

SC21-7658-1

File No. S34-33

IBM System/34

Sort

Reference Manual

Program Number 5726-UT1



SC21-7658-1

File No. S34-33

IBM System/34

Sort

Reference Manual

Program Number 5726-UT1

Second Edition (July 1978)

This is a major revision of, and obsoletes, SC21-7658-0 and Technical Newsletter SN21-7911. Changes or additions to the text and illustrations are indicated by a vertical line to the left of the change or addition.

This edition applies to release 2, modification 0 of IBM System/34 Utilities Program Product (Program 5726-UT1), and to all subsequent releases and modification levels until otherwise indicated in new editions or technical newsletters.

Changes are periodically made to the information herein; before using this publication, refer to the latest *IBM System/34 Bibliography*, GH30-0231, for the editions that are applicable and current.

Use this publication only for the purposes stated in the *Preface*.

Publications are not stocked at the address below. Requests for copies of IBM publications and for technical information about the system should be made to your IBM representative or to the branch office serving your locality.

This publication could contain technical inaccuracies or typographical errors. Use the Reader's Comment Form at the back of this publication to make comments about this publication. If the form has been removed, address your comments to IBM Corporation, Publications, Department 245, Rochester, Minnesota 55901. Comments become the property of IBM.

This *Sort Reference Manual* is written to familiarize readers with the types of sort that can be run on the System/34, and to serve as a reference for the programmer who must define and run sort programs.

This manual describes how to:

- Identify and select the type of sort needed
- Complete sequence specifications sheets
- Use the SORT command
- Supply OCL and sequence specifications when using your procedures
- Optimize sort run time

This manual is not intended to teach an inexperienced sort programmer how to code sort programs. This manual is not a substitute for sort portions of System/34 classes offered by IBM or for equivalent sort training.

The primary audience for the *Sort Reference Manual* is the account programmer. The secondary audience includes third-party programmers and IBM field support personnel.

How This Manual is Organized

This manual has seven chapters, five appendixes, a glossary, and an index. The topics described in the chapters and appendixes are:

Chapter	Topics
1	Sort functions How sort works Overview of sequence specifications
2	Header specification
3	Record type specifications
4	Field specifications
5	How to run sort
6	Performance considerations
7	Sample sort jobs
Appendix	Topics
A	Calculating the sizes of files for sort
B	Collating sequence
C	Header, record type, and field specification column summaries
D	Integer (binary) and real number sorting
E	Sort printed messages

System Requirements

Refer to the *IBM System/34 Planning Guide*, GC21-5154, for a list of system requirements.

Prerequisite System/34 Publications

- *IBM System/34 Introduction*, GC21-5153
- *IBM System/34 Planning Guide*, GC21-5154
- *IBM System/34 System Support Reference Manual*, SC21-5155

Sort Coding Material

- *Sequence Specifications*, GX21-9089

Related System/34 Publications

- *IBM System/34 Operator's Guide*, SC21-5158
- *IBM System/34 Displayed Messages Guide*, SC21-5159
- *IBM System/34 Source Entry Utility Reference Manual*, SC21-7657
- *IBM System/34 Program Product Installation and Modification Reference Manual*, SC21-7689
- *IBM System/34 Master Index*, SC21-7739

The *Publications Summary* of the *IBM System/34 Introduction*, GC21-5153, contains a complete list and brief description of all available System/34 publications.

CHAPTER 1. OVERVIEW OF SORT	1-1	CHAPTER 4. FIELD SPECIFICATIONS	4-1
Sort Functions	1-1	Field Specification Column Summary	4-2
Types of Sort: Addrout, Tagalong, and Summary Tagalong	1-2	Column Descriptions	4-3
Addrout Sort	1-2	Columns 1-2 (Page Number) and 3-5 (Line Number)	4-3
Tagalong Sort	1-3	Column 6 (Line Type)	4-3
Summary Tagalong Sort	1-4	Column 7 (Field Type or Comments)	4-3
How Sort Works.	1-5	Column 8 (C/Z/D/P/U/V)	4-6
Sort Sequence Specifications	1-6	Columns 7-8	4-9
Sort Execution	1-7	Columns 9-16 (Field Location)	4-10
Sort Operation	1-7	Column 17 (Conditionally Forced Character)	4-11
Overview of Sequence Specifications	1-8	Column 18 (Substitute Character)	4-11
Order of Sequence Specifications	1-8	Column 19 (Specifying Replacements for More Than One Character)	4-13
Specification Columns to Consider for Addrout Sort (SORTA)	1-9	Examples of Using Forced Control Fields	4-13
Specification Columns to Consider for Tagalong Sort (SORTR)	1-10	Columns 20-22 (Overflow Field Length)	4-19
Specification Columns to Consider for Summary Tagalong Sort (SORTRS)	1-11	Example of Summary Overflow Indicator Field.	4-19
CHAPTER 2. HEADER SPECIFICATION	2-1	Columns 23-39 (Reserved)	4-20
Header Specification Column Summary	2-1	Columns 40-72 (Comments)	4-20
Column Descriptions	2-2	CHAPTER 5. HOW TO RUN SORT	5-1
Columns 1-2 (Page Number) and 3-5 (Line Number)	2-2	File Information	5-1
Column 6 (Line Type)	2-2	File Characteristics	5-1
Columns 7-12 (Job)	2-2	File Programming Considerations	5-2
Columns 13-17 (Largest Sum [in bytes] of Control Field Lengths for Any Record Type)	2-2	Running Sort by Using the SORT Command	5-3
Column 18 (Ascending or Descending Sequence)	2-2	SORT Command Parameters	5-3
Column 26 (Collating Sequence)	2-2	SORT Command Keying	5-3
Column 27 (Print Option)	2-3	SORT Command Limitations	5-4
Column 28 (Output Option for Tagalong Sorts)	2-3	Running SORT by Using Your Own Sort Procedures	5-5
Columns 29-32 (Output Record Length [in bytes] for Tagalong Sorts)	2-3	File Information Needed When Using Your Own Sort Procedures	5-5
Column 34 (Non-Verify Option)	2-4	Source Member Information Needed When Using Your Own Sort Procedures	5-6
Column 36 (Null Output—Bypass Halt for Message SORT—7724)	2-4	Executing Sort Procedures Stored on Disk.	5-6
Columns 40-72 (Job Description)	2-4	Entering Sort Procedures Through a Display Station Keyboard	5-7
CHAPTER 3. RECORD TYPE SPECIFICATIONS	3-1	CHAPTER 6. PERFORMANCE CONSIDERATIONS	6-1
Record Type Specification Column Summary	3-1	Number of Records to be Sorted	6-1
Column Descriptions	3-2	Record Size	6-1
Columns 1-2 (Page Number) and 3-5 (Line Number)	3-2	Number of Sequence Specifications	6-1
Column 6 (Line Type)	3-2	Alternate Collating Sequence	6-1
Column 7 (Continuation or Comments)	3-4	Disk Location of Files	6-1
Column 8 (C/Z/D/P/U)	3-5	File Placement On a Single Disk System	6-2
Columns 9-16 (Factor 1 Field Location)	3-9	File Placement On a Multiple Disk System.	6-2
Columns 17-18 (Relation)	3-10	Automatic Work File Allocation	6-3
Column 19 (Field, Constant, or Keyword)	3-10	Single Disk System	6-3
Columns 20-27 (Factor 2 Field Location)	3-10	Multiple Disk System	6-3
Columns 20-39 (Factor 2 Constant)	3-11	Order of Records in the Input File	6-3
Columns 20-39 (Factor 2 Keyword)	3-13	Non-Verify Option	6-3
Columns 40-72 (Comments)	3-13	Region Size	6-3
		System Environment in Which a Sort Job Executes	6-3

CHAPTER 7. SAMPLE SORT JOBS.	7-1
Sample Job 1: Producing a File of Inventory Information for	
Reordering Stock Items.	7-3
Header Specification	7-5
Record Type Specifications	7-5
Field Specifications.	7-5
Sample Job 2: Producing a Relative Record Address File for	
an Entire Inventory File	7-7
Header Specification	7-8
Record Type Specifications	7-8
Field Specifications.	7-8
Sample Job 3: Producing a File of Inventory Information	
About Certain Items	7-9
Header Specification	7-10
Record Type Specifications	7-10
Field Specifications.	7-11
Sample Job 4: Producing a History File of Transaction	
Information	7-12
Header Specification	7-14
Record Type Specifications	7-14
Field Specifications.	7-14
Sample Job 5: Producing a List of Inventory Items	
Requiring Adjustments	7-17
Header Specification	7-18
Record Type Specifications	7-18
Field Specifications.	7-18
Sample Job 6: Producing a Summary History File of Issues	
by Customer Number	7-19
Header Specification	7-20
Record Type Specifications	7-20
Field Specifications.	7-20
Sample Job 7: Sorting Summary History File by Descending	
Total Dollars Issued (Negative Values Possible)	7-22
Header Specification	7-24
Record Type Specifications	7-24
Field Specifications.	7-24
 APPENDIX A. CALCULATING THE SIZES OF FILES	
FOR SORT.	A-1
Input File	A-1
Output File	A-1
Work File	A-1
Work Record Length	A-1

APPENDIX B. COLLATING SEQUENCEB-1
Standard Collating SequenceB-3
Alternate Collating SequenceB-3
Coding RulesB-3
Input Order When ALTSEQ Statements Are UsedB-3
Programming ConsiderationsB-3
Effect of ALTSEQ Statements on Other CodingB-4
Sample ALTSEQ StatementsB-4
Programming Aids	B-4
Translation Table and Alternate Collating Sequence	
Coding Sheet	B-5

APPENDIX C. HEADER, RECORD TYPE, AND FIELD	
SPECIFICATION COLUMN SUMMARIESC-1
Header Specification Column SummaryC-1
Record Type Specification Column Summary.C-2
Field Specification Column SummaryC-3

APPENDIX D. INTEGER (BINARY) AND REAL NUMBER	
SORTING	D-1
Sample Sort Job 1. Sorting on a Control Field that Contains	
Either a Positive Integer or a Positive Real Number	D-1
Header Specification	D-2
Record Type Specifications	D-2
Field Specifications	D-2
Sample Sort Job 2. Sorting on a Control Field that Contains	
Only a Positive or Negative Integer	D-3
Header Specification	D-5
Record Type Specifications	D-5
Field Specifications	D-5
Sample Sort Job 3. Sorting on a Control Field that Contains	
Only a Positive or Negative Real Number	D-6
Header Specification	D-8
Record Type Specifications	D-8
Field Specifications	D-8
Sample Sort Job 4. Record Selection Based on Binary Values .	D-9
Type 1 Include Sequence	D-9
Type 2 Include Sequence.	D-9
Sample Sort Job 4 Description.	D-10
Header Specification	D-12
Record Type Specifications	D-12
Field Specifications	D-12

APPENDIX E. SORT PRINTED MESSAGES.	E-1
---	------------

GLOSSARY	F-1
---------------------------	------------

INDEX	X-1
------------------------	------------

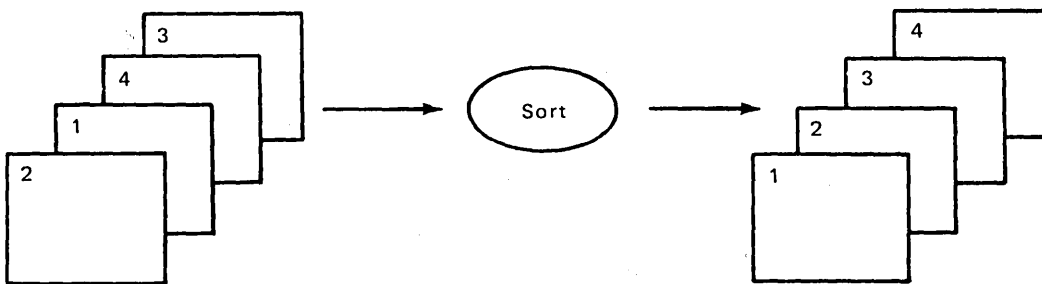
The sort program is part of the System/34 Utilities Program product. You can use the sort program to rearrange, drop, and reformat records in your files.

Multiple copies of the sort program can run concurrently on System/34.

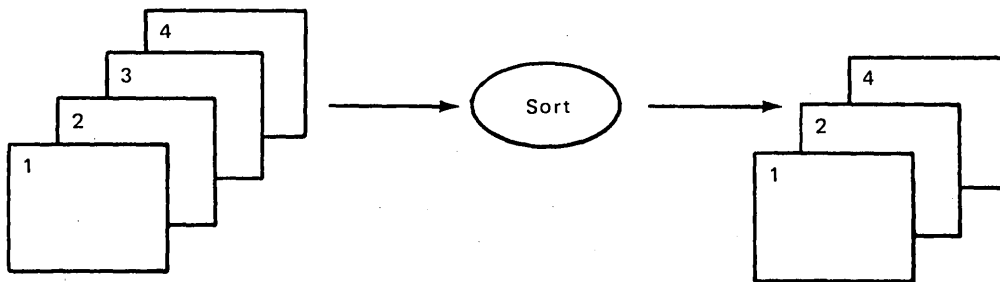
SORT FUNCTIONS

The three basic functions of sort are:

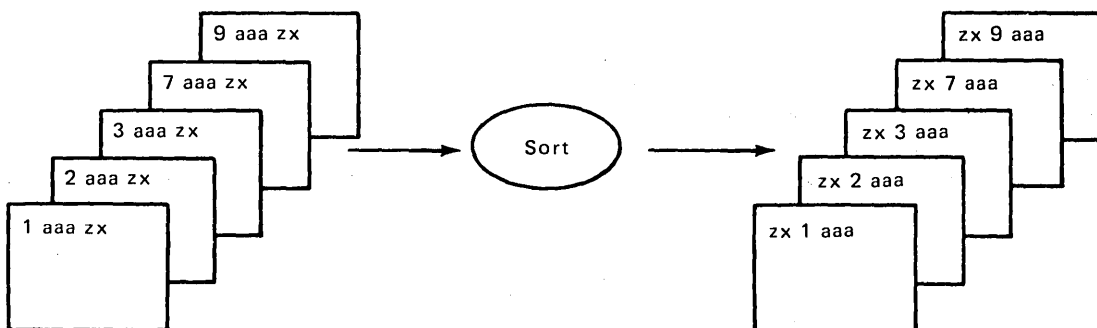
- To rearrange records in a file



- To drop records from a file



- To reformat records in a file

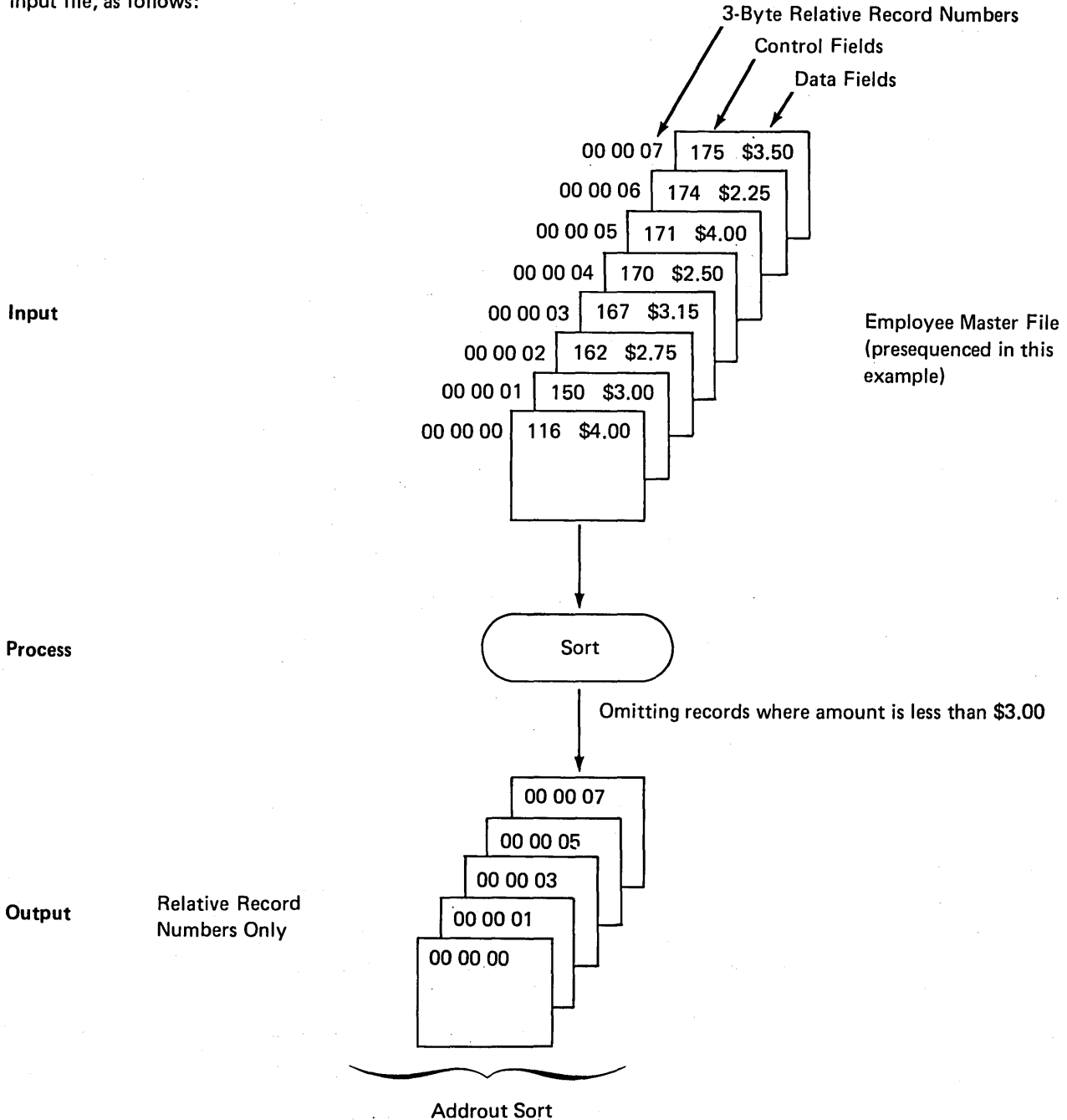


TYPES OF SORT: ADDRROUT, TAGALONG, AND SUMMARY TAGALONG

The three types of sort jobs are: *addrout* (address out); *tagalong* (specified data fields can be included with control fields in the sorted records); and *summary tagalong* (a total or totals can be accumulated in the sorted records).

Addrout Sort

The output from an addrout sort job consists of 3-byte relative record numbers of some or all of the records in the input file, as follows:

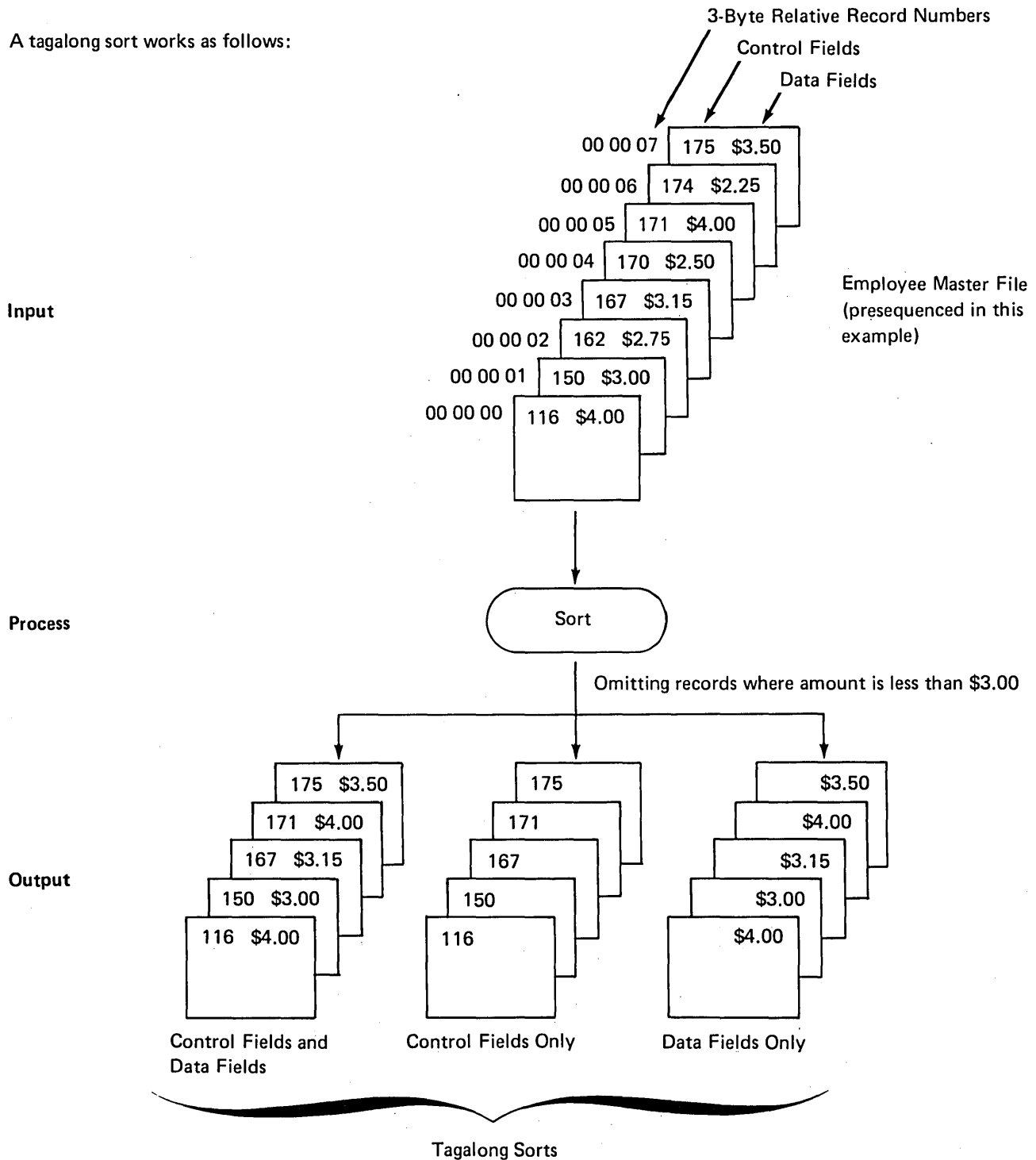


Tagalong Sort

The output from a tagalong sort is a file of sorted records that can contain:

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

A tagalong sort works as follows:



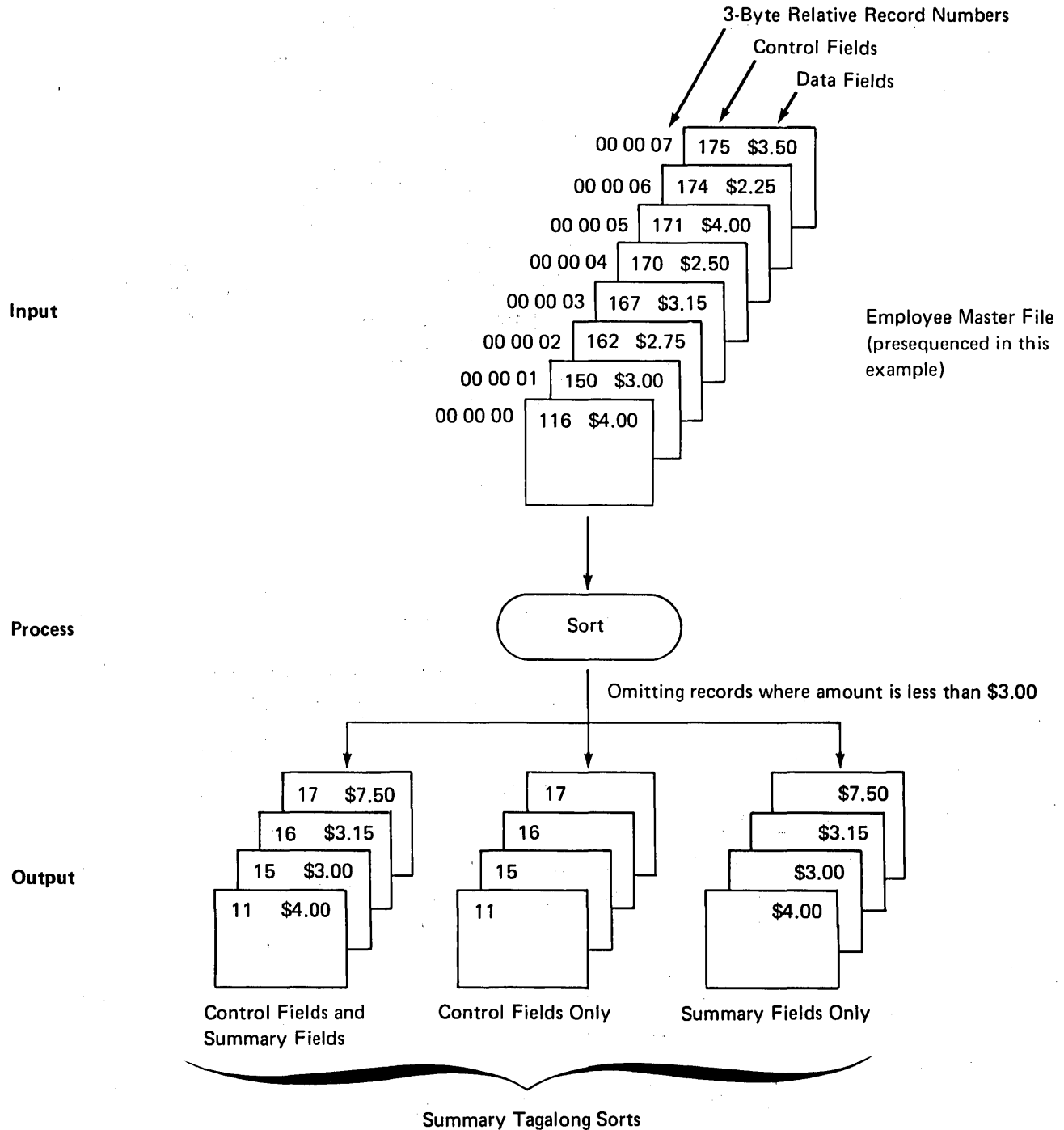
Summary Tagalong Sort

The output from a summary tagalong sort is a file of sorted summary records that usually contains:

- Control fields only
- Summary fields only
- Control fields and summary fields

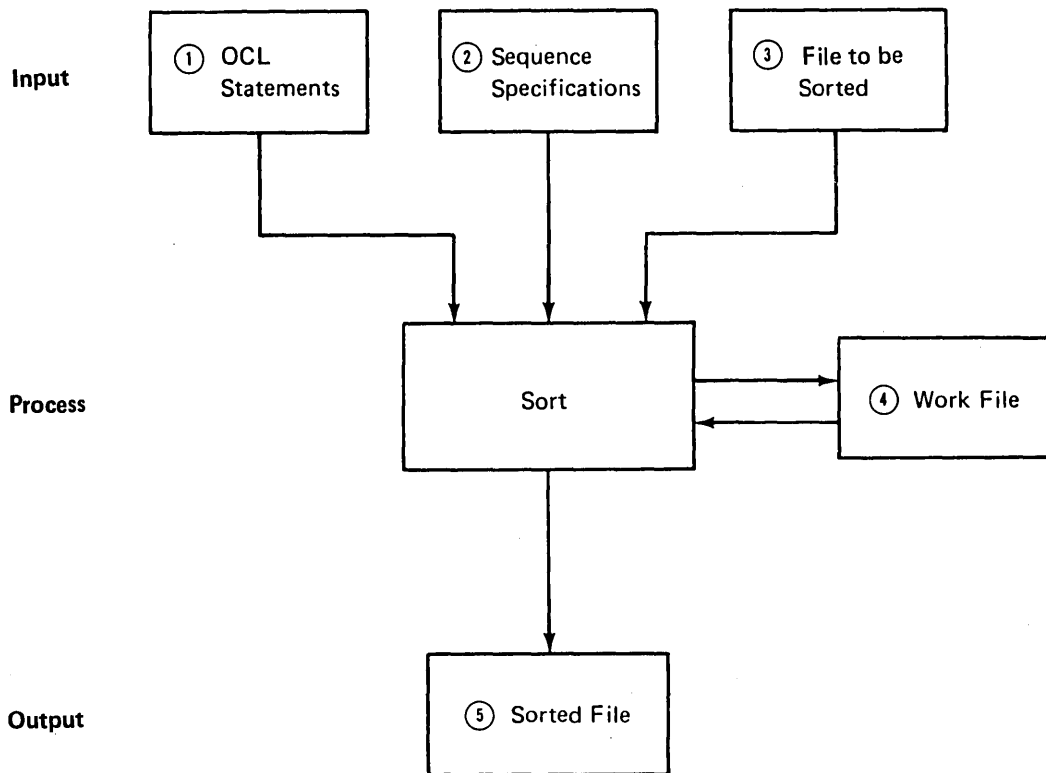
Summary sort output can also include data fields. However, you cannot determine which input record's data field(s) will appear in the summary record.

A summary tagalong sort that sorts only on the first two digits of the control field works as follows (note the summarizing that occurs in the data fields that have a control field of 17):



HOW SORT WORKS

The input needed to run a sort job is: operation control language (OCL) statements, sequence specifications, and an input file to be sorted. Sort works as follows:



① *OCL statements* are your instructions to the system.

② *Sequence specifications* are your instructions to the sort program.

③ *File to be sorted* can be an indexed, a sequential, or a direct disk file with fixed length records, but sort processes each sequentially.

④ *Work file* is a work area on disk where sort writes all of the records you want to sort. The sort program can then work with the records in the work area without disturbing the input file.

⑤ *Sorted files* can contain:

- The relative record numbers of the records in the input file.
- Parts, or all, of the records contained in the input file.
- Summarized fields for each record type in the input file.

Note: Unformatted sequential files created by FORTRAN do not have fixed length records. Therefore, they *should not* be used as input to the sort program.

Sort Execution

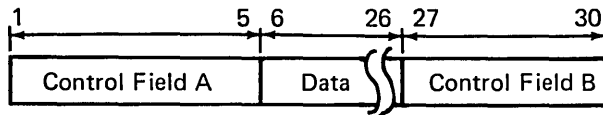
You can use the SORT command statement to execute certain sort jobs. See *Running Sort by Using the SORT Command* in Chapter 5 for a detailed description of how the SORT command works.

You can also create your own procedure member to execute sort. This procedure member can be stored in the system library or in a user library, or the procedure can be entered directly from a display station keyboard. The contents of a procedure member that contains your OCL are explained in Chapter 5.

Sort Operation

Sort works with the records in your file as follows:

1. Sort reads a record from the input file.



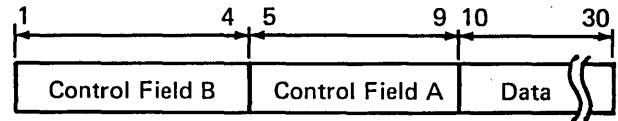
Note: Control fields are the fields you want the sort program to use to sort the records. See *Column 7* under *Column Descriptions* in Chapter 4 for a detailed discussion of control fields.

2. The sort program checks your record type specifications to see whether the record is one you want to sort. (Often you may not want to sort all of the records in the file.)
3. If the record is one you want to sort, the sort program builds a work record, formatting it according to your field specifications. (The format of the work record is important because it controls the format of the output record.)

Assume that your field specifications say:

- a. Put the contents of positions 27 through 30 in the input record into positions 1 through 4 of the work record.
- b. Put the contents of positions 1 through 5 in the input record into positions 5 through 9 of the work record.
- c. Put the contents of positions 6 through 26 in the input record into positions 10 through 30 of the work record.

