 characters.

NUMBER-STATEMENTS is the maximum number of control statement records to be sent to Sort/Merge (INTEGER or COMP-1).

STATIS is the parameter in which the status code will be returned following initialization (INTEGER or COMP-1). STATIS may contain the following values:

Value	Meaning
0	Indicates initialization successful.
1	Indicates error in control statements.
2	Indicates error in initialization of Sort/Merge.
3	Indicates Sort/Merge not installed on system.
4	Indicates no /* (EOF mark) found.
5	Indicates Sort/Merge abnormal termination.
6	Indicates SRTINT cannot allocate a buffer for the intertask communication message.

9.2.2 Send Records to Sort/Merge

Following initialization, you can transmit records from the calling program to Sort/Merge by using subroutine SENREC. Use the following format to call SENREC:

```
CALL SENREC(RECORD-AREA,RECORD-LENGTH,STATIS)
```

where:

RECORD-AREA is the array containing the record to be sent. The array must begin on a word boundary in memory.

RECORD-LENGTH is the length in bytes of the record to be sent. Zero indicates an EOF to Sort/Merge (INTEGER or COMP-1). The record length must be an even number of bytes.

STATIS is the parameter in which the status code will be returned (INTEGER or COMP-1). STATIS may contain the following values:

Value	Meaning
0	Indicates that Sort/Merge successfully received the record.
1	Indicates an error in the send operation. Check to be sure that SRTINT was called prior to calling SENREC. Sort/Merge terminates.
2	Indicates that the Sort/Merge processor is no longer an active task. The system log may contain an error entry for task $2A_{16}$ (if Sort/Merge terminated abnormally). Check all control statements and calling sequences carefully, and use the debug aid explained in a subsequent paragraph in this section.

9.2.3 Receive Records from Sort/Merge

Use subroutine RCVREC to enable the calling program to receive records from Sort/Merge. Use the following format to call RCVREC:

```
CALL RCVREC (RECORD-AREA,RECORD-AREA-LENGTH,BYTES-RECEIVED,STATIS)
```

where:

RECORD-AREA is the array where the received record is to be stored. The array must begin on a word boundary in memory.

RECORD-AREA-LENGTH is the length in bytes of the record area (INTEGER or COMP-1). The length must be an even number of bytes.

BYTES-RECEIVED is the actual number of bytes placed in the record area. A zero in this area indicates that an EOF mark was received from Sort/Merge (INTEGER or COMP-1).

STATIS is the parameter in which the status code is returned following the operation. STATIS may contain the following values:

Value	Meaning
0	Indicates successful receipt of data.
1	Indicates an error in receipt of data or that SRTINT was not called prior to calling REVREC. The system log may contain an error entry for task $2A_{16}$ (if Sort/Merge terminated abnormally). Check all control statements and calling sequences carefully, and use the debug aid explained in a subsequent paragraph in this section.
2	Indicates that an EOF mark was received.

NOTE

STATIS or BYTES-RECEIVED can be monitored to determine if an EOF mark has been received.

9.2.4 Wait for Sort/Merge Completion

Use the SMSTAT call only if you specified a disk file for the output file in the control specifications. The SMSTAT call suspends the calling task until Sort/Merge completes transmitting records to the output disk file and finishes execution. The calling task is then reactivated, and completion status can be checked. Sort/Merge must finish or terminate execution before the calling task is reactivated; thus, if improperly used, this call can indefinitely suspend the calling program.

The format of the SMSTAT call is as follows:

```
CALL SMSTAT(STATIS)
```

where:

STATIS is the parameter (INTEGER or COMP-1) in which the status code will be returned upon completion of the Sort/Merge processing. STATIS may contain the following values:

Value	Meaning
0	Indicates Sort/Merge completed with no errors.
1	Indicates Sort/Merge encountered control statement errors.
2	Indicates Sort/Merge encountered fatal I/O errors.
3	Indicates Sort/Merge terminated for unknown reasons.
8	Indicates Sort/Merge completed but encountered locked key-indexed file records.

9.2.5 Set Time Delay Count

The SETDLY subroutine allows you to set the time delay between checks, thus minimizing the number of times the application task resides in memory.

Subroutine SMSTAT checks the execution of the Sort/Merge task to be sure that execution is continuing. Since SMSTAT is part of the application task, the application task must be in memory when this checking occurs. In a small memory system in which both the application task and Sort/Merge cannot be in memory at the same time, this monitoring function might suspend and roll out the Sort/Merge task so that the SMSTAT subroutine can execute. This can have a significant impact on the speed at which Sort/Merge operates, in that swapping the application task and Sort/Merge may take more time than the sort operation itself.

The format for the SETDLY call is as follows:

```
CALL SETDLY(SECNDS)
```

where:

SECNDS is the parameter (INTEGER or COMP-1) containing the number of seconds that SMSTAT should wait before checking the status of the Sort/Merge task. The limit of this parameter is 30 seconds. If a value of 0 is passed, a default of 26 seconds is used. If a value greater than 30 is passed, a value of 30 is used.

For example, consider the time required to sort 3000 records in a 128K-byte system. If the subroutine SETDLY is not called, the time required to sort the records is approximately 12 minutes; if the subroutine SETDLY is called with a value of 0, the time required to sort the records is approximately 2 minutes and 14 seconds.

NOTE

If the Sort/Merge task terminates prematurely, the application does not reactivate until after the specified time delay occurs.

9.2.6 Linking FORTRAN, COBOL, Pascal, and Assembly Language Programs

The library .SMLIB contains the following Sort/Merge interface subroutines. The subroutine SETDLY is contained in the module SRTINT.

FORTRAN, COBOL, Pascal, or Assembly Language

SRTINT

SENREC

RCVREC

SMSTAT

In addition, COBOL and assembly language programs use a subroutine called COBINT and Pascal uses a subroutine called PSCINT.

NOTE

SRTINT must be either in the root segment in overlay structures or in the same segment as SENREC, RCVREC, and SMSTAT.

SENREC, RCVREC, and SMSTAT each call internal subroutines contained in SRTINT. The other subroutines call COBINT and PSCINT and must be placed in the same manner as SRTINT.

In the following paragraphs, .MAINPROG denotes the user-written code. In each example, all subroutines in .SMLIB are included, but only subroutines that are called need to be linked.

9.2.6.1 Linking Assembly Language Programs. Assembly language requires that COBOL subroutine COBINT be included in the link control file, as follows:

```
NOSYMT

PHASE O,MAIN

INCLUDE .MAINPROG

INCLUDE .SMLIB.SRTINT

INCLUDE .SMLIB.SENREC

INCLUDE .SMLIB.RCVREC

INCLUDE .SMLIB.SMSTAT

INCLUDE .SMLIB.COBINT
```

Only SRTINT and COBINT are required. COBINT must follow the same rules that apply to SRTINT (as stated previously); that is, COBINT must be in either the root segment in overlay structures or in the same segment as SENREC, RCVREC, and SMSTAT.

9.2.6.2 Linking FORTRAN Programs. Perform the linking procedures for FORTRAN, as outlined in the *FORTRAN Programmer's Guide*. INCLUDE statements are necessary for SRTINT, SENREC, RCVREC, and SMSTAT. F$RGMY is part of the library .FORTRAN.STLOBJ and does not need an INCLUDE statement because it is included by the library commands required to link FORTRAN.

9.2.6.3 Linking COBOL Programs. Perform the linking procedures for COBOL are outlined in the *COBOL Programmer's Guide*. The INCLUDE statements for SRTINT, SENREC, RCVREC, and SMSTAT are necessary. COBINT follows the same rules that apply to SRTINT in that it must be in either the root segment in overlay structures or in the same segment as SENREC, RCVREC, and SMSTAT.

9.2.6.4 Linking Pascal Programs. Perform the linking procedures for Pascal as outlined in the *Pascal Programmer's Guide*. The INCLUDE statements for SRTINT, SENREC, RCVREC, and SMSTAT are necessary. Pascal also requires the following:

```
INCLUDE .SMLIB.PSCINT
```

This is a functional replacement for the COBOL COBINT. PSCINT follows the same rules that apply to SRTINT, as noted in paragraph 9.2.6.

9.2.7 Interfacing TI 990 BASIC Programs with Sort/Merge

The file BASCSM in the Sort/Merge library .SMLIB contains the subroutines required to interface TI 990 BASIC to the Sort/Merge utility. The first parameter in the argument list of the calling sequence must be an integer value between 1 and 5, indicating which subroutine is being called. The following is a list of the Sort/Merge subroutines and the corresponding integer values used when calling them:

Value	Subroutine
1	SRTINT
2	SETDLY
3	SENREC
4	RCVREC
5	SMSTAT

The SRTINT subroutine uses the following format:

```
nnn SRTINT = 1
nnn CALL ".SMLIB.BASCSM"(SRTINT,A$(1),N,S)
```

where:

SRTINT contains the integer value indicating that the subroutine SRTINT is being called.

A$(1) is a string array containing the control statements in the calling program.

N is the integer containing the maximum number of control statements.

S is the integer STATIS return parameter.

The SENREC subroutine uses the following format:

```
nnn SENREC = 3
nnn CALL ".SMLIB.BASCSM"(SENREC,S$,L,S)
```

where:

SENREC contains the integer value indicating that the subroutine SENREC is being called.

S$ is the string that contains the record to be sent.

L is the integer length (in bytes) of the string to be passed to Sort/Merge. L may be less than or equal to the actual length of S$. When L equals zero, it specifies the EOF to Sort/Merge. L must be an even number.

S is the integer STATIS return parameter.

The RCVREC subroutine uses the following format:

```
nnn RCVREC = 4
nnn S$ = RPT$("",L
nnn CALL ".SMLIB.BASCSM"(RCVREC,S$,L,B,S)
```

where:

RCVREC contains the integer value indicating that the subroutine RCVREC is being called.

S$ is the string that will receive the record returned from Sort/Merge. You must initialize the string to the expected return length by using the TI 990 BASIC intrinsic function RPT$ before calling the subroutine RCVREC.

L is the integer length in bytes of the record to be received in S$. L must be an even number.

B is the integer length in bytes of the string returned in S$. A zero indicates the EOF. B will be less than or equal to the length of S$.

S is the integer STATIS return parameter.

The SMSTAT subroutine uses the following format:

```
nnn SMSTAT = 5
nnn CALL ".SMLIB.BASCSM"(SMSTAT,S)
```

where:

SMSTAT contains the integer value indicating that the subroutine SMSTAT is being called.

S is the integer STATIS return parameter.

The SETDLY subroutine uses the following format:

```
nnn SETDLY = 2
nnn CALL ".SMLIB.BASCSM"(SETDLY,N)
```

where:

SETDLY contains the integer value indicating that the subroutine SETDLY is being called.

N is the integer variable containing the number of seconds that the application task should delay before checking the status of Sort/Merge.

You must load the file BASCSM before execution of the TI 990 BASIC program; use the following statement:

```
nnn LIBRARY"*.SMLIB.BASCSM"
```

All numeric variables used as parameters to the Sort/Merge subroutines must be declared as INTEGER type.

NOTE

Use of commas and blanks in strings in TI 990 BASIC may cause position problems within a string when using Sort/Merge. For example, if blanks are present in the first few columns of a record, the string begins with the first nonblank column. This may cause control and data fields to be out of their normal positions. Consult the *TI 990 BASIC Reference Manual* for further details.

The Sort/Merge utility supports only the alphanumeric and integer data types of TI 990 BASIC.

9.3 PROGRAM EXAMPLES

The following paragraphs give examples and explanations of FORTRAN, COBOL, Pascal, and BASIC listings.

9.3.1 FORTRAN Examples

Figure 9-2 is a flowchart and Figure 9-3 is a FORTRAN listing of a task that initializes Sort/Merge, passes records to Sort/Merge, and receives the records returned from Sort/Merge via intertask communication. In this example, both input and output file pathnames are @PROC@ to indicate to Sort/Merge that this intertask communication is to be done. Figure 9-4 is an example of a FORTRAN task that allows Sort/Merge to read the input file from the disk and return the sorted output records to the calling task.

9.3.2 COBOL Examples

Figure 9-5 shows a COBOL routine that calls Sort/Merge to input records from a disk file and output the sorted records to a second disk file. Note the proper use of SMSTAT.

Figure 9-6 shows a COBOL routine that calls Sort/Merge and passes records read by the COBOL routine to Sort/Merge. The sorted records are output to a disk file. Note the proper use of SMSTAT.

Figure 9-7 is an example of a COBOL routine that executes a SORTA on a relative record file containing more than 65,536 records. Note how the COBOL routine must manipulate the output from a SORTA (a binary relative record number) so that the correct record is read from the relative record file.

NOTE

You can specify control statements in COBOL as shown in these examples or by using a PIC X(44) VALUE IS construction in a manner similar to that of the FORTRAN examples.

Figure 9-8 compares interfacing Sort/Merge by using the 990 COBOL CALL verb with interfacing Sort/Merge by using the ANSI COBOL SORT verb; the latter is not supported on the 990. The CALL interface gives the COBOL user additional capabilities not found with the ANSI SORT statement. By using the CALL, you specify selection reformatting, summation, and alternate sequence specifications for use in sorting or merging. Figure 9-9 shows the combinations available with the COBOL CALL verb.

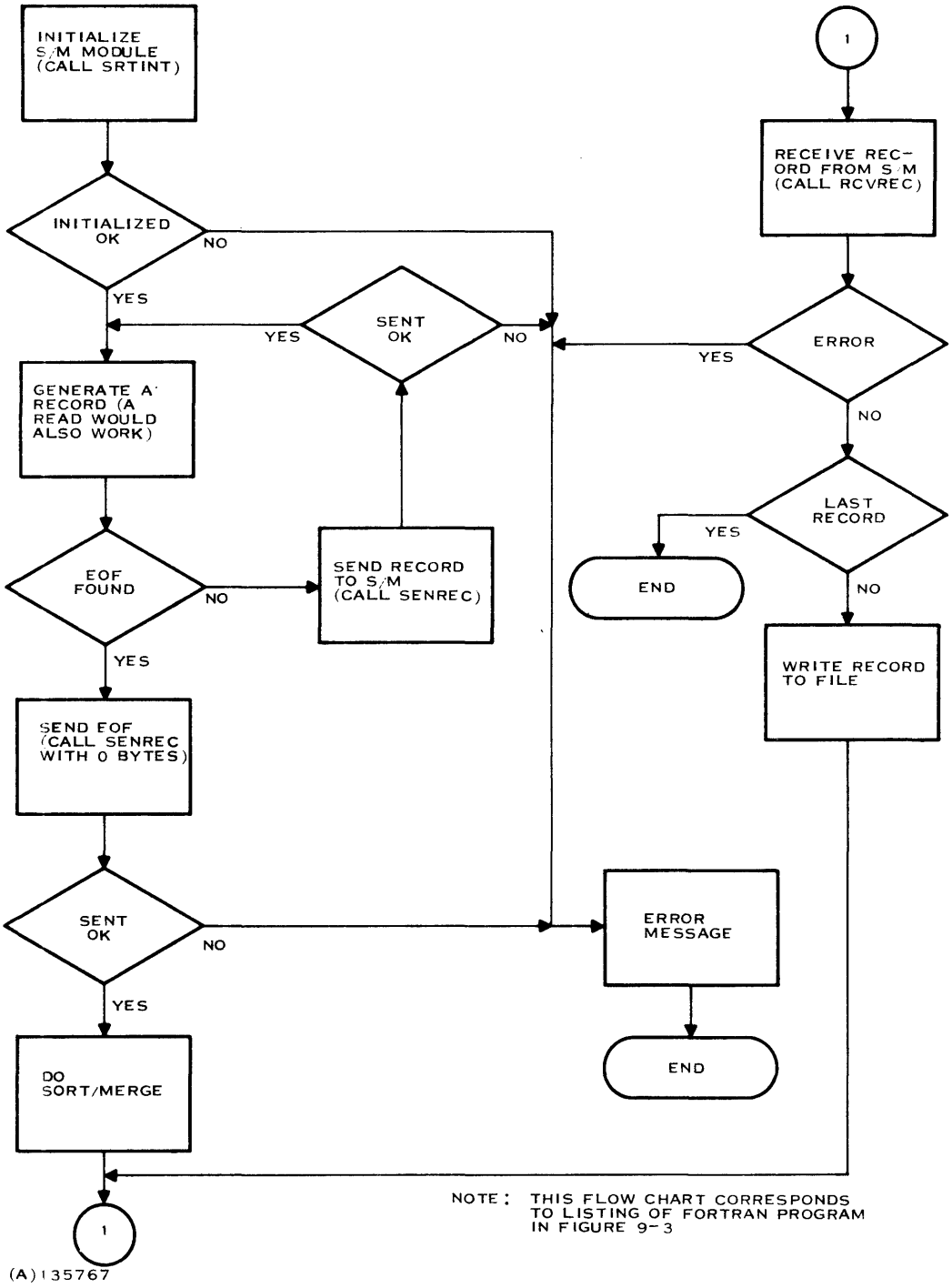


Figure 9-2. Flowchart of FORTRAN Program that Sends Records to Sort/Merge

```
C*  INTERTASK COMMUNICATION TEST PROGRAM   (PPP)
C*          PRINTED OUTPUT IS UNIT 6
C*
      INTEGER BUFFER(220), LIN1(22), LIN2(22), LIN3(22)
      INTEGER LIN4 (22),LIN5 (22),LIN6 (22),LIN7 (22)
      INTEGER LIN8 (22),LIN9 (22),LIN10(22), STATUS, IMAGE(40))
      EQUIVALENCE(LIN1 (1),BUFFER( 1)),(LIN2(1),BUFFER(23))
      EQUIVALENCE(LIN3 (1),BUFFER(45)),(LIN4(1),BUFFER(67))
      EQUIVALENCE(LIN5 (1),BUFFER(89)),(LIN6(1),BUFFER(111))
      EQUIVALENCE(LIN7 (1),BUFFER(133)),(LIN8(1),BUFFER(155))
      EQUIVALENCE(LIN9 (1),BUFFER(177)),(LIN10(1),BUFFER(199))
C  THIS FORTRAN TEST OF THE INTERTASK COMMUNICATIONS
C  GENERATES A RANDOM NUMBER BETWEEN 0 AND 1000.
C   THIS NUMBER IS ENCODED AND PASSED VIA THE
C   @PROC@ INTO THE SM.   SM SORTS THE DATA AND
C   PASSES THE DATA BACK THROUGH THE @PROC@.
C-------------------------- --------------------------
C*  SORT CONTROL BLOCK DEFINITION
C                  000000000111111111122222222223333333333444444
C                  123456789012345678901234567890123456789012345
      DATA LIN1 /'01000HSORTR     6A             80   8000      '/
      DATA LIN2 /'01010DOS@PROC@                                '/
      DATA LIN3 /'01030DWNWORKDISK                              '/
      DATA LIN4 /'01050DIS@PROC@                                '/
      DATA LIN5 /'01055DA  80          100                      '/
      DATA LIN6 /'01060I                                        '/
      DATA LIN7 /'01065FNC  32  37                              '/
      DATA LIN8 /'01070FDC   1  31                              '/
      DATA LIN9 /'01080FDC  38  80                              '/
      DATA LIN10/''/*                                           '/
      DATA NUMSP /10/
C------------------------CALL SRTINT TO TRANSFER SORT/MERGE
C----------------------SPECIFICATIONS TO THE SORT/MERGE TASK
      CALL SRTINT(BUFFER,NUMSP,STATUS)
      IF(STATUS.EQ.0) GOTO 20
C-------------STATUS INDICATES AN ERROR, DISPLAY AND EXIT
11    WRITE(6,12)STATUS
12    FORMAT(' SORT ERROR STATUS  = ',I4)
      STOP 1
C------------------------GENERATE 100 RECORDS, DISPLAY
20    DO 45  I=1,100
      JVAL=IFIX(1000.0*RANF(I))
      ENCODE(80,22,IMAGE,NUMC)JVAL,I
22    FORMAT('**',29X,I6,20X,I3,19X,'*')
      WRITE(6,32) IMAGE
32    FORMAT(1X,40A2)
C---------------------------------- PASS THE RECORDS
      CALL SENREC(IMAGE,80,STATUS)
      IF(STATUS.EQ.0)GOTO 45
      WRITE(6,12)STATUS
```

Figure 9-3. FORTRAN Program Calls Sort/Merge, Calling Program Sends and Receives Records (Sheet 1 of 2)

```
      STOP 2
45    CONTINUE
C-------------------------------------------------SEND EOF
55    ICOUNT=0
      CALL SENREC(IMAGE, ICOUNT, STATUS)
      IF(STATUS.EQ.0)GO TO 65
      WRITE(6,12)STATUS
      STOP 3
C--------------------RECEIVE THE SORTED RECORDS AND DISPLAY
65    DO 115 I=1,200
C---------------------WE WILL TRY TO RECEIVE MORE THAN WE SENT
      CALL RCVREC(IMAGE,80,IBYTE,STATUS)
       IF (STATUS.EQ.1) GO TO 95
       IF (STATUS.EQ.2) GO TO 77
       ICOUNT = ICOUNT + 1
       WRITE (6,112) IBYTE, IMAGE
          GO TO 115
C-----------------------------------------END OF FILE ENCOUNTERED
77     WRITE (6,82) ICOUNT
82     FORMAT (' RECORDS RECEIVED BACK = ',I7)
      STOP
95     WRITE(6,12)STATUS
       STOP 4
112   FORMAT(1X,I2,40A2)
115   CONTINUE
C-------------------------RECEIVED TOO MANY RECORDS *ERROR*
       WRITE(6,122)
122   FORMAT(' RECEIVED TOO MANY RECORDS BACK !!!')
       STOP 100
       END
```

Figure 9-3. FORTRAN Program Calls Sort/Merge, Calling Program Sends and Receives Records (Sheet 2 of 2)

```
C*   INTERTASK COMMUNICATION TEST PROGRAM     (FNPP)
C*          OUTPUT IS TO UNIT 6
C*
        DIMENSION IHD(22),INFL(22),INFLCT(22),IWRKFL(22)
        DIMENSION IOUTFL(22),IOUTCT(22),IFLD1(22),IREC(40)
        DIMENSION IFLD(22),IFLD3(22),IENDCD(22),ISPECS(220)
        EQUIVALENCE (ISPECS(1),IHD(1)),(ISPECS(23),IOUTFL(1)),
      +       (ISPECS(45),IOUTCT(1)),(ISPECS(67),IWRKFL(1)),
      +       (ISPECS(89),INFL(1)),(ISPECS(111),INFLCT(1)),
      +       (ISPECS(133),IFLD1(1)),(ISPECS(155),IFLD(1)),
      +       (ISPECS(177),IFLD3(1)),(ISPECS(199),IENDCD(1))
C  THIS TASK USES SM ROUTINE RCVREC AND PERFORMS A SORTR.
C  THE INPUT IS READ BY THE SORT FROM A FILE.
C  THE FILE TO BE SORTED IS A 400, 80-BYTE RECORD FILE.
C  THE OUTPUT IS RETURNED TO THE FORTRAN PROGRAM.
```