The work record sort builds would look like this:



See *Column 7* under *Column Descriptions* in Chapter 4 for a more detailed discussion of the work record.

If you do not drop control fields, they always precede the data fields in the work and output records. (See *Column 28* under *Column Descriptions* in Chapter 2 for a discussion of how to drop control fields.)

4. The sort program writes all of the input records specified by your record type specifications into the work file.
5. The sort program checks your header and field specifications to see how you want to arrange the records in the output file. (You arrange the records in ascending or descending order by control fields.)
6. The sort program writes the records into the output file in the order you have specified.

OVERVIEW OF SEQUENCE SPECIFICATIONS

Order of Sequence Specifications

The normal order of sequence specifications is:

1. Header line
2. Record type lines
3. Field lines

This order can vary, however, depending on how many records you want to sort and the format of those records. (Format refers to the locations, lengths, and types of fields in a record.) If, for example, you want to sort all of the records in a file—and they all have the same format—you do not have to fill out record type specifications. Not filling out record type specifications is often referred to as an implied include-all.)

Number of Records to be Sorted	Format of Records to be Sorted	Order of Sequence Specifications																					
All of the records in the file (implied include-all)	All the same format	1. Header line 2. Field lines																					
Some of the records in the file	All the same format	1. Header line 2. Record type lines 3. Field lines																					
All or some of the records in the file	Several different formats	<table style="border: none; width: 100%;"> <tr> <td style="border: none; width: 30%;">1. Header line</td> <td style="border: none; width: 10%; text-align: center;">}</td> <td style="border: none; width: 60%;">For first type of record format</td> </tr> <tr> <td style="border: none;">2. Record type lines</td> <td style="border: none;"></td> <td style="border: none;"></td> </tr> <tr> <td style="border: none;">3. Field lines</td> <td style="border: none;"></td> <td style="border: none;"></td> </tr> <tr> <td style="border: none;">4. Record type lines</td> <td style="border: none; text-align: center;">}</td> <td style="border: none;">For second type of record format</td> </tr> <tr> <td style="border: none;">5. Field lines</td> <td style="border: none;"></td> <td style="border: none;"></td> </tr> <tr> <td style="border: none;">6. Record type lines</td> <td style="border: none; text-align: center;">}</td> <td style="border: none;">One set for each additional type of record format¹</td> </tr> <tr> <td style="border: none;">7. Field lines</td> <td style="border: none;"></td> <td style="border: none;"></td> </tr> </table>	1. Header line	}	For first type of record format	2. Record type lines			3. Field lines			4. Record type lines	}	For second type of record format	5. Field lines			6. Record type lines	}	One set for each additional type of record format ¹	7. Field lines		
1. Header line	}	For first type of record format																					
2. Record type lines																							
3. Field lines																							
4. Record type lines	}	For second type of record format																					
5. Field lines																							
6. Record type lines	}	One set for each additional type of record format ¹																					
7. Field lines																							
<p>¹This does not mean that the records in the file must be grouped by format type. The sequence specifications must be grouped (one set of record type and field lines for each format type). The records themselves can be in any order.</p>																							

For detailed information on how to include or omit certain records when sorting a file, see *Column 6* under *Column Descriptions* in Chapter 3.

Specification Columns to Consider for Tagalong Sort (SORTR)

The shaded columns are the ones you must consider for a tagalong sort job.

Type of Sort

Control Field and Sequence Information

Page Sequence

International Business Machines Corporation
SEQUENCE SPECIFICATIONS
Header

GX21-9089
Printed in U.S.A.
Program Identification 75 76 77 78 79 80

Line	Job	Card Match	Sequence (A/D/S)	Stacker Select	Number (I/N)	Alternate Coll. Seq. (S)	Print Option	Output Option (X)	Output Record Length	Checkpoint Option (C)	Non-Verify Option (N)	Reserved	Null Output (N)	Storage Size to Execute	Length Control (C)	Record Length	Job Description
3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
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	45	46	47	48	49	50	51	52	53	54	55
56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73

Record Type

Line	Factor 1	Rel.	Factor 2 (Field, Constant, or Keyword)	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

Field

Line	Forced	Overflow Field Length	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	40	41	42

Format for Sorted File

Comparison of an Input Record Field and a Constant

Comparison of Two Input Record Fields

Comparison of an Input Record Field and the Program Date.

Definition of Normal Control Fields

Definition of Opposite Control Fields

Definition of Forced Control Fields

Definition of Data Fields

Specification Columns to Consider for Summary Tagalong Sort (SORTRS)

The shaded columns are the ones you must consider for a summary tagalong sort job.

Type of Sort

Control Field and Sequence Information

Page Sequence

IBM

International Business Machines Corporation
SEQUENCE SPECIFICATIONS
Header

GX21-9089
 Printed in U.S.A.
 75 76 77 78 79 80

Line	Job	Card Match	Stacker Select	Sequence (A/D/S)	U	S	M	P	P	S	O	O	2	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	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72
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	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72														
Format for Sorted File																																																																																			

Record Type

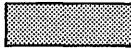
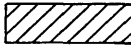
Line	Factor 1	Rel.	Factor 2 (Field, Constant, or Keyword)	Comments																																																																	
3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72
Comparison of an Input Record Field and a Constant																																																																					
Comparison of Two Input Record Fields																																																																					
Comparison of an Input Record Field and the Program Date.																																																																					

Line	Forced	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	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72
Definition of Normal Control Fields																																																																					
Definition of Opposite Control Fields																																																																					
Definition of Forced Control Fields																																																																					
Definition of Data Fields																																																																					
Definition of Summary Data Fields																																																																					

Only one header specification line is used for each sort job. Possible column entries, summarized on this page and in Appendix C for quick reference, are explained in detail in this chapter.

HEADER SPECIFICATION COLUMN SUMMARY

Column Summary		
Columns	Entries	Explanation
1-2	00-99	Page number.
3-5	000	Header line number.
6	H	Header line identification.
7-12	SORTA	Addrout sort job.
	SORTR	Tagalong sort job.
	SORTRS	Summary tagalong sort job.
13-17	1-256	Largest sum (in bytes) of control field lengths for any record type.
18	A	Records in sorted file to be in ascending order by control fields.
	D	Records in sorted file to be in descending order by control fields.
	S	Not used in System/34 sort jobs.
19-25	Blank	Not used in System/34 sort jobs.
26	Blank	Use standard collating sequence in compare operations.
	S	Use an alternate collating sequence in compare operations. ALTSEQ statements will define the collating sequence to be used.
27	0 or blank	Print: Sequence specifications Diagnostic messages Program-status messages Display: Action messages Displayed messages
	1	Print: Program-status messages Display: Action messages Displayed messages
	2	Print: Action messages only Display: Displayed messages
	3	Display: Displayed messages
28	Blank	Keep control fields in output records in tagalong sort jobs.
	X	Drop control fields from output records in tagalong sort jobs.
29-32	1-4096	Length of output records in tagalong sort jobs.
33	Blank	Not used in System/34 sort jobs.
34	N	Data written on the work file will not be verified.
	Blank or character other than N	Data written on the work file will be verified.
35	Blank	Reserved for system use.
36	Blank	Operator message SORT-7724 will be issued.
	N	Operator message SORT-7724 will not be issued.
37-39	Blank	Not used in System/34 sort jobs.
40-72	Any characters	Comments.

	Columns that must be filled in for all sort jobs.
	Columns that must be filled in for tagalong sort jobs.

COLUMN DESCRIPTIONS

Columns 1-2 (Page Number) and 3-5 (Line Number)

Page number (columns 1 and 2) and line number (columns 3 through 5) form a five-digit sequence number. As the sort program reads sequence specifications, it checks the sequence numbers to make sure they are not in descending order. If the numbers are in descending order (if page 02 specifications come before page 01 specifications, for example) and specifications are being issued, the sort program places a warning (the letter S) next to the line (the S stands for sequence error). After issuing a warning, the sort program continues reading the rest of the specification lines, then issues a message and waits for further instructions from the operator. The operator can either continue or end the job.

Because page number applies to all lines on a page, columns 1 and 2 appear only once, in the upper right corner of the page. You number the pages in ascending order.

The line number of the header line is always 000 and is preprinted on the coding sheet.

Column 6 (Line Type)

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

Columns 7-12 (Job)

Columns 7 through 12 tell the sort program what type of sort job you want to do. SORTA means addrout sort. SORTR means tagalong sort. SORTRS means summary tagalong sort.

Columns 13-17 (Largest Sum [in bytes] of Control Field Lengths for Any Record Type)

To calculate this entry:

1. Add together the lengths of the control fields (N, O, and F in column 7 of field specifications) for each type of input record.
2. Enter the largest of these totals in columns 13 through 17. (The total must not exceed 256.)

For more information, see *Control Fields, Column 7* under *Column Descriptions* in Chapter 4.

Column 18 (Ascending or Descending Sequence)

Control fields control the sequence of records in the sorted output file. Your column 18 entry indicates the sequence in which you want the records sorted.

Column 18 Entry	Sequence
A	Ascending sequence by control field
D	Descending sequence by control field
S	Not used in System/34 sort jobs

Column 26 (Collating Sequence)

Column 26 specifies the collating sequence you want the sort program to use in compare operations. (Compare operations determine whether one character is equal to, greater than, or less than another character.)

Standard Collating Sequence

No entry in column 26 tells the sort program to use the standard collating sequence. There are slight variations in the standard collating sequence, depending on whether you are using both the zone and digit portions of the characters in your records, the zone portions only, or the digit portions only. Appendix B shows the complete collating sequence of each situation.

Alternate Collating Sequence

An S in column 26 tells the sort program you want to change the standard collating sequence. To do this you must supply ALTSEQ statements immediately following the header specifications. Appendix B explains how to code ALTSEQ statements.)

Note: Do not use packed or zoned factors in an include or omit record type specification (P or U in column 8) if you specify an alternate collating sequence.

Column 27 (Print Option)

The sort program can issue:

- Sequence specification lines
- Diagnostic messages (S-, T-, and W-type messages) for any errors in sequence specifications
- Program-status messages (I-type messages) to identify various stages of the job
- Action messages (A-type messages, followed by displayed messages) to identify circumstances requiring attention before the job can continue
- Displayed messages that appear on the display screen and that are included and explained in the *Displayed Messages Guide*

Note: Sort printed messages are included in Appendix E of this manual.

Column 27 indicates which of the preceding information you want the sort program to issue during a job.

Column 27

Entry

Program Issues

0 or Blank

Sequence specifications
Diagnostic messages
Program-status messages
Action messages
Displayed messages

1

Program-status messages
Action messages
Displayed messages

2

Action messages
Displayed messages

3

Displayed messages

Column 28 (Output Option for Tagalong Sorts)

Column 28 applies to tagalong sort (SORTR, SORTRS) jobs only. It indicates whether or not you want the sort program to drop control fields from output records after the records are sorted. A blank in column 28 means keep the control fields; X means drop them.

Considerations for Dropping Control Fields

Control fields are normally dropped when you use opposite control fields or an alternate collating sequence. In these two cases, the sort program changes the control information (during the sorting process) so that it is meaningless.

Using Fields Both as Control and Data Fields

If you are using opposite control fields or an alternate collating sequence and you want to keep the control information in a meaningful form in the output records, describe the fields twice: once as control fields and once as data fields. Data fields are not involved in the sorting process and are not changed by the sort program.

Columns 29-32 (Output Record Length [in bytes] for Tagalong Sorts)

Columns 29 through 32 apply to tagalong sort (SORTR, SORTRS) jobs only. The entry in these columns tells the sort program the length of records in the final sorted file.

If you do not drop control fields, the length includes both control and data fields. If you drop control fields, the length includes only data fields. (The record length must not exceed 4096.)

Calculating Output Record Length When Dropping Control Fields

For each type of record, total the lengths of all the data fields you are including in the job. Select the largest total. Enter this number in columns 29 through 32.

Calculating Output Record Length When Not Dropping Control Fields

For each type of record, total the lengths of the data fields. Select the largest total, add this total to the number in columns 13 through 17, and put the sum in columns 29 through 32.

Column 34 (Non-Verify Option)

Column 34 applies to all sort runs. If an N is placed in this column of the sort header statement, none of the data written on the work file will be verified. A blank or a character other than N will cause the data written on the work file to be verified.

Note: If you use the non-verify option (to minimize run time) when your output file is to overlay the input file, the input file might be destroyed if a terminal error occurs before end of job. Therefore, if the input file cannot be easily recreated, you should have a duplicate copy for backup. The input file can be destroyed under identical circumstances whether the non-verify option is used or not. However, the chance for input file destruction is greater when you use the non-verify option; therefore, you should use this option with care.

Column 36 (Null Output—Bypass Halt for Message SORT—7724)

Column 36 applies to all sort runs. A blank can cause the sort program to issue message SORT—7724 NO INPUT RECORDS INCLUDED. The operator then can select either option 0, which creates an empty output file, or option 3, which cancels the job without creating an output file. An N in column 36 causes the sort program to create an empty output file without issuing message SORT—7724.

Columns 40-72 (Job Description)


Columns 40 through 72 are for your comments. You can use any characters you want in these columns. If the sort program prints specification lines (the column 27 entry is a zero or blank), the comments you include in these columns are printed. Comments have no effect on the program.

Record type specifications tell the sort program which of the records in a file are to be sorted. Remember, if all of the records in a file are to be sorted, and they all have the same format, record type specifications are not required.

Possible column entries, summarized on this page and in Appendix C for quick reference, are explained in detail in this chapter.

RECORD TYPE SPECIFICATION COLUMN SUMMARY

Column Summary		
Columns	Entries	Explanation
1-2	00-99	Page number.
3-5	01x-06x	Line number. You can leave column 5 (x) blank, or enter any value to keep the specifications in ascending order.
6	I	Include line.
	O	Omit line.
7	Blank	This is the first line of a set of I or O record type lines.
	A	AND lines. These specifications continue the definition of the record described on the preceding line.
	O	OR line. These specifications define a different type of record than the one on the previous line.
	*	Comment line.
8	C	Use both zone and digit portions of characters.
	Z	Use only zone portion of 1-character field.
	D	Use only digit portions of characters.
	P	Signed packed decimal data.
	U	Signed zoned decimal data.
9-12	1-4096	The input record position in which the factor 1 field begins (blank if field is only one position long).
13-16	1-4096	The input record position in which the factor 1 field ends.
17-18	EQ	Factor 1 must equal factor 2.
	NE	Factor 1 must not equal factor 2.
	LT	Factor 1 must be less than factor 2.
	GT	Factor 1 must be greater than factor 2.
	LE	Factor 1 must be less than or equal to factor 2.
	GE	Factor 1 must be greater than or equal to factor 2.
19	F	Factor 2 is another field in the same input record.
	C	Factor 2 is a constant.
	K	Factor 2 is a keyword.
20-23	1-4096	The input record position in which the factor 2 field begins (blank if field is only one position long).
24-27	1-4096	The input record position in which the factor 2 field ends.
28-39	Any characters	The factor 2 constant
40-72	Any characters	Comments.

 Columns that must be filled in.

COLUMN DESCRIPTIONS

Columns 1-2 (Page Number) and 3-5 (Line Number)

Page number (columns 1 and 2) and line number (columns 3 through 5) form a five-digit sequence number. As the sort program reads sequence specifications, it checks the sequence numbers to make sure they are not in descending order. If the numbers are in descending order (if page 02 specifications come before page 01 specifications, for example) and specifications are being issued, the sort program places a warning (the letter S) next to the line. (The S stands for sequence error.) After issuing a warning, the sort program continues reading the rest of the specification lines, then issues a message and waits for further instructions from the operator. The operator can either continue or end the job.

Because page number applies to all lines on a page, columns 1 and 2 appear only once, in the upper right corner of the page. You number the pages in ascending order.

Record type line numbers are 01 through 06. The numbers in columns 3 and 4 are preprinted on the coding sheet. You can leave column 5 blank or enter any value to keep your specifications in ascending order. If you have more than six record type lines, use another coding sheet and start at line 01.

Out-of-Sequence Lines

Use column 5 when you want to insert a specification without renumbering the other specifications. For example, to insert a specification line between lines 01010 and 01020, you can number it 01015, code it, and then continue to fill out the rest of the sheet.

Be sure any lines that are out of sequence on your coding sheet are clearly marked. You can do this by writing a note in the margin of the page with an arrow pointing to where the insert belongs (*Sample Job 3* in Chapter 7).

Column 6 (Line Type)

Column 6 identifies the line type. An I in this column stands for either an include or an include-all line; an O stands for an omit line.

If the type of sort job you are running requires you to code record type lines, you must use include or include-all lines to describe the records you want sorted. Records not described in include lines are not sorted.

Include Line

Include lines identify records you want the sort program to sort by describing particular record fields.

Include-All Line

Include-all is a special form of include line; it has no record description (columns 7 through 39 are blank). An include-all line tells the sort program to sort all of the records that have not been described by any preceding include or omit line for the job. Records referred to by an include-all line must have the same field specifications.

Note: Only one include-all line can be used per job. If used, it must be the last record type line for that job.

Omit Line

Omit lines identify records you do *not* want the sort program to sort. Omit lines are not required, but can be helpful when you have many types of records you want the sort program to use and just a few you want omitted. Omit lines are normally followed by an include-all line, telling the sort program to sort all of the records that are not described by omit lines.

Sets

There are two types of sets: include sets and omit sets. An include set identifies one or more record types you want to include in your sort job. An omit set identifies one or more record types you want to omit from your sort job. (The records in any record type always have at least one characteristic in common—such as an X in position 5.)

Here are four rules to remember when you use include and omit sets:

1. All include sets must end with a field line.
2. Omit sets never have field lines.
3. Every omit set must be followed by an include set.
4. The last set must be an include set.

There are five types of include sets and three types of omit sets.

Include Sets ¹				
Type	Column 6	Column 7	Explanation	
Include AND lines	H, F, or O		Header line, field line, or omit line.	} Record type lines
	I	∅	New record type indicated by a blank in column 7.	
	I	A	Lines that describe the same record type (as the previous line) have an A in column 7.	
Include OR lines	H, F, or O		Header line, field line, or omit line.	} Record type lines
	I	∅	New record type indicated by a blank in column 7.	
	I	O	Lines that describe a different record type (than the previous line) have an O in column 7.	
Include AND and OR lines	H, F, or O		Header line, field line, or omit line.	} Record type lines
	I	∅	New record type indicated by a blank in column 7.	
	I	O	This line designates a record type that is different from, but has the same field lines as, the record type described in the previous line(s).	
	I	A	This line continues the same record type of a previous line or lines. This record type can be continued (IA), or a different record type can be started (IO), provided all record types have the same field lines. Record types with different field lines would have to start a new include set.	
Include only one record type (implied include-all)	H		Header line.	
	F		No record type lines. Field line(s).	
Include-all	H, F, or O		Header line, field line, or omit line.	} Record type lines
	I		This line tells the sort program to sort all of the records that have not been described by any preceding include and omit lines. Records referred to in this manner must have identical field specifications. Field line(s).	

¹ Every include set must end with field lines. An include set can be followed by another include set, an omit set or // END.

Note: Records not described in include sets will not be sorted.

Omit Sets ¹			
Type	Column 6	Column 7	Explanation
Omit AND lines (one record type)	H or F O O	∅ A	Header line or field line (last line of include set). New record type indicated by a blank in column 7. Lines that describe the same record type (as the previous line) have an A in column 7.
Omit OR lines (different record types)	H or F O O	∅ O	Header line or field line (last line of include set). New record type indicated by a blank in column 7. Lines that describe a different record type (than the previous line) have an O in column 7.
Omit AND and OR lines (different record types)	H or F O O O	∅ A O	Header line or field line (last line of include set). New record type indicated by a blank in column 7. Lines that describe the same record type (as the previous line) have an A in column 7. Lines that describe a different record type (than the previous line) have an O in column 7.

} Record type lines
} Record type lines
} Record type lines

¹ There are no field lines in omit sets. Each omit set must be followed by an include or an include-all set.

Guide to Using Include and Omit Sets

When to Use Include Sets: If you want to sort only a few records in a file, use an include set for each type of record you want to sort.

When to Use Omit Sets: If you want to sort all but a few records in a file, use omit sets followed by either an include set for each type of record you want to sort or an include-all set.

Mixing Include and Omit Sets: You can mix include and omit sets. But because the sort program processes the sets in the order they are coded, you must be particularly careful when you do this. For example, if you wanted to omit all records with a 2 in position 10 but sort those with a 2 in positions 10 and 15, you would have to specify the include set before the omit set. If you specified the omit set first, all the records you wanted to sort would be omitted from the job.

Column 7 (Continuation or Comments)

Column 7 indicates the line's relationship to the preceding line.

Column 7 Entry	Explanation
Blank	This line is the first of a set of include or omit lines. (The type of set is indicated by the column 6 entry: I for include or O for omit.)
A	This line is a continuation of the preceding line. The A stands for AND.
O	This line applies to a different record type than the preceding line, but the control field specifications for both are the same. The O stands for OR.
*	This line is a comment line. Comment lines do not affect program operation in any way. Their only purpose is to help you remember what you were doing in a certain section of coding. Comments are printed only if column 27 of the header line contains a zero or a blank.

Column 8 (C/Z/D/P/U)

Your column 8 entry tells the sort program how to interpret data in the factor 1 and factor 2 fields during compare operations. When the fields contain alphameric data, a C, Z, or D entry tells the sort program what portions of the characters to use. When the fields contain signed numeric data, a P or U entry tells the sort program whether the data is packed or zoned.

Type of Data	Column 8 Entry	Compare Operations	Maximum Field Length ¹
Alphameric	C	Use both zone and digit portions of the characters	256 characters
	Z	Use only the zone portion of the character	1 character
	D	Use only the digit portions of the characters	16 characters
Signed Numeric	P ²	The numeric data is packed	8 bytes or 15 digits and a sign
	U ²	The numeric data is zoned	16 digits

¹For both factor 1 and factor 2 fields, which must be the same length.
²Do not use packed or zoned factors in an include or omit record type specification (P or U in column 8) if you specify an alternate collating sequence (S in column 26 of the header line).

Significance of the Column 8 Entry

You use record type specifications to tell the sort program which records you want to sort. You do this by instructing the sort program to test each record by comparing the data in a specific field with the data in another field in the same record, with a constant, or with all or part of the program date. The data you are comparing is the factor 1 field; the data you are comparing it with (the contents of another field in the same record, a constant, or all or part of the program date) is the factor 2 field. The result of the comparison determines whether or not that record will be sorted. Since the sort program sees your data as nothing more than a series of electronic bits, you must tell the sort program how to interpret the data:

- If the data is alphameric, what part of the characters should be compared.
- If the data is signed numeric, is it in packed or zoned format.

Interpreting Data

EBCDIC characters are represented by a byte composed of 8 bits. Each character has two parts: a zone portion and digit portion. Some characters have identical zone portions; some have identical digit portions. No two characters have identical zone and digit portions.

Character	How It Looks Inside the Computer	
	Zone Portion	Digit Portion
*	0101	1100
1	1111	0001
2	1111	0010
3	1111	0011
K	1101	0010
?	0110	1111
P	1101	0111
Blank	0100	0000
0	1111	0000

¹ Notice that the digit portions of a zero and a blank are exactly the same.

If you instruct the sort program to use only the digit portions of characters (by putting a D in column 8), characters with identical digit portions will look alike and compare as equal. Likewise, if you instruct the sort program to use only the zone portion of characters (by putting a Z in column 8), characters with identical zone portions will look alike and compare as equal. So your column 8 entry is critical in ensuring that your compare operations produce the results you intend.

Suppose, for example, that you want to sort only records with a 2 in column 15 and a 2 in column 50. To get those records included in your sort, you would have to put a C in column 8. The C tells the sort program to use both the zone and digit portions of characters in its compare operations, and no other character has the same zone and digit portions as a 2.

If you put a D in column 8, you would also get the records with a 2 in column 15 and a 2 in column 50. However, you would also get records you did not want, because several characters have the same digit portion as a 2.

Alphameric Data

When the factor 1 and factor 2 fields contain alphameric data, the column 8 entry must specify one of the following:

1. The zone and digit portions of the characters (C entry)
2. Only the zone portion of the character (Z entry)
3. Only the digit portions of the characters (D entry)

Numeric Data

When the factor 1 and factor 2 fields contain numeric data, the column 8 entry must answer two questions:

1. Are the numbers signed or unsigned?
2. Are the numbers packed or zoned?

Format of Numeric Data		Column 8 Entry	Definition	Consideration ¹
Unsigned		D	Number does not have a sign.	Absolute values of the numbers are used. If a number has a sign, it is ignored. For example, -3 would be considered equal to +3.
Signed	Packed	P	Number always carries a sign. When the number is placed in main storage it has a digit portion only.	The sign controls the comparison. For example, -3 is less than 0 and +5 is more than -6.
	Zoned	U	Number always carries a sign. When the number is placed in main storage, it has both a zone and a digit portion.	The sign controls the comparison. For example, -3 is less than 0 and +5 is more than -6.

¹ Before comparing numeric data, the sort program automatically converts any leading blanks to zeros.

Signed Numbers: Signed numbers can be either positive or negative. The sign of a number is indicated by a 4-bit binary code.

Packed and Zoned Numbers: A zoned digit is represented by 8 bits; a packed digit is represented by 4 bits.

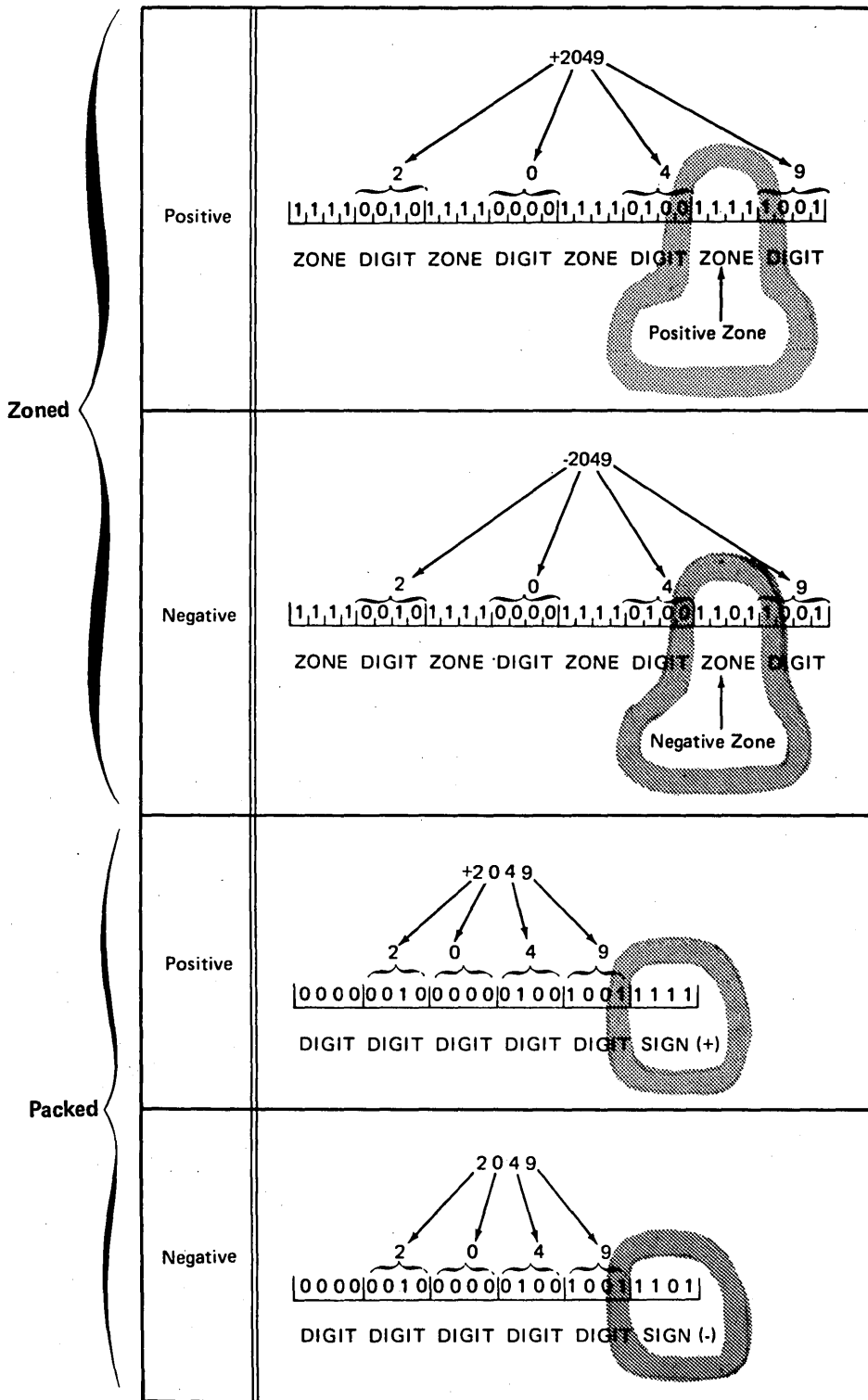
Sign	Binary Code
+	1111 ¹
+	1010
+	1100
+	1110
-	1101
-	1011

¹ Standard form. The sort program accepts all four forms of the plus sign. Before sorting the file, however, the sort program converts all plus signs to the standard form. If a main storage dump is printed, the plus sign will always be expressed as a hex F.

Zoned Digits			Packed Digits	
Digit	Binary Form		Digit	Binary Form
0	1111	0000	0	0000
1	1111	0001	1	0001
2	1111	0010	2	0010
3	1111	0011	3	0011
4	1111	0100	4	0100
5	1111	0101	5	0101
6	1111	0110	6	0110
7	1111	0111	7	0111
8	1111	1000	8	1000
9	1111	1001	9	1001

} Zone Portion
} Digit Portion
} Digit Portion Only

In zoned numbers, the 4-bit sign code replaces the zone portion of the last digit in the number. In packed numbers the sign code takes up the last 4 bits of the number.



Columns 9-16 (Factor 1 Field Location)

Factor 1 fields identify your records. (If all of your inventory records contain an I in column 2, for example, column 2 is a factor 1 field.) The sort program identifies records you want sorted by comparing factor 1 fields with other fields in the same record, with constants, or with all or part of the program date. Columns 9 through 16 identify the locations of the factor 1 fields in the records. If there is more than one factor 1 field for the records you are describing, you must:

- Describe each field in a separate record type line
- Put an A in column 7 of every line (except the first) to tell the sort program that all of the lines apply to the same record type

Columns 9 through 12 (From) identify where the factor 1 field begins in the record. Columns 13 through 16 (To) identify where the field ends.

Length of Factor 1 Fields

A factor 1 field can contain anywhere from 1 to 256 characters. No factor 1 field, however, can be longer than the length of the records you are working with. (For example, when you are working with 96-character records, the longest possible factor 1 field you can have is a 96-character field.) The length of factor 1 fields is also controlled by the column 8 entry.

Column 8	Maximum Factor 1 Field Length
C	256 characters ¹
Z	1 character
D	16 characters
P	8 characters ²
U	16 characters

¹When factor 2 is a constant, the length of the factor 1 field must not exceed 20 characters. When factor 2 is a keyword, the length of the factor 1 field must be 6 if the keyword is UDATE, and must be 2 if the keyword is UMONTH, UDAY, or UYEAR. (See *Columns 20-39* for more information.)

²Because the factor 1 field is packed, the field can actually represent 15 decimal digits and a sign.

Coding Rules

1. Entries must be right-justified: the From entry must end in column 12; the To entry must end in column 16.
2. To describe factor 1 fields that are only 1 character long, leave columns 9 through 12 (From) blank and enter the number of the record position that contains the character in columns 13 through 16 (To).