Figure 9-4. FORTRAN Program Calls Sort/Merge, Inputs Files from Disk, and Outputs Files to Calling Program (Sheet 1 of 2)

```
C  ------------------------------------------------------
C  SORT CONTROL BLOCK DEFINITION
C                          00000000011111111112222222222333333333344444
C                          12345678901234567890123456789012345678901234
      DATA IHD        /'00000HSORTR 00006A           0080    8000      '/
      DATA IOUTFL     /'00001DOS@PROC@                                 '/
      DATA IOUTCT     /'00002DA0080080000000101                        '/
      DATA IWRKFL     /'00004DWNWORKDISK                               '/
      DATA INFL       /'00006DISDS01,MARY.EIGH4                        '/
      DATA INFLCT     /'00008DA0080    00000400                        '/
      DATA IFLD1      /'0001CFNC00320037                               '/
      DATA IFLD       /'00012FDC00010031                               '/
      DATA IFLD3      /'00014FDC00380080                               '/
      DATA IENDCD     /'/*                                             '/
      DATA KNTST,IRECLN/10,80/
C-------------------------------------------------INITIALIZE SORT
C    SORT WILL READ INPUT FILE AND SORT RECORDS.
C    RCVREC WILL WAIT FOR RECEIVE THE RECORDS
C    RETURNED FROM THE SORT.
      CALL SRTINT(ISPECS,KNTST,ISTAT)
      IF (ISTAT.NE.0)                  GOTO 909
C-----------------------------------------------RECEIVE RECORDS
      ICOUNT = 0
C  RECEIVE THE NEXT RECORD.
55    CONTINUE
      CALL RCVREC(IREC,IRECLN,IRECBY,ISTAT)
C  CHECK IF LAST RECORD HAS BEEN SENT.
      IF (IRECBY.EQ.0)                 GOTO 75
      IF (ISTAT.NE.0)                  GOTO 929
      ICOUNT = ICOUNT + 1
      WRITE(6,61) IRECBY,IREC
61    FORMAT(1X,I4,1X,40A2)
      GOTO 55
75    WRITE(6,101) ICOUNT
101   FORMAT(' NUMBER OF RECORDS ',I5)
      STOP
C-------------------------------------------------ERROR
 909  WRITE(6,911) ISTAT
 911  FORMAT(' ERROR IN SRTINT',I4)
      STOP
C-----------------------------------------------ERROR
 929  WRITE(6,930) ISTAT
 930  FORMAT(' ERROR IN RCVREC',I4)
      STOP 2
C-----------------------------------------------ERROR
 939  WRITE(6,940) ISTAT
 940  FORMAT(' SORT/MERGE COMPLETED ABNORMALLY',I4)
      STOP 3
      END
```

Figure 9-4. FORTRAN Program Calls Sort/Merge, Inputs Files from Disk, and Outputs Files to Calling Program (Sheet 2 of 2)

```
 IDENTIFICATION DIVISION.
 PROGRAM-ID.  CTRSE.
*THIS IS A SORT/MERGE INFORMATION QUEUE TEST.  IT
*TESTS THE CASE WHERE BOTH INPUT AND OUTPUT ARE
*PASSED DIRECTLY FROM A FILE TO THE SORT (NO @PROC@).
 AUTHOR.  TEXAS INSTRUMENTS.
 DATE-WRITTEN.  11-6-80.
 ENVIRONMENT DIVISION.
 CONFIGURATION SECTION.
 SOURCE-COMPUTER.  TI-990-10.
 OBJECT-COMPUTER.  TI-990-10.
 DATA DIVISION.
 WORKING-STORAGE SECTION.
 77 MAX-NO-RECS          PIC 9(5)  VALUE IS 10
                                  USAGE IS COMP-1.
 77 STATIS               PIC 9(5)  USAGE IS COMP-1.
 77 RECORD-LENGTH        PIC 9(5)  VALUE IS 80
                                  USAGE IS COMP-1.
 77 RECORD-AREA-LENGTH   PIC 9(5)  USAGE IS COMP-1.
 77 BYTES-RECEIVED       PIC 9(5)  USAGE IS COMP-1.
 77 OFILRCRD             PIC X(80).
 77 ALLDONE              PIC 9(5) VALUE IS ZERO
                                  USAGE IS COMP-1.
*SORT-CONTROL-BLOCK CONTAINS THE SORT/MERGE
*CONTROL SPECIFICATIONS.
 01 SORT-CONTROL-BLOCK.
      03 HEADER          PIC X(44)  VALUE IS
           "00000HSORTR 00006A          0080          ".
* OUTPUT IS PASSED DIRECTLY FROM THE SORT TO THE FILE.
      03 OUT-FILE-SPEC   PIC X(44) VALUE IS
           "00002DOSDS01.OUTX                         ".
      03 CNT-OUT-FILE-SPEC  PIC X(44) VALUE IS
           "00004DA00800800                           ".
      03 WRK-FILE-SPEC   PIC X(44) VALUE IS
           "00006DWEDS01                              ".
* INPUT IS DIRECTLY FROM A FILE TO THE SORT.
      03 INPT-FILE-DESCRPT PIC X(44) VALUE IS
           "00003DISDS01.SRTDAT                       ".
      03 INPT-FILE-CONTIN PIC X(44) VALUE IS
           "00010DA         00000400                  ".
      03 REFORMAT-DESCRIPT-0 PIC X(44) VALUE IS
           "00012FNC00320037                          ".
      03 REFORMAT-DESCRIPTION PIC X(44) VALUE IS
           "00015FDC00010031                          ".
      03 REFORMAT-DESCRPT-2 PIC X(44) VALUE IS
           "00017FDC00380080                          ".
      03 ENDKRD     PIC X(44)  VALUE IS
           "/*                                        ".
```

Figure 9-5. COBOL Program Calls Sort/Merge, Inputs Files from Disk, and Outputs Files to Disk

```
 PROCEDURE DIVISION.
*---------------------------------------COBOL EXAMPLE
 MAIN-PROGRAM.
*INITIALIZE SORT/MERGE:
     CALL "SRTINT" USING SORT-CONTROL-BLOCK,
                         MAX-NO-RECS, STATIS.
     IF STATIS NOT EQUAL ZERO GO TO ERRSTRT.
*---------------------------------------COBOL EXAMPLE
 CKSTATS.
*WAIT UNTIL THE SORT/MERGE IS DONE.
    CALL "SMSTAT" USING STATIS.
     IF STATIS EQUAL ZERO
        DISPLAY " SORT/MERGE COMPLETE, NO ERRORS. "
     ELSE DISPLAY " SORT/MERGE COMPLETED ABNORMALLY. ".
     GO TO END-IT.
*---------------------------------------COBOL EXAMPLE
 ERRSTRT.
      DISPLAY " ERROR IN STRINT CALL. ".
 END-IT.
      STOP RUN.
 END PROGRAM.
```

Figure 9-5. COBOL Program Calls SORT/MERGE, Inputs Files from Disk, and Outputs Files to Disk (Sheet 2 of 2)

```
 IDENTIFICATION DIVISION.
 PROGRAM-ID. CPNP.
*THIS IS A SORT/MERGE INFORMATION QUEUE TEST. IT TESTS THE
*CASE WHERE INPUT IS DIRECTED BY THE COBOL PROGRAM (@PROC@)
*AND OUTPUT IS DIRECTLY FROM SORT/MERGE TO A FILE (NO @PROC@).
 AUTHOR. TEXAS INSTRUMENTS.
 DATE-WRITTEN. 11-6-80.
 ENVIRONMENT DIVISION.
 CONFIGURATION SECTION.
 SOURCE-COMPUTER. TI-990-10.
 OBJECT-COMPUTER. TI-990-10.
 INPUT-OUTPUT SECTION.
 FILE-CONTROL.
      SELECT INFILNAME ASSIGN TO INPUT "INFILE";
             ACCESS MODE IS SEQUENTIAL.
 DATA DIVISION.
 FILE SECTION.
 FD INFILNAME
      DATA RECORD IS INFILRCRD
      LABEL RECORDS ARE STANDARD
      RECORD CONTAINS 80 CHARACTERS
      BLOCK CONTAINS 10 RECORDS.
 01 INFILRCRD       PIC X(80).
```

Figure 9-6. COBOL Program calls Sort/Merge, Inputs Files from Calling Task, and Outputs Files to Disk (Sheet 1 of 4)

```
 WORKING-STORAGE SECTION.
 77 MAX-NO-RECS          PIC 9(5)  VALUE IS 10
                         USAGE IS COMP-1.
 77 STATIS               PIC 9(5)  USAGE IS COMP-1.
 77 RECORD-LENGTH        PIC 9(5)  VALUE IS 80
                                   USAGE IS COMP-1.
 77 RECORD-AREA-LENGTH   PIC 9(5)  USAGE IS COMP-1.
 77 BYTES-RECEIVED       PIC 9(5)  USAGE IS COMP-1.
 77 OFILRCRD             PIC X(80).
 77 ALLDONE              PIC 9(5) VALUE IS ZERO
                         USAGE IS COMP-1.
*SORT-CONTROL-BLOCK CONTAINS THE SORT/MERGE
*CONTROL SPECIFICATIONS.
 01 SORT-CONTROL-BLOCK.
      03 HEADER.
         05 SEQ           PIC X(5)  VALUE IS "00000".
         05 FILLER        PIC A     VALUE IS "H".
         05 SORT-TYPE     PIC A(6)  VALUE IS "SORTR ".
         05 MAX-TOT-CONTL-LEN PIC 9(5) VALUE IS 6.
         05 ASCND-DSCND PIC A       VALUE IS "A".
         05 FILLER        PIC X(7)  VALUE IS SPACES.
         05 COLLATNG-SEQ PIC X      VALUE IS SPACE.
         05 PRINT-OPTION PIC X      VALUE IS SPACE.
         05 OUTPUT-OPTION PIC X     VALUE IS SPACE.
         05 OUTPUT-REC-LEN PIC X(4) VALUE IS "0080".
         05 VERIFY-OPTN     PIC X   VALUE IS SPACE.
         05 WRK-SPACE     PIC X(5)  VALUE IS SPACES.
         05 FILLER        PIC X(6)  VALUE IS SPACES.
*OUTPUT IS PASSED DIRECTLY FROM THE SORT TO A FILE.
      03 OUT-FILE-SPEC.
         05 SEQ           PIC X(5)  VALUE IS "00001".
         05 FILLER        PIC A     VALUE IS "D".
         05 FILE-USE      PIC A     VALUE IS "O".
         05 FILE-TYPE     PIC A     VALUE IS "S".
         05 PATHNAME      PIC X(36) VALUE IS "DS01.OUTX".
      03 CNT-OUT-FILE-SPEC.
         05 SEQ           PIC X(5)  VALUE IS "00002".
         05 FILLER        PIC A     VALUE IS "D".
         05 FILE-USE      PIC A     VALUE IS "A".
         05 LOG-REC-SIZ PIC 9(4)    VALUE IS 80.
         05 PHY-REC-SIZ PIC 9(4)    VALUE IS 800.
         05 NUM-LOG-REC PIC X(8)    VALUE IS SPACES.
         05 FILLER        PIC X(21) VALUE IS SPACES.
      03 WRK-FILE-SPEC.
         05 SEQ           PIC X(5)  VALUE IS "00003".
         05 FILLER        PIC A     VALUE IS "D".
         05 FILE-USE      PIC A     VALUE IS "W".
         05 EXPAND-ALLOC-FLG PIC X VALUE IS "E".
         05 VOLUME        PIC X(8)  VALUE IS "DS01".
         05 FILLER        PIC X(28) VALUE IS SPACES.
```

Figure 9-6. COBOL Program Calls Sort/Merge, Inputs Files from Calling Task, and Outputs Files to Disk (Sheet 2 of 4)

```
*INPUT IS DIRECTED BY THE COBOL PROGRAM.
      03 INPT-FILE-DESCRPT.
         05 SEQ           PIC X(5)  VALUE IS "00004".
         05 FILLER        PIC A     VALUE IS "D".
         05 FILE-USE      PIC A     VALUE IS "I".
         05 FILE-TYPE     PIC A     VALUE IS "S".
         05 PATHNAME      PIC X(36) VALUE IS "@PROC@".
      03 INPT-FILE-CONTIN.
         05 SEQ           PIC X(5)  VALUE IS "00008"
         05 FILLER        PIC A     VALUE IS "D".
         05 FILE-USE      PIC A     VALUE IS "A".
         05 LOG-SIZE      PIC X(4)  VALUE IS "0080".
         05 FILLER        PIC X(4)  VALUE IS SPACES.
         05 NUM-SRT-RECS PIC 9(8) VALUE IS   400.
         05 FILLER        PIC X(21) VALUE IS SPACES.
      03 REFORMAT-DESCRIPTION-0.
         05 SEQ           PIC X(5)  VALUE IS "00010".
         05 FILLER        PIC A     VALUE IS "F".
         05 FIELD-TYPE-CMMT PIC X VALUE IS "N".
         05 CHARACTR-USE PIC A    VALUE IS "C".
         05 FIELD-LOC.
            07 BEG-RECRD-POS PIC X(4) VALUE IS "0032".
            07 END-RECRD-POS PIC X(4) VALUE IS "0037".
         05 CONDTN-FORCD-CHAR PIC X   VALUE IS SPACE.
         05 FORCD-CHAR  PIC X      VALUE IS SPACE.
         05 CONTIN-LIN  PIC X      VALUE IS SPACE.
         05 OUFLW-FLD-LEN PIC X(3) VALUE IS SPACES.
         05 FILLER        PIC X(22) VALUE IS SPACES.
      03 REFORMAT-DESCRIPTION.
         05 SEQ           PIC X(5)  VALUE IS "00014".
         05 FILLER        PIC A     VALUE IS "F".
         05 FIELD-TYPE-CMMT PIC X VALUE IS "D".
         05 CHARACTR-USE PIC A    VALUE IS "C".
         05 FIELD-LOC.
            07 BEG-RECRD-POS PIC X(4) VALUE IS "0001".
            07 END-RECRD-POS PIC X(4) VALUE IS "0031".
         05 CONDTN-FORCD-CHAR PIC X   VALUE IS SPACE.
         05 FORCD-CHAR  PIC X         VALUE IS SPACE.
         05 CONTIN-LIN  PIC X         VALUE IS SPACE.
         05 OUFLW-FLD-LEN PIC X(3)    VALUE IS SPACES.
         05 FILLER        PIC X(22)   VALUE IS SPACES.
      03 REFORMAT-DESCRIPTION-3.
         05 SEQ           PIC X(5)   VALUE IS "00016".
         05 FILLER        PIC A      VALUE IS "F".
         05 FIELD-TYPE-CMMT PIC X  VALUE IS "D".
         05 CHARACTR-USE PIC A     VALUE IS "C".
         05 FIELD-LOC.
            07 BEG-RECRD-POS PIC X(4) VALUE IS "0038".
            07 END-RECRD-POS PIC X(4) VALUE IS "0080".
         05 CONDTN-FORCD-CHAR PIC X   VALUE IS SPACE.
         05 FORCD-CHAR  PIC X       VALUE IS SPACE.
```

Figure 9-6. COBOL Program Calls Sort/Merge, Inputs Files from Calling Task, and Outputs Files to Disk (Sheet 3 of 4)

```
          05 CONTIN-LIN  PIC X       VALUE IS SPACE.
          05 OUFLW-FLD-LEN PIC X(3) VALUE IS SPACES.
          05 FILLER      PIC X(22)   VALUE IS SPACES.
       03 ENDKRD         PIC X(44)   VALUE IS "/*".
 PROCEDURE DIVISION.
*-----------------------------------------------COBOL EXAMPLE
 MAIN-PROGRAM.
*INITIALIZE SORT/MERGE.
     CALL "SRTINT" USING SORT-CONTROL-BLOCK,
                         MAX-NO-RECS, STATIS.
     IF STATIS NOT EQUAL ZERO GO TO ERRSTRT.
*-----------------------------------------------COBOL EXAMPLE
* START THE INPUT SECTION.
      OPEN INPUT INFILNAME.
 NEXREC.
      READ INFILNAME AT END GO TO BEGWRT.
      CALL "SENREC" USING INFILRCRD,
                    RECORD-LENGTH, STATIS.
       IF STATIS NOT EQUAL ZERO GO TO ERRSEN.
      GO TO NEXREC.
*-----------------------------------------------COBOL EXAMPLE
* START THE OUTPUT SECTION.
 BEGWRT.
*-----------------------------------------------COBOL EXAMPLE
*  BEGIN SORT PHASE. SENDING A RECORD LENGTH OF 0 (ALLDONE)
*  INDICATES THAT THE LAST RECORD HAS BEEN SENT.
      CALL "SENREC" USING INFILRCRD, ALLDONE, STATIS.
      IF STATIS NOT EQUAL ZERO GO TO ERRSEN.
*-----------------------------------------------COBOL EXAMPLE
      CLOSE INFILNAME.
 CHKSORT.
      CALL "SMSTAT" USING STATIS.
      IF STATIS NOT EQUAL ZERO GO TO ERRWRT.
* SORT IS DONE.
      GO TO END-IT.
*-----------------------------------------------COBOL EXAMPLE
 ERRSTRT.
      DISPLAY " ERROR IN STRINT CALL. ".
      GO TO END-IT.
 ERRSEN.
      DISPLAY " ERROR IN SENREC CALL. ".
      GO TO END-IT.
 ERRWRT.
      DISPLAY " ERROR IN SMSTAT. ".
*-----------------------------------------------COBOL EXAMPLE
 END-IT.
      STOP RUN.
 END PROGRAM.
```

Figure 9-6. COBOL Program Calls Sort/Merge, Inputs Files from Calling Task, and Outputs Files to Disk (Sheet 4 of 4)

```
 IDENTIFICATION DIVISION.
 PROGRAM-ID.  CSORTA.
*THIS IS A SORT/MERGE INFORMATION QUEUE TEST.  IT
*TESTS THE CASE WHERE INPUT IS DIRECTLY FROM A
*FILE (NO @PROC@) AND OUTPUT IS RETRIEVED BY THE
*COBOL TASK.  (@PROC@).
 AUTHOR.  TEXAS INSTRUMENTS  FDT.
 DATE-WRITTEN.  04-9-79.
 ENVIRONMENT DIVISION.
 CONFIGURATION SECTION.
 SOURCE-COMPUTER.  TI-990-10.
 OBJECT-COMPUTER.  TI-990-10.
 INPUT-OUTPUT SECTION.
 FILE-CONTROL.
      SELECT OUTFLNAM ASSIGN TO OUTPUT "OUTFL";
             ACCESS MODE IS SEQUENTIAL.
      SELECT INFILNAM ASSIGN TO RANDOM "INFILE";
             ORGANIZATION IS RELATIVE;
             ACCESS MODE IS DYNAMIC;
             RELATIVE KEY IS REC-NUMBER;
             FILE STATUS IS RR-STATUS.
 DATA DIVISION.
 FILE SECTION.
 FD OUTFLNAM
      DATA RECORD IS OUTREC
      LABEL RECORDS ARE STANDARD
      RECORD CONTAINS 20 CHARACTERS
      BLOCK CONTAINS 10 RECORDS.
 01 OUTREC.
    03  RRNUM       PIC XXXXXX.
    03  ONEREC      PIC X(14).
 FD INFILNAM
      DATA RECORD IS INREC
      LABEL RECORDS ARE STANDARD
      RECORD CONTAINS 6 CHARACTERS
      BLOCK CONTAINS 1000 RECORDS.
 01 INREC           PIC X(6).
 WORKING-STORAGE SECTION.
 77 MAX-NO-RECS          PIC 9(5)  VALUE IS 8
                         USAGE IS COMP-1.
 77 STATIS               PIC 9(5)  USAGE IS COMP-1.
 77 RECORD-LENGTH        PIC 9(5)  VALUE IS 80
                                   USAGE IS COMP-1.
 77 RECORD-AREA-LENGTH   PIC 9(5)  USAGE IS COMP-1.
 77 BYTES-RECEIVED       PIC 9(5)  USAGE IS COMP-1.
 77 OFILRCRD             PIC X(80).
 77 ALLDONE              PIC 9(5) VALUE IS ZERO
                         USAGE IS COMP-1.
 77 OPN-FLAG             PIC 9(5) VALUE IS 0
                         USAGE IS COMP-1.
 77 REC-NUMBER           PIC 99999 USAGE IS COMP.
```

Figure 9-7. COBOL Routine that Executes Sort/Merge on Relative Record File Containing More Than 65,536 Records (Sheet 1 of 3)

```
  77 MAX-REC                PIC 99999 VALUE IS 65536
                            USAGE IS COMP.
  77 RR-STATUS              PIC XX.
  01 RECS.
     03 REC-NUMBER1          PIC 9(5) USAGE IS COMP-1.
     03 REC-NUMBER2          PIC 9(5) USAGE IS COMP-1.
  01 ERR-REC.
     03 ERR-RR               PIC XXXXXX.
     03 FILLER               PIC X(9) VALUE IS
                            " ERROR = ".
     03 ERR-STATUS           PIC XX.
*SORT-CONTROL-BLOCK CONTAINS THE SORT/MERGE
*CONTROL SPECIFICATIONS.
 01 SORT-CONTROL-BLOCK.
    03 HEADER PIC X(44) VALUE IS
    "00000HSORTA     5A         0        30000".
    03 OUTFILE-SPEC PIC X(44) VALUE IS
    "00010DOS@PROC@".
    03 OUTFILE-SPEC 2 PIC X(44) VALUE IS
    "00020DA          ".
    03 WORKFILE-SPEC PIC X(44) VALUE IS
    "00030DWEDS01".
    03 INPUT-SPEC-1 PIC X(44) VALUE IS
    "00040DIR.REL80".
    03 INPUT-SPEC-2 PIC X(44) VALUE IS
    "00050DA".
    03 REFORMAT-1 PIC X(44) VALUE IS
    "00080FNC    1    5".
    03 ENDKRD          PIC X(44)  VALUE IS "/*".
 PROCEDURE  DIVISION.
*-----------------------------------------COBOL EXAMPLE
 MAIN-PROGRAM.
      OPEN OUTPUT OUTFLNAM.
*INITIALIZE SORT/MERGE.
     CALL "SRTINT" USING SORT-CONTROL-BLOCK,
                   MAX-NO-RECS, STATIS.
     IF STATIS NOT EQUAL ZERO GO TO ERRSTRT.
*-----------------------------------------COBOL EXAMPLE
 CHKSORT.
* RECEIVE RECORDS.
     CALL "RCVREC" USING RECS, RECORD-LENGTH,
                  BYTES-RECEIVED, STATIS.
     IF BYTES-RECEIVED EQUAL ZERO GO TO END-IT.
* OPEN INPUT FILE, IF NOT ALREADY OPENED.
     IF OPN-FLAG EQUAL ZERO
        OPEN INPUT INFILNAM.
     MOVE 1 TO OPN-FLAG.
```