For example, here is the entry you would make to describe the position of a factor 1 field that consists of a B in column 2.

Record Type

Line	Type (U/O)	Factor 1	Rel.	Factor 2 (Field, Constant, or Keyword)
Number	1		EQ NE LT GT	← Keyword → ← Constant →
3		From To		Location
0 1		9 10 11 12 13 14 15 16	20 21 22 23	24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39
0 2			2EQCB	
0 3				
0 4				
0 5				
0 6				

Columns 17-18 (Relation)

The sort program identifies records you want to sort by comparing the factor 1 field (columns 9 through 16) with another field (columns 20 through 27) in the same record, with a constant (columns 20 through 39), or with all or part of the program date (columns 20 through 39). The other field in the same record, the constant, or the program date is called factor 2. Columns 17 and 18 tell the sort program what the results of the comparison must be. (If alternate collating sequence is used, both factor 1 and factor 2 are modified before the comparison is made.)

Column 17-18 Entry	Meaning
EQ ¹	Factor 1 must equal factor 2.
NE ¹	Factor 1 must not equal factor 2.
LT	Factor 1 must be less than factor 2.
GT	Factor 1 must be greater than factor 2.
LE	Factor 1 must be less than or equal to factor 2.
GE	Factor 1 must be greater than or equal to factor 2.

¹ If you want the sort program to compare zone portions of characters (Z in column 8), EQ and NE are the only entries you can use.

Column 19 (Field, Constant, or Keyword)

The sort program identifies records you want to sort by comparing the factor 1 field (columns 9 through 16) with another field in the same record, with a constant, or with all or part of the program date. The other field in the same record, the constant, or the program date is called factor 2. Column 19 tells the sort program whether factor 2 is another field, a constant, or a keyword that represents all or part of the program date. F in column 19 means factor 2 is a field; C means factor 2 is a constant; K means factor 2 is a keyword.

When you put an F in column 19, you use columns 20 through 27 to identify the location of the factor 2 field in the records. When you put a C in column 19, you use columns 20 through 39 for the constant. When you put a K in column 19, the keyword must begin in column 20.

Columns 20-27 (Factor 2 Field Location)

The factor 2 field must be the same length as the factor 1 field. It also must be in the same record as the factor 1 field.

Columns 20 through 27 are used to record the location of the factor 2 field. Columns 20 through 23 (From) identify the starting position of the field; columns 24 through 27 (To) identify where the field ends.

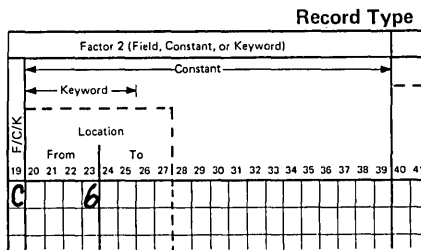
Coding Rules

1. Entries must be right-justified: the From entry must end in column 23; the To entry must end in column 27.
2. To describe fields that are only 1 character long, leave columns 20 through 23 (From) blank, and enter the number of the record position that contains the character in columns 24 through 27 (To).

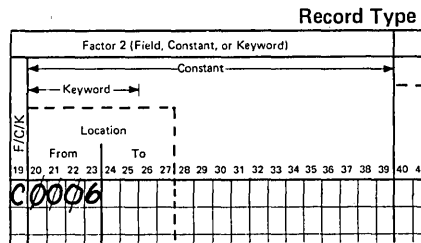
Columns 20-39 (Factor 2 Constant)

When factor 2 is a constant, you use columns 20 through 39 to write in the constant you want to use. The constant can be any arrangement of characters and blanks.

The constant must be the same length as the factor 1 field. For example, if you have a four-position numeric factor 1 field, your constant field must take up four positions. If your constant is the number 6, you put the 6 in column 23, and either leave columns 20, 21, and 22 blank or fill them with zeros.



OR



If the factor 1 field contains a packed number, the length of the constant (including the sign) must be twice the length of the factor 1 field.

Alphameric Constants (Column 8 Entry is C, Z, or D)

The constant must be the same length as the factor 1 field and must always begin in column 20.

Numeric Constants (Column 8 Entry is P, U, or D)

Format: Numeric constants must be right-justified within the field length specified in factor 1 (within twice the field length if factor 1 is a packed number). For example, assume that factor 1 defines a six-position field in the input record, and that factor 2 is the numeric constant 123. To right-justify the constant within six positions, you would have to put the constant in columns 23, 24, and 25. Leading zeros are not required. Blanks and zeros look the same to the sort program. In the preceding example, columns 20 through 25 could contain either 000123 or ~~000~~123 (with ~~0~~ representing a blank).

Note: For character constants, a D in column 8 indicates that only the digit portion of a character will be used in the compare operations. For numeric constants, a D in column 8 indicates that the numbers are unsigned.

Signed Constants: If factor 1 is a packed number, the last character in the constant must be its sign (+ or -). If factor 1 is a zoned number 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.

Coding Negative Zoned Constants			
If Last Digit in Constant is	Character That You Code	How the Number Looks Inside the Computer	
		Zone Portion ¹	Digit Portion ²
0	- (minus code)	0110	0000
1	J	1101	0001
2	K	1101	0010
3	L	1101	0011
4	M	1101	0100
5	N	1101	0101
6	O	1101	0110
7	P	1101	0111
8	Q	1101	1000
9	R	1101	1001

¹The zone portion indicates the negative sign of the entire number.
²The digit portion indicates the numeric value of the last digit in the number.

For example, here are the entries you would make to sort records that have a packed negative 1 (-1) in positions 1 and 2, a zoned negative 24 (-24) in positions 5 through 8, and a zoned negative 10 (-10) in positions 11 through 16.

Line		Factor 1		Factor 2 (Field, Constant, or Keyword)		Record Name	
Number	Type (I/O)	Location	Rel.	Location	Rel.	Record Name	
		From To		From To			
0 1	I P	1 2	ECC	1 -		PACKED -1	
0 2	IAU	5 8	ECC	24		ZONED -24	
0 3	IAU	11 16	ECC	1 -		ZONED -10	
0 4							
0 5							
0 6							

Columns 20-39 (Factor 2 Keyword)

When factor 2 is a keyword, the column 8 entry must be a C. The keyword must begin in column 20; all unused columns through column 39 should be left blank. The keyword tells the sort program whether all or just part of the program date should be compared with the factor 1 field (columns 9 through 16).

Keyword	Part of Program Date	Factor 1 Field Length ¹
UPDATE ²	Entire program date	6 characters
UMONTH	Month portion of program date	2 characters
UDAY	Day portion of program date	2 characters
UYEAR	Year portion of program date	2 characters

¹ The factor 1 field length must be 6 for UPDATE, and must be 2 for UMONTH, UDAY, or UYEAR.

² If the UPDATE keyword is used, the program date must be in the same format as the date contained in the input records. (Refer to the description of the DATE procedure in the *System Support Reference Manual* for information on the possible formats of the program date and for information on how to change the program date if necessary.)

Columns 40-72 (Comments)

Columns 40 through 72 are for your comments. If you instruct the sort program to print sequence specifications (column 27 of the header line is either blank or contains a zero), the comments will be printed along with your sequence specifications. The comments have no effect on the program's operation.



Many programmers like to use columns 40 through 47 to write the names of the records described in the record type specifications. This is why columns 40 through 47 are enclosed in dotted lines on the coding sheet.

Chapter 4. Field Specifications

Field specifications tell the sort program how to arrange and format (build) records in the work file and the output file. Possible column entries, summarized on the next page and in Appendix C for quick reference, are explained in detail in this chapter.

FIELD SPECIFICATION COLUMN SUMMARY

Column Summary		
Columns	Contents	Explanation
1-2	00-99	Page number.
3-5	07x-14x	Line number. You can leave column 5 (x) blank, or enter any value to keep the specifications in ascending order.
6	F	Field specification line.
7	N	Normal control field.
	O	Opposite control field.
	F	Forced control field.
	D	Data field.
	S	Summary data field.
	*	Comment line.
8	C	Use both zone and digit portions of characters in the field.
	Z	Use only zone portion of 1-character field.
	D	Use only digit portions of characters in the field.
	P	Signed packed decimal data.
	U	Signed zoned decimal data.
	V	Force a data character into the data field.
9-12	1-4096	Start position of a field in the record (can be blank if the field is 1 character long).
13-16	1-4096	End position of a field in the record.
17	Any character	Forced control fields; the character you want the sort program to change. Also used for summary overflow indicator fields; the character to be used for the overflow indicator.
18	Any character	Forced control or data field; the character you want the sort program to substitute. Also used for summary overflow indicator fields; the character to which the overflow field is initialized.
19	Blank	Forced control field line is not a continuation of the preceding line.
	Any character other than blank	Forced control field line is a continuation of the preceding line.
20-22	1-256	Overflow field length entry for summary tagalong sort only.
23-39	Blank	Reserved for system use.
40-72	Any characters	Comments.

	Columns that must be filled in for all sort jobs.
	Columns that must be filled in when forced control fields are being used.

COLUMN DESCRIPTIONS

Columns 1-2 (Page Number) and 3-5 (Line Number)

Page number (columns 1 and 2) and line number (columns 3 through 5) form a five-digit sequence number. As the sort program reads sequence specifications, it checks the sequence numbers to make sure they are not in descending order. If the numbers are in descending order (if page 02 specifications come before page 01 specifications, for example) and specifications are being issued, the sort program places a warning (the letter S) next to the line. (The S stands for sequence error.) After issuing a warning, the sort program continues to read the rest of the specification lines, then issues a message and waits for further instructions from the operator. The operator can either continue or end the job.

Because page number applies to all lines on a page, columns 1 and 2 appear only once, in the upper right corner of the page. You number the pages in ascending order.

Here is the recommended procedure for field description line numbers. Field line numbers are 07 through 14. The numbers in columns 3 and 4 are preprinted on the coding sheet. You can leave column 5 blank, or enter any value to keep your specifications in ascending order. If you have more than eight field lines, use another coding sheet and start at line 07.

Use column 5 when you want to insert a specification line without renumbering the other lines. For example, to insert a specification line between lines 01070 and 01080, you can number it 01075, code it, and then continue to fill out the rest of the sheet.

Be sure any lines that are out of sequence are clearly marked. You can do this by writing a note in the margin of the page with an arrow pointing to where the insert belongs. (*Sample Job 3* in Chapter 7.)

Column 6 (Line Type)

Column 6 contains a preprinted F, identifying the line as a field line. For addrout sorts (SORTA), field lines describe control fields that the sort program uses to sort record addresses. For tagalong sorts (SORTR) and summary tagalong sorts (SORTRS), field lines describe the fields that the sort program uses to create the records in the sorted output file. The fields can be either control fields (used to sort the records) or actual data fields. In addition, for summary tagalong sorts, field lines describe the fields that the sort program summarizes (adds together).

Column 7 (Field Type or Comments)

Your column 7 entry tells the sort program whether you are describing a control field, a data field, or a comment line. If you describe a control field, the column 7 entry indicates how the field is to be used.

See *Columns 7-8* to find the ways in which you can combine this entry with the column 8 entry.

Column 7 Entry	Tells the Sort Program
N	This is a normal control field. Sort this field so that the data from the field is in the sequence specified in column 18 of the header line.
O	This is an opposite control field. Sort this field so that the data from the field is in the sequence opposite that specified in column 18 of the header line.
F	This is a forced control field. Change the control field according to the entries in columns 9 through 19.
D	This is a data field. ¹
S	This is a summary data field. ²
*	This is a comment line.

¹ Use this entry for tagalong (SORTR, SORTRS) sorts only. (If you use a D entry during an addrout sort, the line will be treated like a comment line.)

² Use this entry for summary tagalong (SORTRS) only. (If you use an S entry during an addrout sort, the line will be treated like a comment line. If you use an S entry during a tagalong sort, the field will be treated like a normal data field.)

Note: The first control field defined on a form for a record type is the major control field; subsequent control fields for that record type are minor control fields.

Control Fields (N, O, or F in Column 7)

When your file has more than one type of record:

- The number of control fields does not have to be the same for all record types.
- The total lengths of the control fields do not have to be the same for all record types.

When records with duplicate control fields (records with all control fields equal) are sorted, their order as output is unpredictable.

Normal and Opposite Control Fields (N or O in Column 7):

These control fields are fields in your input records that the sort program uses to sort on. However, you can define 1-character control fields that are not in the records by using an unconditional force (see *Forced Control Fields*).

Forced Control Fields (F in Column 7): There are three types of forced control fields:

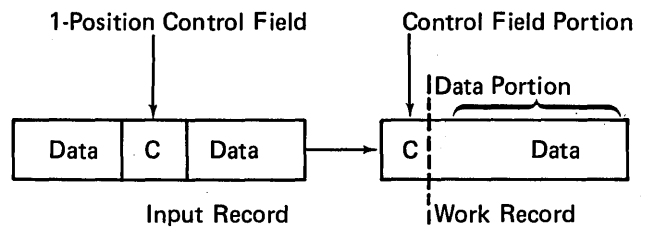
- Conditional
- Force-all
- Unconditional

Forced control fields affect the work and output records only. (It is important to remember that the sort program never changes your input records.)

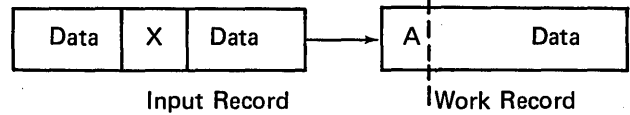
See *Column 17*, *Column 18*, and *Column 19* for information on how to fill out the field specifications when you are using forced control fields.

A conditional force occurs only if a control field in the input record contains a particular entry. Suppose, for example, that you want to sort a file of records, each of which has a 1-position control field. If the character in the control field is an X, you want to replace it with an A before you sort the records. To do this, you would use a conditional force. Your conditional force field would tell the sort program:

- Build a work record from the input record.



- If the control field contains an X, change it to an A.

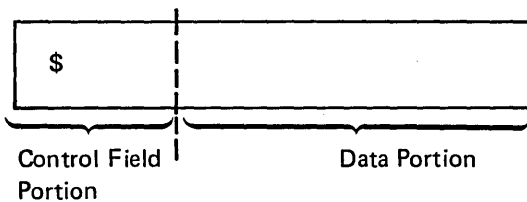


Force-all is a special type of conditional force. Force-all occurs only when a control field in an input record does *not* contain a particular entry. Suppose, for example, that you want to sort a file of records, each of which has a 1-position control field. If the character in the control field is *not* a C, an F, or a \$, you want to put an X in the field before you sort the records. To do this, you would use a force-all line to force the X into the control field.

A force-all line follows a series of conditional force lines. For example, you may want to tell the sort program:

- If the control field contains a C, replace it with a 1.
- If the control field contains an F, replace it with a 2.
- If the control field contains a \$, replace it with a 3.
- If the control field does *not* contain a C, an F, or a \$, put an X in the control field. (You are forcing out all other possible entries by using a force-all line.)

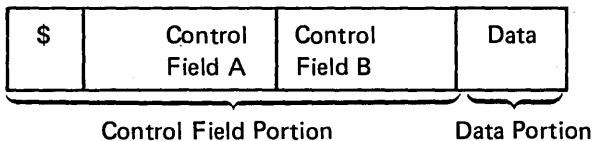
An unconditional force does not depend on entries in the input records. If, for example, you want to put a dollar sign (\$) in the first position of every output record, you would use an unconditional force. Your unconditional forced control field would tell the sort program to put a dollar sign in the first available control field position of the work record.



Any other control fields will go after the dollar sign. Assume the input records are in this format:

Data	Control Field A	Data	Control Field B	Data
------	-----------------	------	-----------------	------

The work record will then look like this:



See *Column 19* for actual coding examples of forced control fields.

Data Fields (D in Column 7)

Data fields apply to tagalong (SORTR, SORTRS) sort jobs only. They are fields you want the sort program to include in the sorted records, but which you do not want the sort program to use in sorting the records. Within each include set (include and field lines), control field lines must be placed before data field lines.

When your file has more than one type of record:

- The number of data fields does not have to be the same for all record types.
- The total lengths of all of the data fields do not have to be the same for all record types. Sort places blanks to the right of shorter data fields so that all total lengths are equal.

Note: For a summary tagalong sort (SORTRS), you cannot determine which input record's data field(s) will appear in the summary record.

Summary Data Fields (S in Column 7)

An S in column 7 of the field specifications defines a summary data field. You can define summary fields for all three types of sort jobs, but the fields will be summarized (added together) only in summary sorts. In tagalong sorts (SORTR), summary fields are treated like normal data fields. In addrout sorts (SORTA), summary fields are treated as comments.

In a summary sort, the summary data fields in the output records for all record types must always be in the same position. These data fields are summarized for all record types, even if all record types are not defined as containing summary fields.

These output fields do *not* have to be in the same position that they occupied in the input records. See messages SORT-7278 and SORT-7280 in Appendix E.

Note: The maximum number of fields that can be summarized is 24 for each record type in a sort job.

Comment Lines (* in Column 7)

Comment lines help document the sort program. They do not affect program operation in any way. You can code comment lines anywhere in the sequence specifications; however, comments will be printed only if column 27 of the header line contains a zero or a blank.

Column 8 (C/Z/D/P/U/V)

Your column 8 entry indicates what portion of the input record characters you want the sort program to use in building and sorting the work records. The column 8 entry is critical in assuring that the sort produces the results you intend. See *Columns 7-8* to learn how you can combine this column 8 entry with the column 7 entry.

Column 8 Entry	Character Portion Used	Maximum Field Length
C	Use both zone and digit portions of the characters	256 characters
Z	Use only the zone portion of the character	1 character
D	Use only the digit portions of the characters	16 characters
P	The characters are a signed, packed decimal number	8 characters or 15 digits and sign
U	The characters are a signed, zoned decimal number	16 characters
V	Force a data character into the data field	1 character

Suppose, for example, you have a 1-character control field in your input records which can be either an *, 1, 2, or 3. The zone and digit portion of each character is:

Character	Zone	Digit
*	0101	1100
1	1111	0001
2	1111	0010
3	1111	0011

If you want the records sorted into ascending order by using the digit portion of the control field characters (by putting a D in column 8), they will be in this order:

- 1
- 2
- 3
- *

If you want the records sorted into ascending order using both the zone and digit portions (by putting a C in column 8), they will be in this order:

- *
- 1
- 2
- 3

Suppose you placed a Z in column 8 and wanted the records sorted into ascending order. You can then be sure that the records with an * control field will precede the records with a 1, 2, or 3 control field. Since 1, 2, and 3 have identical zone portions, records with any of these numbers as a control field will not be in any special order after the sort.

If you want to force characters into your data field, place a V in column 8, and in column 18 specify the character to be forced. That character will be placed in the first available data field position of the work record.

As you can see, your column 8 entry can drastically affect your sorted file.

Do not confuse this column 8 entry with the column 8 entry on the record type specifications. Column 8 of the field specifications tells the sort program what portion of a character to use to sort the records. The column 8 entry on the record type specifications helps select which records you want to be sorted.

Opposite Control Fields

If you want to sort records so that some control fields are in ascending order and other control fields are in descending order, use opposite control fields. An opposite control field is sorted in ascending order (if you specify descending order on the header line), or in descending order (if you specify ascending order on the header line).

If your file contains different record types, all of which have an opposite control field in the same record position, your column 8 entries for these control fields must be one of the following:

- All Ds
- All Cs
- All Zs
- Any combination of Cs and Zs

With any other combination of entries (for example, Ds and Cs), you will not be able to predict the results of the sort.

When you use opposite control fields, the sort program changes them in building the work record. Therefore, you usually drop this meaningless control field information (by coding an X in column 28 on the header line) for tagalong or summary sorts. If the opposite control fields are all Ds, you do not need to drop the control field. If you wish to retain the original control field data in the output record, repeat the information as a data field.

Packed or Zoned Control Fields (Normal or Opposite)

If you specify packed or zoned control fields, the sort program changes the control fields while building the work record. Therefore, you may drop the control field information by coding an X in column 28 on the header line. If you wish to retain the original control field data in the output record, repeat the information as a data field.

Forced Control Fields

Here are a few rules to remember when you are using forced control fields:

1. You can have only 1 character in your forced control field.
2. You can force only entire characters into your control field.
3. You can indicate either a conditional or an unconditional force.
4. A force-all line must be preceded by a conditional force line.
5. You define a forced control field by placing an F in column 7 of the field specifications.

See *Columns 9-16, Column 17, Column 18, and Column 19* for further information on how to complete the field specifications.

Using Control Fields to Sequence the Sorted Records

The order in which you describe control fields in the field specification lines determines the sequence of the records (tagalong sort) or the record addresses (addrout sort) in the sorted file.

Suppose each record in your file that is sorted in ascending order (A in column 18 of the header line) has a normal control field in positions 1 and 2 and an opposite control field in positions 5 through 7. Each record represents one customer's order for a separate item. The part number is in positions 1 and 2; the number of parts ordered is in positions 5 through 7. Your unsorted file might look like this:

Input Record Position

	1	2	3	4	5	6	7
0	1	7			1	2	6
1	3	4			5	1	2
2	1	7			1	2	1
3	1	5				9	6
4	3	4				1	8
5	1	7				2	0
6	3	4			2	5	0

Input Record Number
Part Number
Number Ordered

You can use the first control field to sort the records in ascending order according to the part number. You can then use the second control field to sort the number of parts ordered in descending order within each group of parts. Your sorted file would look like this:

Output Record Position

	1	2	3	4	5
0	1	5		9	6
1	1	7	1	2	6
2	1	7	1	2	1
3	1	7		2	0
4	3	4	5	1	2
5	3	4	2	5	0
6	3	4		1	8

Output Record Number
Part Number
Number Ordered

Columns 7-8

Here are all of the possible combinations for columns 7 and 8.

Column 7	Column 8	Maximum Field Length
N or O	C	256
	Z	1
	D	16
	P	8
	U	16
F	C ¹	1
	Z ¹	1
	D ¹	1
D	C	256
	Z	1
	D	16
	P	8
	U	16
	V	1
S	C	256
	Z	1
	D	16
	P	8
	U	16
	V	1
*		Does not apply (comment line)

¹ For an unconditional force line and a force-all line, column 8 must contain a C. For a conditional force line, the column 8 entry refers to the comparison between the control field in the input record and the record character being replaced (contained in the column 17 entry) rather than to the substitute character (contained in the column 18 entry).

See *Column 7* and *Column 8* for detailed discussions of each entry.

As another example, if you want the 1-character field in position 20 of your input record to be in the first position of the sorted output record, you would describe the field in the first line of your field specifications. In this example, the primary control field for the output file is determined by position 20 of the record.

Line		Type (I/O/I/D/S/*) (C/Z/D/P/U/V)	Location				Forced		Overflow Field Length	Reserved
Number	3		4	5	6	From	To	Record Character Substitute Character Combination		
0 7	F	N	C					20		
0 8	F									
0 9	F									
1 0	F									
1 1	F									
1 2	F									
1 3	F									
1 4	F									

Field Length

The length of the field depends on the column 8 entry.

Column 8 Entry	Maximum Field Length
C	256 characters
Z	1 character
D	16 characters
P	8 characters or 15 digits and sign
U	16 characters
V	1 character

Coding Rules

1. Entries must be right-justified. The From entry must end in column 12; the To entry must end in column 16.
2. To describe fields that are only 1 character long, leave columns 9 through 12 (From) blank, and enter the number of the record position that contains the character in columns 13 through 16 (To).

Column 17 (Conditionally Forced Character)

See *Column 7* for a general discussion of forced control fields. You make an entry in column 17 when you want to use a conditional force. (For example, you might want to put an * in a control field only if the present entry is an A.) When you use conditional force, the input record does not change, but the work and output records will contain an * instead of an A. (See *Column 7* or *Column 18* for more information about conditional force.)

The column 17 entry tells the sort program which character in the control field (defined in columns 13 through 16) you want to replace. The sort program checks to see if the control field in the work record contains the character you specified in column 17. If it does, the character in column 18 replaces the control field character.

If a control field can contain any one of several characters, and you want to specify a forced character replacement for each one, you must include a column 19 entry in all of the forced field lines (except the first). See *Column 19* for instructions.

Besides being used for conditional force lines, column 17 can also be used as a summary overflow indicator field. The column 17 entry, in this case, is the character that is the overflow indicator. For more information on summary overflow indicator fields, see *Columns 20-22*.

Column 18 (Substitute Character)

See *Column 7* for general discussion of forced control fields. You make an entry in column 18 only when you are using a forced control field or a forced data field. The character in column 18 either replaces the control field character you specify in column 17, adds a new character to the control field, or adds a new character to the data field.

Remember that a change to the control or data field does not alter your input record. Also, if you want to use substitute characters, you can force a 1-character field (only an entire character can be forced).

Besides being used for forced field lines, column 18 can be used as a summary overflow indicator field. The column 18 entry, in this case, is the character to which the overflow field is initialized. For more information on summary overflow indicator fields, see *Columns 20-22*.

Defining a Conditionally Forced Character

1. Fill in columns 1 through 6 as you would for any control field.
2. Put an F in column 7.
3. Put a C, Z, or D in column 8.
4. Leave columns 9 through 12 blank.
5. Define the position of the control field in the input record in columns 13 through 16.
6. Enter the character you want to replace in column 17.
7. Enter the character you want to replace it with in column 18. (You can use any character in column 18.)

Defining a Force-All Character

1. Fill in columns 1 through 6 as you would for any control field.
2. Put an F in column 7.
3. Put a C in column 8.
4. Leave columns 9 through 17 blank.
5. Put the character that replaces the control field in column 18.
6. Put any character in column 19. (The character in column 19 tells the sort program that the line refers to the same control field in the work record as the preceding line.)

If you do *not* place a force-all line after conditional force lines and the sort program does *not* find the specified characters in the control field of the input record, the sort program:

- Replaces the control field character with hex FF (if you specified ascending sequence in the header line).
- Replaces the control field character with hex 00 (if you specified descending sequence in the header line).

Defining an Unconditionally Forced Character

1. Fill in columns 1 through 6 as you would for any control field.
2. Put an F in column 7.
3. Put a C in column 8.
4. Leave columns 9 through 17 blank.
5. Put the character you are forcing in column 18.

Sample Job 4 in Chapter 7 illustrates the use of unconditionally forced characters to group record types.

Specifying an Overflow Indicator Field

1. Fill in columns 1 through 6 as you would for any control field.
2. Put an S in column 7.
3. Put a V in column 8.
4. Leave columns 9 through 16 blank.
5. Enter a character in column 17 if you do not wish to use an * as a replacement. If overflow occurs in any summary data field in the record, the initial value of the overflow field will be replaced by the character in column 17.
6. Enter a character in column 18. If none of the summary data fields overflow, the overflow field will contain the character specified in column 18.

Column 19 (Specifying Replacements for More Than One Character)

If a control field in the input record can contain any one of several characters and you want to specify forced character replacements for more than 1 character, use a separate line to define each character you want to replace and the forced character you want to replace it with.

Coding Rules

1. Fill in columns 1 through 6 as you would for any control field.
2. Put an F in column 7.
3. Put a C, Z, or D in column 8.
4. Leave columns 9 through 12 blank.
5. Define the control field in columns 13 through 16.
6. Enter the first character you want to replace in column 17.
7. Enter the character you want to replace it with in column 18.
8. Fill in columns 1 through 6 (of the next line) as you would for any control field.
9. Put an F in column 7.
10. Put a C, Z, or D in column 8.
11. Leave columns 9 through 12 blank.
12. Define the control field in columns 13 through 16.
13. Enter the next character you want to replace in column 17.
14. Enter the character you want to replace it with in column 18.
15. Enter any character (except blank) in column 19. Your entry tells the sort program that the line refers to the same control field in the work record as the preceding line.

Repeat steps 8 through 15 for any other characters to be replaced.