Figure 9-7. COBOL Routine that Executes Sort/Merge on Relative Record File Containing More Than 65,536 Records (Sheet 2 of 3)

```
* COMPUTE ABSOLUTE RELATIVE RECORD NUMBER.
     MULTIPLY REC-NUMBER1 BY MAX-REC GIVING
      REC-NUMBER.
     ADD REC-NUMBER2 TO REC-NUMBER.
* MAKE RECORD NUMBER RELATIVE TO ONE.
     ADD 1 TO REC-NUMBER.
* READ ONLY THOSE ABOVE 65536.
     IF REC-NUMBER LESS THAN MAX-REC GO TO CHKSORT.
*
     READ INFILNAM WITH NO LOCK INVALID KEY GO TO
        KEYERR.
* SET UP TO OUTPUT RECORD.
     MOVE INREC TO ONEREC.
* MAKE RECORD NUMBER RELATIVE TO ZERO.
     SUBTRACT 1 FROM REC-NUMBER.
     MOVE REC-NUMBER TO RRNUM.
     WRITE OUTREC.
*
      GO TO CHKSORT.
*---------------- ----------------------COBOL EXAMPLE
 ERRSTRT.
      DISPLAY " ERROR IN STRINT CALL. ".
      GO TO END-IT.
 KEYERR.
      SUBTRACT 1 FROM REC-NUMBER.
      MOVE REC-NUMBER TO ERR-RR.
      MOVE RR-STATUS TO ERR-STATUS.
      WRITE OUTREC FROM ERR-REC.
      GO TO CHKSORT.
 ERRWRT.
      DISPLAY " ERROR IN RCVREC CALL. ".
*---------------------------------------COBOL EXAMPLE
 END-IT.
      CLOSE OUTFLNAM.
      STOP RUN.
 END PROGRAM.
```

Figure 9-7. COBOL Routine that Executes Sort/Merge on Relative Record File Containing More Than 65,536 Records (Sheet 3 of 3)

TI 990 COBOL

```
WORKING-STORAGE SECTION.
01 SMIB.

   03  INPUT. . .
          .
          .
          .
   CALL "SRTINT" USING SMIB, NO-CTRL-REC,
   STATIS.




ASR.
     READ FN INTO RN AT END GO TO SOR.

     CALL "SENREC" USING RECORD-AREA,
     RCRD-LENGTH, STATIS.

     GO TO ASR.

     CALL "SENREC" USING RECORD-AREA, 0, STATIS.
SOR.
BSR.
     CALL "RCVREC" USING RECORD-AREA
     RCRD-LENGTH, RETURN-SIZE, STATIS.

     IF STATIS = 2 GO TO DONE.

     WRITE RN2.
     GO TO BSR.


DONE.
```

ANSI COBOL

```
FILE SECTION.
SD   SORTF
     DATA RECORD IS SRN.

01   SRN.
          .
          .
          .
     SORT SORTF ON ASCEND-
     ING KEY X
     INPUT PROCEDURE IS ASR
     OUTPUT PROCEDURE IS BSR
     GO TO DONE.

ASR  SECTION.
AR.  TO READ FN INTO RN AT
     END GO TO A-EX.
     RELEASE SRN FROM RN.

     GO TO AR.

A-EX.     EXIT.

BSR SECTION.
BR.  RETURN SORTF INTO RN2
AT END GO TO B-EX.

     WRITE RN2.
     GO TO BR.

     B-EX. EXIT.

DONE SECTION.
          .
          .
          .
```

Figure 9-8. Comparison of 990 COBOL Calling Sort/Merge and ANSI COBOL SORT Verb

```
SORT file-name-1 ON { DESCENDING } KEY  identifier-1 ...
                    { ASCENDING  }

[ ON { DESCENDING } KEY  identifier-2 ...]...
     { ASCENDING  }

INPUT PROCEDURE IS section-name-1 [THRU section-name-2]

OUTPUT PROCEDURE IS section-name-3 [THRU section-name-4]

990 COBOL     Input  :  @PROC@ SENREC
  CALL        Output :  @PROC@ RCVREC

SORT file-name 1 ON { DESCENDING } KEY  identifier-1  ...
                    { ASCENDING  }

[ ON { DESCENDING } KEY  identifier-2 ...]...
     { ASCENDING  }

USING file-name-2

OUTPUT PROCEDURE IS section-name-3 [THRU section-name-4]

990 COBOL     Input  :  File
  CALL        Output :  @PROC@ RCVREC

SORT file-name-1 ON { DESCENDING } KEY  identifier-1 ...
                    { ASCENDING  }

[ ON { DESCENDING } KEY  identifier-2 ...]...
     { ASCENDING  }

INPUT PROCEDURE IS section-name-1 [THRU section-name-2]

GIVING file-name-3

990 COBOL     Input  :  @PROC@ SENREC
  CALL        Output :  FILE   (SMSTAT)

SORT file-name-1 ON { DESCENDING } KEY  identifier-2 ...
                    { ASCENDING  }

[ ON { DESCENDING } KEY  identifier-2 ...]...
     { ASCENDING  }

USING file-name-2

GIVING file-name-3

990 COBOL     Input  :  FILE
  CALL        Output :  FILE   (SMSTAT)
```

Figure 9-9. CALL Verb Combinations

9.3.3 Pascal Examples

Figure 9-10 is an example of a Pascal routine that calls Sort/Merge to input records from a disk file and output the sorted records to a second disk file. Note the use of SETDLY to modify the time delay count.

Figure 9-11 shows a Pascal routine that calls Sort/Merge and passes records read by the Pascal routine to Sort/Merge. The sorted records are returned to the Pascal routine, which outputs them to a disk file.

You must declare external procedures by using the FORTRAN linkage convention.

```
PROGRAM PCNPNP;
 (*
  PROGRAM TITLE: PCNPNP

  ABSTRACT:
    THIS IS A SORT/MERGE INFORMATION QUEUE TEST.  IT
    TESTS THE CASE WHERE BOTH INPUT AND OUTPUT FILES
    ARE ACCESSED DIRECTLY BY SORT/MERGE.

  *)


CONST  MAX_NO_RECS = 10;
       SECNDS = 26;

TYPE   CONTROL_RECORD = PACKED ARRAY [1..44] OF CHAR;

VAR
   (* THE FOLLOWING STATEMENTS ARE THE CONTROL RECORDS FOR
      SRTINT.  THEY MUST BE DECLARED AS ONE CONTIGUOUS BLOCK *)
       SCB_HEADER             : CONTROL_RECORD;
       SCB_OUTFILE_SPEC       : CONTROL_RECORD;
       SCB_OUTFILE_SPEC2      : CONTROL_RECORD;
       SCB_WRKFILE_SPEC       : CONTROL_RECORD;
       SCB_INPFILE_SPEC       : CONTROL_RECORD;
       SCB_INPFILE_SPEC2      : CONTROL_RECORD;
       SCB_REF_DESC_1         : CONTROL_RECORD;
       SCB_REF_DESC_2         : CONTROL_RECORD;
       SCB_REF_DESC_3         : CONTROL_RECORD;
       SCB_ENDKRD             : CONTROL_RECORD;
     (*  END OF SORT CONTROL RECORDS    *)

       STATIS                 : INTEGER;

PROCEDURE SRTINT (VAR SORT_CONTROL_BLOCK : CONTROL_RECORD;
                  MAX_NO_RECS : INTEGER;
                  VAR STATIS : INTEGER); EXTERNAL FORTRAN;

PROCEDURE SETDLY (SECNDS : INTEGER); EXTERNAL FORTRAN;
```

Figure 9-10. Pascal Program Calls Sort/Merge to Input and Output Files from Disk (Sheet 1 of 2)

```
PROCEDURE SMSTAT (VAR STATIS : INTEGER); EXTERNAL FORTRAN;

BEGIN

   (* INITIALIZE SORT CONTROL BLOCK  *)

       SCB_HEADER :=
         '00000HSORTR      6D          4    80   12000      ';
       SCB_OUTFILE_SPEC :=
         '00001DOS.OUTX                                    ';
       SCB_OUTFILE_SPEC2 :=
         '00002DA00800864                                  ';
       SCB_WRKFILE_SPEC :=
         '00003DWEDS01                                     ';
       SCB_INPFILE_SPEC :=
         '00004DIS.SRTDAT                                  ';
       SCB_INPFILE_SPEC2 :=
         '00008DA0080           400                        ';
       SCB_REF_DESC_1 :=
         '00010FNC   32  37                                ';
       SCB_REF_DESC_2 :=
         '00014FDC    1  31                                ';
       SCB_REF_DESC_3 :=
         '00016FDC   38  80                                ';
       SCB_ENDKRD :=
         '/*                                               ';

  (*  SET UP TIME DELAY COUNT FOR SMSTAT *)

       SETDLY(SECNDS);

  (*  INITIALIZE SORT/MERGE  *)

       SRTINT(SCB_HEADER, MAX_NO_RECS, STATIS);
       IF STATIS <> 0 THEN
          MESSAGE ( 'ERROR IN SRTINT CALL.  ')

       ELSE
       BEGIN
  (*  WAIT FOR SORT/MERGE TO COMPLETE   *)
          SMSTAT(STATIS);
          IF STATIS <> 0 THEN
            MESSAGE ( 'ERROR IN SMSTAT CALL.  ')

       END;
END.
```

Figure 9-10. Pascal Program Calls Sort/Merge to Input and Output Files from Disk (Sheet 2 of 2)

```
PROGRAM PCPP;
 (*
  PROGRAM TITLE:  PCPP

  ABSTRACT:
    THIS IS A SORT/MERGE INFORMATION QUEUE TEST.  IT
    TESTS THE CASE WHERE BOTH INPUT AND OUTPUT ARE
    DIRECTED BY THE PASCAL PROGRAM  ( @PROC@ ).

  *)

CONST  MAX_NO_RECS = 10;
       RECORD_LENGTH = 80;
       ALLDONE = 0;

TYPE   CONTROL_RECORD = PACKED ARRAY [1..44] OF CHAR;
       DATA_REC = PACKED ARRAY [1..80] OF CHAR;

VAR
     (* THE FOLLOWING STATEMENTS ARE THE CONTROL RECORDS FOR
        SRTINT.  THEY MUST BE DECLARED AS ONE CONTIGUOUS BLOCK
       SCB_HEADER             : CONTROL_RECORD;
       SCB_OUTFILE_SPEC       : CONTROL_RECORD;
       SCB_OUTFILE_SPEC2      : CONTROL_RECORD;
       SCB_WRKFILE_SPEC       : CONTROL_RECORD;
       SCB_INPFILE_SPEC       : CONTROL_RECORD;
       SCB_INPFILE_SPEC2      : CONTROL_RECORD;
       SCB_REF_DESC_1         : CONTROL_RECORD;
       SCB_REF_DESC_2         : CONTROL_RECORD;
       SCB_REF_DESC_3         : CONTROL_RECORD;
       SCB_ENDKRD             : CONTROL_RECORD;
  (*  END OF SORT CONTROL RECORDS                              *)

       INFILREC               : DATA_REC;
       OUTFILREC              : DATA_REC;
       STATIS                 : INTEGER;
       BYTES_RECEIVED         : INTEGER;
       ESC_FLAG               : BOOLEAN;
       INFILE                 : FILE OF DATA_REC;
       OUTFL                  : FILE OF DATA_REC;

PROCEDURE SRTINT (VAR SORT_CONTROL_BLOCK : CONTROL_RECORD;
                  MAX_NO_RECS : INTEGER;
                  VAR STATIS : INTEGER); EXTERNAL FORTRAN;
PROCEDURE SENREC (VAR INFILREC : DATA-REC;
                  RECORD_LENGTH : INTEGER;
                  VAR STATIS : INTEGER); EXTERNAL FORTRAN;
PROCEDURE RCVREC (VAR OUTFILREC : DATA_REC;
                  RECORD_LENGTH : INTEGER;
                  VAR BYTES_RECEIVED : INTEGER;
                  VAR STATIS : INTEGER ); EXTERNAL FORTRAN;
```

Figure 9-11. Pascal Program Calls Sort/Merge to Input and Output Files from Calling Task (Sheet 1 of 3)

```
BEGIN

   (* INITIALIZE SORT CONTROL BLOCK  *)

      SCB_HEADER :=
        '00000HSORTR      6A          4    80   12000       ';
      SCB_OUTFILE_SPEC :=
        '00001DOS@PROC@                                     ';
      SCB_OUTFILE_SPEC2 :=
        '00002DA00300864                                    ';
      SCB_WRKFILE_SPEC :=
        '00003DWEDS01                                       ';
      SCB_INPFILE_SPEC :=
        '00004DIS@PROC@                                     ';
      SCB_INPFILE_SPEC2 :=
        '00008DA0030           400                          ';
      SCB_REF_DESC_1 :=
        '00010FNC   32   37                                 ';
      SCB_REF_DESC_2 :=
        '00014FDC    1   31                                 ';
      SCB_REF_DESC_3 :=
        '00016FDC   38   80                                 ';
      SCB_ENDKRD :=
        '/*                                                 ';

      RESET(INFILE);

  (*  INITIALIZE SORT/MERGE  *)

      SRTINT(SCB_HEADER, MAX_NO_RECS, STATIS);
      IF STATIS <> 0 THEN
         MESSAGE ( 'ERROR IN SRTINT CALL.  ')
      ELSE
      BEGIN
                  VAR STATIS : INTEGER); EXTERNAL FORTRAN;
  (*  READ RECORDS FROM INPUT FILE AND SEND TO SORT/MERGE *)
        REPEAT
          READ(INFILE, INFILREC);
          SENREC(INFILREC, RECORD_LENGTH, STATIS);
        UNTIL EOF(INFILE) OR (STATIS <> 0);

        IF STATIS <> 0 THEN
          MESSAGE ( 'ERROR IN SENREC CALL.  ')
        ELSE
        BEGIN
          (* SEND RECORD LENGTH OF ZERO  *)
          SENREC(INFILREC, ALLDONE, STATIS);
          IF STATIS <> 0 THEN
            MESSAGE ( 'ERROR IN SENREC CALL.  ')
```

Figure 9-11. Pascal Program Calls Sort/Merge to Input and Output Files from Calling Task (Sheet 2 of 3)

```
                ELSE
                BEGIN
                 (* OPEN OUTPUT FILE  *)
                  REWRITE (OUTFL);
                  ESC_FLAG := FALSE;
                  WHILE ESC_FLAG = FALSE DO
                  BEGIN
    (*  WRITE SORTED RECORDS TO OUTPUT FILE  *)
                   RCVREC(OUTFILREC, RECORD_LENGTH,
                          BYTES_RECEIVED, STATIS);
                   IF (STATIS = 0) AND (BYTES_RECEIVED <> 0 )
                      THEN WRITE (OUTFL, OUTFILREC)
                      ELSE ESC_FLAG := TRUE
                 END;
                 IF STATIS = 1 THEN
                  MESSAGE ( 'ERROR IN RCVREC CALL.  ')
              END
            END
          END;
END.
```

Figure 9-11. Pascal Program Calls Sort/Merge to Input and Output Files from Calling Task (Sheet 3 of 3)

9.3.4 TI 990 BASIC Examples

Figure 9-12 is an example of a TI 990 BASIC program that calls Sort/Merge and passes records read by the BASIC program to Sort/Merge.

Figure 9-13 shows a TI 990 BASIC program that calls Sort/Merge to input records from a disk file and output the sorted records to a second disk file.

9.4 DEBUG AID

The intertask communication subroutines suppress the values for the normal print option (0, 1, 2, or 3 in column 27 of the Header Specification). This suppression occurs because you cannot specify a listing device or file. Problems can occur in debugging these programs since the STATUS parameters that are returned in many cases do not diagnose the problem.

To aid in the debugging process, you can place a 4 in column 27 of the Header Specification; as a result, Sort/Merge returns to the device named ME a listing of the information, error, and warning messages. This allows easy diagnosis of problems during debugging, although you must ensure that there is no contention for the device ME.

```
100  REM  PROGRAM NAME = BPP
110  REM  THIS PROGRAM TESTS THE INTERFACE BETWEEN BASIC
120  REM AND  SORT/MERGE WHERE BOTH INPUT AND OUTPUT ARE
130  REM  DIRECTED BY THE BASIC PROGRAM (@PROC@)
140  REM
150  LIBRARY "*.SMLIB.BASCSM"
160  INTEGER ALL
170  DIM A$(10)
180  REM  INITIALIZE CALLS TO SORT/MERGE ROUTINES
190  SRTINT = 1
200  SENREC = 3
210  RCVREC = 4
220  REM  INITIALIZE SORT CONTROL BLOCK
230  A$(1) = "00000HSORTR 00006A         4 0080  16000     "
240  A$(2) = "00002DOS@PROC@                               "
250  A$(3) = "00004DA00800800                              "
260  A$(4) = "00006DWEDS01                                 "
270  A$(5) = "00008DIS@PROC@                               "
280  A$(6) = "00010DA0080                                  "
290  A$(7) = "00012FNC00320037                             "
300  A$(8) = "00014FDC00010031                             "
310  A$(9) = "00016FDC00380080                             "
320  A$(10) = "/*                                          "
330  N = 10
340  S = -1
350  CALL ".SMLIB.BASCSM"(SRTINT,A$(1),N,S)
360  IF S = -1 THEN 650
370  IF S > 0 THEN 710
380  OPEN #20:"INFILE"
390  IF EOF (20) THEN 470
400  INPUT #20:S$
410  L = LEN(S$)
420  S = -1
430  CALL ".SMLIB.BASCSM"(SENREC,S$,L,S)
440  IF S = -1 THEN 670
450  IF S > 0 THEN 730
460  GOTO 390
470  L = 0
480  S = -1
490  CALL ".SMLIB.BASCSM"(SENREC,S$,L,S)
500  IF S = -1 THEN 670
510  IF S > 0 THEN 730
520  CLOSE #20
530  REM  OPEN FILE FOR OUTPUT
540  OPEN #22:".SMTST.SMOUT.BPP",UPDATE,SEQUENTIAL
550  L = 80
560  S = -1
570  REM  INITIALIZE STRING LENGTH TO 80
580  S$ = RPT$(" ",L)
590  CALL ".SMLIB.BASCSM"(RCVREC,S$,L,B,S)
600  IF S = -1 THEN 690
```

Figure 9-12. BASIC Program Calls Sort/Merge to Input and Output Files from Calling Task (Sheet 1 of 2)

```
610  IF B = 0 THEN 770
620  IF S <> 0 THEN 750
630  PRINT #22: S$
640  GOTO 550
650  PRINT "INVALID PARAMETER LIST IN SRTINT CALL"
660  GO TO 790
670  PRINT "INVALID PARAMETER LIST IN SENREC CALL"
680  GO TO 790
690  PRINT "INVALID PARAMETER LIST IN RCVREC CALL"
700  GOTO 790
710  PRINT "ERROR IN SRTINT ",S
720  GO TO 790
730  PRINT "ERROR IN SENREC ",S
740  GO TO 790
750  PRINT "ERROR IN RCVREC ",S
760  GOTO 780
770  PRINT "SORT/MERGE TERMINATED NORMALLY"
780  CLOSE #22
790  STOP
800  END
```

Figure 9-12. BASIC Program Calls Sort/Merge to Input and Output Files from Calling Task (Sheet 2 of 2)

NOTE

The designation of 4 as the Print Option value is not available in utility specifications and will result in an error if used.

When using this debugging aid from a COBOL program, be sure that the MESSAGE ACCESS NAME prompt field does not contain ME.

9.5 SYSTEM GENERATION

You must specify an intertask communication area during system generation (sysgen) to use Sort/Merge with COBOL, FORTRAN, Pascal, BASIC, or assembly language. In response to the INTERTASK parameter requested at sysgen, specify the number of words of system table area that may be used for intertask communication.

```
100  REM  PROGRAM NAME = BNPNP
110  REM  THIS PROGRAM TESTS THE INTERFACE BETWEEN BASIC AND
120  REM  SORT/MERGE WHERE BOTH INPUT AND OUTPUT IS DIRECTLY
130  REM  TO A FILE
140  REM
150  LIBRARY "*.SMLIB.BASCSM"
160  INTEGER SRTINT,SMSTAT,SETDLY
170  INTEGER L,N,S
180  DIM A$(10)
190  REM  INITIALIZE CALLS TO SORT/MERGE ROUTINES
200  SRTINT = 1
210  SETDLY = 2
220  SMSTAT = 3
230  REM  INITIALIZE SORT CONTROL BLOCK
240  A$(1) = "00000HSORTR 00006A         4 0080  16000       "
250  A$(2) = "00002DOS.SMTST.SMOUT.BNPNP                     "
260  A$(3) = "00004DA00800800                                "
270  A$(4) = "00006DWEDS01                                   "
280  A$(5) = "00008DIS.SMTST.FILE.SRTDAT                     "
290  A$(6) = "00010DA0080                                    "
300  A$(7) = "00012FNS00320037                               "
310  A$(8) = "00014FDC00010031                               "
320  A$(9) = "00016FDC00380080                               "
330  A$(10) = "/*                                            "
340  REM  INITIALIZE SORT/MERGE
350  N = 10
360  S = -1
370  CALL ".SMLIB.BASCSM"(SRTINT,A$(1),N,S)
380  IF S = -1 THEN 530
390  IF S > 0 THEN 490
400  REM  UPDATE TIME DELAY COUNT FOR SMSTAT
410  N = 26
420  CALL ".SMLIB.BASCSM"(SETDLY,N)
430  REM  WAIT FOR SORT/MERGE TO COMPLETE
440  S = -1
450  CALL ".SMLIB.BASCSM"(SMSTAT,S)
460  IF S = -1 THEN 550
470  IF S > 0 THEN 510
480  GOTO 570
490  PRINT "ERROR IN SRTINT ",S
500  GOTO 580
510  PRINT "ERROR IN SMSTAT ",S
520  GOTO 580
530  PRINT "INVALID PARAMETER LIST IN SRTINT CALL"
540  GOTO 580
550  PRINT "INVALID PARAMETER LIST IN SMSTAT CALL"
560  GOTO 580
570  PRINT "SORT/MERGE TERMINATED NORMALLY "
580  STOP
590  END
```

Figure 9-13. BASIC Program Calls Sort/Merge to Input and Output Files from Disk

Sort/Merge uses a minimum of two 44-word buffers. Use the following formula to compute the value to specify in the INTERTASK response field.

$$I_{(words)} = \frac{(MRL_{(bytes)} + 96)}{2} \times NS$$

where:

MRL is the maximum record length to be passed, in bytes. You must specify a minimum value of 80 to give Sort/Merge enough space for its control communication; and also, the specified value must be an even number.

NS is the maximum number of programs that may interface with Sort/Merge, using intertask communication, at one time.

For example, if you use 80-character records and only one program using Sort/Merge via intertask communication is to be active at one time, the following formula applies:

$$\frac{(80 + 96) \times 1}{2} = 88 \text{ words}$$

If you use 120-character records and two programs will be active using Sort/Merge via intertask communication, the following applies:

$$\frac{(120 + 96) \times 2}{2} = 216 \text{ words}$$

If sufficient memory is available, you can increase the size of the intertask communication area so that more records can be put in the intertask area at one time. This decreases the amount of time required to send and/or receive records through the intertask communication area.

10

Examples

10.1 GENERAL

This section contains examples of sort and merge processes. The examples include completed specification sheets, examples of both the interactive and batch methods of entering these control specifications, and examples of file and listing formats. This section does not show the complete design or use of Sort/Merge within an application system; instead, it shows a variety of ways to use this processor.

In the examples presented in this section, a simplified inventory system is used in a variety of sort and merge operations. For the examples, assume that a distributing company tracks its parts inventory daily. It uses a 46-byte inventory transaction file that contains information on parts that are sold, received from a vendor, or returned by a customer. The following column format is used in this data file to categorize information:

CODE CV	CUST/ VEND NO.	DATE	PART NO.	PART QTY	VENDOR INVOICE NO.	AMOUNT $
1	2 5	6 11	12 21	22 29	30 35	36 46

(A)135981

This one file contains both customer transactions and vendor transactions. The CODE in column 1 is a C for customer transactions or a V for vendor transactions. The parts maintained in inventory are assigned a positive value; the parts disbursed from inventory are assigned a negative value. The values for PART (columns 22 through 29) and AMOUNT (columns 36 through 46) are signed ASCII. In any one record, the sign value (+ or −) for the entry under PART matches the sign value for the entry under AMOUNT.

Thus, in the customer record, positive quantities indicate the return of the items to inventory, and negative quantities indicate the sale of parts from the inventory. In the vendor record, positive quantities indicate receipt of a shipment of parts, which are added into the inventory; negative quantities indicate the return of parts from the inventory to the vendor (Table 10-1).

Table 10-1. Sign Relationship

SIGN of QTY and AMT	TRANSACTION TYPE	
	CUSTOMER (C)	VENDOR (V)
POSITIVE	RETURN	RECEIPT
NEGATIVE	SALE	RETURN

At any time, a summation by part number produces the number of parts in inventory as well as the total dollar value of those parts. Also, you can obtain a listing that summarizes the transactions for each day, or you can list only information pertaining to parts that were returned.

The daily transactions are entered from a terminal into a file called INV.INPUT. This file is sorted by CODE, CUST/VEND NO., and PART NO. to produce a daily transaction file, INV.DAILY. From this file, you can select and list returns. You can also use this file to update the master file INV.CUMB to produce INV.CUMN. Then you can perform a summary sort to provide an inventory status report.

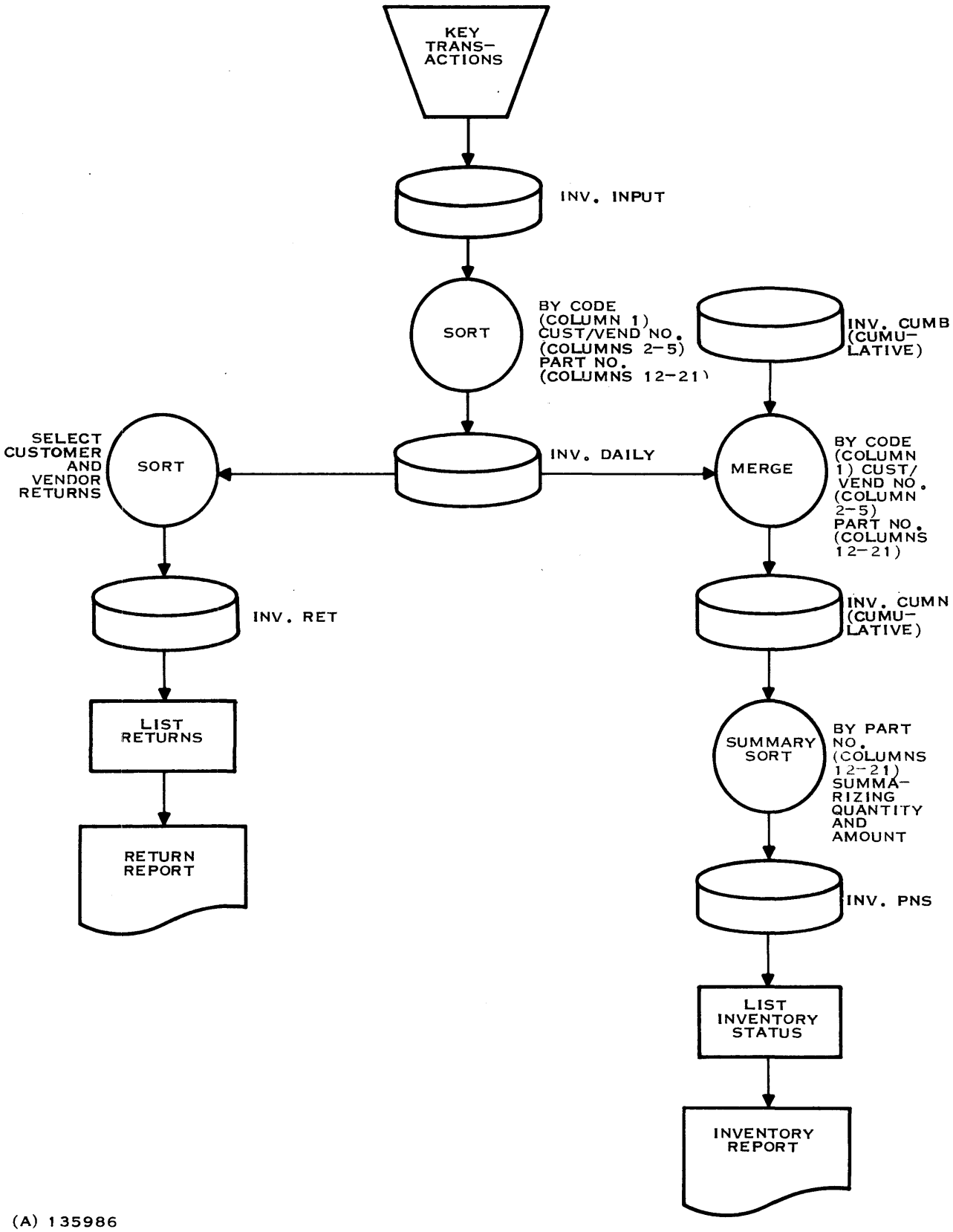
Although the purpose of this section is not to provide details about an inventory system, many of the functions described in the following paragraphs can be used in such a system. Figure 10-1 shows the basic flow of files and processes described in this section.

10.2 EXAMPLE 1: SORT DAILY TRANSACTION FILE

This example shows the preparation of the daily transaction file for updating the master file. A simple Sort is performed by CODE (column 1), CUST/VEND NO. (columns 2 through 5), and PART NO. (columns 12 through 21).

You can use the Sort/Merge Specifications form to plan the sort or merge operation and organize the information into the required order. When planning a sort or merge, you might fill in the the Reformatting Specification first to obtain the information necessary for the Header Specification. However, the explanations in this section are presented from the beginning to the end of the form. When the form is completed, the information is then entered in batch form or in response to the prompting messages presented by the Sort/Merge utility.

The following paragraphs describe each entry required on Sort/Merge Specifications Sheets A and B to specify the sorting of the daily transaction file. Remember that all integers are entered, right-justified, in the appropriate columns. Figure 10-2 shows the completed specifications sheets; refer to them as you read through this example.



(A) 135986

Figure 10-1. Inventory Example

SORT/MERGE SPECIFICATIONS, SHEET A

Page 0 0

Header Specification

Line Number	Spec Type	Merge: MERGE / MERGES / Sort: SORTA / SORTR / SORTRS	Largest Total of Control Fields of Any Record Type (Bytes)	Sequence (A/D)	Reserved	Alternate Coll. Seq. (S)	Print Option (0/1/3)	Drop Ctrl. Flds. (X)	Output Record Length	Variable Lngth. Recs. (V)	Verify Option (V)	Memory Work-Space	Comments
3 4 5	6	7 8 9 10 11 12	13 14 15 16 17	18	19 20 21 22 23 24 25	26	27	28	29 30 31 32	33	34	35 36 37 38 39	(Columns 40 to 72)
0 0 0	H	SORTR	15	A				X	46			16000	

File Description Specifications

OUTPUT FILE

Line Number	Spec Type	File Use	F.T. (S/R/I)	File Pathname	Logical Record Length	Physical Record Length	Number of Logical Records	Comments
3 4 5	6	7	8	9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44				(Columns 45 to 72)
0 1	D	O		INV.DAILY				
0 2	D	A			46	576		

WORK FILE

Line Number	Spec Type	File Use	F.T. (E/N)	Volume Name	Comments
3 4 5	6	7	8	9 10 11 12 13 14 15 16	(Columns 17 to 80)
0 3	D	W	E	DS01	

INPUT FILE

Line Number	Spec Type	File Use	F.T. (S/R/I)	File Pathname	L.R. Length	No. of Records	Comments
3 4 5	6	7	8	9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44			(Columns 45 to 72)
0 4	D	I		SINV.INPUT			
0 5	D	A					
0 6	D	I					
0 7	D	A					
0 8	D	I					
0 9	D	A					
1 0	D	I					
1 1	D	A					
1 2	D	I					
1 3	D	A					

Figure 10-2. Example 1, Specifications Sheets and Listing Output (Sheet 1 of 3)

SORT/MERGE SPECIFICATION, SHEET B

Page 02

Record Selection Specification

Line Number	TYPE (I/O)	Continuation (A/O)	(C/Z/D/S/F/U)	Operand 1 Location From	Operand 1 Location To	Rel. EQ NE LT GT LE GE	FC	Operand 2 (Field or Constant) Location From	Operand 2 Location To	Operand 2 Constant	Comments (Columns 40 to 72)
14											
15											
16											
17											
18											
19											
20											
21											
22											
23											
24											
25											
26											

Reformatting Specification

Line Number	Type	TYPE (N/O/F/D/S)	(C/Z/D/S/I/F/V/F/U)	Location From	Location To	Forced Record Character	Forced Substitute Character	Forced Continuation	Overflow	Reserved	Comments (Columns 40 to 72)
270	F	NC			1						CODE
280	F	NC		2	5					CONTROL FIELD 15 BYTES	CUST/VEND NO.
290	F	NC		12	21						PART NO.
300	F	DC		1	46					DATA FIELD 46 BYTES	
310	F										
320	F										
330	F										
340	F										
350	F										
360	F										
370	F										
380	F										
390	F										
400	F										

(A) 135992

Figure 10-2. Example 1, Specifications Sheets and Listing Output (Sheet 2 of 3)

```
      PAGE    1              SORT/MERGE    *A      00:15:08

           0    1    1    2    2    3    3    4    4    5    5    6
      ....5....0....5....0....5....0....5....0....5....0....5....0.
      00000HSORTR     15A          0X  46  16000
      00010DDSINV. DAILY
      00020DA  46 576
      00030DISINV. INPUT
      00040DA
      00070FNC         1
      00080FNC    2    5
      00090FNC   12   21
      00100FDC    1   46
INFO    4:    END OF GENERATION PHASE
INFO    3:    NUMBER OF SPECIFICATIONS=      9
INFO    0:    NO ERRORS
INFO   23:    BYTES OF WORKING STORAGE AVAILABLE= 16000
INFO   24:    BYTES REQUIRED FOR SPECIFICATIONS =   352
INFO   14:    INPUT FILE RECORD SIZE=     80
INFO   15:    INPUT FILE BLOCK SIZE =    288
INFO   16:    OUTPUT FILE RECORD SIZE =     46
INFO   17:    OUTPUT FILE BLOCK SIZE=    576
INFO    5:    SORT EXECUTION HAS BEGUN
INFO   18:    WORK FILE RECORD SIZE =     64
INFO    9:    NUMBER OF INPUT FILE RECORDS READ =     20
INFO   10:    NUMBER OF INPUT FILE RECORDS SELECTED =     20
INFO   30:    NUMBER OF WORK FILE TRACKS USED =      1
INFO   29:    NUMBER OF PASSES REMAINING=      0
INFO   25:    SORT FINAL PASS HAS BEGUN
INFO   26:    SORT FINAL PASS COMPLETED
INFO   13:    NUMBER OF OUTPUT RECORDS WRITTEN=    20
```

Figure 10-2. Example 1, Specifications Sheets and Listing Output (Sheet 3 of 3)

10.2.1 Header Specification

The following is a column-by-column description of the information required in the Header Specification to perform this sort operation:

Columns 1-6
The entries for these columns are preprinted on the form and are automatically specified.

Type of Sort or Merge, Columns 7-12
Specify SORTR, a tag-along sort. The data will tag-along with the control fields.

Total Control Field Length, Columns 13-17
Specify the integer 15. This is the combined column total of the three fields from the input file that are to be the control fields for the sort (that is, CODE, column 1; CUST/VEND NO., columns 2 through 5; and PART NO., columns 12 through 21).

Sequence, Column 18
Specify an A to designate ascending sequence.

Columns 19 through 15
These columns are reserved.

Alternate Collating Sequence, Column 26
Leave this column blank; no alternate collating sequence is specified.

Print Option, Column 27
Leave this column blank; the default value specifies that the listing output is to contain control specifications and information, error, and warning messages.

Drop Control Fields, Column 28
Enter an X to specify that control fields are to be dropped from the output records after the records are sorted.

Output Record Length, Columns 29-32
Specify 46 bytes, the longest record to be output. In this example, all data fields from the input record are to be included, so this is the same length as the longest input record. (The inventory transaction file provides 46 bytes, or 46 character positions, for each record.)

Variable Length Records, Column 33
Leave this column blank; the default specifies fixed-length records.

Verify Option, Column 34
Leave this column blank; the default specifies that data sent to the work file need not be verified during this sort operation.

Memory Size, Columan 35-39
Specify 16,000 bytes. In most cases, the larger the memory size, the faster the sort will process.

10.2.2 Output File Specification

The following entries are made on Specifications Sheet A (Figure 10-2) to describe the output file:

Columns 1-7
: Entries for these columns are preprinted on the form and are automatically specified. They designate the page and line numbers and identify the information as data relating to the output file or to the attributes of the file.

File Type, Column 8
: Specify S since the file is a sequential file. Since the default for this column also designates a sequential file, you can leave this columns blank.

File Pathname, Line 1, Columns 9-44
: Specify the pathname of the output file INV.DAILY. (The pathname may contain a maximum of 36 characters.)

Logical Record Length, Line 2, Columns 8-11
: Specify that the output file records are to be 46 bytes (characters) long. This designation becomes commentary if the file was previously created with a different LRL specified.

Physical Record Length, Line 2, Columns 12-15
: Specify the physical output file records to be 576 characters. This means that 12 logical records will be placed in each physical record.

Number of Logical Records, Line 2, Columns 16-23
: Leave this column blank; the default specifies that the file is to be created expandable.

10.2.3 Work File Specification

The following entries are made on Specifications Sheet A (Figure 10-2) to describe the work file:

Columns 1-7
: Entries for these columns are preprinted on the form and are automatically specified. They designate the page and line numbers and identify the information as data relating to the work file.

File Type, Column 8
: Specify E to designate that the file is to be expandable. Since the default value specifies expandable, you can leave this column blank.

Volume Name, Columns 9-16
: For this example, specify the system disk, DS01. Any volume name of a disk that has been installed is an acceptable response.

  The work file will be on a separate volume from the input and output files. This is the most efficient arrangement for a sort operation.

10.2.4 Input File Specification

The following entries are made on Specifications Sheet A (Figure 10-2) to describe the input file:

Columns 1-7
: Entries for these columns are preprinted on the form and are automatically specified. They designate the page and line numbers and identify the information as data relating to the input file or the attributes of that file.

File Type, Column B
: Specify S to designate that the file is sequential. Since the default value also specifies sequential, you can leave this column blank.

File Pathname, Line 4, Columns 9-44
: Enter the name of the input file, INV.INPUT. (The pathname may be the name of any file or device that contains the input file.)

Logical Record Length, Line 5, Columns 8-11
: Leave these columns blank; the values specified at the creation of the input file will be used.

Number of Logical Records, Line 2, Columns 16-23
: Leave these columns blank; the default allocation will be used.

10.2.5 Record Selection Specification

All records are to be included in this sort operation. Since the default value specifies include all, this part of the form requires no entries.

10.2.6 Reformatting Specification

The order in which the control fields are given on the Reformatting Specification determines each record's sequence in the output file. Likewise, the order in which data fields are given on the Reformatting Specification determines the order in which the fields appear in each output record.

The daily input file records are to be sorted by CODE first. All records having the same CODE are then sorted by CUST/VEND NO. All records having the same CODE and CUST/VEND NO. are then sorted by PART NO. Line numbers 270, 280, and 290 of Specifications Sheet B (Figure 10-2) specify the control fields for each of these input file categories. The total length of the combined control fields is 15 bytes.

The order of the categories in each record of the input file is to remain the same in the output file records. Since the control fields are to be dropped (as specified in the Header Specification), the output record length of the data field is the same as the input record length, 46 bytes. The following entries are made on the Specifications Sheet:

Type, Column 7
: Specify N for each control field to designate that it is to be sorted in normal sequence. Since ascending sequence was specified in the Header Specification, normal means ascending sequence.

  Specify D for the data field.

Character Portion, Column 8
: Specify a C for the control fields and the data field to indicate a full ASCII character sort.

Location, Column 9-16

In columns 9 through 12, specify the beginning column location for each control field. In columns 13 through 16, specify the ending column location for each control field. Note that when only one column is specified, the integer is placed in the To column.

On line number 300, specify the beginning and ending column locations of the data field. In this case, the entire input record is designated as the data field.

Figure 10-2 shows the listing of the control file and messages that Sort/Merge produces when the information from the specifications form is entered either interactively or in batch form.

Figure 10-3 shows the format of the work file and output file records and their relationship. Note that the control fields and data fields are placed in the work file and written to the output file in the order specified in the Reformatting Specification. Sort/Merge adds the pad field and length field; these fields are not available to you. In this example, the control fields were dropped; consequently, the first 18 characters of the work file record do not appear in the output file.

10.3 EXAMPLE 2: SELECTION OF PARTICULAR RECORDS

In example 1, the file INV.DAILY was output. Example 2 uses INV.DAILY as the input file to create the output file INV.RET. This output file will contain only items that have been returned either by a customer or to a vendor. As shown in Table 10-1, a positive quantity indicates a customer return in the customer records; a negative quantity indicates a return in the vendor records. Thus, multiple selection is required for this example. The selection of records pertaining to parts returned by a customer is specified on sheet B of the form; an additional sheet B is then used to select records pertaining to parts returned to a vendor.

The purpose of this example is to show how you can select various records to be used in the sort and how you can reformat data fields within each record during the sort operation. Two different types of records are in this file and each contains different information. In this example, vendor records (V) are to be listed separate from customer records (C), and the sequence of data fields in vendor records is to be reformatted different from customer records. Also, blanks are to be inserted between the fields in the output file and a dollar sign is to be inserted before the AMOUNT field.

Figure 10-4 shows the specification sheets for example 2. The following paragraphs detail these specifications.

10.3.1 Header Specification

The Header Specification is similar to that used for example 1. The only difference is that the Total Control Field Length (columns 13 through 17) is 5 for this example since only the CODE field in column 1 and the CUST/VEND NO. field in columns 2 through 5 are used as control fields.

10.3.2 File Descriptions

The File Description Specifications are the same as those described in example 1, except the pathnames are changed to INV.RET for the output file and INV.DAILY for the input file.

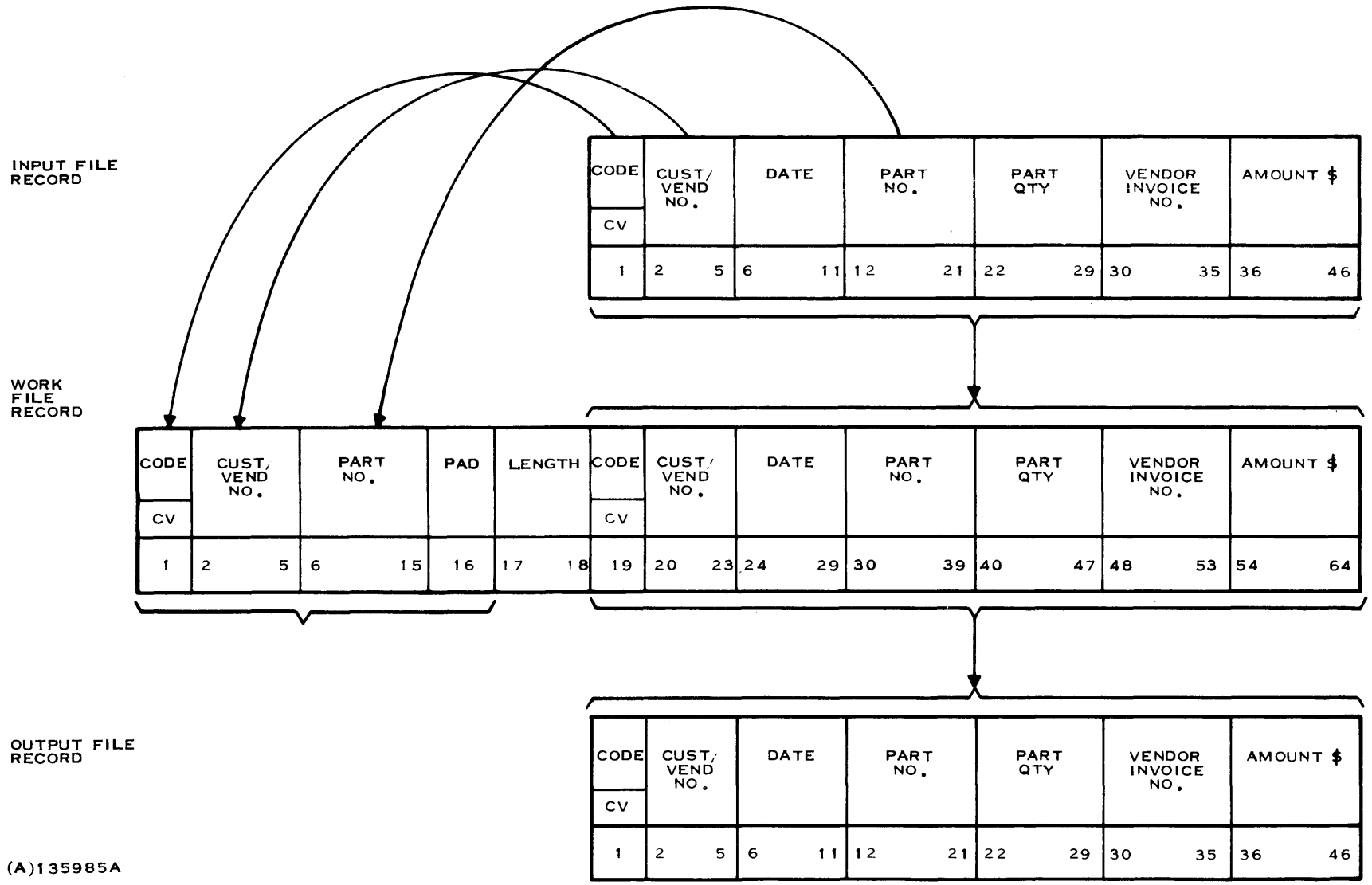


Figure 10-3. Input, Work, and Output File Records

SORT/MERGE SPECIFICATIONS, SHEET A

Page 0 0

Header Specification

Line Number	Spec Type	Merge: MERGE / MERGES / Sort SORTA / SORTR / SORTRS	Largest Total of Control Fields of Any Record Type (Bytes)	Sequence (A/D)	Reserved	Alternate Coll. Seq. (S)	Print Option (0/1/3)	Drop Ctrl. Flds. (X)	Output Record Length	Variable Lngth. Recs. (V)	Verify Option (V)	Memory Work Space	Comments
3 4 5	6	7 8 9 10 11 12	13 14 15 16 17	18	19 20 21 22 23 24 25	26	27	28	29 30 31 32	33	34	35 36 37 38 39	(Columns 40 to 72)
0 0 0	H	SORTR		SA				X	A6			16000	

File Description Specifications

OUTPUT FILE

Line Number	Spec Type	File Use	F.T. (S/R/I)	File Pathname	Logical Record Length	Physical Record Length	Number of Logical Records	Comments
3 4 5	6	7	8	9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44				(Columns 45 to 72)
0 1	O	O		INV.RET				
0 2	O	A			46	576		