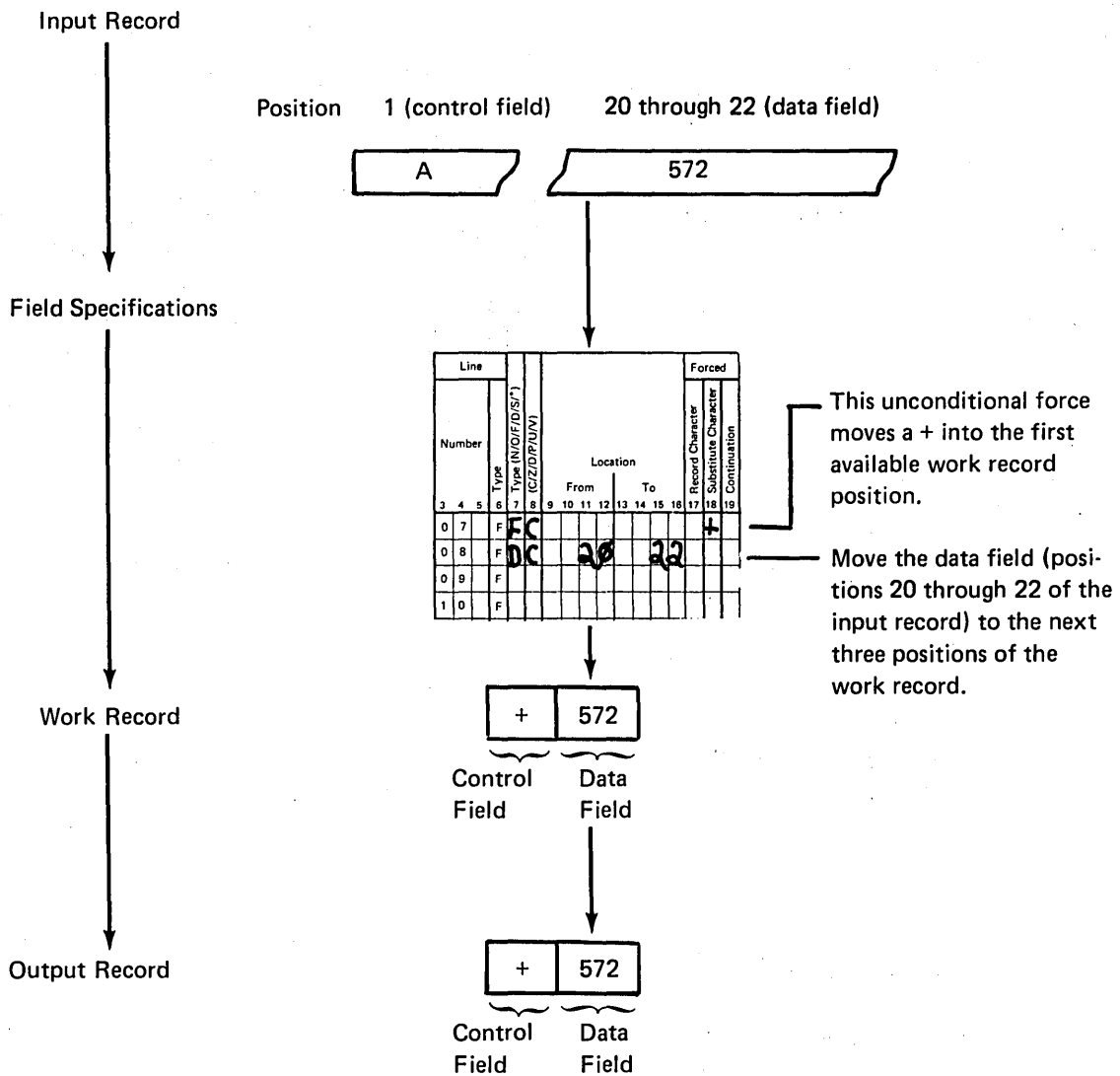
Examples of Using Forced Control Fields

You can use forced control fields to change (add to or replace) a character in a 1-position control field. The following examples represent these types of forced control characters:

- Unconditional force
- Conditional force using only forced control fields
- Conditional force using normal or opposite control fields and forced control fields
- Force-all

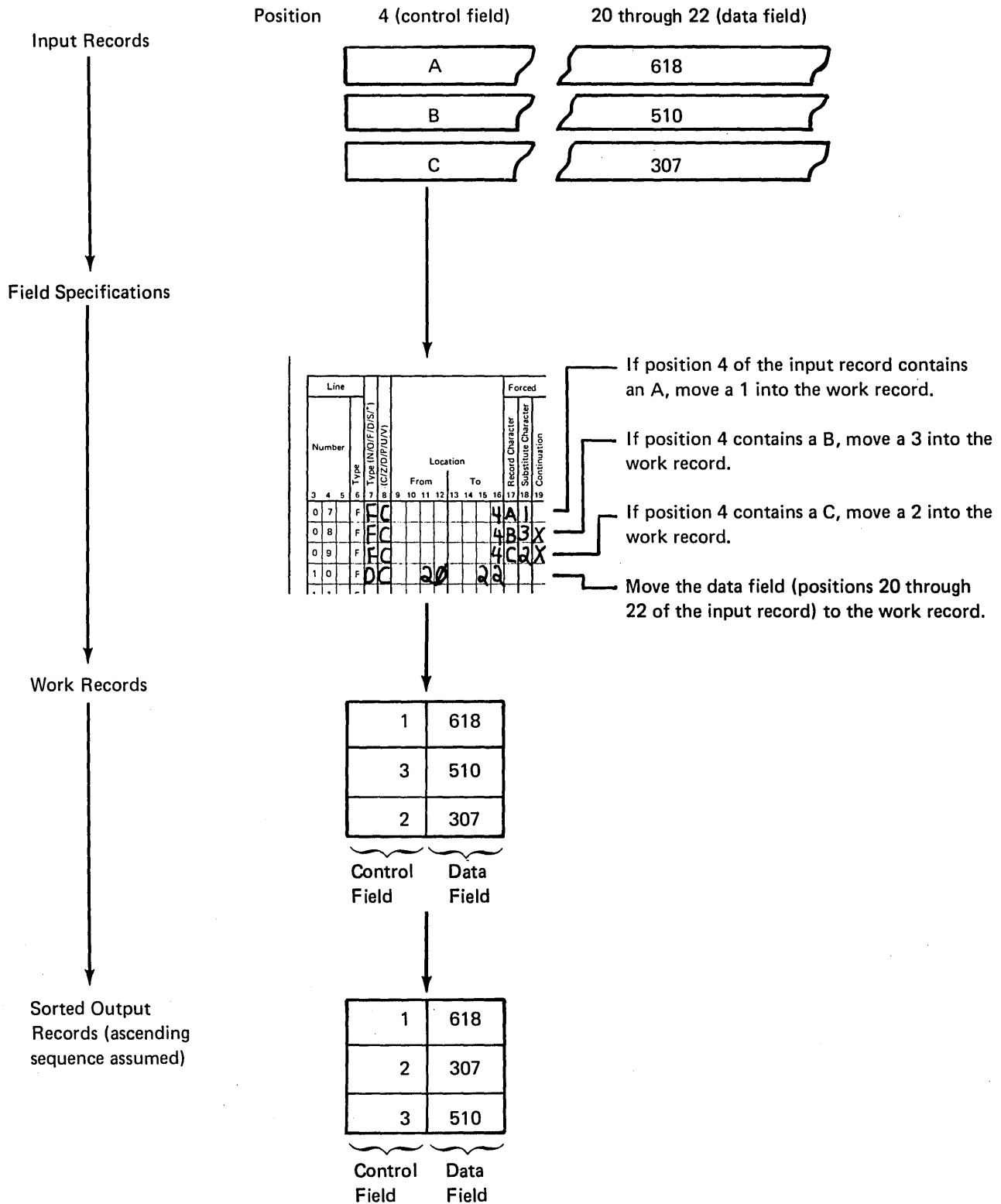
Unconditional Force

This example illustrates how you can unconditionally place a control field into the work and output records.



Conditional Force Using Only Forced Control Fields

This example illustrates how control field characters change according to the conditions you specify.

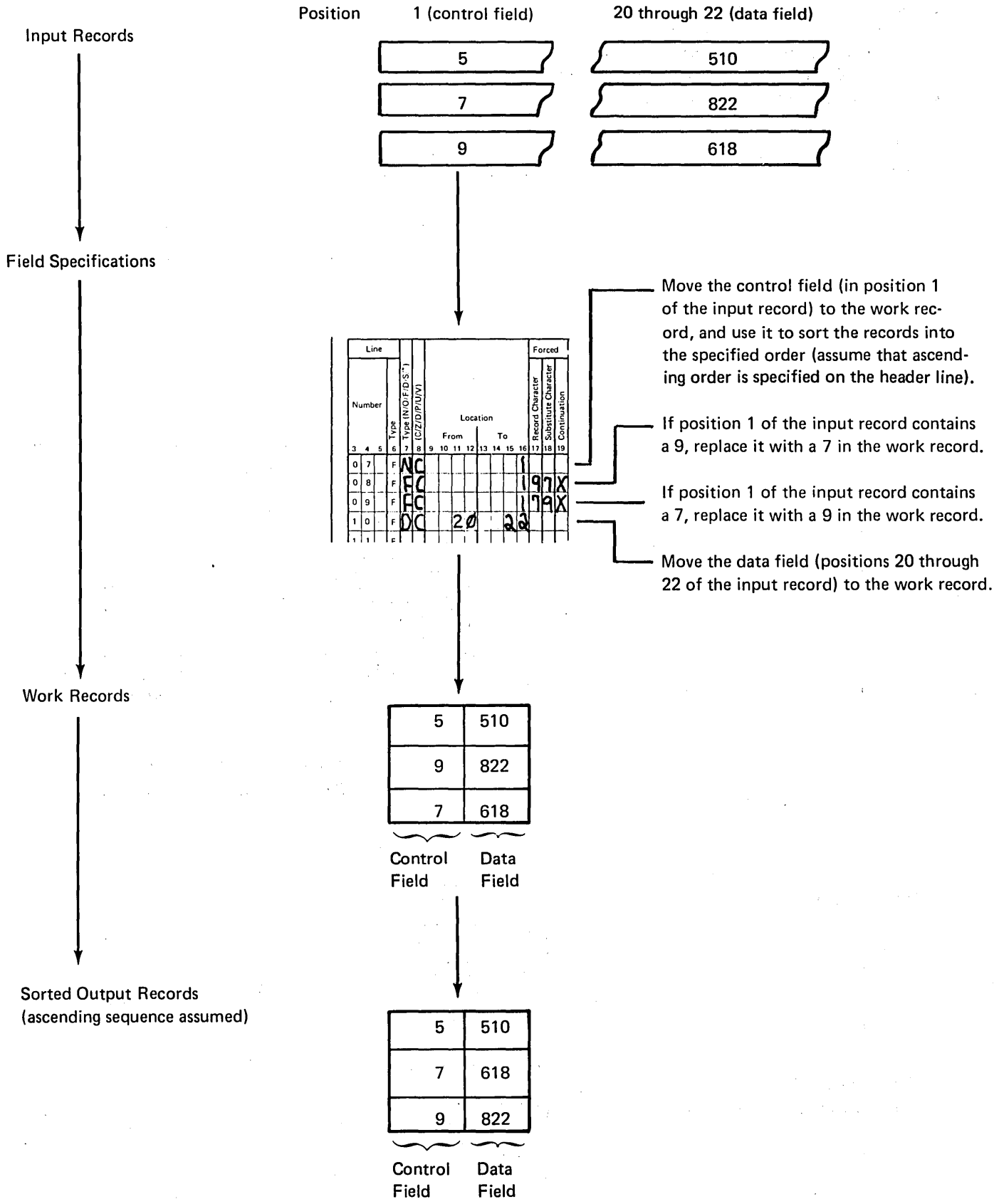


This example also shows how you can use column 19 to specify conditional replacements for more than one character. Because continuation lines are used, only one position in the work and output records is defined by the first three lines. If column 19 is left blank, each line would define a new position in the work and output records.

If you are sorting records into ascending order (A in column 18 of the header line), the sort program places hex FF into the work record before you force any characters. If you are sorting records into descending order (D in column 18 of the header line), the sort program places hex 00 into the work record before you force any characters. Therefore, if the sort program does not find an A, B, or C in position 4 of the input record, the hex 00 or FF is left unchanged in the work record.

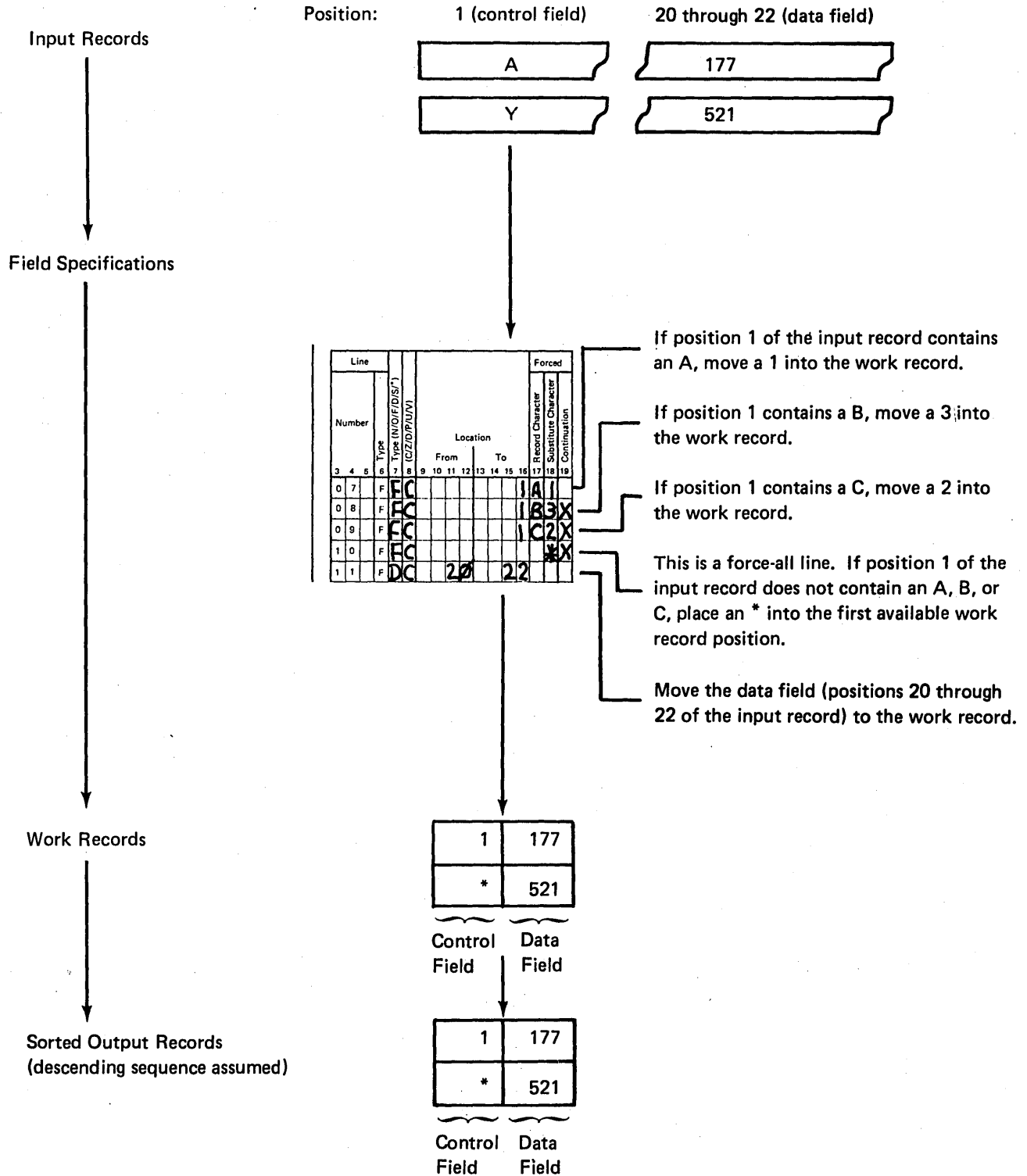
Conditional Force Using Normal or Opposite Control Fields and Forced Control Fields

This example is similar to the previous one in that the control field in the input record causes changes in the work and output records. In this case, however, the sort program first moves the control field to the work record (because the first control field is a normal control field) and then changes it if necessary. Notice that any forced character replacements are made before the records are sorted.



Force-All

Suppose your control field can contain characters other than A, B, or C. If you do not want a hex FF or 00 in the work and output records, use a force-all line.



Columns 20-22 (Overflow Field Length)

These columns are used only by a summary tagalong sort to eliminate the possibility of an overflow condition.

You can eliminate the possibility of an overflow condition by increasing the length of the summary data field to allow for any anticipated overflow. To get this increase, you code an overflow field length entry in columns 20 through 22 of the field specifications. The overflow field length entry should be long enough to contain the sum of the summary data field length and the anticipated overflow length. The entry must be right-justified to column 22, and cannot exceed the maximum field length.

For example, if you want to summarize a zoned field in positions 6 through 10 of your input record and you know that the output will exceed the five-position summary field by two positions, specify a 7 in column 22.

Line Number	Type (C/Z/D/P/U/V)	Location From To	Forced		Overflow Field Length	Reserved
			Record Character Substitute Character Continuation			
0 7	F	SU 6 10			7	
0 8	F					
0 9	F					
1 0	F					
1 1	F					
1 2	F					
1 3	F					
1 4	F					

If packed fields are summarized, columns 20 through 22 should specify the number of bytes of packed data.

For example, if you want to summarize a packed field in positions 1 through 3 of your input record and you know that the output will exceed the three-position packed summary field (five numbers plus sign) by one position, specify a 4 in column 22 (seven numbers plus sign).

Line Number	Type (C/Z/D/P/U/V)	Location From To	Forced		Overflow Field Length	Reserved
			Record Character Substitute Character Continuation			
0 7	F	SP 1 3			4	
0 8	F					
0 9	F					
1 0	F					
1 1	F					
1 2	F					
1 3	F					
1 4	F					

Example of Summary Overflow Indicator Field

Sometimes the summary data field size is too small to hold the sum of the corresponding data fields. Suppose the sorted file for a mail order company looked like this:

A552	23	A552	75
Catalog Number	Number Sold	Catalog Number	Number Sold

A552	5	C200	10
Catalog Number	Number Sold	Catalog Number	Number Sold

C200	16
Catalog Number	Number Sold

If the company used a summary sort but specified only two positions for the summary data field, its sorted file would look like this:

A552	03	C200	26
Catalog Number	Number Sold	Catalog Number	Number Sold

The number sold for the first catalog number should be 103, but since you have only a two-position summary field, the third digit overflowed and was lost. Since the company must know when an overflow occurs, its programmer will define an overflow indicator field that will print a summary overflow indicator whenever an overflow occurs.

Note: You can have only one summary overflow field for each record type.

To define a summary overflow field, put an S in column 7 and a V in column 8. The overflow field will be initialized to the character specified in column 18. If overflow occurs in any summary data field in the record, the initial value of the overflow field will be replaced by the character in column 17. (If you did not put an entry in column 17, the initial value of the overflow field will be replaced by an *.) If none of the summary data fields overflow, the overflow field will contain the character specified in column 18.

Here are the entries you would make to:

1. Initialize an overflow field to +.
2. Replace the + with an * whenever a summary data field overflows.

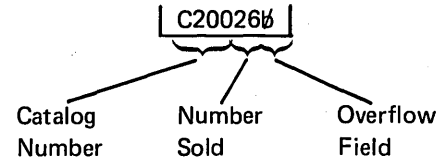
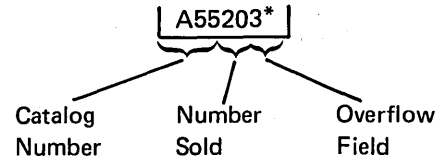
Line		Type (N/O/F/D/S/*) (C/Z/D/P/D/V)										Location		Forced		Reserved																		
Number	Type											From	To	Record Character	Substitute Character	Continuation	Overflow Field Length																	
3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	
07	F				S	V									+																			
08	F																																	
09	F																																	
10	F																																	
11	F																																	

Suppose the following field specifications were used for the previous example:

Line		Type (N/O/F/D/S/*) (C/Z/D/P/D/V)										Location		Forced		Reserved																		
Number	Type											From	To	Record Character	Substitute Character	Continuation	Overflow Field Length																	
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	
07	F				N	C				1				4																				
08	F				S	C				11				12																				
09	F				S	V									*																			
10	F																																	
11	F																																	

The output records would then contain the catalog number, the number of items sold (a summary data field), and the overflow field. The overflow field is initially blank, but it will contain an * after an overflow.

The summary output records would then look like this:



You can also use the V entry in column 8 with a D entry in column 7. The character you specify in column 18 is unconditionally forced into your work record. See *Sample Job 6* in Chapter 7 for an example of using these entries.

Columns 23-39 (Reserved)

These columns are not used. Leave them blank.

Columns 40-72 (Comments)

Columns 40 through 72 are for your comments. If you instructed the sort program to print sequence specifications (column 27 of the header line is either blank or contains a zero), comments are printed along with your sequence specifications. The comments have no effect on program operation.

Columns 40 through 45 are enclosed in dotted lines on the coding sheet, because many programmers like to use these columns to write the names of fields they described in the field lines.

To run any sort job, you must:

- Know the types of files you will use (input file, work file, output file)
- Supply operation control language (OCL) and sequence specifications to run your job

You can enter OCL statements to load and execute a sort job into the system in three ways: by using the SORT command; by executing your own sort procedure stored on disk, or by entering your own sort procedure through a display station keyboard.

FILE INFORMATION

File Characteristics

Input File

The input file is an area on disk that contains records to be sorted. Any existing indexed, sequential, or direct file with fixed length records can be used as input to the sort program; for example, an RPG II file or a file from a previous sort job. If you use the direct access method to create a file for sort, you might want to test for and omit blank records because direct file areas are set to blanks before the file is created.

Note: Unformatted sequential files created by FORTRAN do not have fixed length records. Therefore, they *should not* be used as input to the sort program.

Work File

The work file is an area on disk that is set aside as a work area. The sort program uses this area as working storage during the sorting process. You can either specify the work file area on disk or you can let the sort program automatically allocate it.

You should usually give a work file scratch status (RETAIN-S or RETAIN-J) because the work file is created during a sort and usually has no function after the sort job (or sort job step in a multiple-step procedure) is completed.

Output File

The output file contains the sorted file. If the output file exists before the sort, it cannot have permanent status (RETAIN-P) because the sort program cannot write into an existing permanent file. However, if the output file is built during the sort, it can have scratch (RETAIN-J), temporary (RETAIN-T), or permanent (RETAIN-P) status. RETAIN-S should not be used because the file will not exist at the end of the sort job.

Automatic Work File Allocation

If you omit the work file FILE statement from your sort OCL statements, the sort program will try to automatically allocate a scratch area on the disk. This scratch area (which is allocated if enough disk space is available) will be large enough to contain all of the work records if all input records are selected for sorting.

File Programming Considerations

You must consider several file characteristics when setting up your files or selecting the type of sort you will use. Some basic considerations are:

Considerations	Addrout Sort (SORTA)	Tagalong Sorts (SORTR, SORTRS)
Work file size	Work file needs only be big enough to contain control fields and relative record numbers.	Work file must be big enough to contain the entire file to be sorted. ¹ (See Appendix A for information on how to calculate work file size.)
Sort output	Relative record numbers only.	Control fields, data fields, and summary data fields. (See <i>Types of Sort: Addrout, Tagalong, and Summary Tagalong</i> in Chapter 1.
Input file overlay	Output file should not overlay the input file. ²	Output file can overlay the input file; that is, the input file can also be the output file. ³
Sort job running time	Usually shorter than SORTR or SORTRS.	Usually longer than SORTA.

¹ If the input file is a shared file (DISP-SHR) to which records are being added, or to which records have been added but that has not been closed, a FILE statement for the sort work file must be supplied before the records in the input file can be sorted. If the number of records in the input file is known, use that number to calculate the work file size. If the number of records in the input file is not known, use the maximum number of records that the input file can contain to calculate the work file size. (See Appendix A for information on how to calculate work file size.)

² If your input file is overlaid, you must have a backup copy of your input file that you can access using your addrout sort output.

³ The space for the output file can be the same as the space for the input file. However, you should not overlay a file you want to sort unless you first make a backup copy of your input file. When you use the input file as the output file, either you must specify (for your output file) the location and size, or you must indicate that the output file is an existing file by specifying DISP-OLD in your output file FILE statement. (See description of FILE statement parameters in the *System Support Reference Manual*.)

RUNNING SORT BY USING THE SORT COMMAND

The SORT command permits you to run certain sort jobs without supplying your own OCL statements (see *SORT Command Limitations*). When using the SORT command, your sequence specifications must be stored in a source member on disk. The SORT command format is:

```
SORT input file label,source member
output file label,number of records, [user library name],
[ Y ]
[ N ]
```

SORT Command Parameters

SORT command parameters are:

Input file label: Label of the existing data file on disk to be sorted.

Source member: Name of the source member that contains the sort sequence specifications.

Output file label: Label of the file that will contain the sorted data. This label must not be an existing file label.

Number of records: Number of records that the new output file will contain.

User library name: Name of the user library that contains the source member. If the user library name is omitted, the sort program searches the system library (#LIBRARY) for the source member.

Y: Place the sort job on the input job queue. Y specifies *N* that the sort job should be placed on the input job queue. N (default) specifies that the sort job should not be placed on the input job queue. If this parameter is used to place a sort job on the input job queue the sort program must be contained in the system library (# LIBRARY).

Input file label, source member, output file label, and number of records are always required. User library name is required only if the source member is contained in a user library. (See the *System Support Reference Manual* for more detailed information on source members and user libraries.)

SORT Command Keying

The SORT command statement can be keyed in two ways:

1. It can be keyed in its entirety, for example:

```
SORT MASTER,MASPECS,SORTMAST,
1000,LIBR1,Y
```

If any of the first four parameters are not keyed in, a display appears prompting for those parameters (see 2 which follows). If the user library name parameter is not entered, the default is to #LIBRARY. If the input job queue parameter is not entered, the default is to N. Any parameter entered with the SORT command appears in the display prompting for the parameters. You can rekey any of the parameters when the screen appears.

2. The word SORT can be keyed, which causes the following display to appear. The display prompts for all the parameters:

LABEL OF FILE TO BE SORTED

NAME OF SOURCE MEMBER CONTAINING
THE SORT SPECS

LABEL OF THE OUTPUT FILE

NUMBER OF RECORDS TO BE PLACED
IN THE OUTPUT FILE

NAME OF USER LIBRARY CONTAINING
THE SOURCE MEMBER

PLACE JOB ON THE INPUT JOB QUEUE
(N OR Y)

If any required parameter is omitted in response to the display, they will be prompted for individually on the display screen:

ENTER FILE LABEL OF FILE TO BE SORTED

ENTER SORT SPECS SOURCE MEMBER NAME

ENTER FILE LABEL FOR SORT OUTPUT FILE

ENTER NUMBER OF RECORDS TO BE SORTED

After the SORT command statement is entered in one of the preceding ways, all parameters are checked to ensure their accuracy. A message is displayed when an error is encountered in a parameter. If none of the parameters contain an error, the SORT command is either placed on the input job queue or causes the following procedure to be executed:

```
// MEMBER PROGRAM1-#GS#MM  
  
// MEMBER USER1-#GS#MM  
  
// LOAD #GSORT  
  
// FILE NAME-INPUT,LABEL-input file label,  
DISP-SHR  
  
// FILE NAME-OUTPUT,LABEL-output file label,  
RECORDS-number of records  
  
// RUN  
  
// SOURCE source member[,user library name]
```

SORT Command Limitations

Use of the SORT command places limitations on the input file and the output file, and also requires that the sequence specifications be placed in a source member before the sort job can be run.

Input File Limitations

- Only one file on the disk should have the input file label. If the SORT command is used and more than one file have the given input file label, the file that has the latest date is sorted.
- The sorted file cannot be written back over the input file.

Output File Limitations

- The output file must be a new file. The sorted file cannot be written over a file that already exists on the disk.
- The output file is always a temporary file. Therefore, if the output must be something other than a temporary file, the SORT command cannot be used.

Besides these limitations, the SORT command automatically allocates a work file, which generally increases the time needed to run a sort job. If you want to minimize sort running time, do not use the SORT command. For more information on sort timing considerations, see Chapter 6.

RUNNING SORT BY USING YOUR OWN SORT PROCEDURES

File Information Needed When Using Your Own Sort Procedures

The following text contains general file information. (For detailed FILE statement parameter descriptions, see the *System Support Reference Manual*.)

If you enter your own OCL statements, you must supply the sort program with the following information:

About the Input File:

- Label of the file you will sort. The FILE statement NAME must be INPUT. Use the LABEL keyword to give the sort program the label of your file if it is not INPUT. For example:

```
// FILE NAME-INPUT,LABEL-MASTER
```

where MASTER is the label of your input file.

About the Work File (Optional):

- Label of your work file. The FILE statement NAME must be WORK. Use the LABEL keyword to give the sort program the label of your file if it is not WORK. For example:

```
// FILE NAME-WORK,LABEL-SORTWORK,
```

```
{ BLOCKS- } number,RETAIN- { S }  
{ RECORDS- }
```

where SORTWORK is the label of your work file.

- Number of blocks or records in the work file. In the work file FILE statement, the number after either BLOCKS or RECORDS indicates the total number of blocks or records in the work file. See Appendix A for information on how to calculate the number for either BLOCKS or RECORDS.
- Status of the work file. RETAIN-S (scratch at end of sort job step) or RETAIN-J (scratch at end of job) indicates work file status. The work file is usually a scratch file because you usually do not need its information after a sort job (or sort job step in a multiple-step procedure) is run.

Note: If the input file is a shared file (DISP-SHR) to which records are being added, or to which records have been added but that has not been closed, a FILE statement for the sort work file must be supplied.

About the Output File:

- Label of the sorted file. The FILE statement NAME must be OUTPUT. Use the LABEL keyword to give the sort program the label of your file if it is not OUTPUT. For example:

```
// FILE NAME-OUTPUT,LABEL-SORTMAST,
```

```
{ BLOCKS- } number,RETAIN- { J }  
{ RECORDS- }
```

where SORTMAST is the label of your output file.

- Number of blocks or records in the output file. In the output file FILE statement, the number after either BLOCKS or RECORDS indicates the total number of blocks or records in the output file. See Appendix A for information on how to calculate the number for either BLOCKS or RECORDS.
- Status of the output file. RETAIN-J (scratch at end of job), RETAIN-T (temporary), or RETAIN-P (permanent) indicates output file status. A file built during the sort job can have any of these parameters; however, an output file that exists before the sort job is run cannot have permanent status because the sort program cannot write into an existing permanent file. RETAIN-S should not be used because the file will not exist at the end of the sort job.

Note: If the work or output file you want to use is an existing scratch (RETAIN-J) or temporary (RETAIN-T) file, you must either supply the location where the file begins and its originally allocated size or you must indicate that the file is an existing file by specifying DISP-OLD in your work or output file FILE statement. (See descriptions of FILE statement parameters in the *System Support Reference Manual*.)

Source Member Information Needed When Using Your Own Sort Procedures

A source member can contain the sort sequence specifications for a sort job. If this source member is in a library other than the system library, the user library name must be placed on the SOURCE statement in order for the source member to be located by the sort program.

If a user library is specified on the SOURCE statement but is not found on disk, sort will search the system library for the source member.

Note: The wrong sequence specifications will be used by the sort program if the user library is not on disk and the source member in the system library is not the one desired.

The format of the SOURCE statement is:

```
// SOURCE source member [,user library name]
```

where the user library name is an optional parameter.

For a detailed description of source members and user libraries, and for information on how source members and procedures can be created by using the \$MAINT system utility program, see the *System Support Reference Manual*.

Source members and procedures can also be created and updated by the Source Entry Utility (SEU). See the *Source Entry Utility Reference Manual* for information on how to use SEU.

Executing Sort Procedures Stored on Disk

If you use a procedure that is on disk, sort sequence specifications either can be contained in a source member or can be contained within the procedure.

To execute the procedure, enter either:

```
name
```

or

```
// INCLUDE name
```

where name is the name of the procedure.

If the sort sequence specifications are contained in a source member, the procedure includes:

```
// LOAD #GSORT

// FILE NAME-INPUT, ...

// FILE NAME-WORK, ... (optional)

// FILE NAME-OUTPUT, ...

// RUN

// SOURCE source member [,user library name]
```

where the referenced source member includes:

```
(header specification)

(record type specifications)

(field specifications)

// END
```

Note: If the procedure is one that does not call a terminal, an END statement must also follow the SOURCE statement.

The sort sequence specifications can be included in the procedure, which then includes:

```
// LOAD #GSORT

// FILE NAME-INPUT

// FILE NAME-WORK, ... (optional)

// FILE NAME-OUTPUT, ...

// RUN

(header specification)

(record type specifications)

(field specifications)

// END
```

Note: The procedure should contain an END statement after the last sort sequence specification statement. If there is no END statement, undesirable results may occur. For example:

- OCL statements following the last sort sequence specification statement will be diagnosed as sort sequence specification statements.
- The PROGRAM1 message member will be reset so that no sort error messages are issued if the END statement is keyed in at the display station keyboard after the procedure is done executing.

Entering Sort Procedures through a Display Station Keyboard

If you enter the OCL for a procedure through a display station keyboard, sort sequence specifications either can be contained in a source member or you can key them in with the sort procedure OCL.

When sort sequence specifications are contained in a source member, the following OCL statements are keyed:

```
// LOAD #GSORT  
  
// FILE NAME-INPUT, ...  
  
// FILE NAME-WORK, ... (optional)  
  
// FILE NAME-OUTPUT, ...  
  
// RUN  
  
// SOURCE source member[,user library name]
```

where the referenced source member includes:

```
(header specification)  
  
(record type specifications)  
  
(field specifications)  
  
// END
```

The sort sequence specifications can be keyed in with the sort procedure OCL statements as follows:

```
// LOAD #GSORT  
  
// FILE NAME-INPUT, ...  
  
// FILE NAME-WORK, ... (optional)  
  
// FILE NAME-OUTPUT, ...  
  
// RUN  
  
(header specification)  
  
(record type specifications)  
  
(field specifications)  
  
// END
```

Note: The END statement indicates to the sort program that the last sort sequence specification statement has been entered.

The time required to run a sort job can vary greatly, and depends on the following factors:

- Number of records to be sorted
- Record size
- Number of sequence specifications
- Whether or not alternate collating sequence is being used.
- Disk location of the input, work, and output files
- Whether or not automatic work file allocation is being used
- Order of records in the input file
- Use of the non-verify option
- Region size
- System environment in which a sort job executes

The following text describes how each factor affects the time required to run a sort job.

NUMBER OF RECORDS TO BE SORTED

The more records you want to sort, the longer the sort job takes to run. For the most efficient sort job, therefore, use the record omit option whenever possible.

RECORD SIZE

The larger the records, the longer it takes to sort a file. Therefore, if program run time is important to you, do not include unnecessary fields in your sort job. Every field you do not include decreases the record size by the length of that field. Remember, the sort program builds a work record for each input record included in the sort.

NUMBER OF SEQUENCE SPECIFICATIONS

The more sequence specifications you use, the less main storage space will be available for records.