WORK FILE

Line Number	Spec Type	File Use	F.T. (E/N)	Volume Name	Comments
3 4 5	6	7	8	9 10 11 12 13 14 15 16	(Columns 17 to 80)
0 1	W		E	DS01	

INPUT FILE

Line Number	Spec Type	File Use	F.T. (S/R/I)	File Pathname	L.R. Length	No. of Records	Comments
3 4 5	6	7	8	9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44			(Columns 45 to 72)
0 1	I			SINV.DAILY			

Figure 10-4. Example 2, Specification Sheets (Sheet 1 of 3)

SORT/MERGE SPECIFICATION, SHEET B

Page 1 2

Record Selection Specification

Line Number	TYPE (I/O)	Continuation (A/O)	(C/Z/D/S/P/U)	Operand 1 Location From	Operand 1 Location To	RH LO NE LT GT LE GE	C AL	Operand 2 (Field or Constant) Location From	Operand 2 Location To	Constant	Comments (Columns 40 to 72)
14	O	I	C			IEQ	CV				VENDOR
15	O	I	AS	22	29	LT	C	00000000		+	RETURN
16											
17											
18											
19											
20											
21											
22											
23											
24											
25											
26											

Reformatting Specification

Line Number	Type	TYPE (N/O/F/D/S)	(C/Z/D/S/I/F/V/P/U)	Location From	Location To	Forced Record Character	Forced Substitute Character	Forced Continuation	Overflow	Reserved	Comments (Columns 40 to 72)
27	O	F	NC		1					CONTROL FIELD	BY CODE
28	O	F	NC	2	5					5 BYTES	BY VEND NO.
29	O	F	DC		1						CODE
30	O	F	DV				b				BLANK
31	O	F	DC	2	5						VEND NO.
32	O	F	DV				b				BLANK
33	O	F	DC	12	21						PART NO.
34	O	F	DV				b			DATA FIELD	BLANK
35	O	F	DC	22	29					46 BYTES	PART QTY.
36	O	F	DV				b				BLANK
37	O	F	DC	30	35						INVOICE NO.
38	O	F	DV				b				BLANK
39	O	F	DV				$				$
40	O	F	DC	36	46						AMOUNT

(A)135991A

Figure 10-4. Example 2, Specification Sheets (Sheet 2 of 3)

SORT/MERGE SPECIFICATION, SHEET B

Page 1 2

Record Selection Specification

Line Number	TYPE (I/O)	Continuation (A/O)	(C/Z/D/S/P/U)	Operand 1 Location From	Operand 1 Location To	Rel EQ NE LT GT LE GE	F.C	Operand 2 (Field or Constant) Location From	Operand 2 Location To	Operand 2 Constant	Comments (Columns 40 to 72)
14	O	I	C		16	EQ	C	C			CUSTOMER
15	O	I	A	S	22	29	GT	C	00000000	+	RETURN
16											
17											
18											
19											
20											
21											
22											
23											
24											
25											
26											

Reformatting Specification

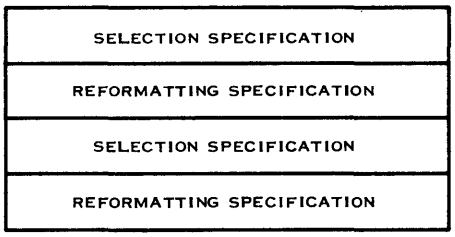
Line Number	Type	TYPE (N/O/F/D/S)	(C/Z/D/S/H/E/V/P/U)	Location From	Location To	Forced Record Character	Forced Substitute Character	Forced Continuation	Overflow	Reserved	Comments (Columns 40 to 72)
27	O	F	NC		1					CONTROL FIELD	BY CODE
28	O	F	NC	2	5					5 BYTES	BY CUST NO.
29	O	F	DC		1						CODE
30		F	DV				b				BLANK
31		F	DC	2	5						CUST NO.
32		F	DV				b				BLANK
33		F	DC	12	21						PART NO.
34		F	DV				b			DATA FIELD	BLANK
35		F	DC	22	29					46 BYTES	PART QTY
36		F	DV				b				BLANK
37		F	DC	6	11						DATE
38		F	DV				b				BLANK
39		F	DV				$				$
40		F	DC	36	46						AMOUNT

(A) 135994A

Figure 10-4. Example 2, Specification Sheets (Sheet 3 of 3)

10.3.3 Record Selection Specification

Vendor records (V) are to be listed separately from customer records (C) and are to be reformatted differently. To allow different records to be reformatted in different ways, the reformatting specification may follow each record selection specification, as shown in the following diagram:



(A)135979

Therefore, one record selection and reformatting form should be completed for vendor records and another for customer records. First, select those vendor records that have a negative value (a parts return) in the PART QTY category and reformat the record fields as follows:

Type, Column 6

Enter an I on both lines 140 and 150 to include only records where column 1 of the inventory file contains a V **and** where columns 22 through 29 of the inventory file are less than the signed ASCII value of zero.

Continuation, Column 7

Specify an A on line 150 to denote ANDing of the two selection criteria.

Character Portion, Column 8

Specify a C for ASCII character comparison on line 140 (that is, column 1 of the input file equal to V). Specify an S for signed ASCII comparison on line 150 (that is, columns 22 through 29 of the inventory file show a value less than 0).

Operand 1, Columns 9-16

Specify the columns to be used in the selection process. For line 140, operand 1, only column 1 of the input file (the customer/vendor CODE category) is selected. Note that single column entries are made only in the To portion of the field. For line 150, operand 1, columns 22 through 29 of the input file (the PART QTY category) is selected.

Relation, Columns 17-18

Specify the relation of operand 1 to operand 2 on line 140 as EQ (that is, CODE category equal to vendor). Specify the relation of operand 1 to operand 2 on line 150 as LT (that is, PART QTY category value less than 0, indicating a parts return).

Field or Constant, Column 29

Specify whether another field in the input record is to be compared with operand 1 or if a constant is to be compared. This example compares constants, so enter a C in column 29 for both line 140 and 150.

Operand 2, Columns 19-39
Specify the value of the constant. In this case, enter a V on line 140 and an ASCII zero on line 150.

The information specified on the Record Selection Specification designates the following: include all records in the sort that have column 1 equal to an ASCII V and that have columns 22 through 29 less than a signed ASCII 0.

The preparation of an additional Sheet B to define record selection for customer returns is similar to the vendor return specification, except column 1 of the inventory file is specified as equal to C (customer) and columns 22 through 29 of the inventory file are specified as greater than a signed ASCII 0, since parts returned to inventory have a positive value. See Figure 10-34.

10.3.4 Reformatting Specification

For vendor records, the original seven categories in the inventory file are reduced to six categories by eliminating the DATE field when the record is reformatted. For customer records, no entries are included under the VENDOR INVOICE NO. category, so the DATE field can be moved to this field to reduce the number of categories in customer records to six also. Figure 10-5 shows the format of the original input file and the intended format of the output file.

Note that blanks are to be forced between fields in the records and that a dollar sign is forced before the AMOUNT field to allow printing with a simple list program, as explained in example 3.

The following entries are made on the vendor version of the Reformatting Specification:

Type, Column 7
Specify N for each control field to designate that it is to be sorted in normal sequence first by the V designation (column 1 of the input file) and then by VENDOR NO. (columns 2 through 5 of the input file). Since ascending sequence was specified in the header specification, normal means ascending sequence.

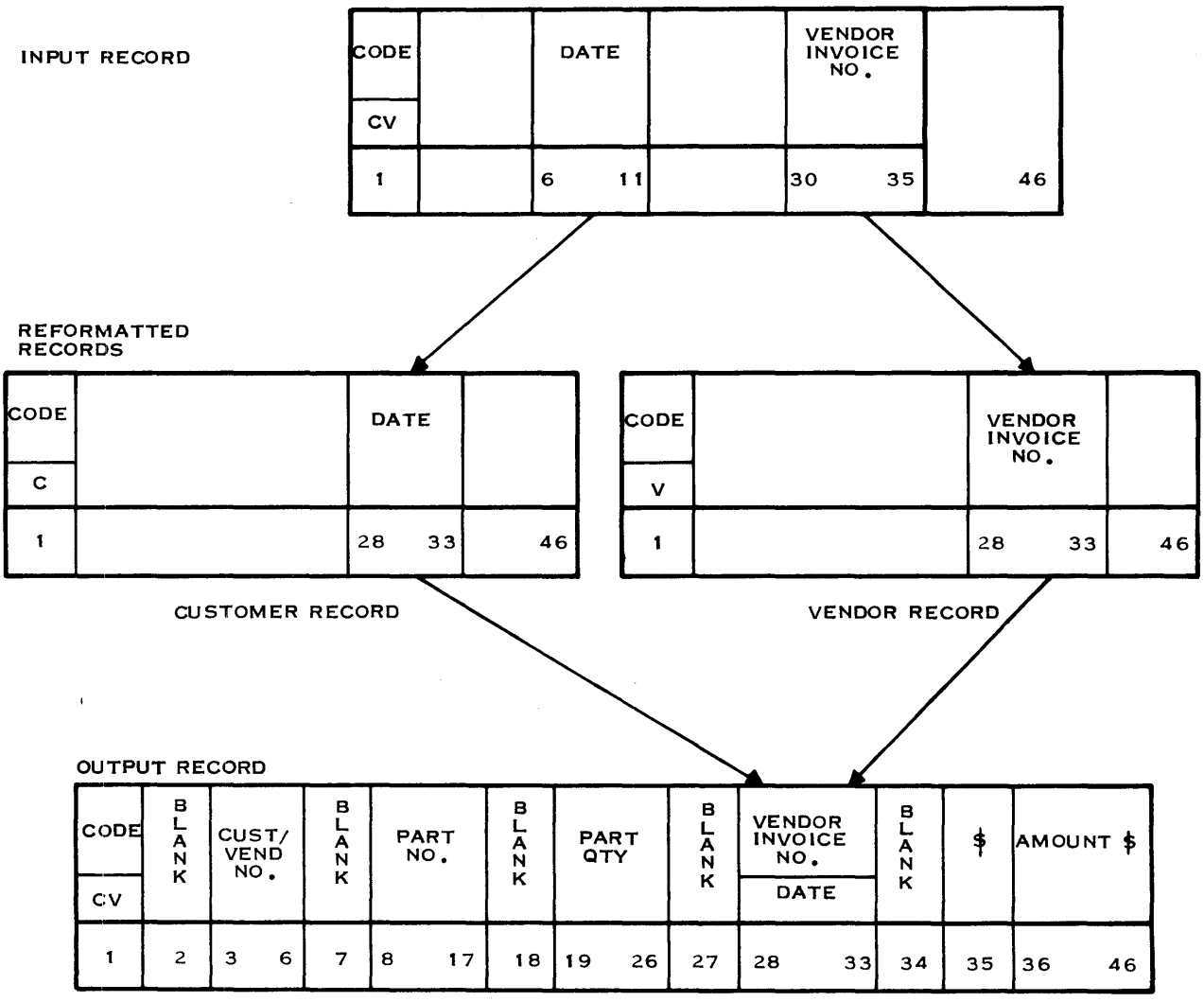
Specify D for the data fields, the blanks, and the dollar sign.

Character Portion, Column 8
Specify a C for the control fields and the data fields that describe input file column positions to indicate a full ASCII character sort. Specify a V on those lines specifying a blank space or dollar sign to indicate a forced character.

Location, Column 9-16
In columns 9 through 12, specify the beginning column location for each control field. In columns 13 through 16, specify the ending column location for each control field. Note that when only one column is specified, the integer is placed in the To column.

Specify the beginning and ending column locations for each data field in the input file that is to be included in the output file. The order in which the data fields are given determines the order of the data fields in the output file. Note that the DATE field (columns 6 through 11 of the input file) is omitted in the Reformatting Specification.

The b in column 18 represents a blank; however, the character b is not typed when you actually enter the control specifications.



(A)135982A

Figure 10-5. Selection for Reformatting and Resulting Reformatted Record

The customer version of the Reformatting Specification is completed in a similar manner, except that the DATE field (columns 6 through 11 of the input file) is moved out of sequence and replaces the VENDOR INVOICE NO. field (columns 30 through 35 of the input file).

Figure 10-6 shows a printout of the control file and a segment of the output file.

```
PAGE    1            SORT/MERGE    *A       00:16:42

     0    1    1    2    2    3    3    4    4    5    5    6    6
....5....0....5....0....5....0....5....0....5....0....5....0....5.
00000HSORTR      5A          0X  46  16000
00010DOSINV.RET
00020DA  46 576
00030DISINV.DAILY
00040DA
00050I C        1EQCV
00060IAS  22  29LTC0000000+
00070FNC        1
00080FNC   2    5
00090FDC        1
00100FDV
00110FDC   2    5
00120FDV
00130FDC  12   21
00140FDV
00150FDC  22   29
00160FDV
00170FDC  30   35
00180FDV
00190FDV          $
00200FDC  36   46
00210I C        1EQCC
00220IAS  22  29GTC0000000+
00230FNC        1
00240FNC   2    5
00250FDC        1
00260FDV
00270FDC   2    5
00280FDV
00290FDC  12   21
00300FDV
00310FDC  22   29
00320FDV
00330FDC   6   11
00340FDV
00350FDV          $
00360FDC  36   46
```

Figure 10-6. Printout and Output File (Sheet 1 of 2)

```
INFO   4:   END OF GENERATION PHASE
INFO   3:   NUMBER OF SPECIFICATIONS=     37
INFO   0:   NO ERRORS
INFO  23:   BYTES OF WORKING STORAGE AVAILABLE= 16000
INFO  24:   BYTES REQUIRED FOR SPECIFICATIONS =   550
INFO  14:   INPUT FILE RECORD SIZE=     46
INFO  15:   INPUT FILE BLOCK SIZE =    576
INFO  16:   OUTPUT FILE RECORD SIZE =     46
INFO  17:   OUTPUT FILE BLOCK SIZE=    576
INFO   5:   SORT EXECUTION HAS BEGUN
INFO  18:   WORK FILE RECORD SIZE =      54
INFO   9:   NUMBER OF INPUT FILE RECORDS READ =     20
INFO  10:   NUMBER OF INPUT FILE RECORDS SELECTED =     14
INFO  30:   NUMBER OF WORK FILE TRACKS USED =      1
INFO  29:   NUMBER OF PASSES REMAINING=      0
INFO  25:   SORT FINAL PASS HAS BEGUN
INFO  26:   SORT FINAL PASS COMPLETED
INFO  13:   NUMBER OF OUTPUT RECORDS WRITTEN=     14
C    45     111111 0000851+ 022677 $0000004444+
C   123   12365498 0001895+ 020777 $0000014269+
C   124       2222 0000101+ 022077 $0000005952+
C   872   87651321 0004444+ 021077 $0000000215+
C  1082    4568521 0000011  022277 $0000000752+
C  1082   45000123 0010004+ 022877 $0000003645+
C  1082   87654321 0001122+ 022877 $0000000217+
C  3289     111111 0008592+ 021277 $0000014521+
C  5000        128 0000045+ 021477 $0000006396+
C  5000      45575 0008889+ 020277 $0000002251+
V   855     265111 0000010-     10 $0000000547-
V   855   12365498 0000001-   5021 $0000000335-
V   855   45685215 0000011-   1002 $0000011172-
V  8555   87654321 0002222- 178582 $0000198845-
```

Figure 10-6. Printout and Output File (Sheet 2 of 2)

10.4 EXAMPLE 3: MERGE DAILY FILE INTO CUMULATIVE FILE

The Specification Sheets shown in Figure 10-7 show the merge of the daily transaction file into the cumulative file. This example is simple in that the selection and reformatting are the same for each input file. If desired, you can select records and reformat them from each file in a different manner, as described in the previous example.

10.4.1 Header Specifications

The Header Specifications are the same as in the previous examples, except that Merge is specified in columns 7 through 11 of the form.

SORT/MERGE SPECIFICATIONS, SHEET A

Page 0 0

Header Specification

Line Number	Spec Type	Merge: MERGE, MERGES / Sort: SORTA, SORTR, SORTRS	Largest Total of Control Fields of Any Record Type (Bytes)	Sequence (A,D)	Reserved	Alternate Coll. Seq. (S)	Print Option (0/1/3)	Drop Ctrl. Flds. (X)	Output Record Length	Variable Length. Recs. (V)	Verify Option (V)	Memory Work Space	Comments
3 4 5	6	7 8 9 10 11 12	13 14 15 16 17	18	19 20 21 22 23 24 25	26	27	28	29 30 31 32	33	34	35 36 37 38 39	(Columns 40 to 72)
0 0 0	H	MERGE	15A					X	46			16000	

File Description Specifications

OUTPUT FILE

Line Number	Spec Type	File Use	F.T. (S/R/I)	File Pathname / Logical Record Length	Physical Record Length	Number of Logical Records	Comments
3 4 5	6	7	8	9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44			(Columns 45 to 72)
0 1	D	O		INV.CUMN			
0 2	D	A		46	576		

WORK FILE

Line Number	Spec Type	File Use	F.T. (E/N)	Volume Name	Comments
3 4 5	6	7	8	9 10 11 12 13 14 15 16	(Columns 17 to 80)
0 3	D	W	E	EDS01	

INPUT FILE

Line Number	Spec Type	File Use	F.T. (S/R/I)	File Pathname / L.R. Length	No. of Records	Comments
3 4 5	6	7	8	9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44		(Columns 45 to 72)
0 4	D	I		SINV.CUMB		
0 5	D	A				
0 6	D	I		SINV.DAILY		
0 7	D	A				
0 8	D	I				
0 9	D	A				
1 0	D	I				
1 1	D	A				
1 2	D	I				
1 3	D	A				

Figure 10-7. Example 3, Specifications Sheets (Sheet 1 of 2)

SORT/MERGE SPECIFICATION, SHEET B

Page 1 2

Record Selection Specification

Line Number	TYPE (I/O)	Continuation (A/O) (C/Z/D/S/P/U)	Operand 1 Location From	Operand 1 Location To	Rel. EQ NE LT GT LE GE	F C	Operand 2 (Field or Constant) Location From	Operand 2 Location To	Operand 2 Constant	Comments (Columns 40 to 72)
3 4 5	6	7	8	9 10 11 12 13 14 15 16	17 18	19	20 21 22 23	24 25 26 27	28 29 30 31 32 33 34 35 36 37 38 39	
1 4										
1 5										
1 6										
1 7										
1 8										
1 9										
2 0										
2 1										
2 2										
2 3										
2 4										
2 5										
2 6										

Reformatting Specification

Line Number	Type	TYPE (N/O/F/D/S) (C/Z/D/S/I/V/F/V/P/U)	Location From	Location To	Forced Record Character	Forced Substitute Character	Forced Continuation	Overflow	Reserved	Comments (Columns 40 to 72)
3 4 5	6	7 8	9 10 11 12	13 14 15 16	17	18	19	20 21 22	23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39	
2 7	0 F	NC		1						BY C/V CODE
2 8	0 F	NC	2	5					CONTROL FIELDS 15 BYTES	BY C/V NO.
2 9	0 F	NC	12	21						BY PART NO.
3 0	0 F	DC	1	46					DATA FIELD 46 BYTES	
3 1	F									
3 2	F									
3 3	F									
3 4	F									
3 5	F									
3 6	F									
3 7	F									
3 8	F									
3 9	F									
4 0	F									

(A) 135990A

Figure 10-7. Example 3, Specifications Sheets (Sheet 2 of 2)

10.4.2 File Description Specifications

Specify INV.CUMN as the output file. The file attributes on line 2 of the form remain the same.

Specify DS01 as the expandable work file.

Specify INV.CUMB and INV.DAILY as the input files for the merge operation. Since these are both sequential files, place an S in column 8 for each of these files. Note that from two to five input files are specified for a merge. The file specified first will have its records output first when an equal sequence condition is sensed. Both files must be in the same sequence (ascending or descending) before the merge begins.

10.4.3 Record Selection Specification

Since all records from each input file are to be included in the merge, no record selection specification is required. The default value includes all records.

10.4.4 Reformatting Specification

Both of the input files for this merge have the same record format. The file INV.CUMB contains the cumulative inventory information; the file INV.DAILY was sorted in example 1 and contains new records to be added to the cumulative inventory file.

Three control fields are used; the N in column 7 of the form designates that these fields are to be merged in normal (ascending) order. The file records are to be merged by CODE first (column 1 of the input files). All records having the same CODE are then merged by CUST/VEND NO. (columns 2 through 5 of the input files). All records having the same CODE and CUST/VEND NO. are then merged by PART NO. (columns 12 through 21 of the input files).

All data fields in the input records are to be in the output records. Thus, the data field is specified as columns 1 through 46 of the input file, and the D in column 7 identifies line 300 as a data field specification.

10.4.5 Entering the Specifications Interactively and in Batch Form

After you have organized the necessary information on the worksheets for the merge operation, prepare the control file. This can be done interactively by using the Execute Sort/Merge (XSMF) command and entering the information in response to the field prompts. The control file could also be prepared by batch a stream.

Figure 10-8 shows the interactive entry of the merge specifications used in this example. Since you can repeat the input file specification prompts, you can specify several files.

Figure 10-9 shows a batch control stream for the same merge.

```
[] XSMF
EXECUTE SORT/MERGE
 NEW CONTROL FILE?:    YES
 CONTROL FILE NAME:    .CONTM
**SORT/MERGE HEADER SPECIFICATION**
 SORT MERGE TYPE: SORTR  MERGE
 TOTAL CONTROL LENGTH:    15
 SORT MERGE SEQUENCE: A
 ALTERNATE SEQUENCE:
 PRINT OPTION: 1  0
 DROP CONTROL FIELDS: X
 OUTPUT RECORD LENGTH:    46
 VARIABLE LENGTH RECORDS:
 VERIFY OPTION:
 MEMORY SIZE:   16000
**SORT/MERGE OUTPUT FILE SPECIFICATION**
 FILE PATHNAME:    INV.CUMN
 FILE TYPE: S
 LOGICAL RECORD LENGTH:    46
 PHYSICAL RECORD LENGTH:    576
 NUMBER OF LOGICAL RECORDS:
**SORT/MERGE INPUT FILE SPECIFICATION**
 FILE PATHNAME:    INV.CUMB
 FILE TYPE: S
 LOGICAL RECORD LENGTH:
 NUMBER OF LOGICAL RECORDS:
ENTER NEXT SPECIFICATION - INPUT,SELECTION,REFORMATTING
 NEXT:    INPUT
**SORT/MERGE INPUT FILE SPECIFICATION**
 FILE PATHNAME:    INV.DAILY
 FILE TYPE: S
 LOGICAL RECORD LENGTH:
 NUMBER OF LOGICAL RECORDS:
ENTER NEXT SPECIFICATION - INPUT,SELECTION,REFORMATTING
 NEXT:    R
**SORT/MERGE REFORMATTING SPECIFICATION**
 FIELD TYPE:    N
 CHARACTER PORTION: C
 BEGINNING LOCATION:
 ENDING LOCATION:    1
ENTER NEXT SPECIFICATION - INPUT,SELECTION,REFORMATTING
 NEXT:    R
**SORT/MERGE REFORMATTING SPECIFICATION**
 FIELD TYPE:    N
 CHARACTER PORTION: C
 BEGINNING LOCATION:    2
 ENDING LOCATION:    5
ENTER NEXT SPECIFICATION - INPUT,SELECTION,REFORMATTING
 NEXT:    R
**SORT/MERGE REFORMATTING SPECIFICATION**
 FIELD TYPE:    N
```

Figure 10-8. Interactive Control Stream Example (Sheet 1 of 2)

```
 CHARACTER PORTION: C
 BEGINNING LOCATION:    12
 ENDING LOCATION:    21
ENTER NEXT SPECIFICATION - INPUT, SELECTION, REFORMATTING
 NEXT:    R
**SORT/MERGE REFORMATTING SPECIFICATION**
 FIELD TYPE:    D
 CHARACTER PORTION: C
 BEGINNING LOCATION:    1
 ENDING LOCATION:    46
ENTER NEXT SPECIFICATION - INPUT, SELECTION, REFORMATTING
 NEXT:
 RUN SORT MERGE?: YES  NO
[] Q
```

Figure 10-8. Interactive Control Stream Example (Sheet 2 of 2)

```
BATCH
SM$SMC CFN = DS01.CONTM
SM$HD SMT = MERGE,TCL = 15,PO = 0,ORL = 46,MS = 16000
SM$OUT FP = INV.CUMN,FT = S,LRL = 46,PRL = 576
SM$IN FP = INV.CUMB,FT = S
SM$IN FP = INV.DAILY,FT = S
SM$REF FT = N,CP = C,EL = 1
SM$REF FT = N,CP = C,BL = 2,EL = 5
SM$REF FT = N,CP = C,BL = 12,EL = 21
SM$REF FT = D,CP = C,BL = 1,EL = 46
SM$CLS
XBSM CFN = DS01.CONTM,LDN = LP01
EBATCH
```

Figure 10-9. Batch Control Stream Example

10.5 EXAMPLE 4: SUMMARY BY PART NUMBER

This example shows a summary sort. The output will be one record for each part number, showing the total quantity on hand and the dollar value of that quantity. Blanks are forced between fields to allow the output to be printed with a simple listing program. Overflow fields are provided for each summary field. Figure 10-10 shows the format of the input and summary output record. Each part number will have only one record output.