- Code required by the select/build routine for any sort job uses about 100 bytes of storage. (The select/build routine is the machine language program created by the sequence specifications.)
- Each sequence specification uses about 25 bytes of storage. (Remember that comment lines use no main storage.)
- If you specify includes and/or omits (referring to packed data) for your sort job, the select/build routine will use an additional 200 bytes of storage.
- If you specify field statements (referring to packed data) for your sort job, the select build routine will use an additional 80 bytes of storage.
- An alternate collating sequence uses an additional 375 bytes of storage.

Consider using additional sequence specifications to exclude records you do not want to sort. Your sort job may run faster than if you use fewer specifications but process more records.

ALTERNATE COLLATING SEQUENCE

Use of an alternate collating sequence increases the time it takes to run a sort job (see *Alternate Collating Sequence* in Appendix B).

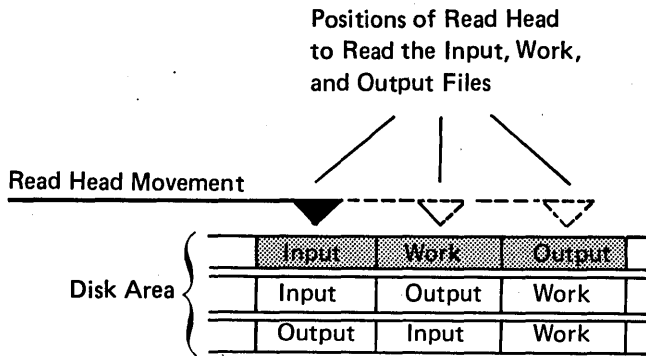
DISK LOCATION OF FILES

Because the sort program must move each record you want to sort at least two times (first from the input file to the work file, then from the work file to the output file), file location affects how long it takes to run a job. The distance the disk arm (or arms) must travel between the input and work files and between the work and output files can be minimized to reduce running time.

File Placement On a Single Disk System

The following shows how the input, work, and output files for a sort job can be placed on a single disk system. The first method shown is usually the fastest:

Methods of File Placement On a Single Disk System

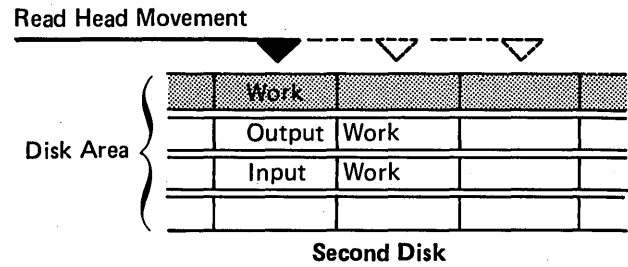
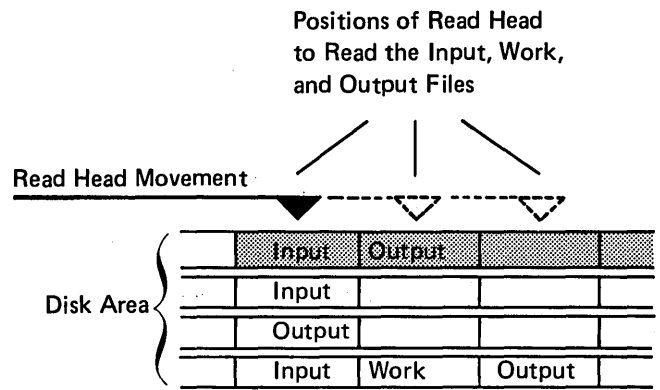


Usually Processes the Fastest

File Placement On a Multiple Disk System

The following shows how input, work, and output files for a sort job can be placed on a multiple disk system. The first method shown is usually the fastest:

Methods of File Placement On a Multiple Disk System



Usually Processes the Fastest

The first disk can be either disk A1 or A2; the second disk is the other one. If all files must be contained on one disk in a multiple disk system, use file placement as described under *File Placement on a Single Disk System*.

Note: On a system that is running in a multiprogramming mode, a sort job may run faster if one spindle is totally dedicated to accessing the sort input, work, and output files.

AUTOMATIC WORK FILE ALLOCATION

Single Disk System

Automatic work file allocation usually increases the time needed to run a sort job on a single disk system because an automatically allocated work file is not arranged for a fast sort run. If you want to minimize sort running time, determine the best location for your work file and use the FILE statement to specify your work file location rather than using automatic work file allocation.

Multiple Disk System

Automatic work file allocation usually does not increase the time needed to run a sort job on a multiple disk system. Run time increases only if both input and output files are on one disk and space is not available for the work file on the other disk. In this case, sort job run time is similar to the time required to run the same sort job on a single disk system.

ORDER OF RECORDS IN THE INPUT FILE

The order of input file records can affect how long it takes to run a sort job. Suppose that two files have about the same number of records to be sorted. Suppose further that the two files contain identical information; however, the first file has more records in the sequence you want than the second file has. The records in the first file can usually be sorted faster than those in the second file if the same sequence specifications are used to sort both files.

NON-VERIFY OPTION

If an N is entered in column 34 of the sort header specification, the sort program does not verify data written on the work file. The amount of performance improvement depends on file size, number of records, and file location.

Because you might create a file of invalid data when you specify the non-verify option, using this option is usually not the best way to improve your sort program performance. You should consider other factors that affect performance before you use this option.

REGION SIZE

You can increase the region size that the sort program uses whenever you have enough main storage available to do so. The larger region size usually allows the sort program to process more records per disk access, which usually reduces the time required to sort records in a file. (The minimum region size for the sort program is 14K bytes.)

SYSTEM ENVIRONMENT IN WHICH A SORT JOB EXECUTES

Two identical sort jobs operating in equal size regions may have different run times, depending on their respective system environments. A sort job that competes for system resources in a multiprogramming environment usually requires more running time than the same sort job would require if it were the only job executing on the system.

Each of the seven jobs in this section has:

1. An introduction explaining its purpose
2. Filled-out sequence specification sheets
3. Discussion of the sequence specifications

The first six jobs use the following files and input records.
(*Sample Job 7* uses other files and input records, which are explained in the introduction to that job.)

Files	Records in Files	Contents of Records
Inventory file	Inventory records	Quantities of items in stock
		Number of transactions for each stock item
Transaction file	Issue records	Shipments of items to customers
	Receipt records	Purchases of more items from suppliers (vendors)
	Adjustment records	Corrections to inventory quantities

The formats of the four types of records are shown in the following charts.

Record Formats for the Inventory File				
	Record Positions	Field Names	Contents¹	
Inventory Record	1-6	ITEM	Number of stock item to which the record applies.	Item Information
	7-10	CLASS	Class of the item (men's clothing, jewelry, and so on).	
	11-24	DESC	Description of the item.	
	25-27	LOC	Location of the warehouse where the item is stored.	Stock Status and Reorder Information
	28-29	UNIT	Unit of measure for ordering the item (by the pound, by the dozen, and so on).	
	30-34	COST	Cost of the item per unit.	
	35-39	PRICE	Price (per unit) at which the item is sold.	
	40-45	REORD	Reorder point (lowest stock quantity allowed).	
	46-50	QTY	Number of units to order when reordering.	
	51-56	BAL	Number of units in stock.	
	57		Not used.	
	58-62	ORDER	Number of units ordered, but not yet received.	
	63-68		Not used.	
	69-74	AVAIL	Number of units available (BAL + ORDER).	
	75-80	PORDER	Purchase order number.	
81-86	PDATE	Date of purchase order.		
87-91	VEND1	First vendor.		
92-96	VEND2	Second vendor.		
97-99	TRANS	Number of transactions for this period.	Transaction Activity	
100-105	TDATE	Date of last transaction.		
106-111	ISSUES	Number of units issued during this period.		

¹ Any unsigned numeric field (packed or zoned) can be treated as character information.

Record Formats for the Transaction File			
	Record Positions	Field Names	Contents ¹
Issue Record	1	ID	I (identifies this record as an Issue record).
	2-7	ITEM	Number of the stock item to which record applies.
	8-12	QTY	Number of units ordered (5 dozen, 32 pounds, and so on).
	13-19	CUST	Customer number.
	20-27	INV	Invoice number.
	28-33	DATE	Date of the order.
	34	DIS	Discount information.
	35-39	PRICE	Price of the item per unit.
	40-42	CON1	The constant 001.
	43-50	COST	Total cost of items ordered.
Receipt Record	1	ID	R (identifies this record as a Receipt record).
	2-5	CLASS	Class of the item ordered (men's clothing, jewelry, and so on).
	6-11	ITEM	Number of the item ordered.
	12-16	PRICE	Price of the item per unit (by the dozen, by the pound, and so on).
	17-22	PORDER	Purchase order number.
	23-27	QTY	Number of units ordered.
Adjustment Record	1	ID	A (identifies this record as an adjustment record).
	2-7	ITEM	Number of the stock item to which record applies.
	8	CODE	Adjustment code (identifies reason for the adjustment).
	9-13	QTY	Number of units (dozens, pounds, and so on) being added or subtracted.
	14-17	OK	Authorization for the adjustment.
	18-23	DATE	Date of the adjustment.

¹ Any unsigned numeric field (packed or zoned) can be treated as character information.

SAMPLE JOB 1: PRODUCING A FILE OF INVENTORY INFORMATION FOR REORDERING STOCK ITEMS

- The sort program selects inventory for stock items that need to be reordered. These are items for which the quantity on hand (AVAIL field) is less than or equal to the reorder point (REORD field).
- Output records are to contain information needed to reorder the items. The information is in these fields: CLASS, ITEM, DESC, UNIT, COST, QTY, AVAIL, PDATE, VEND1, VEND2, and PORDER.
- Output records are to be in ascending order by class (CLASS field) and by item numbers (ITEM field) within each class.

GX21-9089
Printed in U.S.A.

SEQUENCE SPECIFICATIONS

Header

Page **01** Program Identification 75 76 77 78 79 80

Line	Job	Card Match	Sequence (A/D/S)	Stacker Select	Number (I/N)	Alternate Coll. Seq. (S)	Print Option	Output Option (X)	Output Record Length	CCP/Disk Sort Only	Record Length	Job Description
3	0	0	H	SORTR	10A				64	N		

Line	Factor 1	Rel.	Factor 2 (Field, Constant, or Keyword)	Comments
01	ID 69 74	LEF	40 45	AVAIL FLD LESS OR EQ TO REORD FLD

Line	Record Character	Substitute Character	Continuation	Overflow Field Length	Reserved	Field Name	Comments		
07	F	NC	7	10	Total-10 Characters	CLASS	CLASS OF ITEM--JEWELRY, ETC		
08	F	NC	1	6				ITEM	STOCK ITEM NUMBER
09	F	DC	11	24				DESC	DESCRIPTION OF ITEM
10	F	DC	28	29	Total-54 Characters (including VEND1, VEND2, and PORDER on page 02)	UNIT	UNIT OF MEASURE		
11	F	DC	30	34		COST	COST PER UNIT		
12	F	DC	46	50		QTY	RECORD QUANTITY		
13	F	DC	69	74		AVAIL	UNITS AVAILABLE--BAL+ORDER		
14	F	DC	81	86		PDATE	PURCHASE ORDER DATE		

Since there is not enough space on the first coding sheet for the last three fields, they are coded on a second sheet. To show the proper sequence for the last three fields, put a 02 in columns 1 and 2 of the second coding sheet.

GX21-9089
Printed in U.S.A.

SEQUENCE SPECIFICATIONS

Header

Page **02** Program Identification 75 76 77 78 79 80

Line	Job	Card Match	Sequence (A/D/S)	Stacker Select	Number (I/N)	Alternate Coll. Seq. (S)	Print Option	Output Option (X)	Output Record Length	CCP/Disk Sort Only	Record Length	Job Description
3	0	0	H	SORTR	10A				64	N		

Line	Record Character	Substitute Character	Continuation	Overflow Field Length	Reserved	Field Name	Comments
07	F	DC	87	91	Total-54 Characters (including VEND1, VEND2, and PORDER on page 02)	VEND1	FIRST VENDOR
08	F	DC	92	96		VEND2	SECOND VENDOR
09	F	DC	75	80		PORDER	PURCHASE ORDER NUMBER

Header Specification

Columns 1-2

01 identifies this sheet as the first sequence specifications sheet.

Column 6

H identifies the line as a header line.

Columns 7-12

SORTR identifies the job as a tagalong sort.

Columns 13-17

The control fields used to sort records in this job are CLASS and ITEM. The total of their lengths is 10, which is the entry for columns 13 through 17.

Column 18

The A stands for ascending order. Output records are to be sorted into ascending order by CLASS and ITEM fields.

Columns 29-32

The output records contain 11 fields, including two control fields (CLASS and ITEM) and nine data fields. Control fields are not being dropped (column 28 is blank). Therefore, the length of the output record is the total length (64) of all 11 fields described in field specifications.

Column 36

N in column 36 indicates that the sort program will create an empty output file if no records are found in the input file or if no records are selected for sorting, and will *not* issue message SORT-7724 NO INPUT RECORDS INCLUDED.

Record Type Specifications

Column 6

I identifies the line as an include line. The input records described by the line are used by the sort program. All other input records are omitted.

Column 8

Both factor 1 and factor 2 fields are numeric positive fields. The D in column 8 causes zeros and blanks to be considered equal if leading zeros were not recorded.

Columns 9-16, 17-18, and 20-27

Inventory records for which the number of units available (AVAIL) is less than or equal to the reorder point (REORD) are selected:

- Columns 9 through 16 identify the location of the AVAIL field (record positions 69 through 74).
- Columns 20 through 27 identify the location of the REORD field (record positions 40 through 45).
- Columns 17 and 18 define the comparison made between the two fields.

Column 19

An F in column 19 indicates that factor 1 is compared to the field located by the entries in columns 20 through 27.

Columns 40-72

Columns 40 through 72 are comment columns.

Field Specifications

Column 6

F identifies the lines as field specification lines.

Column 7

The first two fields described (CLASS and ITEM) are normal control fields. The rest are data fields. All are included in the output record (column 28 of the header line is blank).

The records are sorted into ascending order by the CLASS field. Within each class, the item number (ITEM field) are in ascending order.

CLASS Sequence	ITEM Sequence	Sequence of Records in Sorted Output File
Lowest class no.	Lowest item no. Next higher item no. Third higher item no. . . . Highest item no.	Lowest class no. - Lowest item no. Lowest class no. - Next higher item no. Lowest class no. - Third higher item no. . . . Lowest class no. - Highest item no.
Each higher class no.	Lowest item no. Next higher item no. Third higher item no. . . . Highest item no.	Class no. - Lowest item no. Class no. - Next higher item no. Class no. - Third higher item no. . . . Class no. - Highest item no.
Highest class no.	Lowest item no. Next higher item no. Third higher item no. . . . Highest item no.	Highest class no. - Lowest item no. Highest class no. - Next higher item no. Highest class no. - Third higher item no. . . . Highest class no. - Highest item no.

Column 8

Zone and digit portions of all characters in all fields are used as they appear in the input records.

Columns 9-16

Columns 9 through 16 identify the locations of fields in the input records.

Columns 40-72

Columns 40 through 72 are comment columns.

SAMPLE JOB 2: PRODUCING A RELATIVE RECORD ADDRESS FILE FOR AN ENTIRE INVENTORY FILE

- Assume that the inventory file is an indexed file. The ITEM field is the record key used in the index. (The record keys are in ascending order.)
- This job creates a record address file containing relative record numbers of all records in the file. It sorts them into ascending order by the CLASS field. Within each class, addresses are also sorted into ascending order by ITEM field.
- As a result of this job, inventory records can be processed consecutively in either of two ways:
 1. By using the file index, which is in ascending order by ITEM field.
 2. By using the record address file created by the sort program, which is in ascending order by CLASS and also by ITEM within each class.

Line		Job		Card Match		Stacker Select		Output Record Length		CCP/Disk Sort Only		Job Description									
Number	Type	Card: MATCH, SORT, SELECT, MERGE	Largest sum (in bytes) of Control Field Lengths for Any Record Type	S	P	M	U	S	P	Output Record Length	Record Length	Job Description									
000	H	SORTA	10A																		

Line		Factor 1		Factor 2 (Field, Constant, or Keyword)		Comments									
Number	Type (I/O)	Location	Rel. EQ NE LT GE	Location	Record Name										
01				Constant											
02				Keyword											
03				Location											
04															
05															
06															

Line		Location		Forced		Comments											
Number	Type (N/O/D/S/*)	From	To	Record Character	Overflow Field Length	Field Name											
07	F	7	10			CLASS	CLASS OF ITEM--JEWELRY, ETC										
08	F	1	6					ITEM	STOCK ITEM NUMBER								
09	F					Total 10 Characters											
10	F																
11	F																
12	F																
13	F																
14	F																

Header Specification

Column 6

H identifies the line as a header line.

Columns 7-12

SORTA identifies the job as an addrount sort.

Columns 13-17

The control fields used to sort relative record numbers in this job are CLASS and ITEM. The total of their lengths is 10, which is the entry for columns 13 through 17.

Column 18

The A stands for ascending order. The records are sorted into ascending order by item class (CLASS field), and by item number (ITEM field) within each class.

Record Type Specifications

Because all input records are being used and all have the same field specifications, no record type specifications are needed (include-all is implied).

Field Specifications

Column 6

F identifies the lines as field specification lines.

Column 7

Both fields (CLASS and ITEM) are normal control fields. The relative record numbers are sorted into ascending order by CLASS field and by ITEM field within each class:

CLASS Sequence	ITEM Sequence	Sequence of Records in Sorted Output File
Lowest class no.	Lowest item no. Next higher item no. Third higher item no. . . . Highest item no.	Lowest class no. - Lowest item no. Lowest class no. - Next higher item no. Lowest class no. - Third higher item no. . . . Lowest class no. - Highest item no.
Each higher class no.	Lowest item no. Next higher item no. Third higher item no. . . . Highest item no.	Class no. - Lowest item no. Class no. - Next higher item no. Class no. - Third higher item no. . . . Class no. - Highest item no.
Highest class no.	Lowest item no. Next higher item no. Third higher item no. . . . Highest item no.	Highest class no. - Lowest item no. Highest class no. - Next higher item no. Highest class no. - Third higher item no. . . . Highest class no. - Highest item no.

Column 8

C means that both zone and digit portions of the characters in the CLASS control field are used in the sorting process.

D means that only the digit portions of the characters in the ITEM control field are used in the sorting process. This causes blanks to be treated as zeros.

Columns 9-16

The CLASS control field is located in positions 7 through 10 of the input records.

The ITEM control field is located in positions 1 through 6 of the input records.

SAMPLE JOB 3: PRODUCING A FILE OF INVENTORY INFORMATION ABOUT CERTAIN ITEMS

- The sort program selects inventory records for stock items in classes 0126 through 0130.
- Output records are to contain only inventory information that indicates the activity (transactions) involving the stock items. The information is in these fields: ITEM, CLASS, TRANS, and ISSUES.
- Output records are to be sorted in ascending order by class (CLASS field), and in descending order by number of transactions (TRANS field) within each class.

Line		Job		Card Match		Stacker Select		Output Record Length		CCP/Disk Sort Only		Job Description	
Number	Type	Card: MATCH, SORT, SELECT, MERGE	Largest sum (in bytes) of Control Field Lengths for Any Record Type	U	S	P	M	U	O	Print Option	Output Option (X)	Record Length	
3	4	5	6	7	8	9	10	11	12	13	14	15	16
00	H	SORTR	7A							X		19	

Line		Factor 1		Factor 2 (Field, Constant, or Keyword)		Comments	
Number	Type (IO)	Location	Rel.	Location	Constant	Record Name	
3	4	5	6	7	8	9	10
01	ID	7	10	EQ	LECO130		
02	IAD	7	10	EQ	LECO126		INCLUDE CLASS FLOS 0126-0130

Line		Type (NOIF/DS/)		Forced		Reserved		Comments	
Number	Type	Location	Record Character	Substitute Character	Overflow Field Length	Field Name			
3	4	5	6	7	8	9	10	11	12
07	F	NC	7	10					
08	F	OD	97	99				Total—7 Characters	CLASS CLASS OF ITEM--JEWELRY,ETC
09	F	DC	1	6					TRANS NUMBER OF TRANSACTIONS
10	F	DC	97	99				Total—19 Characters	ITEM STOCK ITEM NUMBER
095	F	DC	7	10					TRANS NUMBER OF TRANSACTIONS
12	F	DC	106	111					CLASS CLASS OF ITEM--JEWELRY,ETC
13	F								ISSUES NUMBER OF UNITS ISSUED

Header Specification

Column 6

H identifies the line as a header line.

Columns 7-12

SORTR identifies the job as a tagalong sort.

Columns 13-17

The control fields used to sort records in this job are CLASS and TRANS. The total of their lengths is 7, which is the entry for columns 13 through 17.

Column 18

The A stands for ascending order. One control field (CLASS) is defined as a normal control field in field specifications. The other control field (TRANS) is an opposite control field. The records, therefore, are sorted into ascending order by item class (CLASS field), and into descending order by number of transactions (TRANS field) within each class.

Column 28

X indicates that control fields are being dropped from the output records because the TRANS field is an opposite control field. The sort program must convert the information from the TRANS field into a special form so that the records can be sorted properly. The sort program does not return the information to its original form after sorting. The information, therefore, would be meaningless.

The information from these two fields, however, is necessary in the output records. Therefore, the fields are also defined as data fields (see the explanation of column 7 of the field specifications for this example).

Columns 29-32

Because control fields are dropped from the output records, the length of the output records is the total length of the fields defined as data fields in field specifications. The length is 19, which is the entry in columns 29 through 32.

Record Type Specifications

Column 6

I identifies the record type lines as include lines. The input records described by these lines are used by the sort program. All other input records are omitted.

Column 7

The A in column 7 of line 02 means that line 02 is a continuation of the definition of the record described in line 01. The A means AND. An input record must meet all conditions listed in lines 01 and 02 before it is included.

Column 8

D in column 8 of lines 01 and 02 means that the digit portions of the characters in positions 7 through 10 of the records (factor 1) are compared with the digit portions of the constants 0126 and 0130 (factor 2). Leading blanks in positions 7 through 10 are treated like zeros. Therefore, b126 (b is a blank) and 0126 are considered equal.

Columns 9-16, 17-18, and 20-39

Inventory records that indicate the activity (transactions) of stock classes 0126 through 0130 are to be selected:

- Columns 9 through 16 (factor 1) identify the location of the CLASS field (record positions 7 through 10). The CLASS field contains four-digit class numbers.
- Columns 20 through 23 list constants (C in column 19), which in this example define the upper and lower limits of the classes being selected from the inventory file.
- Columns 17 and 18 define the relationship between the input record field being used to build the file (factor 1) and the constants in factor 2. All inventory classes greater than or equal to (GE) 0126 and less than or equal to (LE) 0130 are to be included in the sorted file.

Column 19 and Columns 20-39

C in column 19 indicates that factor 1 is compared to a constant in columns 20 through 39. The constant in line 01 is 0130. The constant in line 02 is 0126.

Field Specifications

Column 6

F in column 6 of lines 07 through 12 indicates that the lines are field specification lines.

Column 7

Column 7 indicates the types of fields being described. The fields described in lines 07 and 08 (CLASS and TRANS, respectively) are control fields. CLASS is a normal control field. TRANS is an opposite control field. Output records are sorted so that CLASS fields are in the order indicated in column 18 of the header specifications (ascending order). Records having the same CLASS field are in descending order according to the TRANS field.

The fields described in lines 09 through 12 are data fields. They are included in the output record but have no effect on the sequence of the records.

Note that the CLASS and TRANS fields are described once as control fields and once as data fields because TRANS is an opposite control field. Its contents in the control field portion of the output records are changed in the sorting process and are meaningless for later processing. Therefore, the control fields are dropped from the output records (X in column 28 of the header specifications), and CLASS and TRANS are defined as data fields so that they still appear in the output records.

The form of the output record is as follows:

ITEM	CLASS	TRANS	ISSUES
1 ←→ 6	7 ←→ 10	11 ←→ 13	14 ←→ 19

Positions

Column 8

C in column 8, line 07, and in column 8, lines 9 through 12, means that both the zone and digit portions of all characters in all fields are used as they appear in the input records. D in column 8, line 08, means that only the digit portions of the characters in the TRANS control field are used in the sorting process.

Columns 9-16

Columns 9 through 16 identify the locations of fields in the input records.

SAMPLE JOB 4: PRODUCING A HISTORY FILE OF TRANSACTION INFORMATION

- The sort program selects records for all transactions in the transaction file.
- The entire file of input records is used to create output records. The output records are used to produce a history of the activity involving various stock.
- Output records are sorted by record type: RECEIPT records first, ISSUE records second, and ADJUST (adjustment) records last. Each type of record is sorted into ascending order by item number (ITEM field). The ISSUE records also have a subcontrol field—DATE OF ORDER.

Line		Job		Card Match		Stacker Select		Output Record Length		CCP/Disk Sort Only		Job Description	
Number	Type	Card: MATCH, SORT, SELECT, MERGE Disk: SORTR, SORTA, SORTRS Tape: SORTT	Largest sum (in bytes) of Control Field Lengths for Any Record Type	S	P	P	S	P	Output Record Length	Record Length			
3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	0	H	SORTR							39			

Line		Factor 1		Factor 2 (Field, Constant, or Keyword)		Comments	
Number	Type (I/O)	From	To	Rel.	Location	Record Name	
3	4	5	6	7	8	9	10
0	1	I	C	1	EQCR	RECEIPT	

Line		Location		Forced		Reserved		Comments	
Number	Type (N/O/F/D/S/*) (C/Z/D/P/U/V)	From	To	Record Character	Substitute Character	Overflow Field Length		Field Name	
3	4	5	6	7	8	9	10	11	12
0	7	F	FC			1			
0	8	F	ND	6	11		Total—7 Characters	ITEM	NUMBER OF ITEM ORDERED
0	9	F	DC	2	5		Total—20 Characters	CLASS	CLASS OF ITEM—JEWELRY, ETC
1	0	F	DC	12	16			PRICE	PRICE OF ITEM PER UNIT
1	1	F	DC	17	22			ORDER	PURCHASE ORDER NUMBER
1	2	F	DC	23	27			QTY	NUMBER OF UNITS ORDERED

Line	Job	Card Match	CCP/Disk Sort Only	Record Type	Comments
Line	Factor 1	Rel.	Factor 2 (Field, Constant, or Keyword)		
Number	Location	EQ NE LT GT LE GE	Constant Keyword Location	Record Name	
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 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72	From To	F/C/K	From To		
0 1	I C	1EQCT		ISSUE	
0 2					

Line	Field	Comments
Number	Location	Field Name
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 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72	From To	
0 7	F FC	FORCED CONTROL FIELD
0 8	F ND	STOCK ITEM NUMBER
0 9	F ND	DATE OF ORDER
1 0	F DC	NUMBER OF UNITS ORDERED
1 1	F DC	CUSTOMER NUMBER
1 2	F DC	INVOICE NUMBER
1 3	F DC	DISCOUNT INFORMATION
1 4	F DC	PRICE OF ITEM PER ITEM
Total-13 Characters		
Total-26 Characters		

Line	Job	Card Match	CCP/Disk Sort Only	Record Type	Comments
Line	Factor 1	Rel.	Factor 2 (Field, Constant, or Keyword)		
Number	Location	EQ NE LT GT LE GE	Constant Keyword Location	Record Name	
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 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72	From To	F/C/K	From To		
0 1	I C	1EQCA		ADJUST	
0 4					
0 5					
0 6					

Line	Field	Comments
Number	Location	Field Name
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 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72	From To	
0 7	F FC	FORCED CONTROL FIELD
0 8	F ND	STOCK ITEM NUMBER
0 9	F DC	ADJUSTMENT CODE
1 0	F DC	NUMBER OF UNITS
1 1	F DC	AUTHORIZATION
1 2	F DC	DATE OF ADJUSTMENT
Total-7 Characters		
Total-16 Characters		

Header Specification

Column 6

H identifies the line as a header line.

Columns 7-12

SORTR identifies the job as a tagalong sort.

Columns 13-17

Three types of records are described: one on each sheet. The total length of the control fields for each type of record is 7, 13, and 7 characters, respectively. The largest of these totals (13) is the entry used in columns 13 through 17.

Column 18

The A stands for ascending order. It indicates the order into which output records (described in the three sets of field lines) will be sorted.

Columns 29-32

The total length of control fields for each type of record described is 7, 13, and 7 respectively. The total length of data fields for each type of record is 20, 26, and 16 respectively. The largest control-field total (13) is added to the largest data-field total (26) and the result (39) is used in columns 29 through 32.

Record Type Specifications

Column 6

Column 6 in line 01 of each of the three sheets contains an I indicating that the lines are include lines. The input records described by these lines are used by the sort program. All other input records are omitted.

Column 8, Columns 9-16, Columns 17-18, Column 19, and Columns 20-39

Column 8 in line 01 of each of the three sheets contains a C indicating that the entire character in position 1 of the input records is to be compared with the constants R, I, and A respectively. Only input records with one of those characters in position 1 will be included in the job.

In the record type lines, columns 9 through 39 have the following meanings:

- Column 9 through 16 identify the input-record position (Position 1) being compared against R, I, and A.
- Columns 17 and 18 indicate the comparison result that determines whether an input record is used (the character in position 1 must equal R, I, or A).
- C in column 19 of each line indicates that the characters with which the input-record character is compared are constants supplied in columns 20 through 39.
- Columns 20 through 39 supply the constants (R, I, and A, respectively) used in the comparison.

Field Specifications

Field specifications differ for the three types of records being described. The records, therefore, are described on separate coding sheets.

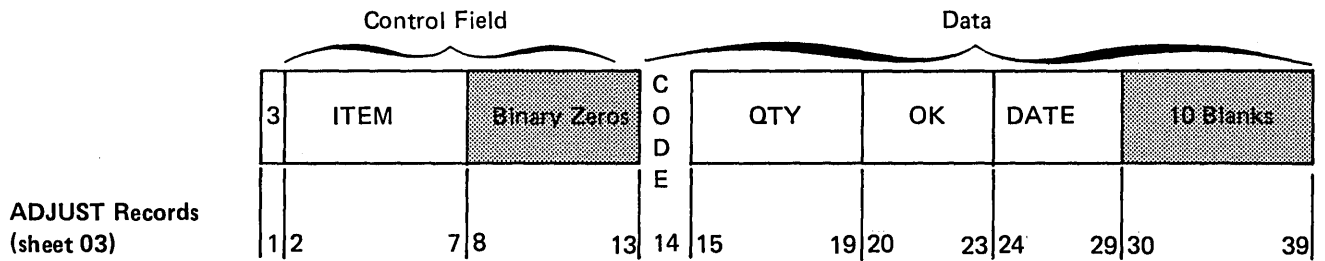
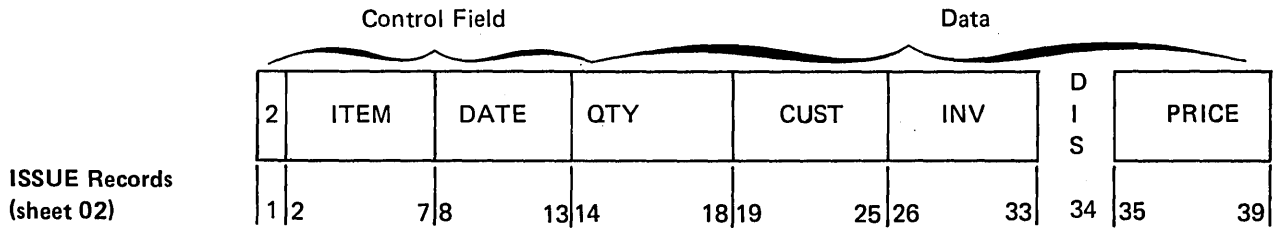
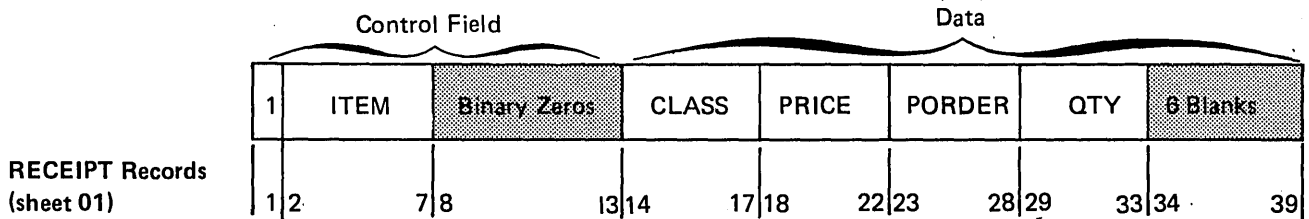
Column 6

F in column 6 of lines 07 through 12 on sheet 01, lines 07 through 14 on sheet 02, and lines 07 through 12 on sheet 03 indicate that the lines are field lines.