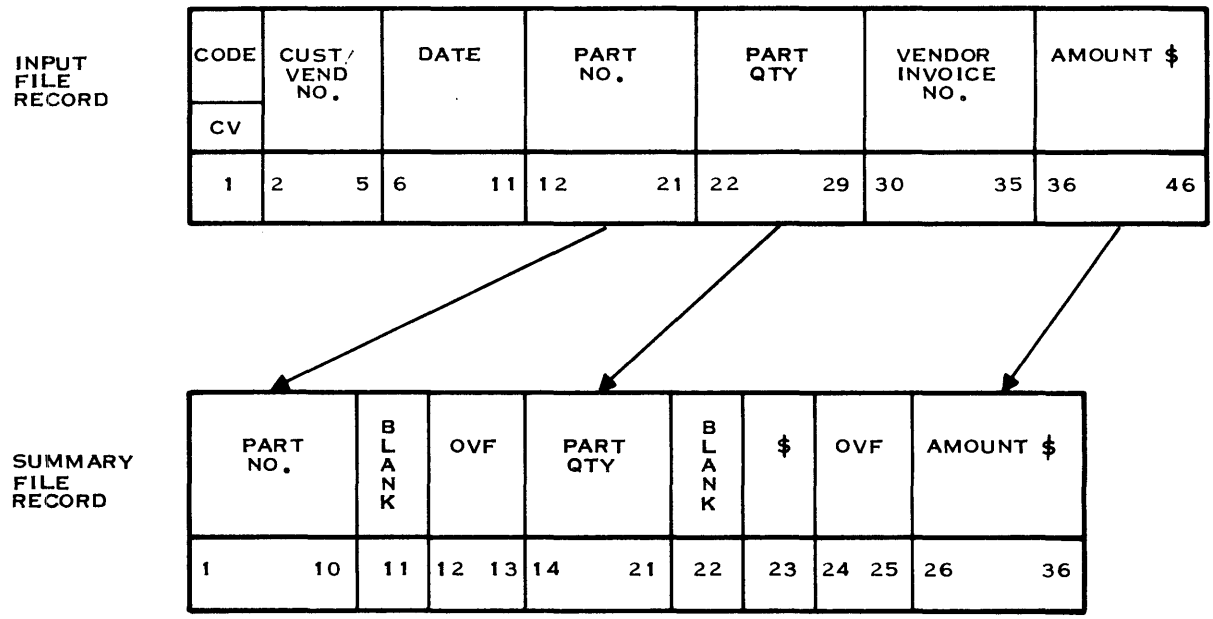


Figure 10-10. Summary Record Format

10.5.1 Header Specification

Two parameters differ from previous examples on this Header Specification, as shown in Figure 10-11 and explained by the following:

- A SORTRS is specified in columns 7 through 12 to specify summary sort.
- Control fields are not dropped (no X in column 28). Since the control field is in the desired position for output and is not modified, it is not dropped. As a result, the work file contains shorter records.

10.5.2 File Description Specifications

The output file pathname is INV.PNS, the work file is assigned to DS01, and the input file pathname is INV.CUMN. The file attributes are the same as previously used.

10.5.3 Record Selection Specification

All records are to be included; no specification is necessary.

10.5.4 Reformatting Specification

The control field is to be the PART NO. field (columns 12 through 21 of the input file, INV.CUMB). This is specified on line 270 of the form (Figure 10-11). A forced blank space is specified on line 280. Dollar signs are specified as forced characters on lines 300 and 310.

SORT/MERGE SPECIFICATIONS, SHEET A

Page 0 0

Header Specification

Line Number	Spec. Type	Merge: MERGE, MERGES / Sort: SORTA, SORTR, SORTRS	Largest Total of Control Fields of Any Record Type (Bytes)	Sequence (A/D)	Reserved	Alternate Coll. Seq. (S)	Print Option (0/1/3)	Drop Ctrl. Flds. (X)	Output Record Length	Variable Lngth. Recs. (V)	Verify Option (V)	Memory Work-Space	Comments
3 4 5	6	7 8 9 10 11 12	13 14 15 16 17	18	19 20 21 22 23 24 25	26	27	28	29 30 31 32	33	34	35 36 37 38 39	(Columns 40 to 72)
0 0 0	H	SORTRS	10A						36			16000	

File Description Specifications

OUTPUT FILE

Line Number	Spec. Type	File Use	F.T. (S/R/I)	File Pathname	Logical Record Length	Physical Record Length	Number of Logical Records	Comments
3 4 5	6	7	8	9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25	26 27 28 29 30 31 32	33 34 35 36 37 38 39	40 41 42 43 44	(Columns 45 to 72)
0 1	O	O		INV.PNS				
0 2	O	A			36	576		

WORK FILE

Line Number	Spec. Type	File Use	F.T. (E/N)	Volume Name	Comments
3 4 5	6	7	8	9 10 11 12 13 14 15 16	(Columns 17 to 80)
0 3	O	W	E	DS01	

INPUT FILE

Line Number	Spec. Type	File Use	F.T. (S/R/I)	File Pathname	L.R. Length	No. of Records	Comments
3 4 5	6	7	8	9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25	26 27 28 29 30 31 32	33 34 35 36 37 38 39 40 41 42 43 44		(Columns 45 to 72)
0 4	O	I		SINV.CUMN				
0 5	O	A						
0 6	O	I						
0 7	O	A						
0 8	O	I						
0 9	O	A						
1 0	O	I						
1 1	O	A						
1 2	O	I						
1 3	O	A						

Figure 10-11. Example 4, Specifications Sheets (Sheet 1 of 2)

SORT/MERGE SPECIFICATION, SHEET 6

Record Selection Specification

Line Number	TYPE (I/O)	Continuation (A/O)	(C/Z/D/S/P/U)	Operand 1 Location From	Operand 1 Location To	Rel. (EQ, NE, LT, GT, LE, GE)	C/O	Operand 2 (Field or Constant) Location From	Operand 2 Location To	Constant	Comments (Columns 40 to 72)
1 3											
1 4											
1 5											
1 6											
1 7											
1 8											
1 9											
2 0											
2 1											
2 2											
2 3											
2 4											
2 5											
2 6											

Reformatting Specification

Line Number	Type	TYPE (N/O/F/D/S) (C/Z/D/S/I/F/V/P/U)	Location From	Location To	Forced Record Character	Forced Substitute Character	Forced Continuation	Overflow	Reserved	Comments (Columns 40 to 72)
2	0	NC	12	21					} CONTROL 10 BYTES	
2 8	0	DV				b				
2 9	0	SS	22	29				10		PART QTY.
3 0	0	DV				b				
	0	DV				$			DATA FIELDS	
	0	SS	36	46				13	26 BYTES	AMOUNT $
			TOTAL RECORD LENGTH = 36 BYTES							

(A)135993A

Figure 10-11. Example 4, Specifications Sheets (Sheet 2 of 2)

The fields defined on lines 290 and 320 are summary fields. Line 290 is a summary field for PART QTY. The summary is specified with an S in column 7 to denote summary and an S in column 8 to denote summary of a signed ASCII field. The columns of the input record are then specified in the location field as columns 22 through 29 of the input file. The overflow in columns 20 through 22 of the form indicates the total number of positions required for the sum. In this example, two additional positions were allowed. Thus on line 290, the specified columns of the input file total 8 columns. Two is added for the overflow, and the resulting 10 is placed in the overflow field. The overflow for line 320 is determined in a similar manner.

When the sort is complete, each PART NO. has one record output with the sum of PART QTY. and the sum of AMOUNT for that part. Figure 10-12 shows an example of the output.

```
     128 000000090+ $000000012792+
     528 000000047+ $000000007786+
    2222 000000202+ $000000013904+
   45575 000033322+ $000000107589+
   65111 000000008- $000000001743-
  111111 000018864+ $000000028221+
  111155 000000751+ $000000000554+
  265111 000013266+ $000000047196+
  568321 000001224- $000000077212-
 2465498 000001995+ $000000000069+
 4568521 000000022+ $000000001504+
 5687215 000000017+ $000000004853+
 7655321 000001422+ $000000005217+
12365498 000003788+ $000000013868+
12554654 000000001- $000000000335-
45000123 000021012+ $000000090935+
45685215 000000019- $000000008906-
87651321 000008888+ $000000110430+
87654321 000002200- $000000212995-
```

Figure 10-12. Example Output from Summary Sort

11

Sort/Merge Structure

11.1 GENERAL

Logically, the Sort/Merge utility is a group of modules working together to perform a sort or merge operation. These modules consist of the following:

- Control Statement Compiler Module — Summarizes the Sort/Merge Specifications Sheets A and B into tables for use by the Record Selection and Reformatting Module or the Summary Module.
- Record Selection and Reformatting Module — Uses the tables that the Control Statement Compiler Module constructs to select records from the input file for inclusion in the work file; records are selected according to criteria in the Record Selection Specification and are reformatted according to the Reformatting Specification.
- Sort Module — Sorts selected records into sorted strings.
- String Merge Module — Merges the sorted strings into a long string of sorted records.
- Record Merge Module — Merges already sorted modules in a merge-only operation.
- Summary Module — Sums input record fields as specified in the Reformatting Specification for either a sort or merge operation.

Figure 11-1 shows a representation of these modules for a sort operation; Figure 11-2 shows a representation of these modules for a merge-only operation. Notice that in a merge-only operation, the Record Merge Module is substituted for the Sort Module and String Merge Module used in a sort operation.

The various files shown on the left side of Figures 11-1 and 11-2 are as follows:

- Input file is the file of input records to be sorted or merged.
- Work file is a temporary storage file that Sort/Merge uses in its sort operation. Strings of input records are sorted, then stored temporarily on the work file, and later retrieved from the work file for merging operations with other sorted record strings.
- Output file is the file on which the sorted and merged records are stored after the operation is complete.

11.2 SORT/MERGE MODULES

The following paragraphs further explain the modules and files listed in paragraph 11.1.

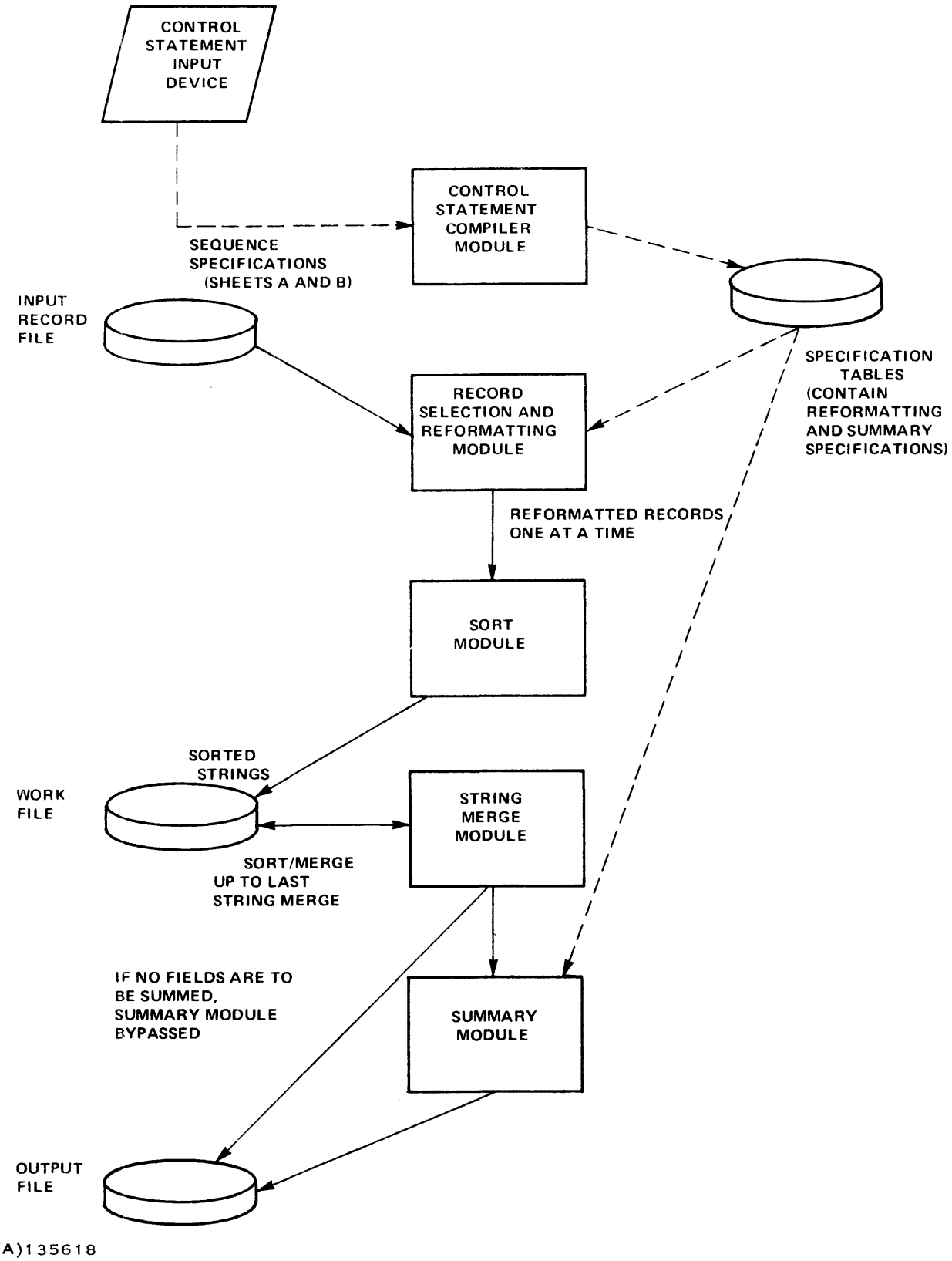


Figure 11-1. Flow Diagram of Sort Operation

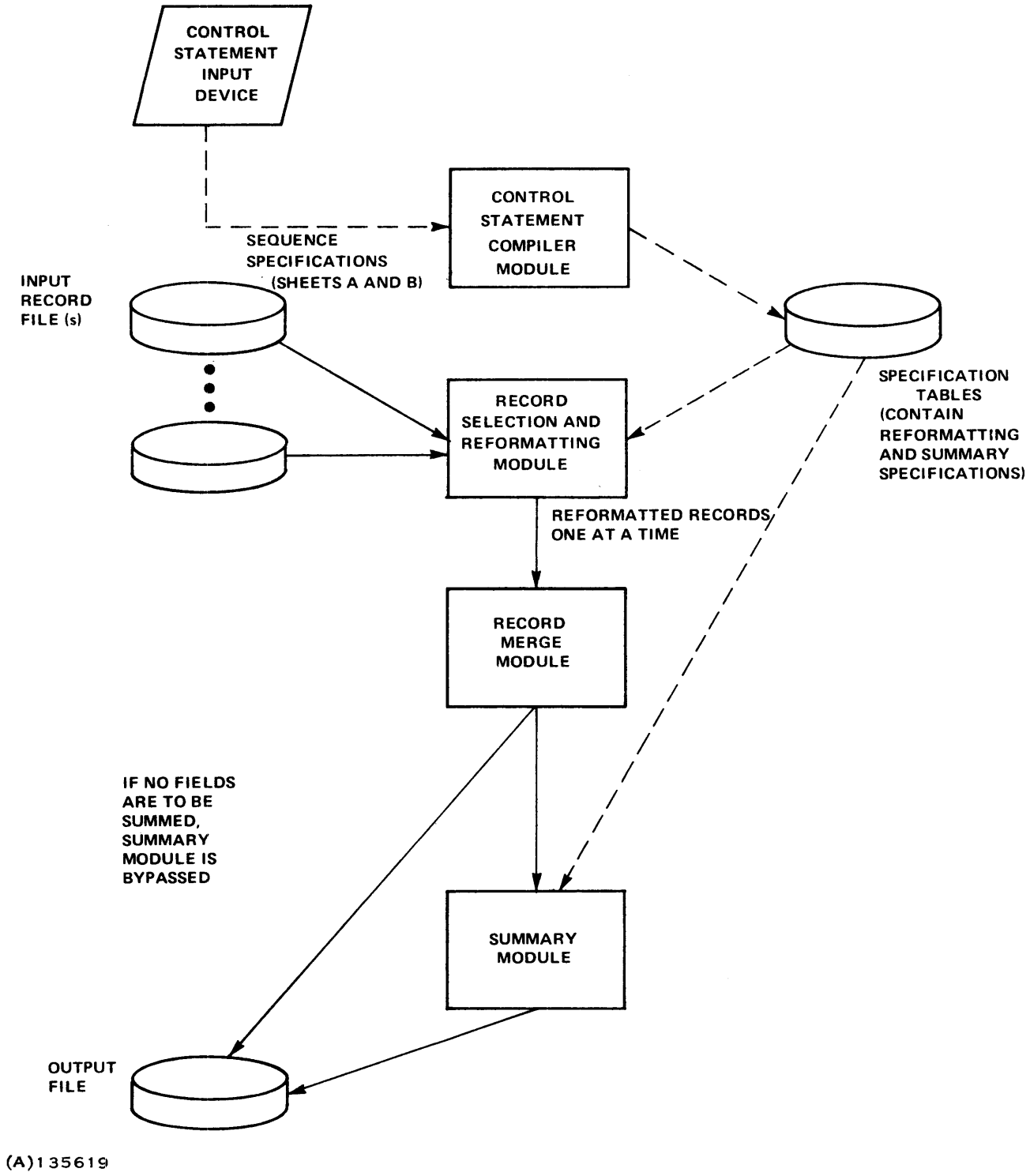


Figure 11-2. Flow Diagram of Merge-Only Operation

11.2.1 Control Statement Compiler Module

The Control Statement Compiler Module accepts the Sort/Merge control specifications and builds the selection, reformatting, and summary tables. It diagnoses errors that appear in the Sort/Merge control specifications and can, as an option, provide to you a listing of the control specifications. The selection, reformatting, and summary tables are packed very tightly to conserve memory space since they must reside in memory during the sort phase of the Sort/Merge process or during the merge phase if a merge-only operation is being executed. The selection and reformatting information is broken down field by field from the Header, Record Selection, and Reformatting Specifications. Each field is then encoded into a bit pattern and packed into a table as tightly as possible. That is, each input field occupies as few bit positions as possible while allowing space for all of the options or possible entries of a particular field. The tables contain complete information, compressed and equivalent to the actual input control specifications.

When all of the Sort/Merge control specifications have been diagnosed, packed into tables, and optionally listed, the Control Statement Compiler Module is no longer required. It is then replaced in memory by the following:

- Record Selection and Reformatting Module and Sort Module for a sort operation
- Record Selection and Reformatting Module and Record Merge Module for a merge-only operation

11.2.2 Record Selection and Reformatting Module

The Record Selection and Reformatting Module reads the sequential input records that are to be sorted and/or merged. It neither sorts nor merges, but prepares the records for the sort or merge process. As each record is read from an input file, the compressed Record Selection Specifications are used to determine the type of the record. Once the type has been determined, a decision is made whether to use the record and reformat it or whether to bypass the record. If it is to be used, the record is broken down into fields and each field is positioned (in a sequence according to specifications) to form a reformatted record. Each reformatted record consists of two portions: a control portion (or key) and a data portion. The control portion is used to determine the record's sequence in the sorted file, while the data portion constitutes the tag-along information for each record.

The control portion can consist of any number of fields from the input record, or it can be constant data from the Sort/Merge Reformatting Specifications. Fields within the same record type can be collated in opposite orders (high to low, then low to high). To increase the speed of sorting and merging, the control portion is always forced to begin and end on a word boundary in memory. Likewise, the data portion can consist of any of the original input record fields in any order; it is also forced to reside on a word boundary so that the record can be compared to other records and moved in memory with word operand instructions. The Record Selection and Reformatting Module remains in memory until all of the input records have been read.

11.2.3 Sort Module

The Sort Module accepts records to be sorted one record at a time from the Record Selection and Reformatting Module. These records are sorted into strings, blocked, and written to a disk work file.

The Sort Module uses a replacement-selection (sometimes called tournament) sorting algorithm. The replacement-selection technique produces sorted strings of records. On the average, these strings are twice as long as the record-holding area in memory for random records. For records

that are nearly sorted, the strings can be extremely long. When all input records have been processed, the buffer is purged and the String Merge Module replaces the Sort Module in memory.

11.2.4 String Merge Module

The String Merge Module provides an n-way merge, where n is based on the amount of available memory. Memory is divided into n areas, each of which is used as an input buffer for one of the sets of sorted strings of records residing in the disk work file. As the strings are merged, the resulting string is then written back to the disk work file. This process continues until there are only n strings remaining in the work file.

When only n strings remain, the last-pass merge begins. The last-pass merge contains many of the same routines as the earlier merge; however, as each record is merged, it is passed out of the String Merge Module rather than being written back to the disk work file.

The Sort/Merge algorithm becomes very efficient for files that are already in a nearly sorted order. It produces only a few long strings during the sort mode. If the number of strings produced is less than or equal to the order of merge n, the records are passed out of the String Merge Module as soon as the last input record has been processed.

11.2.5 Record Merge Module

The Record Merge Module performs up to a five-way merge of previously sorted files. This merge should not be confused with the String Merge Module of the Sort/Merge process, which only merges the strings produced by Sort/Merge. Record selection criteria generated by the Control Statement Compiler Module drives the Record Merge Module.