Column 7

Column 7 indicates the types of fields being described for each of the three types of records. The records described by sheets 01 and 03 (RECEIPT and ADJUST records respectively) have two control fields. The records described by sheet 02 (ISSUE records) have three. The remaining fields described are data fields.

These diagrams show the formats of the output records that will be created from each of the three types of input records. The number underneath the field identification blocks indicate the positions information will occupy in the output record.



The first control field in each type of output record is a forced control field. The number 1 will begin every RECEIPT output record, 2 will begin ISSUE records, and 3 will begin ADJUST records. This forces all RECEIPT output records to be first, all ISSUE records second, and all ADJUST records third.

The output records for this job must be the same length. The total number of characters in the control fields described for RECEIPT and ADJUST was less than the total for ISSUE records (the control field length for ISSUE on the header is used to assign the absolute length of the control field). Therefore, the sort program added binary zeros at the end of the control fields for RECEIPT and ADJUST output records. Blanks were added at the end of the data portion of RECEIPT and ADJUST output records to make the total length of the records equal to the total length of ISSUE output records.

Column 8

C means that both zone and digit portions of the characters in the fields are used. D means that only the digit portions of the characters in the ITEM control field are used in the sorting process.

Columns 9-16 and Column 18

Column 9 through 16 of the field lines identify the locations of the fields in the input records.

The first control field for each type of record is a forced control field. On each sheet, it is described by an unconditional force line. For this reason, columns 9 through 16 are blank in those lines. Column 18 identifies the character to be used in the forced control field: 1 in output records created from RECEIPT records, 2 in ISSUE output records, and 3 in ADJUST output records.

SAMPLE JOB 5: PRODUCING A LIST OF INVENTORY ITEMS REQUIRING ADJUSTMENTS

- The sort program selects all adjustment records in the transaction file.
- The output record is a control field consisting of the stock item number only.
- Since the output record consists of no data field and, in particular, no summary data fields, the result of the SORTRS job will be to eliminate all duplicate adjustment stock item numbers from the sorted file.
- The output records, each consisting of a unique stock item number that had an adjustment, are to be in ascending order.
- The output provides a list of all stock items that had an adjustment.

Line		Job															Card Match										CCP/Disk Sort Only										Header																			
Number		Card: MATCH, SORT, SELECT, MERGE Largest sum (in bytes) of Control Field Lengths for Any Record Type Disk: SORTR, SORTA, SORTRS Tape: SORTT															Stacker Select U M P P S P U M U O O 25 Number (I/N) 26 Alternate Call. Sel. (S) 27 Print Option 28 Output Option (X)										Output Record Length 29 30 31 32 33 Checkpoint Option (C) 34 Non-Verify Option (N) 35 Reserved 36 Null Output (N) 37 Storage Size to Execute 38 Length Code (R/C)										Record Length 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72										Job Description									
0 0 0 H		SORTRS															6A										6																													
Line		Factor 1															Factor 2 (Field, Constant, or Keyword)										Comments																													
Number		Type (I/O) Continuation (A/O)* C/Z/D/P/U Stacker Select Location From To															Rel. EQ NE LT GT LE GE F/C/K From To										Record Name																													
0 1		I C															1EQCA										ADJUST																													
0 2																																																								
0 3																																																								
0 4																																																								
0 5																																																								
0 6																																																								
Line		Field															Comments																																							
Number		Type (N/O/F/D/S*) (C/Z/D/P/U/V) Record Character Substitute Character Continuation Overflow Field Length Reserved															Field Name																																							
0 7		NC 2 7															ITEM																																							
0 8																																																								
0 9																																																								
1 0																																																								
1 1																																																								
1 2																																																								
1 3																																																								
1 4																																																								

Header Specification

Column 6

H identifies the line as a header line.

Columns 7-12

SORTRS identifies the job as a summary sort.

Columns 13-17

Only one control field is used in this job: ITEM. Its length is 6, which is the entry for columns 13 through 17.

Column 18

The A stands for ascending order.

Columns 29-32

The output record consists of one control field only. The output record length is 6, the same as the control field length.

Record Type Specifications

Column 6

I identifies the record type line 01 as an include line. Input records described by this line are to be sorted. Since this is the only include set, all other records are omitted.

Column 7

Line 01 is the first (and only) include line of this include set consisting of lines 01 and 07. Therefore, column 7 must be blank.

Column 8

C in column 8 tells the sort program to use both the zone and digit portions of the factor 1 and factor 2 fields described in line 01.

Columns 13-16, 17-18, 19, and 20

Only adjustment records containing an A in position 1 are to be included in the sort.

Field Specifications

Column 6

F in column 6 of line 07 indicates that this is a field specification line.

Columns 7 and 8

NC in columns 7 and 8 of line 07 indicates that the ITEM field described is a normal control field consisting of both the zone and digit portions of the characters in the ITEM field of the input records included in the sort.

Columns 9-16

Columns 9 through 16 of line 07 identify the location of the ITEM field (having a length of 6) that is to be used as the control field.

SAMPLE JOB 6: PRODUCING A SUMMARY HISTORY FILE OF ISSUES BY CUSTOMER NUMBER

- The sort program selects all ISSUE records from the transaction file.
- A summary sort (SORTRS) is used to create a file consisting of each customer to whom any stock item was issued.
- Each output record consists of a unique customer number, the total number of issues made to that number, and the total dollar value of all the issues.
- The output records are to be in ascending order by customer number.

Line		Job	Card Match		Stacker Select		Output Record Length	CCP/Disk Sort Only	Job Description		
Number	Type	Card: MATCH, SORT, SELECT, MERGE Disk: SORTR, SORTA, SORTRS Tape: SORTT	Largest sum (in bytes) of Control Field Lengths for Any Record Type	S	M	P	U	Number (N)	Print Option	Output Option (X)	Record Length
3	4	5	6	7	8	9	10	11	12	13	14
00	H	SORTRS	7A							23	

Line	Factor 1	Rel.	Factor 2 (Field, Constant, or Keyword)		Comments
Number	Location	F/C/K	From	To	Record Name
3	4	5	6	7	8
01	IC		1E	QC	ISSUE

Line	Record Character	Substitute Character	Continuation	Overflow Field Length	Reserved	Field Name	Comments
Number	Location	From	To				
3	4	5	6	7	8	9	10
07	NC	13	19		Total—7 Characters	CUST	BLANK SPACE
08	DV						#
09	DV						#
10	SD	40	42		Total—16 Characters	CON1	TOTAL # OF ISSUES
11	DM						BLANK SPACE
12	DM						\$
13	SU	43	50			COST	TOTAL \$ OF ISSUES
14	SV						OVERFLOW INDICATOR IS X

Header Specification

Column 6

H identifies the line as a header line.

Columns 7-12

SORTRS identifies the job as a summary sort.

Columns 13-17

The work record as well as the output record consists of a control field made up of the customer number. Its length is 7, which is the entry for columns 13 through 17.

Column 18

The A indicates that the control field is to be sorted in ascending order.

Columns 29-32

The control field is not to be dropped when the sort program is writing the output file. The length of the output record is the control field (7 characters as specified in line 07) added to the data fields (16 characters as specified in lines 08 through 14). This totals 23 characters, and is the entry for columns 29 through 32.

Record Type Specifications

Column 6

I identifies the record type line 01 as an include line. Input records described by this line are to be sorted. Since this is the only include set, all other input records are omitted.

Column 7

Line 01 is the first (and only) include line of this include set consisting of lines 01 through 14; therefore, column 7 must be blank.

Columns 8, 16, 17-18, 19, and 20

Only ISSUE records containing the full character I in position 1 of the input records are to be included in the sort.

Field Specifications

Column 6

F in column 6 of lines 07 through 14 indicates that these lines are field lines.

Columns 7 and 8

NC in line 07 specifies that this is the control field to be sorted in the sequence specified in column 18 of the header line (ascending order) using both the zone and digit portions of the characters.

Lines 08, 09, 11, and 12 specify unconditional 1-character force fields.

Line 10 indicates that this is a summary data field. When the data in the input record (positions 40 through 42) is used to build a work record, only the digit portions of the characters will be used because CON1 is an unsigned decimal number. Using a D in column 8 with a signed number causes the absolute value of that signed number to be used.

Line 13 also is specified as a summary data field. However, U in column 8 indicates that this is signed zoned decimal data. The sign is to be moved to the work record and used in subsequent summary additions.

Line 14 is specified as the summary overflow indicator byte.

Columns 9-16

Entries are placed in the From and To columns for lines 07, 10, and 13 to indicate the control field (CUST), digit summary data field (CON1), and the signed zoned decimal summary field (COST), respectively.

For lines 08, 09, 11, and 12, the entries are blank because these are unconditional 1-character force fields.

For line 14, the entries are blank because this line is an overflow indicator specification.

Column 17

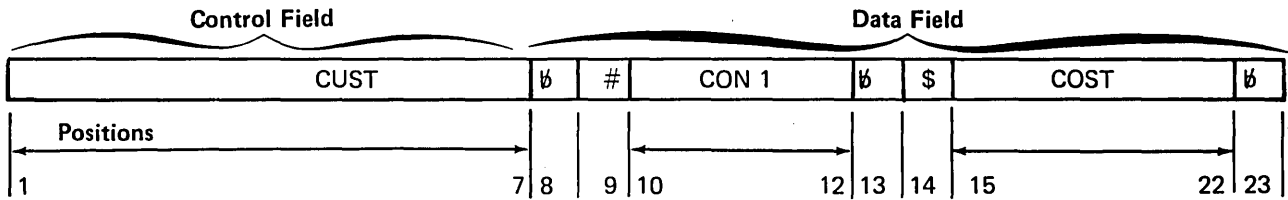
The X in line 14 defines the field as an overflow indicator field.

Column 18

This column is only used for lines 08, 09, 11, 12, and 14.
The line numbers and the forced value in the data fields are:

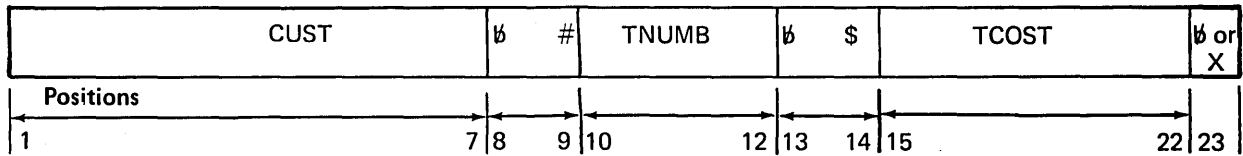
- 08 ¢ (blank)
- 09 #
- 11 ¢ (blank)
- 12 \$
- 14 ¢ (blank)

The initial work record created from the ISSUE records will look like this:



Since this is a summary sort, the CON1 and COST fields will be summarized for records having identical control fields (CUST). These portions 10 through 12 (CON1) in the work records will be summarized to reflect the total number of issues (TNUMB) for a given customer number. Likewise, positions 15 through 22 (COST) will be summarized to reflect the total cost of all issues (TCOST).

The output record will look like this:



A blank in position 23 indicates that TNUMB and TCOST are valid sums. X in position 23 indicates that overflow occurred when the sort program was summarizing TNUMB and/or TCOST for this customer number, resulting in an invalid sum or sums.

**SAMPLE JOB 7: SORTING SUMMARY HISTORY FILE
BY DESCENDING TOTAL DOLLARS ISSUED
(NEGATIVE VALUES POSSIBLE)**

- The input for this job is the output file created by *Sample Job 6*. Each input record has the following format:

Record Position	Field Name	Contents
1-7	CUST	Customer number
8-9		Constants: ¢ #
10-12	TNUMB	Total number of issues to customer
13-14		Constants: ¢ \$
15-22	TCOST	Total cost of all issues to customer
23	OFLOW	Overflow indicator: ¢ – TNUMB and TCOST are valid sums X – Invalid sum or sums; overflow occurred while the sort program was summarizing TNUMB and/or TCOST

- Only records with valid TNUMB and TCOST will be used.
- A record sort of the valid records is desired; that is, the format of the output record is to be identical to the format of the input record.
- The records are to be sorted by total dollars issued (TCOST) in descending order. TCOST could be negative for some customer numbers. For this reason, a sort that considers the sign must occur.

SEQUENCE SPECIFICATIONS
Header

Line	Job	Card Match	Stacker Select	Output Record Length	CCP/Disk Sort Only	Job Description
3	Card: MATCH, SORT, SELECT, MERGE	Largest sum (in bytes) of Control Field Lengths for Any Record Type	C U S M P U O	23		
4	Disk: SORTR, SORTA, SORTRS					
5	Tape: SORTT					
0	000H					

Record Type

Line	Factor 1	Rel.	Factor 2 (Field, Constant, or Keyword)	Comments
3	Location	EQ	Constant	
4	From To	NE	Keyword	
5	From To	LT		
6	From To	GT		
7	From To	LE		
8	From To	GE		
0	0			
1	C			INVALID SUMS
2				INCLUDE ALL POSITIVE TCOST
3				INFORMATION
4				
5				
6				

Field

Line	Location	Forced	Reserved	Comments
3	From To	Record Character Substitution Character Continuation	Overflow Field Length	
4	From To			
5	From To			
6	From To			
7	15			
8	22			
9	1			
23				
0				
7	NU			
8	DC			
9				
10				
11				
12				
13				
14				

Header Specification

Column 6

H identifies the line as a header line.

Columns 7-12

SORTR identifies the job as a tagalong sort.

Columns 13-17

One type of record is specified to be included. The total length of the control fields for this type is 8 (the entry in column 17).

Column 18

D stands for descending order. The largest positive number comes first; the largest negative number (the negative number having the largest absolute value) comes last.

For example, the absolute values of -19 and -99 are 19 and 99. -99 is more negative than -19 because its absolute value is greater than 19.

Column 28

X indicates that the control field is to be dropped when the output file is created.

Columns 29-32

When the control field is dropped, only the data portion remains. This data portion is the entire input record. Thus, 23 is entered in columns 29 through 32.

Record Type Specifications

Columns 6 and 7

Two record types are specified. One type is to be omitted (O in column 6). One is to be included. Each record type is specified by one record type line. Since these lines are the first (as well as the last) record type lines in their omit set and include set, column 7 must be blank. Note that the asterisk in column 7, line 03, indicates that this line is only a comment line.

Columns 8, 9-16, 17-18, 19, and 20-39

For line 01, these columns specify that if position 23 in the input record contains the character constant X, the record is to be omitted.

Line 02 indicates that all records not omitted by line 01 are to be included. For each record included, a work record is formed by the field specifications lines 07 and 08. Note that the control field is an eight-position zoned field (line 07).

For a more complete discussion of numeric data, see *Column 8* and *Columns 9-16* under *Column Descriptions* in Chapter 3.

Field Specifications

Column 6

F in column 6 of lines 07 and 08 indicates that these lines are field specification lines.

Columns 7, 8, 9-16, and 18

- Line 07 indicates unpacked data in the control field from positions 15 through 22 of the input record. (This control field is dropped because an X is coded in column 28 of the header specifications.)
- Line 08 indicates that the entire record is included as data.

Appendix A. Calculating the Sizes of Files for Sort

INPUT FILE

Use this formula to calculate the size (in blocks) of an input file; round the result up to the nearest whole number.

$$\text{BLOCKS} = \frac{\text{number of records in the file} \times \text{input record length}}{\text{number of bytes in a block (2,560)}}$$

OUTPUT FILE

Use this formula to calculate how many blocks the output file requires; round the result up to the nearest whole number.

$$\text{BLOCKS} = \frac{\text{number of records that will be selected for sorting} \times \text{output record length}}{\text{number of bytes in a block (2,560)}}$$

If you use the RECORDS parameter in the output file statement:

$$\text{RECORDS} = \text{number of records that will be selected for sorting}$$

WORK FILE

Use either of the following formulas to approximate how many blocks or records to specify on the FILE statement for the work file; whichever formula you use, round your results up to the nearest whole number.

$$\text{BLOCKS} = 2 + \left(\frac{\text{number of records that will be selected for sorting} \times \text{work record length} \times 1.1}{\text{number of bytes in a block (2,560)}} \right)$$

or

$$\text{RECORDS} = \text{number of records that will be selected for sorting} + \left(\frac{2 \times \text{number of bytes in a block (2,560)}}{\text{work record length}} \right)$$

Note: The sort program includes an 8-byte work block vector in every work block written to the work file. Therefore, the work file must be larger than a file that would hold the exact number of records to be sorted. This is why both formulas approximate the number of records or blocks to specify on the FILE statement for the work file.

Work Record Length

Type of Sort	Type of Output	Work Record Length
SORTA	Address output	Length of control fields + 3
SORTR or SORTRS	Control fields only	Length of control fields as specified in the header line
	Control fields and data	Length of data + length of control fields (output record length specified in the header line)
	Data only	Length of data + length of control fields (output record length specified in the header line + the control field length specified in the header line)

Appendix B. Collating Sequence

Standard EBCDIC Collating Sequence												
When Both Zone and Digit Portions of Characters are Used			When Only the Zone Portion of Characters is Used			When Only the Digit Portion of Characters is Used						
Order in the Sequence ¹	Character	Corresponding Hexadecimal Number ²	Order in the Sequence ¹	Character	Corresponding Hexadecimal Number ²	Order in the Sequence ¹	Character	Corresponding Hexadecimal Number ²				
1 (lowest)	blank	40	1 (lowest)	ϕ	4A	1 (lowest)	blank	40				
2	ϕ	4A		<	4B		&	50				
3	.	4B		(4C		- (minus)	60				
4	<	4C		+	4D		{	C0				
5	(4D			4E		}	D0				
6	+	4E		!	4F		0	F0				
7		4F	2	\$	5A	2	/	61				
8	&	50		*	5B		a	81				
9	!	5A)	5C		j	91				
10	\$	5B		;	5D		~	A1				
11	*	5C		┘	5E		A	C1				
12)	5D		/	5F		J	D1				
13	;	5E	3		61	3	\	E1				
14	┘	5F		/	6A		1	F1				
16	- (minus)	60		,	6B		3	b	82			
17	/	61		%	6C			k	92			
15		6A		_ (underscore)	6D			s	A2			
18	,	6B		>	6E			B	C2			
19	%	6C	?	6F	K	D2						
20	_ (underscore)	6D	4	\	79	4		S	E2			
21	>	6E		:	7A		2	F2				
22	?	6F		#	7B		4	c	83			
23	\	79		@	7C			l	93			
24	:	7A		' (apostrophe)	7D			t	A3			
25	#	7B		=	7E			C	C3			
26	@	7C	"	7F	L	D3						
27	' (apostrophe)	7D	5	a	81	5		T	E3			
28	=	7E		b	82		3	F3				
29	"	7F		c	83		5	d	84			
30	a	81		d	84			m	94			
31	b	82		e	85			u	A4			
32	c	83		f	86			D	C4			
33	d	84	g	87	M	D4						
34	e	85	h	88	U	E4						
35	f	86	6	i	89	4	4	F4				
36	g	87		j	91		5	e	85			
37	h	88		k	92			6	n	95		
38	i	89		l	93				v	A5		
39	j	91		m	94				E	C5		
40	k	92		n	95				N	D5		
41	l	93	o	96	V	E5						
42	m	94	6	p	97	5	5		F5			
43	n	95		q	98		6	f	86			
44	o	96		r	99			7	o	96		
45	p	97		6	j				91	w	A6	
46	q	98			6				k	92	F	C6
47	r	99							6	l	93	O
48	s	90	6			m				94	W	E6
49	t	91				6	n			95	6	F6
50	u	92					6	o		96	6	
51	v	93		6				p		97		
52	w	94			6			q		98		
53	x	95						6	r	99		
54	y	96	6									
55	z	97				6						

¹When several characters share the same position in the sequence, they are considered equal. For example, if you are using only the digit portions of characters, b, k, s, B, K, S, and 2 (position 3) are considered equal.

²This is the number you use in ALTSEQ statements to identify a character that you want to shift to a different order in the sequence.

Standard EBCDIC Collating Sequence (cont'd.)										
When Both Zone and Digit Portions of Characters are Used			When Only the Zone Portion of Characters is Used			When Only the Digit Portion of Characters is Used				
Order in the Sequence ¹	Character	Corresponding Hexadecimal Number ²	Order in the Sequence ¹	Character	Corresponding Hexadecimal Number ²	Order in the Sequence ¹	Character	Corresponding Hexadecimal Number ²		
48	~	A1	7	~	A1	8	g	87		
49	s	A2		s	A2		p	97		
50	t	A3		t	A3		x	A7		
51	u	A4		u	A4		G	C7		
52	v	A5		v	A5		P	D7		
53	w	A6		w	A6		X	E7		
54	x	A7		x	A7		7	F7		
55	y	A8		y	A8					
56	z	A9		z	A9					
57	{	C0	8	&	50	9	h	88		
58	A	C1		{	C0		H	C8		
59	B	C2		A	C1		Q	D8		
60	C	C3		B	C2		Y	E8		
61	D	C4		C	C3		8	F8		
62	E	C5		D	C4					
63	F	C6		E	C5					
64	G	C7		F	C6		10	\	79	
65	H	C8		G	C7			i	89	
66	I	C9	H	C8	r	99				
67	}	D0	I	C9	z	A9				
68	J	D1			I	C9				
69	K	D2			R	D9				
70	L	D3			Z	E9				
71	M	D4			9	F9				
72	N	D5								
73	O	D6	9	-(minus)	60	11	¢	4A		
74	P	D7		}	D0		!	5A		
75	Q	D8		J	D1		:	6A		
76	R	D9		K	D2		:	7A		
77	\	E0		L	D3					
78	S	E2		M	D4		12	.	4B	
79	T	E3		N	D5			\$	5B	
80	U	E4		O	D6			,	6B	
81	V	E5		P	D7			#	7B	
82	W	E6	Q	D8						
83	X	E7	R	D9	<	4C				
84	Y	E8			*	5C				
85	Z	E9			%	6C				
86	0	F0			@	7C				
87	1	F1	10	\	E1	13	(4D		
88	2	F2		S	E2)	5D		
89	3	F3		T	E3			6D		
90	4	F4		U	E4					
91	5	F5		V	E5					
92	6	F6		W	E6					
93	7	F7		X	E7					
94	8	F8		Y	E8					
95	9	F9		Z	E9					
(highest)			11 (highest)	blank	40	14	(underscore)	7D		
				0	F0					
				1	F1		15	+	4E	
				2	F2			:	5E	
				3	F3			>	6E	
				4	F4			=	7E	
				5	F5					
				6	F6			16 (highest)		4F
				7	F7				J	5F
			8	F8	?	6F				
			9	F9	"	7F				

¹ When several characters share the same position in the sequence, they are considered equal. For example, if you are using only the digit portions of characters, b, k, s, B, K, S, and 2 (position 3) are considered equal.

² This is the number you use in ALTSEQ statements to identify a character that you want to shift to a different order in the sequence.

STANDARD COLLATING SEQUENCE

The Standard EBCDIC Collating Sequence chart shows the three standard collating sequences:

- Standard collating sequence when both zone and digit portions of characters are used.
- Standard collating sequence when only the zone portion of characters is used.
- Standard collating sequence when only the digit portion of characters is used.

ALTERNATE COLLATING SEQUENCE

If you want to change the standard collating sequence, you must code an S in column 26 of the header specification and code ALTSEQ statements. You can code as many ALTSEQ statements as you need. Each one must begin with ALTSEQ and can contain a maximum of 96 positions.

Note: Do not use a packed or zoned factor 1 in an include or omit record type specification (P or U in column 8) if you specify an alternate collating sequence.

Coding Rules

1. Code ALTSEQ in the first six positions to tell sort you want to alter the standard collating sequence.

ALTSEQ

2. Leave the next two positions blank.

ALTSEQ

3. Enter the hexadecimal equivalent¹ of the character you are taking out of its normal sequence.

ALTSEQ hn

4. Enter the hexadecimal equivalent¹ of the value that the character specified in columns 9 and 10 is to assume in the collating sequence.

ALTSEQ hnhn

5. Enter as many pairs (from rules 3 and 4) as the number of characters you are taking out of normal sequence.

6. Leave no space between sets of hexadecimal numbers.

ALTSEQ hnhnhnhn

7. When you reach the end of one statement, you can continue on the next specification line (follow rules 1 through 6).

8. Enter a double asterisk in positions 1 and 2 to indicate the end of the ALTSEQ statements.

ALTSEQ	hnhn	maximum of 96 positions
ALTSEQ	hnhn	maximum of 96 positions
ALTSEQ	hnhn	maximum of 96 positions
**		

¹See Standard EBCDIC Collating Sequence chart for hexadecimal equivalents of all characters.

Input Order When ALTSEQ Statements Are Used

ALTSEQ statements follow the header line. When you use an alternate collating sequence, your input must be in this order:

1. OCL statements
2. Sequence specifications
 - a. Header line
 - b. ALTSEQ statements
 - c. **
 - d. Record type and field lines as required
3. // END

Programming Considerations

When you move a character into the sequence position normally assigned to another character, both the new and the original character occupy the same position and are considered equal (see the first sample ALTSEQ statement). If you do not want the two characters to be equal, you must also move the character that normally occupies that position (see the second ALTSEQ statement).

Effect of ALTSEQ Statements on Other Coding

ALTSEQ statements can change:

- Factor 1 and factor 2
- Normal and opposite control fields
- Control field characters before they are replaced or added to by forced fields

ALTSEQ statements never change data fields in records or forced control field characters.

Sample ALTSEQ Statements

To change the Sequence of One character

ALTSEQ ALTSEQ 40F0
Statements **

Explanation Blank (40) is moved to the position in the collating sequence occupied by zero (F0). Blanks and zeros, therefore, are considered equal.

To Change the Sequence of Several Characters

ALTSEQ ALTSEQ 5BC8C8C9C9D1
Statements **

Explanation 5BC8 moves \$ (5B) to the position occupied by H (C8).
C8C9 moves H (C8) to the position occupied by I (C9).
C9D1 moves I (C9) to the position occupied by J (D1).
The \$, therefore, has been inserted between G and H, as follows:
A, B, C, D, E, F, G, \$, H, I, (I and J occupy the same position).

PROGRAMMING AIDS

You can use a *Translation Table and Alternate Collating Sequence Coding Sheet*, GX21-9096, to keep track of changes you make to your collating sequence. A sample sheet is included for your information.



TRANSLATION TABLE AND ALTERNATE COLLATING SEQUENCE CODING SHEET

*No. of sheets per pad may vary slightly

Code	Graphic	Entry	Replaced By/Takes Place Of
00000000		00	
00000001		01	
00000010		02	
00000011		03	
00000100		04	
00000101		05	
00000110		06	
00000111		07	
00001000		08	
00001001		09	
00001010		0A	
00001011		0B	
00001100		0C	
00001101		0D	
00001110		0E	
00001111		0F	
00010000		10	
00010001		11	
00010010		12	
00010011		13	
00010100		14	
00010101		15	
00010110		16	
00010111		17	
00011000		18	
00011001		19	
00011010		1A	
00011011		1B	
00011100		1C	
00011101		1D	
00011110		1E	
00011111		1F	
00100000		20	
00100001		21	
00100010		22	
00100011		23	
00100100		24	
00100101		25	
00100110		26	
00100111		27	
00101000		28	
00101001		29	
00101010		2A	
00101011		2B	
00101100		2C	
00101101		2D	
00101110		2E	
00101111		2F	
00110000		30	
00110001		31	
00110010		32	

Code	Graphic	Entry	Replaced By/Takes Place Of
00110011		33	
00110100		34	
00110101		35	
00110110		36	
00110111		37	
00111000		38	
00111001		39	
00111010		3A	
00111011		3B	
00111100		3C	
00111101		3D	
00111110		3E	
00111111		3F	
01000000	Blank	40	
01000001		41	
01000010		42	
01000011		43	
01000100		44	
01000101		45	
01000110		46	
01000111		47	
01001000		48	
01001001		49	
01001010	£	4A	
01001011	.	4B	
01001100	<	4C	
01001101	(4D	
01001110	+	4E	
01001111		4F	
01010000	&	50	
01010001		51	
01010010		52	
01010011		53	
01010100		54	
01010101		55	
01010110		56	
01010111		57	
01011000		58	
01011001		59	
01011010	!	5A	
01011011	\$	5B	
01011100	*	5C	
01011101	!	5D	
01011110	:	5E	
01011111	~	5F	
01100000	-	60	
01100001	/	61	
01100010		62	
01100011		63	
01100100		64	
01100101		65	

Code	Graphic	Entry	Replaced By/Takes Place Of
01100110		66	
01100111		67	
01101000		68	
01101001		69	
01101010	!	6A	
01101011	,	6B	
01101100	%	6C	
01101101	-	6D	
01101110	>	6E	
01101111	?	6F	
01110000		70	
01110001		71	
01110010		72	
01110011		73	
01110100		74	
01110101		75	
01110110		76	
01110111		77	
01111000		78	
01111001	\	79	
01111010	:	7A	
01111011	#	7B	
01111100	@	7C	
01111101	'	7D	
01111110	=	7E	
01111111	"	7F	
10000000		80	
10000001	a	81	
10000010	b	82	
10000011	c	83	
10000100	d	84	
10000101	e	85	
10000110	f	86	
10000111	g	87	
10001000	h	88	
10001001	i	89	
10001010		8A	
10001011		8B	
10001100		8C	
10001101		8D	
10001110		8E	
10001111		8F	
10010000		90	
10010001	j	91	
10010010	k	92	
10010011	l	93	
10010100	m	94	
10010101	n	95	
10010110	o	96	
10010111	p	97	
10011000	q	98	

Code	Graphic	Entry	Replaced By/Takes Place Of
10011001	r	99	
10011010		9A	
10011011		9B	
10011100		9C	
10011101		9D	
10011110		9E	
10011111		9F	
10100000		A0	
10100001	~	A1	
10100010	s	A2	
10100011	t	A3	
10100100	u	A4	
10100101	v	A5	
10100110	w	A6	
10100111	x	A7	
10101000	y	A8	
10101001	z	A9	
10101010		AA	
10101011		AB	
10101100		AC	
10101101		AD	
10101110		AE	
10101111		AF	
10110000		B0	
10110001		B1	
10110010		B2	
10110011		B3	
10110100		B4	
10110101		B5	
10110110		B6	
10110111		B7	
10111000		B8	
10111001		B9	
10111010		BA	
10111011		BB	
10111100		BC	
10111101		BD	
10111110		BE	
10111111		BF	
11000000	{	C0	
11000001	A	C1	
11000010	B	C2	
11000011	C	C3	
11000100	D	C4	
11000101	E	C5	
11000110	F	C6	
11000111	G	C7	
11001000	H	C8	
11001001	I	C9	
11001010		CA	
11001011		CB	

Code	Graphic	Entry	Replaced By/Takes Place Of
11001100		CC	
11001101		CD	
11001110		CE	
11001111		CF	
11010000	}	D0	
11010001	J	D1	
11010010	K	D2	
11010011	L	D3	
11010100	M	D4	
11010101	N	D5	
11010110	O	D6	
11010111	P	D7	
11011000	Q	D8	
11011001	R	D9	
11011010		DA	
11011011		DB	
11011100		DC	
11011101		DD	
11011110		DE	
11011111		DF	
11100000		EO	
11100001	\	E1	
11100010	S	E2	
11100011	T	E3	
11100100	U	E4	
11100101	V	E5	
11100110	W	E6	
11100111	X	E7	
11101000	Y	E8	
11101001	Z	E9	
11101010		EA	
11101011		EB	
11101100		EC	
11101101		ED	
11101110		EE	
11101111		EF	
11110000	0	FO	
11110001	1	F1	
11110010	2	F2	
11110011	3	F3	
11110100	4	F4	
11110101	5	F5	
11110110	6	F6	
11110111	7	F7	
11111000	8	F8	
11111001	9	F9	
11111010		FA	
11111011		FB	
11111100		FC	
11111101		FD	
11111110		FE	
11111111		FF	

Note: Not all graphic symbols shown here are available on all systems.