The algorithm that the Record Merge Module uses is as follows:

1. Obtain one record from each input file via the Record Selection and Reformatting Module.

2. Write to the output file the lowest key record in memory.

3. Using the Record Selection and Reformatting Module, read a new record into memory from the input file from which the last record written was read.

4. Perform steps 2 and 3 until an EOF mark is encountered on all input files.

5. Write an EOF on the output file.

11.2.6 Summary Module

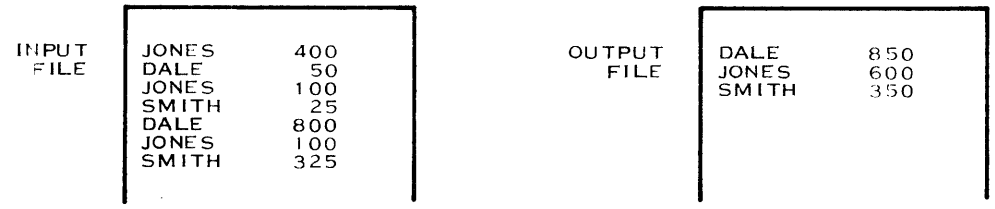
Sort/Merge can sum specified fields in successive records. The field to sum is specified in the Reformatting Specification as a summary field.

The Summary Module determines which fields are to be summed and the form of the summary data from tables built by the Control Statement Compiler Module. These summary tables are the only such tables in memory during this phase of Sort/Merge.

After the records emerge from the String Merge Module (from a sort) or the Record Merge Module (from a merge), they are sent to the Summary Module. The Summary Module keeps a running total of the summation field(s) as the records are passed to it. When all of the summary records have

been received for a particular set of control specifications, the summed total of the field(s) is flushed to the output file as a single record and the running total is reset to zero.

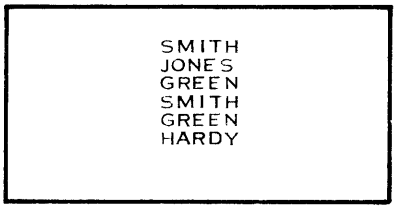
For example, the following input file has different dollar values associated with the names Smith, Dale, and Jones; the output file on the right shows a summary sort of this file, sorted alphabetically by name.



(A)135620

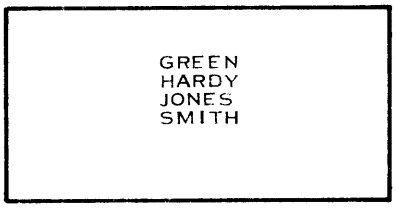
Summary data fields are summed only in a summary sort (SORTRS) or merge (MERGES) operation. In a tag-along sort (SORTR) or normal merge (MERGE), summary fields are treated as ordinary data fields. In address-only sorts, they are treated as commentary.

If no summation field is specified in a summary sort, the summary module outputs all records with unequal control fields and one record from each equal control field set. For example:



(A)135621

A summary sort on the above names with no summation specified produces the following:



(A)135622

Appendix A

Sort/Merge Messages

Messages provide information about the Sort/Merge operation in progress. Included are information messages giving statistics or the present phase of the Sort/Merge operation, warnings indicating program output may not meet the desired specifications, or messages indicating an unrecoverable error was encountered and Sort/Merge will terminate.

The Sort/Merge messages are of three types; each type begins with a code word. Each code word indicates one of three levels of message importance. The code word is followed by a number that identifies the type of error or occurrence. The code words are as follows:

- ERROR — Indicates that the source control statement contains an error and the program will terminate following the generation phase. The source control statements are translated into machine language during the generation phase. Incorrect source information prevents this translation.

- WARNING — Indicates that an unusual condition has been found and that you should check to determine if the desired outcome of Sort/Merge has been affected. A warning does not terminate program operation; however, results of the operation are unpredictable.

- INFO — Provides information that you might need. Some information messages pertain to situations that immediately terminate Sort/Merge. An information message contains a short explanation of the occurrence that caused the termination.

Table A-1 lists the error and warning messages in numerical order; Table A-2 lists information messages; and Table A-3 lists I/O error messages.

Four messages are returned at the completion of the Sort/Merge process:

SORT/MERGE COMPLETED NORMALLY
Indicates normal completion.

SORT/MERGE COMPLETED ABNORMALLY
Indicates that errors occurred during the Sort/Merge process.

SORT/MERGE COMPLETED ABNORMALLY — HEADER MISSING
Indicates that the control file does not contain a valid header statement.

SORT/MERGE COMPLETED NORMALLY — LOCK KIF RECORDS ENCOUNTERED
Indicates that Sort was able to read key-indexed records that were locked by some other task and perhaps updated. Therefore, the validity of the sorted data should be checked.

Table A-1. Error and Warning Messages

Code	Explanation
ERROR 1	Missing control field size
ERROR 2	Invalid control field size
WARNING 3	Invalid sort type
ERROR 4	Invalid alternate sequence statement
WARNING 5	Invalid ascend/descend indicator
WARNING 6	Ignore nonblank characters in this field
WARNING 7	Invalid alternate sequence indicator
WARNING 8	Invalid print option
ERROR 10	Missing output record size
ERROR 11	Invalid output record size
ERROR 12	Invalid hexadecimal constant
WARNING 13	Missing alternate sequence specifications
WARNING 14	Missing terminating asterisk
ERROR 15	Invalid selection specification
WARNING 17	Invalid format: illegal entry in character portion field
ERROR 18	Invalid operand 1 start
ERROR 19	Invalid operand 1 end
ERROR 20	Invalid operand 2 start
ERROR 21	Invalid operand 2 end
ERROR 22	Invalid operator code
WARNING 23	Invalid operand description
ERROR 24	Invalid include-all specification
ERROR 25	Invalid specification
ERROR 26	Invalid field specification
ERROR 27	Missing field specification

Table A-1. Error and Warning Messages (Continued)

Code	Explanation
ERROR 28	Invalid field-end description
ERROR 29	Field-end below minimum
ERROR 30	Field-end above maximum
ERROR 31	Invalid field-start description
ERROR 32	Field-start below minimum
ERROR 33	Field-start above minimum
WARNING 34	Invalid overflow description
WARNING 35	Same record selection type assumed
WARNING 36	No data fields and control field dropped
WARNING 37	Invalid sequence number
WARNING 38	Source images are not in sequence
ERROR 39	Not enough memory specified in header statement
ERROR 40	More than 24 summary fields specified
WARNING 41	Inconsistent summary data fields
ERROR 42	Invalid force-all specification
ERROR 43	Forced data statement with overflow specified
ERROR 44	Field specifications exceed output record size
ERROR 45	Control field specifications exceed maximum control size
WARNING 46	Character constant length exceeds 20
ERROR 48	Invalid continuation
WARNING 49	Packed or unpacked factor 1 with ALTSEQ
ERROR 50	Invalid factor length
WARNING 51	Invalid relation when zone specified
ERROR 52	Operand 2 should be constant; zone specified
WARNING 53	Missing summary data field on SORTS

Table A-1. Error and Warning Messages (Continued)

Code	Explanation
ERROR 54	Modified control field; should be dropped
ERROR 55	Exceeds maximum length for character field
WARNING 56	Exceeds maximum length for Z or V field
WARNING 57	Exceeds maximum length for D field
WARNING 58	Exceeds maximum length for packed field
WARNING 59	Exceeds maximum length for unpacked field
WARNING 60	Control field size less than specified
WARNING 61	Overflow size less than field size
ERROR 62	Out of order control field
WARNING 63	Invalid drop control field; drop assumed
WARNING 64	Invalid verify option; verify assumed
ERROR 65	Unexpected end-of-file (EOF) found
ERROR 66	Output record size exceeds 4096
ERROR 67	Input record size exceeds 4096
WARNING 68	Ignore summary field on SORTA or SORTR
ERROR 69	Missing control field specification
WARNING 70	Field specifications less than output record size
ERROR 71	Inconsistent control and record sizes in the header
ERROR 72	Inconsistent operand lengths
ERROR 73	Invalid memory limit
ERROR 74	Invalid logical record size
ERROR 75	Invalid physical record size
ERROR 76	Invalid physical record count
ERROR 77	Missing output file description
ERROR 79	Missing input file description

Table A-1. Error and Warning Messages (Continued)

Code	Explanation
ERROR 80	Missing file name
ERROR 81	Invalid file use
ERROR 82	Invalid file description
ERROR/WARNING 83	Invalid file type. An error message indicates an incorrect Input or Output File Specification; a warning message indicates an incorrect Work File Specification
ERROR 84	Exceeds maximum number of input files
ERROR 85	Missing file description continuation
ERROR 87	Invalid input record size
ERROR 88	Requested space exceeds available memory
ERROR 89	Integer field length odd or greater than six bytes or invalid format for signed ASCII fields
ERROR 90	Floating-point field length not four or eight bytes
WARNING 91	Input file type does not match file type specified
ERROR 92	Input file type is not a sequential, relative record, or key-indexed file
WARNING 93	Logical record length (LRL) specified for input file is smaller than LRL returned by operating system; Larger specified LRL used as input LRL
WARNING 94	LRL specified for input file is greater than LRL returned from operating system; larger specified LRL used as input LRL
ERROR 95	Missing LRL

Table A-2. INFO Messages

Code	Explanation[1]
INFO 0:	No errors
INFO 1:	Number of errors found[2]
INFO 3:	Number of warnings given
INFO 4:	End of generation phase
INFO 5:	Sort execution has begun
INFO 6:	I/O error reading input file[2]
INFO 7:	I/O error writing output file[2]
INFO 8:	I/O error reading source file[2]
INFO 9:	Number of input file records read
INFO 10:	Number of input file records selected
INFO 11:	No input file records found
INFO 12:	No input file records selected
INFO 13:	Number of output records written
INFO 14:	Input file record size
INFO 15:	Input file block size
INFO 16:	Output file record size
INFO 17:	Output file block size
INFO 18:	Work file record size
INFO 19:	Output file size insufficient[2]
INFO 20:	Work file size insufficient[2]
INFO 21:	Memory size insufficient[2]
INFO 23:	Words of working storage available
INFO 24:	Bytes required for specifications
INFO 25:	Sort final pass has begun
INFO 26:	Sort final pass completed

Table A-2. INFO Messages (Continued)

Code	Explanation[1]
INFO 27:	I/O error on work file[2]
INFO 28:	Number of passes just completed
INFO 29:	Number of passes remaining
INFO 30:	Number of work file tracks used
INFO 31:	Work file record count is in error[2]
INFO 32:	File merge has begun
INFO 33:	File merge completed
INFO 34:	Overlay load error
INFO 35:	Output key-indexed file not created
INFO 36:	Output file type incorrectly specified
INFO 37:	Invalid output file type

Notes:

[1] Both code and diagnostic explanation will be written out.
[2] Terminates program immediately; no further Sort/Merge processing occurs.

Table A-3. File I/O Error Messages

INPUT OPEN ERROR (four-digit error number)
INPUT CLOSE ERROR (four-digit error number)
WORK OPEN ERROR (four-digit error number)
OUTPUT OPEN ERROR (four-digit error number)
FILE (file name) ERROR (four-digit error number)

NOTE

The four-digit error number is an SVC error such as 00D4. The meaning of these error codes is found in the *Model 990 Computer DX10 Operating System Release 3 Reference Manual, Volume VI, Error Reporting and Recovery* .

Appendix B

Alternate Collating Sequence Specifications

B.1 GENERAL

You can define an alternate collating sequence to vary the collating hierarchy of the character set. To do this, place an S in column 26 of the Header Specification to designate that alternate collating sequence records will be supplied to Sort/Merge.

You can use as many alternate collating sequence records as needed. Each record begins with the characters ALTSEQbb in positions 1 through 8; the complete format of an ALTSEQ record is as follows:

Positions	Meaning
1-8	ALTSEQbb
9-10	Two-character hexadecimal representation of the first character whose hierarchy is *being defined.*
11-12	Two-character hexadecimal representation of the first character whose hierarchy is *presently being used.*
13-16, 17-20, 21-24, . . . end-of-record	Used in the same manner as positions 9-12.

Do not leave positions blank between entries in a record. When additional records are required, columns 1 through 8 of the records that follow must also contain ALTSEQbb. A record containing double asterisks (**) in positions 1 and 2 terminates the set of alternate collating sequence specifications.

Alternate collating sequence records immediately follow the Header Specification in the specification sequence.

When a character is moved to a collating sequence position previously held by another character, the two characters are considered equal. If this is not desired, move the original character elsewhere in the sequence.

An alternate collating sequence affects the following:

- Operand 1 and operand 2
- Normal and opposite control fields
- Original control field characters (before modification through force)

Figure B-1 shows an example of alternating sequence specification records.

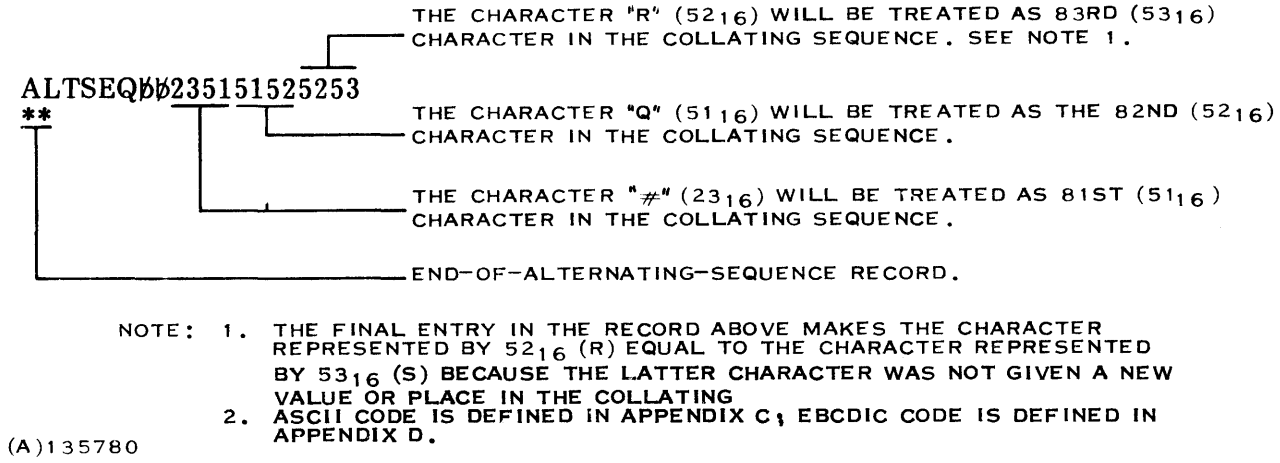


Figure B-1. Example of Alternating Sequence Specification Records

B.2 ALTERNATE SEQUENCE SPECIFICATION USING THE SCI

When using the System Command Interpreter (SCI), you need only enter strings of two-character hexadecimal pairs.

B.2.1 Interactive Mode

If an S is entered as the value for the keyword ALTERNATE COLLATING SEQUENCE, you will be prompted for alternate sequence pairs as follows:

```
ALTERNATE SEQUENCE SPECIFICATION

    ENTER ALTERNATE PAIRS:
```

Enter alternate sequence pairs, up to eight pairs per line, as needed. After entering the last pair, enter a null return in response to the next ENTER ALTERNATE PAIRS prompt that is displayed to indicate the end of the alternate sequence specification. The ** line is automatically entered into the control file.

Figure B-2 gives an interactive example of alternate sequence specification.

```
[] XSMF
EXECUTE SORT/MERGE
 NEW CONTROL FILE?:    YES
 CONTROL FILE NAME:    .CONTRL
**SORT/MERGE HEADER SPECIFICATION**
 SORT/MERGE TYPE:   SORTR
 TOTAL CONTROL LENGTH:    5
 SORT/MERGE SEQUENCE:   A
 ALTERNATE SEQUENCE:     S
 PRINT OPTION:   1    0
 DROP CONTROL FIELDS:   X
 OUTPUT RECORD LENGTH:     80
 VARIABLE LENGTH RECORDS:
 VERIFY OPTION:
 MEMORY SIZE:     16000
**SORT/MERGE ALTERNATE SEQUENCE SPECIFICATION**
 ENTER ALTERNATE PAIRS:    32523353
**SORT/MERGE ALTERNATE SEQUENCE SPECIFICATION**
ENTER ALTERNATE PAIRS:
**SORT/MERGE OUTPUT FILE SPECIFICATION
 FILE PATHNAME:     .OUT2
 FILE TYPE:   S
 LOGICAL RECORD LENGTH:    80
 PHYSICAL RECORD LENGTH:     560
 NUMBER OF LOGICAL RECORDS:
**SORT/MERGE WORK FILE SPECIFICATION**
 WORK FILE VOLUME:   DS01
 WORK FILE TYPE:    E
**SORT/MERGE INPUT FILE SPECIFICATION**
 FILE PATHNAME:      .SRTDAT
 FILE NAME:   S
 LOGICAL RECORD LENGTH:     80
 NUMBER OF LOGICAL RECORDS:    400
ENTER NEXT SPECIFICATION  -- SELECTION OR REFORMATTING
 NEXT:     R
**SORT/MERGE REFORMATTING SPECIFICATION**
 FIELD TYPE:    N
 CHARACTER PORTION:   C
 BEGINNING LOCATION:    23
 ENDING LOCATION:    27
ENTER NEXT SPECIFICATION -- SELECTION OR REFORMATTING
 NEXT:    R
**SORT/MERGE REFORMATTING SPECIFICATION**
 FIELD TYPE:     D
 CHARACTER PORTION:   C
 BEGINNING LOCATION:    1
 ENDING LOCATION:     80
ENTER NEXT SPECIFICATION -- SELECTION OR REFORMATTING
 NEXT:
 RUN SORT/MERGE?:   YES    NO
[]
```

Figure B-2. Interactive Examples

B.2.2 Batch Mode

To specify an alternate sequence in a batch control stream, you must specify the alternate sequence keyword value, AS = S, in the header command statement, SM$HD. Enter the alternate sequence command statement, SM$ALT, as needed immediately following the header command statement. An additional SM$ALT statement that contains no alternate pairs indicates to Sort/Merge the end of the alternate sequence specificaton.

Figure B-3 is an example of batch stream alternate sequence specification.

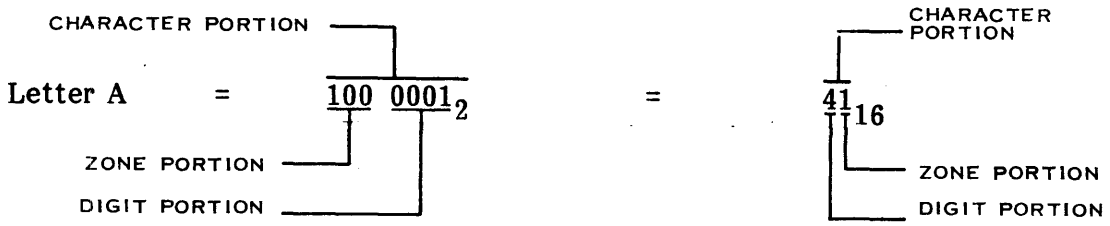
```
BATCH
SM$SMC CFN=.CONTRL
SM$HD AS=S,TCL=80,DCF=" ",PO=0,OR=80,MS=16000
SM$ALT EAP=30F031F132F233F334F435F536F637F738F8
SM$ALT EAP=39F941C142C243C344C445C546C647C748C8
SM$ALT EAP=49C94AD14BD24CD34DD44ED54FD650D751D8
SM$ALT EAP=52D953E254E355E456E557E658E759E860E9
SM$ALT
SM$OUT FP=.OUT33,LRL=80,PRL=560
SM$WKF
SM$IN FP=DS01.SRTDAT,LRL=80,NOLR=400
SM$REF FT=N,BL=1,EL=80
SM$CLS
XBSM CFN=DS01.CONTRL,LDN=LP01
SM$SMC CFN=.CONTRL
SM$HD AS=S,TCL=80,DCF=" ",PO=0,OR=80,MS=16000
SM$ALT EAP=F030F131F232F333F423F535F636F737F838
SM$ALT EAP=F939C141C242C343C444C545C646C747C848
SM$ALT EAP=C949D14AD24BD34CD44DD54ED64FD750D851
SM$ALT EAP=D952E253E354E455E556E657E758E859E960
SM$ALT
SM$OUT FP=.OUT34,LRL=80,PRL=560
SM$WKF
SM$IN FP=DS01.SRTDAT,LRL=80,NOLR=400
SM$REF FT=N,BL=1,EL=80
SM$CLS
XBSM CFN=DS01.CONTRL.LDN=LP01
SM$SMC CFN=DS01.CONTRL
SM$HD SMS=D,TCL=2,PO=0,ORL=80,MS=12000
SM$OUT FP=DS01.OUT35
SM$IN FP=DS01.SRTDAT
SM$SLC CP=C,01E=5,REL=EQ,FOC=C,02C=5
SM$REF FT=N,EL=5
SM$REF FT=0,EL=65
SM$REF FT=D,BL=1,EL=80
SM$CLS
XBSM CFN=.CONTRL,LDN=LP01
EBATCH
```

Figure B-3. Batch Example

Appendix C

ASCII Codes

The following diagram shows the zone and digit character portions of the ASCII codes for binary and for hexadecimal:



(A)135750

Table C-1. ASCII Control Codes[1]

Control		Binary Code	Hexadecimal Code
NUL —	Null	0000 0000	00
SOH —	Start of heading	0000 0001	01
STX —	Start of text	0000 0010	02
ETX —	End of text	0000 0011	03
EOT —	End of transmission	0000 0100	04
ENQ —	Enquiry	0000 0101	05
ACK —	Acknowledge	0000 0110	06
BEL —	Bell	0000 0111	07
BS —	Backspace	0000 1000	08
HT —	Horizontal tabulation	0000 1001	09
LF —	Line feed	0000 1010	0A
VT —	Vertical tab	0000 1011	0B
FF —	Form feed	0000 1100	0C
CR —	Carriage return	0000 1101	0D
SO —	Shift out	0000 1110	0E
SI —	Shift in	0000 1111	0F
DLE —	Data link escape	0001 0000	10
DC1 —	Device control 1	0001 0001	11
DC2 —	Device control 2	0001 0010	12
DC3 —	Device control 3	0001 0011	13
DC4 —	Device control 4 (stop)	0001 0100	14
NAK —	Negative acknowledge	0001 0101	15
SYN —	Synchronous idle	0001 0110	16
ETB —	End of transmission block	0001 0111	17

Table C-1. ASCII Control Codes[1] (Continued)

Control	Binary Code	Hexadecimal Code
CAN — Cancel	0001 1000	18
EM — End of medium	0001 1001	19
SUB — Substitute	0001 1010	1A
ESC — Escape	0001 1011	1B
FS — File separator	0001 1100	1C
GS — Group separator	0001 1101	1D
RS — Record separator	0001 1110	1E
US — Unit separator	0001 1111	1F
DEL — Delete, rubout	0111 1111	7F

Note:

[1] American Standards Institute Publication X3.4-1968

Table C-2. ASCII Character Code[1]