Hexadecimal numbers of characters in the standard collating sequences.

Where you record the hexadecimal number of the character you are going to put in that relative position in the sequence.

Appendix C. Header, Record Type, and Field Specification Column Summaries

HEADER SPECIFICATION COLUMN SUMMARY



Column Summary		
Columns	Entries	Explanation
1-2	00-99	Page number.
3-5	000	Header line number.
6	H	Header line identification.
7-12	SORTA	Addrout sort job.
	SORTR	Tagalong sort job.
	SORTRS	Summary tagalong sort job.
13-17	1-256	Largest sum (in bytes) of control field lengths for any record type.
18	A	Records in sorted file to be in ascending order by control fields.
	D	Records in sorted file to be in descending order by control fields.
	S	Not used in System/34 sort jobs.
19-25	Blank	Not used in System/34 sort jobs.
26	Blank	Use standard collating sequence in compare operations.
	S	Use an alternate collating sequence in compare operations. ALTSEQ statements will define the collating sequence to be used.
27	0 or Blank	Print: Sequence specifications Diagnostic messages Program-status messages
		Display: Action messages Displayed messages
	1	Print: Program-status messages Display: Action messages Displayed messages
	2	Print: Action messages only Display: Displayed messages
28	Blank	Keep control fields in output records in tagalong sort jobs.
	X	Drop control fields from output records in tagalong sort jobs.
29-32	1-4096	Length of output records in tagalong sort jobs.
33	Blank	Not used in System/34 sort jobs.
34	N	Data written on the work file will not be verified.
	Blank or character other than N	Data written on the work file will be verified.
35	Blank	Reserved for system use.
36	Blank	Operator message SORT-7724 will be issued.
	N	Operator message SORT-7724 will not be issued.
37-39	Blank	Not used in System/34 sort jobs.
40-72	Any characters	Comments.
<div style="display: flex; align-items: center; margin-bottom: 5px;"> <div style="width: 20px; height: 10px; background-color: #cccccc; margin-right: 5px;"></div> Columns that must be filled in for all sort jobs. </div> <div style="display: flex; align-items: center;"> <div style="width: 20px; height: 10px; background: repeating-linear-gradient(45deg, transparent, transparent 2px, #cccccc 2px, #cccccc 4px); margin-right: 5px;"></div> Columns that must be filled in for tagalong sort jobs. </div>		

RECORD TYPE SPECIFICATION COLUMN SUMMARY

Column Summary		
Columns	Entries	Explanation
1-2	00-99	Page number.
3-5	01x-06x	Line number. You can leave column 5 (x) blank, or enter any value to keep the specifications in ascending order.
6	I	Include line.
	O	Omit line.
7	Blank	This is the first line of a set of I or O record type lines.
	A	AND lines. These specifications continue the definition of the record described on the preceding line.
	O	OR line. These specifications define a different type of record than the one on the previous line.
	*	Comment line.
8	C	Use both zone and digit portions of characters.
	Z	Use only zone portion of 1-character field.
	D	Use only digit portions of characters.
	P	Signed packed decimal data.
	U	Signed zoned decimal data.
9-12	1-4096	The input record position in which the factor 1 field begins (blank if field is only one position long).
13-16	1-4096	The input record position in which the factor 1 field ends.
17-18	EQ	Factor 1 must equal factor 2.
	NE	Factor 1 must not equal factor 2.
	LT	Factor 1 must be less than factor 2.
	GT	Factor 1 must be greater than factor 2.
	LE	Factor 1 must be less than or equal to factor 2.
	GE	Factor 1 must be greater than or equal to factor 2.
19	F	Factor 2 is another field in the same input record.
	C	Factor 2 is a constant.
	K	Factor 2 is a keyword.
20-23	1-4096	The input record position in which the factor 2 field begins (blank if field is only one position long).
24-27	1-4096	The input record position in which the factor 2 field ends.
28-39	Any characters	The factor 2 constant.
40-72	Any characters	Comments.
<div style="display: flex; align-items: center;"> <div style="width: 100px; height: 15px; background-color: #cccccc; margin-right: 10px;"></div> Columns that must be filled in. </div>		

FIELD SPECIFICATION COLUMN SUMMARY

Column Summary		
Columns	Contents	Explanation
1-2	00-99	Page number.
3-5	07x-14x	Line number. You can leave column 5 (x) blank, or enter any value to keep the specifications in ascending order.
6	F	Field specification line.
7	N	Normal control field.
	O	Opposite control field.
	F	Forced control field.
	D	Data field.
	S	Summary data field.
	*	Comment line.
8	C	Use both zone and digit portions of characters in the field.
	Z	Use only zone portion of 1-character field.
	D	Use only digit portions of characters in the field.
	P	Signed packed decimal data.
	U	Signed zoned decimal data.
	V	Force a data character into the data field.
9-12	1-4096	Start position of a field in the record (can be blank if the field is 1 character long).
13-16	1-4096	End position of a field in the record.
17	Any character	Forced control fields; the character you want the sort program to change. Also used for summary overflow indicator fields; the character to be used for the overflow indicator.
18	Any character	Forced control or data field; the character you want the sort program to substitute. Also used for summary overflow indicator fields; the character to which the overflow field is initialized.
19	Blank	Forced control field line is not a continuation of the preceding line.
	Any character other than blank	Forced control field line is a continuation of the preceding line.
20-22	1-256	Overflow field length entry for summary tagalong sort only.
23-39	Blank	Reserved for system use.
40-72	Any characters	Comments.

	Columns that must be filled in for all sort jobs.
	Columns that must be filled in when forced control fields are being used.

Appendix D. Integer (Binary) and Real Number Sorting

This appendix indicates how either positive and negative integers or positive and negative real numbers in control fields can be used to sort records in a file.

Coded sequence specifications sheets for sample sort jobs are included, followed by a brief discussion of the entries made on the sheets.

Note: For a detailed discussion of sequence specifications sheet entries, see Chapters 2, 3, 4, and 7.

SAMPLE SORT JOB 1. SORTING ON A CONTROL FIELD THAT CONTAINS EITHER A POSITIVE INTEGER OR A POSITIVE REAL NUMBER

Line		Job													Card Match													CCP/Disk Sort Only													Job Description																									
Number		Card: MATCH, SORT, SELECT, MERGE Disk: SORTR, SORTA, SORTRS Tape: SORTT													Largest sum (in bytes) of Control Field Lengths for Any Record Type													Stacker Select S U M M P P S P S U M M P P S P													Output Record Length													Record Length												
Type		Sequence (A/D/S)													Number (I/N) Alternate Coll. Seq. (S) Print Option Output Option (X)													Checkpoint Option (C) Non-Verify Option (N) Reserved													Null Output (N) Storage Size to Execute Length Code (R/C)																									
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 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72																									
0 0		H SORTR													4A													64																																						

Line		Factor 1													Factor 2 (Field, Constant, or Keyword)													Comments																									
Number		Location From To													EQ NE LT GT LE GE F/C/K													Constant Keyword													Record Name												
Type (U/D) Combination (A/O) CZ/D/P/U		Stacker Select																																																			
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 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72																									
0 1																																																					
0 2																																																					
0 3																																																					
0 4																																																					
0 5																																																					
0 6																																																					

Line		Field													Comments																									
Number		Location From To													Record Character Substitute Character Continuation Overflow Field Length Reserved													Field Name												
Type (N/O/F/D/S/I) CZ/D/P/U/V																																								
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 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72												
0 7		NC 1 4																										CONTROL FIELD												
0 8		DC 5 64																										REST OF DATA												
0 9																																								
1 0																																								
1 1																																								
1 2																																								
1 3																																								
1 4																																								

Header Specification

Column 6

H identifies the line as a header line.

Columns 7-12

SORTR identifies the job as a tagalong sort.

Columns 13-17

The control field that contains either a positive integer or a positive real number has a length of 4 (the entry in column 17).

Column 18

The A specifies ascending order. The smallest positive integer or positive real number comes first; the largest positive integer or positive real number comes last in the sort sequence.

Columns 29-32

The control field is not dropped when data is written to the output file (column 28 is blank). Therefore, the length of the output record equals the length of the control field (4 bytes) added to the length of the data field (60 characters) specified by lines 07 and 08. This output record length totals 64, which is the entry for columns 29 through 32.

Record Type Specifications

Because all input records are being used and all have the same field specifications, no record type specifications are needed (include-all is implied).

Field Specifications

Column 6

F identifies the lines as field specification lines.

Columns 7, 8, and 9-16

Line 07 indicates that the control field is in positions 1 through 4 of the input record. This control field is included in the sorted output file (column 28 of the header line is blank).

Line 08 indicates that the other 60 input record characters are included as data in the sorted output file.

SAMPLE SORT JOB 2. SORTING ON A CONTROL FIELD THAT CONTAINS ONLY A POSITIVE OR NEGATIVE INTEGER

Line		Job		Card Match														CCP/Disk Sort Only		Job Description													
Number		Type		Stacker Select														Record Length															
3 4 5		7 8 9 10 11 12		U S M P U O														40 41 42 43 44															
0 0 0		H		S A														X 64															

Line		Factor 1														Factor 2 (Field, Constant, or Keyword)														Comments													
Number		Location														Location														Record Name													
3 4 5		From To														From To																											
0 1	I															" "														ALL POSITIVE INTEGERS													
0 2																																											
0 3																																											
0 4																																											
0 5																																											
0 6																																											

Line		Field														Comments																											
Number		Location														Field Name																											
3 4 5		From To																																									
0 7	F															2														FORCE 2 FOR POSITIVE INTEGERS													
0 8	F	1 4																												CONTROL FIELD													
0 9	F	1 64																												DATA FIELD													
1 0	F																																										
1 1	F																																										
1 2	F																																										
1 3	F																																										
1 4	F																																										

Header Specification

Column 6

H identifies the line as a header line.

Columns 7-12

SORTR identifies the job as a tagalong sort.

Columns 13-17

The total length of the control field (for either a positive or a negative integer) is 5 bytes (the entry in column 17).

Column 18

The A stands for ascending order. Since negative integers are stored in two's complement form, the smallest negative integer comes first; the largest positive integer comes last in the sort sequence (for example: -2, -1, 0, 1, 2).

Column 28

X indicates that the control field is dropped when data is written to the output file.

Columns 29-32

When the control field is dropped, only the data portion remains. The data portion in this sample job is the entire input record. Therefore, 64 is entered in columns 29 through 32.

Record Type Specifications

Column 6

Two record types are specified (positive integers and negative integers). Both record types are to be included (I in column 6 of line 01, pages 01 and 02).

Columns 8, 9-16, 17-18, 19, and 20-39

On line 01 (page 01), these columns specify that if position 1 (the leftmost byte of the 4-byte control field from the input record) is less than or equal to the character constant " (see note), the work record will have a character 2 forced into the first byte of its 5-byte control field (line 07 on page 01). This include line selects only positive integers. Line 01 (page 02), which is an include-all line, indicates that all records not described by the first include line (page 01) are also included in the sort. The first include line (line 01 page 01) selects all positive integers; therefore, this include-all line (page 02) selects all negative integers, which will have a character 1 forced into the first byte of their work record control field (line 07 on page 02). All negative integers are forced in front of all positive integers in the sort sequence.

Note: The special character " has a bit configuration of 01111111, which is equivalent to the high order byte of the maximum positive integer.

Field Specifications

Column 6

F identifies the lines on pages 01 and 02 as field specification lines.

Columns 7, 8, 9-16, and 18

On line 07 (pages 01 and 02), these columns specify that for positive integers (line 07 on page 01), a character 2 will be forced into the first byte of the work record control field; for negative integers (line 07 on page 02), a character 1 will be forced into the first byte of the work record control field.

On line 08 (pages 01 and 02), columns 7, 8, and 9 through 16 specify that the control field is in positions 1 through 4 of the input records. (The control field is not included in the output records because header specification column 28, page 01, contains an X.)

On line 09 (pages 01 and 02), columns 7, 8, and 9 through 16 specify that 64 input record characters are included as data in the sorted output file.

SEQUENCE SPECIFICATIONS
Header

Line	Job	Card Match	Stacker Select	Number (N)	Print Option	Output Option (X)	Output Record Length	Checkpoint Option (C)	Non-Verify Option (N)	Null Output (N)	Storage Size to Execute	Length Code (R/C)	Record Length	Job Description
3	0													
4	0													
5	0													
6	H													

Record Type

Line	Number	Type (I/O)	Factor 1	Rel.	Factor 2 (Field, Constant, or Keyword)	Comments
3	0	1				
4	0	2				
5	0	3				
6	0	4				
7	0	5				
8	0	6				

Field

Line	Number	Type (N/O/F/D/S/*)	Location	Forced	Reserved	Comments
3	0	7				
4	0	8				
5	0	9				
6	0	10				
7	0	11				
8	0	12				
9	0	13				
10	0	14				
11	0	15				
12	0	16				
13	0	17				
14	0	18				
15	0	19				
16	0	20				
17	0	21				
18	0	22				
19	0	23				
20	0	24				
21	0	25				
22	0	26				
23	0	27				
24	0	28				
25	0	29				
26	0	30				
27	0	31				
28	0	32				
29	0	33				
30	0	34				
31	0	35				
32	0	36				
33	0	37				
34	0	38				
35	0	39				
36	0	40				
37	0	41				
38	0	42				
39	0	43				
40	0	44				
41	0	45				
42	0	46				
43	0	47				
44	0	48				
45	0	49				
46	0	50				
47	0	51				
48	0	52				
49	0	53				
50	0	54				
51	0	55				
52	0	56				
53	0	57				
54	0	58				
55	0	59				
56	0	60				
57	0	61				
58	0	62				
59	0	63				
60	0	64				
61	0	65				
62	0	66				
63	0	67				
64	0	68				
65	0	69				
66	0	70				
67	0	71				
68	0	72				

Header Specification

Column 6

H identifies the line as a header line.

Columns 7-12

SORTR identifies the job as a tagalong sort.

Columns 13-17

The total length of the control field (for either a positive or negative real number) is 5 bytes (the entry in column 17).

Column 18

The A stands for ascending order. The smallest negative real number comes first; the largest positive real number comes last in the sort sequence (for example: -2.0, -1.0, 0.0, 1.0, 2.0).

Column 28

X indicates that the control field is dropped when data is written to the output file.

Columns 29-32

When the control field is dropped, only the data portion remains. The data portion in this sample job is the entire input record. Therefore, 64 is entered in columns 29 through 32.

Record Type Specifications

Column 6

Two record types are specified (positive real numbers and negative real numbers). Both types are to be included (I in column 6 of line 01, pages 01 and 02).

Columns 8, 9-16, 17-18, 19, and 20-39

On line 01 (page 01), these columns specify that if position 1 (the leftmost byte of the 4-byte control field from the input record) is less than or equal to the character constant " (see note), the work record will have a character 2 forced into the first byte of its 5-byte control field (line 07 on page 01). This include line selects only positive real numbers. Line 01 (page 02), which is an include-all line, indicates that all records not described by the first include line (page 01) are also included in the sort. The first include line (line 01 on page 01) selects all positive real numbers; therefore, this include-all line (page 02) selects all negative real numbers, which will have a character 1 forced into the first byte of their work record control field (line 07 on page 02). All negative real numbers are forced in front of all positive real numbers in the sort sequence.

Note: The special character " has a bit configuration of 01111111, which is equivalent to the high order byte of the maximum positive real number.

Field Specifications

Column 6

F identifies the lines on pages 01 and 02 as field specification lines.

Columns 7, 8, 9-16, and 18

On line 07 (pages 01 and 02), these columns specify that for positive real numbers (line 07 on page 01), a character 2 will be forced into the first byte of the work record control field; for negative real numbers (line 07 on page 02), a character 1 will be forced into the first byte of the work record control field.

On line 08 (pages 01 and 02), columns 7, 8, and 9 through 16 specify that the control field is in positions 1 through 4 of the input records. Note that real numbers are always carried in true form (never in two's complement form). Therefore, sorting negative real numbers into ascending sequence requires them (when treated as character fields) to be sorted into descending sequence. The O in column 7, line 08 on page 02 indicates an opposite control field, which means that the negative real numbers will be sorted in descending sequence.

On line 09 (pages 01 and 02), columns 7, 8, and 9 through 16 specify that 64 input record characters are included as data in the sorted output file.

SAMPLE SORT JOB 4. RECORD SELECTION BASED ON BINARY VALUES

To select records based on a binary value, use one or both of the following types of include sequences:

Type 1 Include Sequence

For binary values whose hexadecimal equivalents can be represented by one or two characters in the 64-character set (Appendix B, Standard EBCDIC Collating Sequence table), use either an *include character* (I and C in columns 6 and 8 of the record type specifications) or an *include zone* (I and Z in columns 6 and 8 of the record type specifications) ANDed with an *include digit* (IAD in columns 6 through 8 of the record type specifications).

Type 2 Include Sequence

For binary values whose hexadecimal zone portions do not appear in the 64-character set, use two include specifications for the zone portion ANDed with an *include digit* for the digit portion.

Example: Select records that contain a value of 44 in position 2. The binary representation of this number is 0010 1100, which is a hex 2C. For the digit portion, select a character from the table in Appendix B with a digit portion of C (for example, <, *, %, and @). Use this character in an include line, comparing its digit portion to the binary value of 1100. Because there is no character in the 64-character set whose zone portion is equivalent to 2, use two include lines to describe constants with a zone portion greater than 1 and less than or equal to 2. Describe these constants as packed, which reverses the zone and digit portions from hex F1 and F2 to hex 1F and 2F.

Line	Job	Largest sum (in bytes) of Control Field Lengths for Any Record Type	Card Match										Output Record Length	CCP/Disk Sort Only	Record Length	Job Description			
			Stacker Select					Number (I/N)											
Number	Type	Sequence (A/D/S)	S	U	M	P	P	U	S	O	O	Print Option	Output Option (X)	Checkpoint Option (C)	Non-Verify Option (N)	Null Output (N)	Storage Size to Execute	Length Code (R/C)	
00	H	SORTR																	

Record Type

Line	Job	Factor 1	Rel.	Factor 2 (Field, Constant, or Keyword)																Comments
				Constant																
Number	Type (I/O)	Continuation (A/O/?)	Stacker Select	Location	F/C/K	From	To	From	To	From	To	From	To	From	To	From	To	Record Name		
01	I	D		44	EQC+														DIGIT = E **	
02	I	A		44	LEC1														AND < = 'HEX 1F ** TYPE 2	
03	I	A		44	GTC0														AND > HEX 0F **	
04	*	*	*	*															*** AND ***	
05	I	A		45	EQC@														ZONE = 7 ** TYPE 1	
06	I	A		45	EQCB														DIGIT = 8 **	

Field

Line	Job	Type (I/O/F/D/S/?)	Continuation (A/O/?)	Location	Forced	Record Character	Substitute Character	Overflow Field Length	Reserved																Field Name	Comments
									Reserved																	
Number	Type	Continuation	Location	Record Character	Substitute Character	Overflow Field Length	Field Name	Comments																		
07	F			1																SEQUENCE BY ITEM						
08	F			7																REST OF DATA						

	Hexadecimal Value	Binary Value	Decimal Value
Field to be selected, column 44	1E	= 0001 1110	= 30
Character +	4E	= 0100 1110	= 78
Packed character 1	1F	= 0001 1111	= 31
Packed character 0	0F	= 0000 1111	= 15
Field to be selected, column 45	78	= 0111 1000	
Character @	7C	= 0111 1100	
Character 8	F8	= 1111 1000	

These specifications will cause selection of all records containing (in column 44) a value between 15 (hex 0F) and 31 (hex 1F) whose digit portion is equal to E. They will further limit selection of records to those which have a binary value in column 45 whose zone portion equals 7 and whose digit portion equals 8. Therefore, all records selected will have hex 1E78 in columns 44 and 45.

Header Specification

Column 6

H identifies the line as a header line.

Columns 7-12

SORTR identifies the job as a tagalong sort.

Columns 13-17

The total length of the control field is 6 bytes.

Column 18

The A stands for ascending order. The file will be sorted according to the standard collating sequence shown in Appendix B.

Columns 29-32

The control field is not dropped when data is written to the output file (column 28 is blank). Therefore, the length of the output record equals the length of the control field (6 bytes) plus the length of the data field (105 characters) specified by lines 07 and 08. This output record length totals 111, which is the entry for columns 29 through 32.

Record Type Specifications

Line 01 selects records whose digit portion in column 44 is E. Lines 02 and 03 select records whose packed value in column 44 is less than or equal to hex 1F and, at the same time, is greater than a packed value of hex 0F.

The only hexadecimal values that satisfy both specifications for lines 02 and 03 are hex 11 through hex 1F. Line 04 is a comment line. Lines 05 and 06 select records whose zone portion in column 45 is 7 and whose digit portion is 8.

Field Specifications

Column 6

F identifies the lines as field specification lines.

Columns 7, 8, and 9-16

Line 07 indicates that the control field is in positions 1 through 6 of the input records. Only the digit portions of the characters will be used. This control field will be included in the output file.

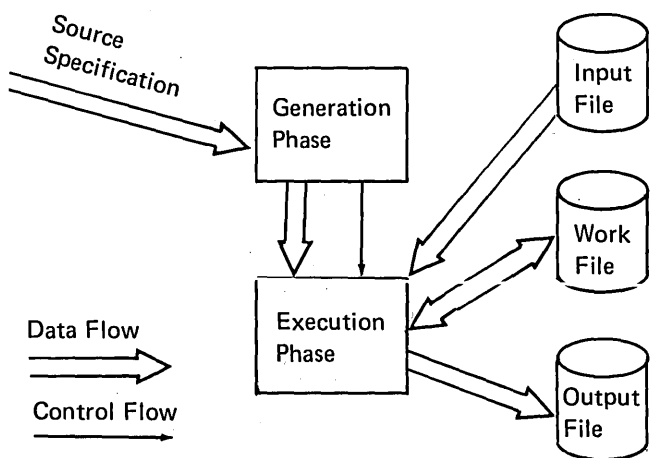
Line 08 indicates that the other 105 input record characters are included as data in the output file.

This appendix describes sort programmer messages which are activated (or suppressed) by the print option in the header statement. These messages, if activated, are issued using the system list function which prints messages when the system list device is a printer and displays messages when the list device is a display station. See the SYSLIST statement in the *System Support Reference Manual* for details on assigning the system list device. For information on how to suppress sort programmer messages, see *Column 27 (Print Option)* in Chapter 2. Sort displayed messages are described in the *Displayed Messages Guide*.

Sort printed messages are issued mainly during the generation phase of the sort program. No operator action is required. Therefore, each message description in this section consists only of a simple explanation for the programmer.

To understand these messages, you must have a basic understanding of how the sort program works. Specifically, it is helpful to know what happens during the generation phase and during the execution phase.

The generation phase translates your sequence specifications into machine language.



The execution phase gets the records you want sorted, sorts the records, and writes the sorted records onto the output file in the order that you want them.

Following is a typical sort diagnostic message as it appears on a listing. Note that a one-character alphabetic code, which indicates the significance of the message, follows the message identification code.

SORT-7264 W SPECIFICATION IN WRONG NUMERICAL ORDER

The codes and their significance are listed below.

Code	Meaning	Significance
T	Terminal	Processing cannot continue. The program ends immediately and no more processing is done. Sort displayed message SORT-7725 follows and the operator must cancel the job.
S	Severe	A sort sequence specification statement contains an error. When the generation phase is complete, sort displayed message SORT-7725 is issued and the operator must cancel the job.
W	Warning	An error condition might exist. Corrective action is required if the condition is unintentional. When the generation phase is complete, sort displayed message SORT-7722 is issued.
A	Action	A condition exists that requires an operator response. A sort displayed message follows.
I	Information	For information only. Messages with an I code indicate the current status of the job.

**SORT-7101 I PRINT OPTION ASSUMED
TO BE 0**

The print option in the header line is not blank, 0, 1, 2, or 3. Therefore, it is assumed to be 0.

**SORT-7102 W SORT JOB ASSUMED TO
BE SORTR**

Neither SORTA, SORTR, or SORTRS job type is specified on the header line. Because an output record length is specified, SORTR is assumed. To avoid this error, correct columns 7 through 12 of the header line.

**SORT-7103 T SORT JOB TYPE NOT
SPECIFIED**

Neither a job type nor an output record length is specified in the header line. Place the proper job type in columns 7 through 12 of the header line. Resubmit the job.

**SORT-7104 T FIRST STATEMENT
NOT A HEADER**

The first sort sequence specification statement is not a header statement or a valid SOURCE statement. Change your first sort sequence specification so it is either a header statement or a valid SOURCE statement.

**SORT-7105 W INVALID ENTRY,
COLUMN 36**

Column 36 of the header line does not contain an N or a blank. A blank is assumed. To avoid this error next time the job is run, specify either N or blank in column 36 of the header line.

**SORT-7150 S CONTROL FIELD
DROPPED—NO DATA
SPECIFICATIONS**

No data specifications (including summary specifications) were entered for this job, but the control field is being dropped. If the job were allowed to continue, the output file would contain all blanks. Add data field specifications to the job and/or retain the control field; then resubmit the job.

**SORT-7151 T CONTROL FIELD LENGTH
NOT GIVEN**

No control field length is specified on the header line for this job. Place the length of your control fields in columns 13 through 17 of the header line; then resubmit the job.

**SORT-7152 S CONTROL FIELD LENGTH
TOO LARGE**

The length specified for the control field is too large. A length of 256 is assumed. Specify a control field length of 256 or less in columns 13 through 17 of the header line; then resubmit the job.

**SORT-7154 W ASCENDING SEQUENCE
ASSUMED A, COLUMN 18**

Column 18 of the header line contains neither D, for descending sequence, nor A, for ascending sequence. Ascending sequence is assumed. To avoid this error next time the job is run, specify either A or D in column 18 of the header line.

**SORT-7155 W OUTPUT OPTION
ASSUMED X, COLUMN 28**

Column 28 of the header line contains neither X nor blank. Output option X is assumed. To avoid this error next time the job is run, specify either X or blank in column 28 of the header line.

SORT-7156 S OUTPUT RECORD LENGTH NOT SPECIFIED

This is a SORTR or SORTRS job, but no output record length is specified on the header line. The maximum length of 4096 is assumed. Specify the length of the output records in columns 29 through 32 of the header line; then resubmit the job.

SORT-7157 S OUTPUT RECORD LENGTH TOO LARGE

This is a SORTR or SORTRS job. The output record length specified in columns 29 through 32 of the header line is larger than 4096. The maximum length of 4096 is assumed. Be sure the header line is not shifted by one or more columns. Specify an output record length of 4096 or less; then resubmit the job.

SORT-7158 S OUTPUT RECORD LENGTH INCONSISTENT

This is a SORTR or SORTRS job and you are not dropping the control field. The output record length (columns 29 through 32) that includes the control field should not be less than the length of the control field (columns 13 through 17). Be sure you do not want to drop the control field. Recompute the output record length; correct the entry on the header line; then resubmit the job.

SORT-7159 S WORK RECORD LENGTH TOO LARGE

The work record length for this job is too large. The maximum length of 4096 is assumed. Specify a work record length of 4096 or less; then resubmit the job. (See Appendix A for information on how to determine the work record length.)

SORT-7160 W ALTERNATE COLLATING SEQUENCE ASSUMED

Column 26 of the header line does not contain an S or a blank. Alternate collating sequence is assumed. If alternate collating sequence is not to be used, specify a blank in column 26 of the header line; then resubmit the job. If alternate collating sequence is correct, place an S in column 26 to avoid this occurring again.

SORT-7161 T INVALID CONTROL FIELD LENGTH

A nonnumeric character or an embedded blank was specified in columns 13 through 17 of the header line. Correct the entry, making sure that no nonnumeric characters or embedded blanks are specified; then resubmit the job.

SORT-7162 T INVALID OUTPUT RECORD LENGTH

A nonnumeric character or an embedded blank was specified in columns 29 through 32 of the header line. Correct the entry, making sure that no nonnumeric characters or embedded blanks are specified; then resubmit the job.

SORT-7176 S INVALID ALTERNATE SEQUENCE STATEMENT

Column 26 of the header line specifies that an alternate sequence is to be used, but the statement just read is not a comment statement or a valid sort specification statement, nor does it contain ALTSEQ in columns 1 through 8. Correct the incorrect statement or delete it if it does not belong in the job; or add a valid alternate sequence statement.

**SORT-7177 W MISSING THE DOUBLE
ASTERISK (**) STATEMENT**

No statement containing asterisks (*) in columns 1 and 2 was between the alternate sequence statement and the include, omit, or field specification statement. The sort program assumes an ** statement following the alternate sequence statement. To avoid this error next time the job is run, place an ** statement between the alternate sequence statement and the include, omit, or field specification statement.

**SORT-7178 S ALTSEQ STATEMENT HAS
INVALID DATA**

An alternate sequence statement entry that is used to alter the normal sequence is in error. One of the four-position groups (columns 9 through 12, 13 through 16, and so on) in the ALTSEQ statement does not contain a valid hex character (0 through 9 or A through F). An asterisk (*) flags the columns in error on the source listing. Processing continues with the next four-position group. Correct the incorrect alternate sequence statement; then resubmit the job.

**SORT-7179 T UNEXPECTED END OF
FILE FOUND**

An end-of-file statement followed the alternate sequence statement. No comment statement or ** statement exists. Check that the last part of the sequence specification is not missing from the job or that an end-of-file statement is not placed incorrectly in the job.

**SORT-7181 W NO VALID ALTSEQ
STATEMENTS FOUND**

ALTSEQ is specified on the header line, but no valid ALTSEQ statements were entered. The sort program ignores the ALTSEQ specification on the header line. If an alternate collating sequence is to be used, correct the invalid ALTSEQ statements, or add them if none were specified; then resubmit the job. If alternate collating sequence is not to be used, leave column 26 of the header line blank to avoid this message.