Character	Binary Code	Hexadecimal Code	Character	Binary Code	Hexadecimal Code
Space	0010 0000	20	P	0101 0000	50
!	0010 0001	21	Q	0101 0001	51
"(double quote)	0010 0010	22	R	0101 0010	52
#	0010 0011	23	S	0101 0011	53
$	0010 0100	24	T	0101 0100	54
%	0010 0101	25	U	0101 0101	55
&	0010 0110	26	V	0101 0110	56
'(single quote)	0010 0111	27	W	0101 0111	57
(	0010 1000	28	X	0101 1000	58
)	0010 1001	29	Y	0101 1001	59
*(asterisk)	0010 1010	2A	Z	0101 1010	5A
+	0010 1011	2B	[	0101 1011	5B
,(comma)	0010 1100	2C	\	0101 1100	5C
-(minus)	0010 1101	2D	]	0101 1101	5D
.(period)	0010 1110	2E	^	0101 1110	5E
/	0010 1111	2F	_ (underline)	0101 1111	5F
0	0011 0000	30	`	0110 0000	60
1	0011 0001	31	a	0110 0001	61
2	0011 0010	32	b	0110 0010	62
3	0011 0011	33	c	0110 0011	63
4	0011 0100	34	d	0110 0100	64
5	0011 0101	35	e	0110 0101	65
6	0011 0110	36	f	0110 0110	66
7	0011 0111	37	g	0110 0111	67
8	0011 1000	38	h	0110 1000	68
9	0011 1001	39	i	0110 1001	69
:	0011 1010	3A	j	0110 1010	6A
;	0011 1011	3B	k	0110 1011	6B
<	0011 1100	3C	l	0110 1100	6C
=	0011 1101	3D	m	0110 1101	6D
>	0011 1110	3E	n	0110 1110	6E
?	0011 1111	3F	o	0110 1111	6F
@	0100 0000	40	p	0111 0000	70
A	0100 0001	41	q	0111 0001	71
B	0100 0010	42	r	0111 0010	72
C	0100 0011	43	s	0111 0011	73
D	0100 0100	44	t	0111 0100	74
E	0100 0101	45	u	0111 0101	75
F	0100 0110	46	v	0111 0110	76
G	0100 0111	47	w	0111 0111	77
H	0100 1000	48	x	0111 1000	78
I	0100 1001	49	y	0111 1001	79
J	0100 1010	4A	z	0111 1010	7A
K	0100 1011	4B	{	0111 1011	7B
L	0100 1100	4C	|	0111 1100	7C
M	0100 1101	4D	}	0111 1101	7D
N	0100 1110	4E	~	0111 1110	7E
O	0100 1111	4F			

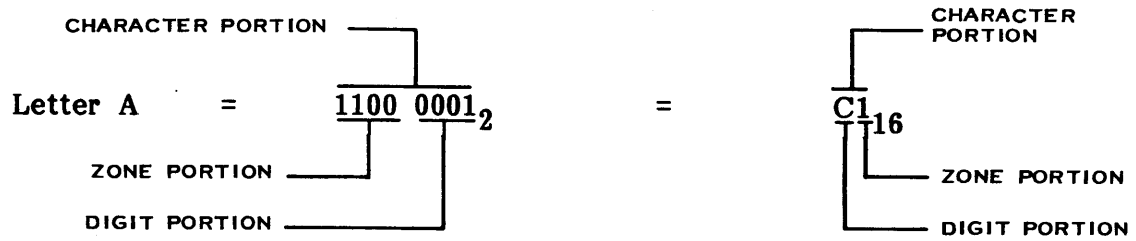
Note:

[1] American Standards Institute Publication X3.4-1968

Appendix D

EBCDIC Codes

The following diagram shows the zone and digit character portions of the EBCDIC codes for binary and for hexadecimal:



(A)135751

Meaning	Binary Code	Hexadecimal Code	Meaning	Binary Code	Hexadecimal Code
NUL	0000 0000	00	CU1	0001 1011	1B
SOH	0000 0001	01	IFS	0001 1100	1C
STX	0000 0010	02	IGS	0001 1101	ID
ETX	0000 0011	03	IRS	0001 1110	1E
PF	0000 0100	04	IUS	0001 1111	1F
HT	0000 0101	05	DS	0010 0000	20
LC	0000 0110	06	SOS	0010 0001	21
DEL	0000 0111	07	FS	0010 0010	22
—	—	—	—	—	—
RLF	0000 1001	09	BYP	0010 0100	24
SMM	0000 1010	0A	LF	0010 0101	25
VT	0000 1011	0B	EOB or ETB	0010 0110	26
FF	0000 1100	0C	PRE or ESC	0010 0111	27
CR	0000 1101	0D	—	—	—
SO	0000 1110	0E	SM	0010 1010	2A
SI	0000 1111	0F	CU2	0010 1011	2B
DLE	0001 0000	10	—	—	—
DC1	0001 0001	11	ENQ	0010 1101	2D
DC2	0001 0010	12	ACK	0010 1110	2E
DC3	0001 0011	13	BEL	0010 1111	2F
RES	0001 0100	14	—	—	—
NL	0001 0101	15	SYN	0011 0010	32
BS	0001 0110	16	—	—	—
IL	0001 0111	17	PN	0011 0100	34
CAN	0001 1000	18	RS	0011 0101	35
EM	0001 1001	19	UC	0011 0110	36
CC	0001 1010	1A	EOT	0011 0111	37

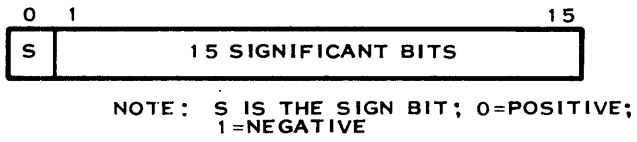
Meaning	Binary Code	Hexadecimal Code	Meaning	Binary Code	Hexadecimal Code
—	—	—	j	1001 0001	91
CU3	0011 1011	3B	k	1001 0010	92
DC4	0011 1100	3C	l	1001 0011	93
NAK	0011 1101	3D	m	1001 0100	94
—	—	—	n	1001 0101	95
SUB	0011 1111	3F	o	1001 0110	96
SP	0100 0000	40	p	1001 0111	97
—	—	—	q	1001 1000	98
¢	0100 1010	4A	r	1001 1001	99
.(period)	0100 1011	4B	—	—	—
<	0100 1100	4C	~	1010 0001	A1
(	0100 1101	4D	s	1010 0010	A2
+	0100 1110	4E	t	1010 0011	A3
|	0100 1111	4F	u	1010 0100	A4
&	0101 0000	50	v	1010 0101	A5
—	—	—	w	1010 0110	A6
!	0101 1010	5A	x	1010 0111	A7
$	0101 1011	5B	y	1010 1000	A8
*(asterisk)	0101 1100	5C	z	1010 1001	A9
)	0101 1101	5D	—	—	—
;	0101 1110	5E	{	1100 0000	C0
¬	0101 1111	5F	A	1100 0001	C1
-(minus)	0110 1111	60	B	1100 0010	C2
/	0110 0001	61	C	1100 0011	C3
—	—	—	D	1100 0100	C4
,(comma)	0110 1011	6B	E	1100 0101	C5
%	0110 1100	6C	F	1100 0110	C6
_(underline)	0110 1101	6D	G	1100 0111	C7
>	0110 1110	6E	H	1100 1000	C8
?	0110 1111	6F	I	1100 1001	C9
—	—	—	—	—	—
			}	1101 0000	D0
:	0111 1010	7A	J	1101 0001	D1
#	0111 1011	7B	K	1101 0010	D2
@	0111 1100	7C	L	1101 0011	D3
'(single quote)	0111 1101	7D	M	1101 0100	D4
=	0111 1110	7E	N	1101 0101	D5
"(double quote)	0111 1111	7F	O	1101 0110	D6
—	—	—	P	1101 0111	D7
a	1000 0001	81	Q	1101 1000	D8
b	1000 0010	82	R	1101 1001	D9
c	1000 0011	83	—	—	—
d	1000 0100	84	S	1110 0010	E2
e	1000 0101	85	T	1110 0011	E3
f	1000 0110	86	U	1110 0100	E4
g	1000 0111	87	V	1110 0101	E5
h	1000 1000	88	W	1110 0110	E6
i	1000 1001	89	X	1110 0111	E7
—	—	—	Y	1110 1000	E8
			Z	1110 1001	E9

Meaning	Binary Code	Hexadecimal Code	Meaning	Binary Code	Hexadecimal Code
—	—	—	5	1111 0101	F5
0	1111 0000	F0	6	1111 0110	F6
1	1111 0001	F1	7	1111 0111	F7
2	1111 0010	F2	8	1111 1000	F8
3	1111 0011	F3	9	1111 1001	F9
4	1111 0100	F4	—	—	—

Appendix E

Data Types for FORTRAN, COBOL, Pascal, BASIC, and RPG II

E.1 INTEGER



(A)135752

This is a 16-bit two's complement value.

Nomenclature:

FORTRAN: INTEGER

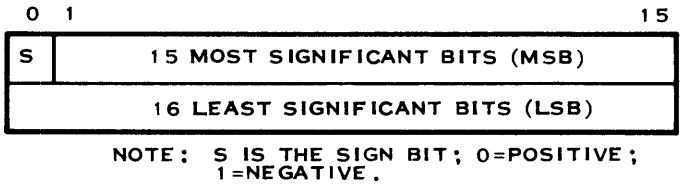
COBOL: COMPUTATIONAL-1

Pascal: INTEGER

BASIC: INTEGER

RPG II: BINARY

E.2 EXTENDED INTEGER



(A)135753

This is a 32-bit two's complement value.

Nomenclature:

FORTRAN: EXTENDED INTEGER

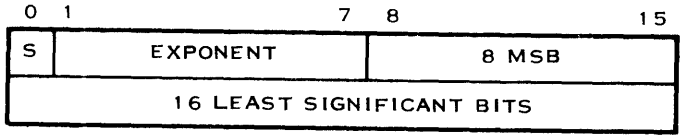
COBOL: COMPUTATIONAL-4

Pascal: LONGINT

BASIC: —

RPG II: —

E.3 FLOATING POINT



NOTE: S IS THE SIGN BIT; 0=POSITIVE; 1=NEGATIVE

(A)135754

Nomenclature:

FORTRAN: REAL

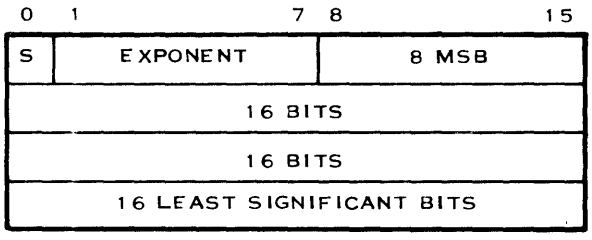
COBOL: —

Pascal: REAL (n), where n ⩽ 7

BASIC: REAL

RPG II: —

E.4 EXTENDED FLOATING POINT



NOTE: S IS THE SIGN BIT; 0=POSITIVE; 1=NEGATIVE

(A)135755

Nomenclature:

FORTRAN: DOUBLE PRECISION

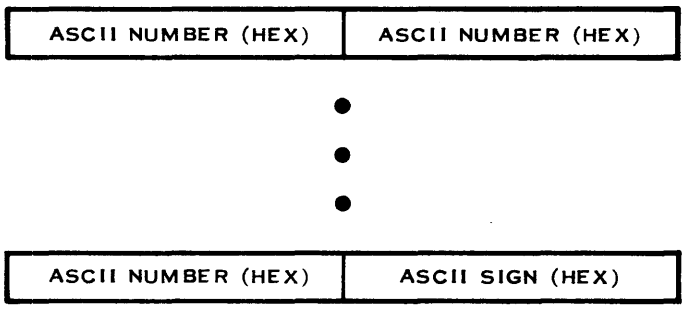
COBOL: —

Pascal: REAL (n), where n ≥8

BASIC: —

RPG II: —

E.5 SIGNED ASCII



(A)135756

The ASCII code for numbers 0 through 9 is 30_{16} through 39_{16}, respectively.

The ASCII code for the sign of a signed ASCII field is $2D_{16}$ for minus, and $2B_{16}$ or 20_{16} for plus.

Nomenclature:

FORTRAN: —

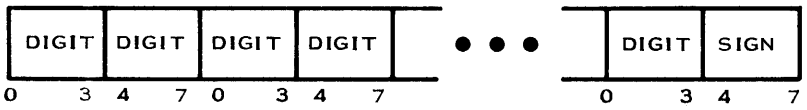
COBOL: Display (9) with sign

Pascal: —

BASIC: —

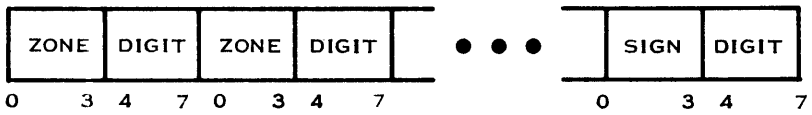
RPG II: —

E.6 RPG II COMPATIBLE PACKED FORMAT



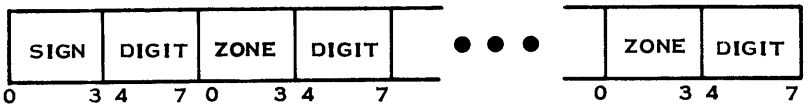
RPG II compatible packed numeric data format contains two decimal digits per byte, except for the rightmost byte which contains a sign in the four right bits of the byte. Sign is F when positive and D when negative.

E.7 RPG II COMPATIBLE UNPACKED FORMAT



RPG II compatible unpacked numeric data format contains one decimal digit per byte and a sign in the rightmost zone field of the data.

E.8 LEADING SIGN UNPACKED FORMAT



(A)143696

Leading sign unpacked format contains one decimal digit per byte and a sign in the leftmost zone field of the data.

Nomenclature:

FORTRAN: —

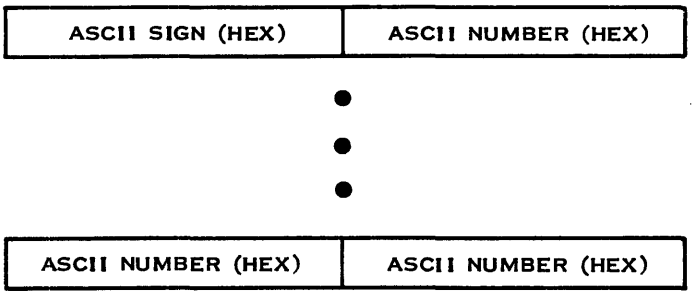
COBOL: Display sign is leading

Pascal: —

BASIC: —

RPG II: —

E.9 BEGINNING SIGNED ASCII



(A)143697

The ASCII code for numbers 0 through 9 is 30_{16} through 39_{16}, respectively.

The ASCII code for the sign is in the leftmost byte: $2D_{16}$ for minus, and $2B_{16}$ for plus.

Nomenclature:

FORTRAN: —

COBOL: Display sign is separate leading

Pascal: —

BASIC: —

RPG II: —

Appendix F

Estimating Sorting Time

You can use the following formula to obtain a rough estimate of sorting time on a dedicated system with all files on Trident disks:

$$\text{Time} = \frac{\text{NOLR} \times \text{LRL}}{\text{MEM}} \times 10$$

where:

Time = sorting time in seconds

NOLR = number of logical records to be sorted

LRL = logical record length of the work file; this is the sum of all reformatting specifications fields plus 2

MEM = bytes of memory; this is the value given on the header line

This gives a value with ± 10 percent accuracy for most combinations of up to 40,000 records.

Appendix G

Sorting with Many Records or Limited Workspace

The DX10 disk storage allocation algorithm uses a set of predefined defaults for allocating disk space. If large files are sorted (70,000 or more records), or you have limited disk storage, the work file and output file might not use the disk space most efficiently. To make the most efficient use of disk space, specify the number of logical records (NOLR) on the input created with initial and secondary allocation sizes of NOLR/8. Also, precreate the output file, specifying an initial allocation of NOLR and a secondary allocation of NOLR/8.

Alphabetical Index

Introduction

HOW TO USE INDEX

The index, table of contents, list of illustrations, and list of tables are used in conjunction to obtain the location of the desired subject. Once the subject or topic has been located in the index, use the appropriate paragraph number, figure number, or table number to obtain the corresponding page number from the table of contents, list of illustrations, or list of tables.

INDEX ENTRIES

The following index lists key words and concepts from the subject material of the manual together with the area(s) in the manual that supply major coverage of the listed concept. The numbers along the right side of the listing reference the following manual areas:

- Sections — Reference to Sections of the manual appear as "Sections x" with the symbol x representing any numeric quantity.

- Appendixes — Reference to Appendixes of the manual appear as "Appendix y" with the symbol y representing any capital letter.

- Paragraphs — Reference to paragraphs of the manual appear as a series of alphanumeric or numeric characters punctuated with decimal points. Only the first character of the string may be a letter; all subsequent characters are numbers. The first character refers to the section or appendix of the manual in which the paragraph may be found.

- Tables — References to tables in the manual are represented by the capital letter T followed immediately by another alphanumeric character (representing the section or appendix of the manual containing the table). The second character is followed by a dash (-) and a number.

  Tx-yy

- Figures — References to figures in the manual are represented by the capital letter F followed immediately by another alphanumeric character (representing the section or appendix of the manual containing the figure). The second character is followed by a dash (-) and a number.

  Fx-yy

- Other entries in the Index — References to other entries in the index preceded by the word "See" followed by the referenced entry.

CUT ALONG LINE

USER'S RESPONSE SHEET

Manual Title: Model 990 Computer DX Sort/Merge User's Guide

(946252-9701)

Manual Date: 15 December 1980 **Date of This Letter:** _____

User's Name: _____ **Telephone:** _____

Company: _____ **Office/Department:** _____

Street Address: _____

City/State/Zip Code: _____

Please list any discrepancy found in this manual by page, paragraph, figure, or table number in the following space. If there are any other suggestions that you wish to make, feel free to include them. Thank you.

Location in Manual	Comment/Suggestion
_____	_____
	_____
	_____
	_____
_____	_____
	_____
	_____
	_____
_____	_____
	_____
	_____
	_____

NO POSTAGE NECESSARY IF MAILED IN U.S.A.
FOLD ON TWO LINES (LOCATED ON REVERSE SIDE), TAPE AND MAIL

FOLD

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 7284 DALLAS, TX

POSTAGE WILL BE PAID BY ADDRESSEE

TEXAS INSTRUMENTS INCORPORATED

DIGITAL SYSTEMS GROUP

ATTN: TECHNICAL PUBLICATIONS
P.O. Box 2909 M/S 2146
Austin, Texas 78769

FOLD

Texas Instruments U.S. District Sales and Service Offices

(A complete listing of U.S. offices is available from the district office nearest your location)

California
831 S. Douglas Street
El Segundo, California 90245
(213) 973-2571

100 California Street
Suite 480
San Francisco, California 94111
(415) 781-9470

776 Palomar Avenue
P.O. Box 9064
Sunnyvale, California 94086
(408) 732-1840*

3186 Airway
Suite J
Costa Mesa, California 92626
(714) 540-7311

Colorado
9725 East Hampden Avenue
Suite 301
Denver, Colorado 80231
(303) 751-1780

Florida
1850 Lee Road
Suite 115
Winter Park, Florida 32789
(305) 644-3535

Georgia
3300 Northeast Expressway
Building 9
Atlanta, Georgia 30341
(404) 458-7791

*Service telephone number

Illinois
515 West Algonquin Road
Arlington Heights, Illinois 60005
(312) 640-2900
(800) 942-0609*

Massachusetts
504 Totten Pond Road
Waltham, Massachusetts 02154
(617) 890-7400

Michigan
24293 Telegraph Road
Southfield, Michigan 48034
(313) 353-0830
(800) 572-8740*

Minnesota
7625 Parklawn Avenue
Minneapolis, Minnesota 55435
(612) 830-1600

Missouri
2368 Schuetz
St. Louis, Missouri 63141
(314) 569-0801*

New Jersey
1245 Westfield Avenue
Clark, New Jersey 07066
(201) 574-9800

Ohio
4124 Linden Avenue
Dayton, Ohio 45432
(513) 258-3877

Pennsylvania
420 Rouser Road
Coraopolis, Pennsylvania 15108
(412) 771-8550

Texas
8001 Stemmons Expressway
P.O. Box 226080
M/S 3108
Dallas, Texas 75266
(214) 689-4460

13510 North Central Expressway
P.O. Box 225214
M/S 393
Dallas, Texas 75265
(214) 238-3881

9000 Southwest Freeway, Suite 400
Houston, Texas 77074
(713) 776-6577

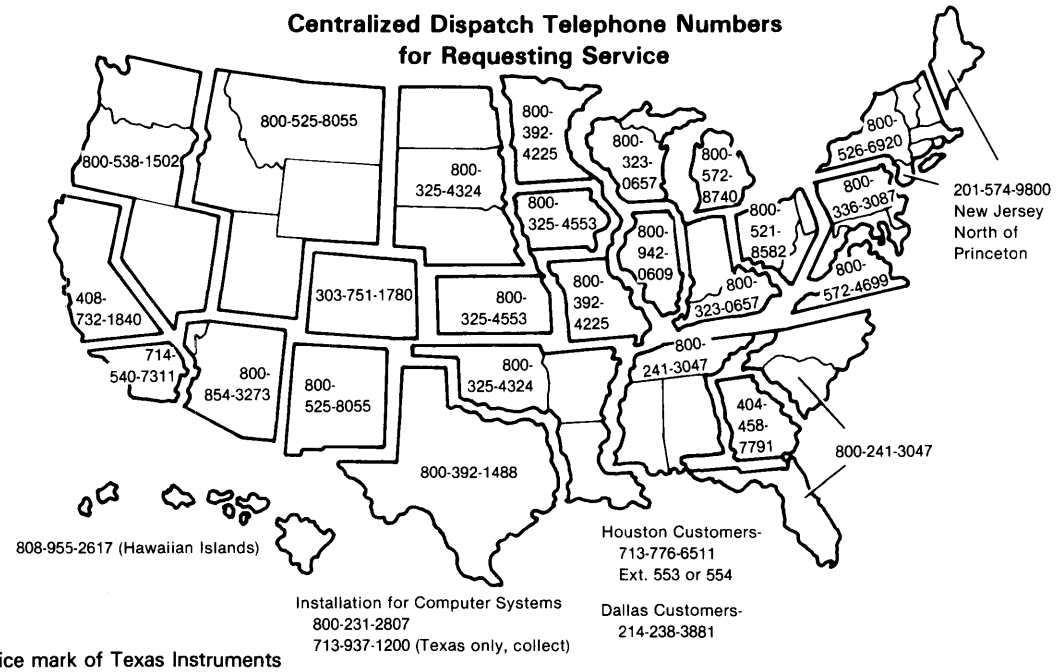
8585 Commerce Drive, Suite 518
Houston, Texas 77036
(713) 776-6531
(713) 776-6553*

Virginia
1745 Jefferson Davis Highway
Crystal Square 4, Suite 600
Arlington, Virginia 22202
(703) 553-2200

Wisconsin
205 Bishops Way
Suite 214
Brookfield, Wisconsin 53005
(414) 784-1323

TI-CARE*

Centralized Dispatch Telephone Numbers for Requesting Service



*Service mark of Texas Instruments

The **TI Customer Support Line** is available to answer our customers' complex technical questions. The extensive experience of a selected group of TI senior engineers and systems analysts is made available directly to our customers. The **TI Customer Support Line** telephone number is **(512) 250-7407.**

TEXAS INSTRUMENTS
INCORPORATED
DIGITAL SYSTEMS GROUP
POST OFFICE BOX 2909 AUSTIN, TEXAS