**SORT-7202 S INVALID SPECIFICATION
TYPE, COLUMN 6**

A specification is not identified in column 6 as a valid type of specification. Valid specification types are comments, include sets, omit sets, or field specifications. The specification is bypassed and the rest of the specifications are processed. The specification was probably specified incorrectly or misplaced in the job. Correct the error; then resubmit the job.

**SORT-7204 W BLANK CONTINUATION
ASSUMED, COLUMN 7**

A nonblank character is specified in column 7 of a specification that is the first in an omit or include set; a blank is expected. The sort program assumes a blank in column 7. To avoid this message, leave column 7 blank.

**SORT-7206 S INVALID CONTINUATION,
COLUMN 7**

A character other than A, O, or blank is specified in column 7 of a specification that is part of an omit or include set. The specification is ignored and the next specification is processed. Correct the specification by placing an A or O in column 7; then resubmit the job.

**SORT-7208 W OR CONTINUATION
ASSUMED, COLUMN 7**

A blank is specified in column 7 of a statement that is part of an omit or include set. Column 7 should be A or O. O is assumed. If O is incorrect, enter an A in column 7; then resubmit the job. If O is correct, place an O in column 7 to avoid this message.

SORT-7210 W INVALID C/Z/D/P/U/V SPECIFICATION, COLUMN 8

An invalid entry is specified in column 8 of a specification statement. The valid entries are:

Specification Type	Column 8 Entry
Include or omit record type	C, Z, D, P, or U
Control field	C, Z, D, P, or U
Data field or summary field	C, Z, D, P, U, or V

The sort program assumes a C in column 8. If C is incorrect, place the correct entry in column 8; then resubmit the job. If C is correct, place a C in column 8 to avoid this message.

Note: When the specification type is an include or omit record type and factor 2 is a keyword, column 8 must contain a C. When the specification type is a control field and you want an unconditional force or a force-all line, column 8 must contain a C.

SORT-7212 S SPECIFICATION IN WRONG LOGICAL ORDER

The specification in error is either:

- A control field specification (F in column 6) that incorrectly follows an omit specification (O in column 6).
- An omit specification (O in column 6) that incorrectly follows an include specification (I in column 6).

The specification is bypassed and the next specification is processed. Put the specifications in the correct order; then resubmit the job.

SORT-7214 S INCLUDE OR OMIT STATEMENT AFTER INCLUDE-ALL

An INCLUDE-ALL specification was read for this job, but an include or omit specification (I or O in column 6) was found. Only control field specifications should follow the INCLUDE-ALL specification. The specifications are processed as though the INCLUDE-ALL specification was not read. Remove either the INCLUDE-ALL specification or the include or omit specifications; then resubmit the job.

SORT-7216 S LAST SET NOT A VALID INCLUDE SET

The last set of sequence specifications should be an include set, but one of the following occurred:

- No specifications were specified after the header line.
- The last set was an omit set.
- The last set was an include set that contained no field specifications.

Make one of the following corrections, depending on the condition that caused the message; then resubmit the job:

- Supply the sequence specifications for the job.
- Eliminate the omit set; the records are omitted by default.
- Supply the missing field specifications.

SORT-7218 S TOO MANY SEQUENCE SPECIFICATIONS

All the main storage allocated by the sort program for the select/build routine and for error information was used, but there are still specifications to be processed. The remaining specifications are checked for errors. Remove all causes for error messages from the specifications; then resubmit the job. If this error occurs again and no other error messages are issued, you must decrease the number of sequence specifications.

SORT-7220 T TOO MANY ERRORS

The main storage allocated by the sort program for the select/build routine and for error information was filled with error information. No more specifications can be processed. Correct all diagnosed errors; then resubmit the job.

SORT-7222 W ZONE OR V FIELD LENGTH EXCEEDS 1

The referenced specification is a field specification (F in column 6). The type specified is a zone field (Z in column 8) or a V field (V in column 8) that should have a length of one byte.

The specified number of bytes (calculated using columns 9 through 12 and 13 through 16) is used in building the work record field for this record type. The byte zone or overflow information generated is placed in the rightmost byte (low-order byte) of this field. The other positions to the left contain binary zeros (control fields) or blanks (data fields). If a one-byte zone or V field is incorrect, correct the field specification; then resubmit the job.

SORT-7224 S COLUMNS 9-16 OR COLUMNS 20-27 ARE INVALID

The location information (columns 9 through 16 and/or columns 20 through 27) in a specification is invalid because:

- The From location is greater than the To location.
- The To location is 0 or blank.
- The lengths of factor 1 and factor 2 are not equal.

Correct the error; then resubmit the job.

SORT-7225 S INVALID KEYWORD BEGINNING IN COLUMN 20

The specification causing this error is an include or omit record type specification. A K was found in column 19, but neither UDATE, UMONTH, UDAY or UYEAR was found beginning in column 20. UDATE is assumed. Correct the specification; then resubmit the job.

SORT-7226 W DIGIT FIELD LENGTH EXCEEDS 16

A digit field length of more than 16 was specified in the specifications. A length of 16 is assumed. If the digit field length is actually 16 or less, correct the specification. If the digit field length is greater than 16, divide it into two different fields and create new specifications for these fields; then resubmit the job.

SORT-7228 W CHARACTER CONSTANT LENGTH EXCEEDS 20

The specification causing this error is an include or omit specification (I or O in column 6). Since a C is specified in columns 19, factor 2 is a character constant. The length implied by factor 1 (columns 9 through 12 and 13 through 16) is greater than 20, the maximum allowed. The 20 rightmost characters in factor 1 are compared against the constant in columns 20 through 39 for this job. Correct the specification statement if a 20-character field is not correct or to avoid this message.

SORT-7230 W SAME SPECIFICATION TYPE ASSUMED, COLUMN 6

The specification causing this message has a blank in column 6, so the specification type is not known. However, column 7 contains an A or O and the previous specification was an include or omit specification (I or O in column 6). Therefore, the sort program assumes that this specification is a record type specification and that the specification type is to be identical to that in the previous specification. If the assumption is incorrect, correct the specification; then resubmit the job.

SORT-7232 S ZONE SPECIFIED-FACTOR 2 IS NOT A CONSTANT

A zone is specified in column 8 of an include or omit record type specification, but column 19 does not have a C specified. The specification is bypassed. Correct the specification; then resubmit the job.

SORT-7234 W ZONE SPECIFIED-RELATIONSHIP NOT EQ OR NE

A zone is specified in column 8 of the include or omit record type specification, but columns 17 and 18 do not contain EQ or NE. EQ is assumed and processing of the specification continues. If EQ is not the correct entry for columns 17 and 18, correct the specification, then resubmit the job. If EQ is correct, correct the specification to avoid this message.

**SORT-7236 S INVALID RELATIONSHIP,
/ COLUMNS 17-18**

The specification causing this error is an include or omit record type specification. The relationship between factor 1 and factor 2, specified in columns 17 and 18, is not EQ, NE, LT, GT, LE, or GE. EQ is assumed. Correct the specification; then resubmit the job.

**SORT-7238 W INVALID FACTOR 2 TYPE,
COLUMN 19**

Column 19 of an include or omit record type specification does not contain an F (field), a C (constant), or a K (keyword). C is assumed; factor 2 is processed as a constant. Check column 19 for a missing or invalid entry. Correct the specification before the job is run again.

**SORT-7240 S UNPACKED DECIMAL-
LENGTH EXCEEDS 16**

The specification causing this error contains a U in column 8, indicating unpacked decimal. Columns 9 through 16 define a factor 1 field that is longer than the allowable 16 bytes. A length of 16 is assumed. Check the To and From columns to determine the error. Correct the error before the job is run again.

**SORT-7242 W ZONE SPECIFIED-LENGTH
EXCEEDS 1**

The specification causing this error is an include or omit record type specification that has a Z in column 8. In this case, factor 1 must be one byte long, but it is not. A length of one is assumed and the constant in column 20 of the specification is processed. Correct the specification before the job is run again.

**SORT-7244 W PACKED DECIMAL-LENGTH
EXCEEDS 8**

A packed field is specified (P in column 8), but the field length is more than 8. A packed field can be only 8 columns long. A field length of 8 is assumed and the rightmost eight bytes of the field are used for the include or omit. If column 19 contains a C, the constant in columns 20 through 35 of the specification is used as factor 2. Correct the specification before the job is run again.

**SORT-7246 W INVALID FIELD TYPE
SPECIFICATION, COLUMN 7**

The specification causing this error is a field specification, but column 7 does not contain an F, N, O, D, or S. An entry of N (normal control field) is assumed if there are no previous data or summary data field specifications for this include set. Otherwise, an entry of D (data field) is assumed. Correct the specification before the job is run again.

**SORT-7248 S ILLEGAL FORCE-ALL
CONTINUATION**

The specification causing this error is a field specification for a force-all specification (F in column 7). This specification should follow a conditional force specification, but it does not. The specification is ignored. Correct the specification; then resubmit the job.

**SORT-7250 S FIELD STATEMENTS IN
WRONG LOGICAL ORDER**

The specification causing this error is a control field specification that follows a data (D) or summary (S) specification in this include set. All control field specifications (N, O, or F in column 7) must precede any data or summary data field specifications. This specification is bypassed and the next specification is processed. The specification might be misplaced or specified incorrectly. Correct the specification; then resubmit the job.

SORT-7252 S CONTROL FIELD LENGTH EXCEEDS HEADER VALUE

The accumulated total of the control field lengths in this include set exceeds the value specified in the header line. The remaining specifications in the include set are processed. Check the length specified in the header line against the specified control fields. Correct the errors; then resubmit the job.

SORT-7254 W CONTROL FIELD LENGTH LESS THAN HEADER VALUE

All control field specifications for this include set were processed, and the accumulated length of control fields is less than the value specified in the header line. The remaining positions are filled with hex zeros. Check the length specified in the header line against the specified control fields. Correct any errors before the job is run again.

SORT-7256 S CHARACTER FIELD SPECIFICATION-LENGTH EXCEEDS 256

A length greater than 256 is specified in a character field specification (C in column 8). The rest of the sequence specifications are processed. Check the character field specifications. If you require more than 256 positions of data, divide the field into two or more field specification statements. Correct the error; then resubmit the job.

SORT-7258 S TO OR FROM FIELD NOT WITHIN INPUT RECORD

Either one or both of the specifications in the To or From columns are zero or contain a value larger than the input record length. Make sure the correct input file was specified and the To and From columns contain the correct values. Correct the error; then resubmit the job.

SORT-7262 S DATA LENGTH EXCEEDS HEADER VALUE

A SORTR or SORTRS job is being processed and the data field specifications (FD or FS in columns 6 and 7) for this include set exceed the value in the header line. An error exists in either the output record length in the header line or in the To and From fields in the field description lines. If you are dropping the control field (X in column 28 of the header line), the data length is the output record length (columns 29 through 32 of the header line). If you are not dropping the control field, the control field length (columns 13 through 17 of the header line) must be subtracted from the output record length to determine the data field length. Correct the error; then resubmit the job.

SORT-7264 W SPECIFICATION IN WRONG NUMERICAL ORDER

Columns 1 through 5 of each sequence specification statement are used to order the specifications. One or more of the specification statements in this job has a value in columns 1 through 5 that is less than the value in the preceding specification statement. On the source listing, an S is printed to the left of the specification statements that are out of order. To avoid this error, correct the entries in columns 1 through 5 of the sequence specifications so that no entry is less than that on the previous specification statement.

SORT-7266 S FACTOR 1 LENGTH EXCEEDS 256

The specification causing this error is an include or omit specification for character data (C in column 8). Factor 2 is a field (F in column 19) and the length of factor 1 is greater than 256, the maximum length. The sort program assumes a length of 256 for factor 1. If the length of factor 1 must be more than 256, divide this specification into two specifications with an AND continuation line (A in column 7); then resubmit the job.

SORT-7268 W P OR U FACTOR 1 USED WITH ALTSEQ

An include or omit record type specification specifies a packed or unpacked factor 1, but the header line specifies an alternate collating sequence. The sort program changes factor 1 and factor 2 as the ALTSEQ specifications indicate. This change can affect the units position (and the sign) of an unpacked decimal number or any one position of a packed decimal number. As a result, the desired records might not be included or omitted. Do not specify P or U in column 8 of the record type entries when you specify an alternate collating sequence.

SORT-7276 W MORE THAN ONE SUMMARY V FIELD IN INCLUDE SET

This is a SORTRS job. The specification causing the error specifies a summary V field (FSV in columns 6 through 8), but at least one other summary V field was already specified for this include set. The statement in error is assumed to specify a data V field (FDV in columns 6 through 8). To avoid this error, change the extra summary V fields to data V fields.

SORT-7278 W SUMMARY V OR DATA FIELD INCONSISTENT

This is a SORTRS job. The specification causing the error is a summary data field specification (S in column 7). To determine which of the following conditions is true, compare this specification with the first include set having S specifications:

- The number of S specifications prior to this specification differs.
- The length of this S specification differs.
- The substitution character of the V specification differs.
- The relative location of the specification within the output record differs.

The S specifications of the first include set containing S specifications define the summary field positions. Correct the specifications; then resubmit the job.

SORT-7280 W NUMBER OF SUMMARY FIELDS INCONSISTENT

This is a SORTRS job. The number of S specifications in this include set is different from the number of S specifications in the first include set. The S specifications of the first include set with S specifications form the summary output record. To avoid this error, correct the include sets so they all have the same number of S specifications.

SORT-7282 W NO SUMMARY SPECIFICATIONS—SUMMARY SORT

The header line specifies that this is a SORTRS job, but there are no summary data field specifications. All records with duplicate control fields are eliminated from the output file. Check the job type to be sure that a SORTRS job was intended.

SORT-7284 W SUMMARY SPECIFICATIONS IGNORED

This is a SORTR job; consequently, no summary data field specifications (FS in columns 6 and 7) were expected. However, at least one was found. All S specifications are treated as data specifications (D in column 7). Make sure this should be a SORTR job. If not, correct the specifications; then resubmit the job.

SORT-7286 S TOO MANY SUMMARY DATA FIELDS

This is a SORTRS job. More than 24 summary data fields were specified (FS in columns 6 and 7 and C, D, P, or U in column 8). Limit the number of summary data fields to 24 or less.

**SORT-7288 W INVALID SUMMARY
SPECIFICATION—
COLUMNS 20, 21, 22**

The overflow field length in columns 20 through 22 is incorrect because it is:

- An invalid entry
- Not right-justified
- Less than the field length given by the To and From entries

The entry is assumed blank. If a valid entry was intended in columns 20 through 22, correct the entry; then resubmit the job.

**SORT-7391 T MAIN STORAGE TOO
SMALL**

The amount of main storage assigned to the sort program is not large enough to process this job. The job cannot be run. Make sure:

- The correct input file was specified.
- If this is a SORTR or SORTRS job, the output length specified in the header line is correct.

Correct the errors; then resubmit the job. If there are no errors, increasing the region size may allow the job to execute.

**SORT-7395 T DISK SPACE NOT AVAILABLE
FOR WORK FILE**

The sort program was attempting to automatically allocate space for the work file, but not enough space was available on the disk. See the description of sort displayed message SORT-7733. Sort displayed message SORT-7733 was issued prior to this message and the operator selected the controlled cancel.

**SORT-7401 I JOB COMPLETED
GENERATION PHASE**

The generation phase is complete. All statements were checked for errors. If severe or terminal errors were found during generation, sort displayed message SORT-7725 is issued; if only warning errors were found, sort displayed message SORT-7722 is issued. If no errors were found, the job proceeds to the execution phase.

SORT-7402 I NO ERRORS FOUND

No errors were found during the generation phase of the job. Processing continues with the execution phase.

**SORT-7403 I OPERATOR CHOSE TO
CANCEL JOB**

At the end of the generation phase, sort displayed message SORT-7722 was issued and the operator canceled the job.

**SORT-7404 I OPERATOR CHOSE TO
CONTINUE JOB**

At the end of the generation phase, sort displayed message SORT-7722 was issued and the operator continued the job.

**SORT-7422 A XXX WARNING ERRORS
FOUND DURING
GENERATION PHASE**

During the generation of this job, no severe or terminal errors were found. However, a number of warning errors were found that could affect the execution of the job. Statements entered or specified incorrectly probably caused the errors. Sort displayed message SORT-7722 is issued after this message.

**SORT-7423 A DISK ERROR WHILE
READING SEQUENCE
SPECIFICATIONS**

A disk error occurred while the sequence specifications were being read. Sort displayed message SORT-7723 is issued after this message.

**SORT-7425 A SEVERE/TERMINAL ERRORS
DURING GENERATION
PHASE**

Severe and/or terminal errors occurred during the generation of this job. Sort displayed message SORT-7725 is issued after this message.

**SORT-7450 I XX,XXX MAIN STORAGE
BYTES ASSIGNED**

This message indicates the amount of main storage assigned for the sort program.

**SORT-7451 I XXXX BYTES—INPUT
RECORD LENGTH**

This message indicates the input record length.

**SORT-7452 I XXXX BYTES—WORK FILE
RECORD LENGTH**

This message indicates the record length used for the work file. The record length is computed from the information in the header line.

**SORT-7453 I XXXX BYTES—OUTPUT
RECORD LENGTH**

This message indicates the record length of the output file. The record length is computed from the information in the header line.

**SORT-7461 I XXXX BYTES—
SELECT/BUILD ROUTINE**

This message indicates the number of bytes in main storage occupied by the sequence specifications (including any alternate collating statements).

**SORT-7462 I XXXX SEQUENCE SPECIFI-
CATION STATEMENTS**

This message indicates the number of sequence specifications that the generation phase processed for this job.

SORT-7600 I INITIAL PHASE STARTED

The execution phase of the job began. Records are being read from the input file and written to the work file in intermediate strings.

SORT-7601 I #XX PASS COMPLETED

Sort pass XX was completed successfully. The first sort pass is 00, the second pass 01, and so on.

**SORT-7602 I XXXXXX INPUT RECORDS
WERE READ IN**

This message indicates the number of input records read for your job.

**SORT-7603 I XXXXXX RECORDS WERE
SELECTED TO BE SORTED**

This message indicates how many input records will be sorted.

**SORT-7620 A NO INPUT RECORDS WERE
SELECTED FOR SORTING**

None of the input records met the include requirements of the sort sequence specifications. Check the specifications or input records that were specified. Sort displayed message SORT-7724 is issued after this message.

**SORT-7681 A WORK RECORD COUNT
IN ERROR**

This message indicates an error in the sort program. Contact IBM for program support services.

**SORT-7690 I SORT STARTING FINAL
PASS**

The final pass of the job began. Records are being written on the output file.

**SORT-7691 I FINAL PASS SUCCESS-
FULLY COMPLETED**

The final pass of the job was completed with no errors.

**SORT-7692 I XXXXXX SORTED RECORDS
PLACED ON OUTPUT FILE**

The last pass of the sort is complete. This message indicates how many output records were sorted and placed on the output file.

**SORT-7693 A NO RECORDS FOUND IN
THE INPUT FILE**

No records are in the input file. If the job is continued, a null output file is created. Make sure the correct input file was specified and that the file contains the desired records. Sort displayed message SORT-7724 is issued after this message.

SORT-7694 I XX PASSES REMAINING

This message indicates the number of passes left for this job.

**SORT-7695 I XXXX BLOCKS USED FOR
WORK FILE**

This message indicates the number of blocks that were used for the work file.

SORT-7901 I *** NORMAL EOJ *******

The sort job was completed.

SORT-7902 I *** ABNORMAL EOJ *******

Errors prevented the sort job from being run to normal completion.

**SORT-7928 A DISK ERROR WHILE
READING THE INPUT FILE**

A disk error occurred while reading records from an input file. Sort displayed message SORT-7728 follows.

**SORT-7929 A DISK ERROR WHILE
READING OR WRITING
THE WORK FILE**

A disk error occurred while reading or writing records on the work file. Sort displayed message SORT-7729 is issued after this message.

**SORT-7930 A DISK ERROR WHILE
WRITING TO THE OUTPUT
FILE**

A disk error occurred while the sorted records were being written to the output file. Sort displayed message SORT-7730 is issued after this message.

SORT-7931 A WORK FILE TOO SMALL

The sort work file is full. See the description of sort displayed message SORT-7731. Sort displayed message SORT-7731 is issued after this message.

SORT-7932 A OUTPUT FILE TOO SMALL

The sort output file is full. See the description of sort displayed message SORT-7732. Sort displayed message SORT-7732 is issued after this message.

addrout file: A record address disk file produced by the sort program. An addrout file contains the binary relative record numbers of records in a disk file, and can be used to process the file designated as the input file to the sort program.

addrout sort: A type of sort where the output consists of 3-byte, binary, relative record numbers that indicate the relative positions (1st, 20th, 99th) of records to be processed in a disk file.

alphabetic character: Any one of the letters A through Z, or one of the special characters #, \$, and @.

alphabetic field: One or more alphabetic characters of related information in a record.

alphanumeric character: An alphabetic character, or one of the digits 0 through 9.

alphanumeric field: One or more alphanumeric characters of related information in a record.

alternate collating sequence: A user-defined collating sequence that alters the standard EBCDIC collating sequence.

ALTSEQ statement: A statement that defines an alternate collating sequence to the sort program.

backup copy: A copy of a file or of a library member that is kept in case the original file or library member is destroyed.

binary: Relating to, being, or belonging to a system of numbers having 2 as its base (for example, the binary digits 0 and 1).

binary digit (bit): In binary notation, either of the characters 0 or 1. See also *byte*.

bit: A binary digit.

block: (1) A record or a collection of contiguous records recorded or processed as a unit. (2) In System/34, a 10-sector unit of disk storage that contains 2,560 bytes.

byte: (1) A sequence of 8 adjacent bits that are operated on as a unit and that comprise the smallest addressable unit in the System/34. (2) The representation of a character by 8 binary bits; the amount of storage required for one EBCDIC character.

cataloged procedure: A set of OCL statements that have been placed in a library.

character: A digit, letter, or other symbol that is used as part of the control, organization, or representation of data.

column: A vertical arrangement of characters, as on a punched card or a coding form.

command: A request for the performance of an operation or the execution of a particular program.

command statement: A statement that requests the performance of a particular function. A command statement always contains the name of the command, and may include parameters.

conditional force: If the control field in the input record contains a particular entry, the control field character is replaced before the record is sorted.

consecutive processing: File processing that reads records in the order in which they exist in the file.

constant: A unit of data that does not change during execution of a program.

control field: One or more specified fields that are compared to determine the record sequence in the output file.

control statement: A statement that provides the SSP or utility program with information about the job being run.

data: A collection of facts, numbers, letters, and symbols that can be processed by a computer.

data field: See *field*.

direct file: A file in which, for purposes of storage and retrieval, there is a relationship between the contents of the records and their positions in the file.

disk file: An organized collection of related records on disk that are treated as a unit.

display screen: The part of a display station on which data, messages, or other information is displayed.

display station: An input/output device containing a display screen on which data is displayed, and an attached keyboard from which data is entered. It can be used to request jobs and/or enter data.

displayed message: A message that appears on the display screen and is documented in the *Displayed Messages Guide*.

documented message: Any message that is explained in a manual. A documented message contains a message identifier code that directs the operator or programmer to an explanation of the message in the message manual or in a reference manual.

EBCDIC (extended binary-coded decimal interchange code): A character set containing 256 eight-bit characters.

execution phase: The part of the sort program that reads the records you want to sort, sorts the records, and writes the sorted records into the output file.

field: One or more bytes of related information in a record.

file: An organized collection of related records treated as a unit.

file label: See *label*.

filename: An arbitrary symbol created by the programmer or program to identify and refer to a collection of related records.

force-all: If the control field in the input record does not contain a particular entry, the control field character is replaced before the record is sorted.

forced control field: A one-position control field that results from replacing one character with another character, or from forcing a character into a control field position.

generation phase: The part of the sort program that translates the sequence specifications into machine language.

hexadecimal: Pertaining to a number system with a base of 16; valid digits range from 0 (zero) through F (fifteen).

implied include-all: Omission of all record type specifications to indicate that all records are to be included in the sort.

include-all: A special form of an include set that tells the sort program to sort all of the records that have not previously been described.

include set: Sort sequence specifications that identify one or more record types that you want to sort.

indexed file: A file in which the position of each record is recorded in a separate portion of the file called an index.

input file: A file containing the data that the sort program will process.

input record: A data record that is transferred to computer storage for processing.

integer: A positive or negative whole number (a number that does not have a decimal point or exponent).

job: One or more related procedures or programs grouped into a procedure.

job step: A unit of work represented by a single program.

keyword: A nonvariable character string in sequence specifications (such as UPDATE).

label: The name in the disk or in the diskette volume table of contents that identifies a file.

library: An area on disk that can contain load members, procedure members, source members, and subroutine members. See also *system library* and *user library*.

library member: A named collection of records or statements in a library.

member: Same as *library member*.

message: A series of words or symbols designed to convey information.

multiple disk system: A system that has more than one disk.

normal control field: Any control field that is specified by an N in column 7 of the field specifications.

numeric: Pertaining to the digits 0 through 9.

OCL: Operation control language.

omit set: Sort sequence specifications that identify one or more record types that you do not want to sort.

operation control language (OCL): A programming language used to identify a job and its processing requirements to the SSP.

opposite control field: Any control field that is specified by an O in column 7 of the field specifications.

output file: A file containing the data that results from the sort program processing.

output record: A sorted record that is written to the output file by the sort program.

overflow field: In a summary tagalong sort, a field that allows for anticipated field expansion.

overlay: To write over (and therefore destroy) an existing file.

packed decimal format: Each byte within a field represents two numeric digits except the rightmost byte, which contains one digit in bits 0 through 3 and the sign in bits 4 through 7. For all other bytes, bits 0 through 3 represent one digit; bits 4 through 7 represent one digit. For example, the decimal value +123 is represented as 0001 0010 0011 1111. Contrast with *zoned decimal format*.

packed field: A field that contains data in the packed decimal format.

parameter: A value specified in a command statement or a control statement.

position: The address (location) of a character in a series, as in a record.

printed message: A message issued by the sort program that is printed on a printer or displayed on the display screen.

procedure: A set of related OCL statements, and possibly utility control statements, that cause a specific function or set of functions to be performed. A procedure in a library is called a procedure member.

procedure member: A procedure that is stored in a library.

program: A sequence of instructions to a computer, written in a special form that the computer can interpret.

program date: The date associated with a program (or job step).

program product: An IBM-written, licensed program for which a monthly charge is made. A program product performs functions related to processing user data.

prompt: A message issued by a program that requests either information or an operator action to continue processing.

real number: A number (which contains a decimal point) stored in floating point format. The leftmost byte of a real number contains the sign and the exponent (decimal point location). The remaining bytes of the whole number contain the fraction (decimal point portion value) or the fraction and mantissa (portion of the number to the left of the decimal point) in left-adjusted, binary format.

record: A collection of related data, treated as a unit. For example, one line of an invoice can comprise a record.

record address file: An input file that indicates to a program which records are to be read from a disk file, and the order in which these records are to be read from the disk file.

record length: The total number of characters (bytes) in a record.

record type: The classification of records in a file. Records are classified according to a specific field or fields within each record.

region: The amount of main storage available for a task.

relative record number: A number that specifies the location of a record in relation to the beginning of the file.

right-adjust: The placement of data in a register or field, or the shifting of the contents of a register or field, so that the last significant byte at the right end of the data is placed in the rightmost position of the register or field. See also *right-justify*.

right justify: The adjustment of positions of characters on a page so that the right margin of the page is regular. See also *right-adjust*.

select field: A field tested for a condition to determine whether a record should be included in, or omitted from, a sort job.

sequence specifications: See *sort sequence specifications*.

sequential file: A file in which records are entered one after the other; a file in which there is no relationship between the contents of the records and their positions in the file.

sort program: A part of the Utilities Program Product used to arrange records (or their relative record numbers) in a predetermined sequence, according to data contained in one or more specific fields within the records.

sort sequence specifications: Source statements for the sort program.

source member: A collection of records (such as sort sequence specifications) that is used as input for a program. Source members are stored in a library.

special character: A character other than a digit, a letter, or #, \$, and @. For example, *, +, and % are special characters.

SSP: System Support Program Product.

summary data field: In a summary tagalong sort, a data field designated for accumulated totals.

summary tagalong sort: A type of sort that arranges records according to a specified control field(s), and can accumulate totals from specified data fields within the sorted records.

system library: The library containing the members that are part of the SSP. See also *library* and *user library*.

tagalong sort: A type of sort that arranges records according to a specified control field(s), and can include data fields, control fields, or both data fields and control fields in the sorted output.

task: A unit of work for the main storage processor.

unconditional force: A character is always forced into the control field before the record is sorted.

user library: A library created by the user. A user library is in addition to the system library, and may contain any type of library member. See also *library* and *system library*.

Utilities Program Product: A multipurpose program product for creating, displaying, maintaining, listing, and sorting data files, and for creating, displaying, maintaining, and listing source members and procedure members in a library.

utility control statement: A control statement that gives a utility program information concerning the way the program is to perform its function or the output it is to produce.

volume table of contents (VTOC): An area on a disk or diskette that describes the location, size, and other characteristics of each data file on the disk or diskette.

work file: An area on disk that is reserved for temporary storage of data being sorted.

work record: A record built by the sort program for subsequent processing.

zoned decimal format: Representation of a decimal value by 1 byte per digit. Bits 0 through 3 of the rightmost byte represent the sign; bits 0 through 3 of all other bytes represent the zone portion; bits 4 through 7 of all bytes represent the numeric portion. For example, in zoned decimal format, the decimal value of +123 is represented as 1111 0001 1111 0010 1111 0011. Contrast with *packed decimal format*.

zoned field: A field that contains data in the zoned decimal format.

- * to indicate comments on field lines 4-6
- * to indicate comments on record type lines 3-4

- action messages (A)
 - issued by the sort program 2-3
- addrout (address out) sort 1-3
 - (see also SORTA)
 - defined F-1
- alphameric constants
 - as factor 2 (columns 20-39—record type specifications) 3-11
 - length and position 3-12
- alphameric data
 - affect on column 8 record type entry 3-6
 - as affected by column 8 record type entry 3-5
- alternate collating sequence
 - affect on factor 1 and factor 2 comparison 3-10
 - affect on sort time 6-1
 - ALTSEQ statement use B-3
 - defined F-1
 - indicated on header line (column 26) 2-2
 - restrictions on using column 8 record type entries B-3
 - storage required for 6-1
- ALTSEQ statements
 - defined F-1
 - examples of B-4
 - how to code B-3
 - placement in sequence specifications B-3
 - programming considerations B-3
 - what they can and cannot change B-4
- AND lines
 - described 3-4
 - describing a record type 3-3
 - describing additional factor 1 fields 3-9
 - in include sets 3-3
 - in omit sets 3-4
- ascending or descending sequence
 - specified in column 18 of header specifications 2-2
- automatic work file allocation 5-1, 6-3
 - affect on sort time 5-4, 6-3
 - multiple disk system 6-3
 - single disk system 6-3

- backup copy
 - access by addrout sort (SORTA) output 5-2
 - defined F-1
 - needed when overlaying a file 5-2
 - binary sorting D-1
- blanks
 - test for and omit 5-1
- bypass halt for message SORT-7724 2-4

- C/Z/D/P/U (column 8—record type specifications) 3-5
 - affect on factor 1 length (columns 9-16) 3-9
 - restrictions when using alternate collating sequence 3-5
 - to indicate alphameric or numeric data 3-5
 - to indicate packed or zoned data 3-5
 - to indicate portions of characters to compare 3-5
 - to indicate records you want to sort 3-5
 - when factor 1 and factor 2 contain alphameric data 3-6
 - with signed and unsigned data 3-7
- C/Z/D/P/U/V (column 8—field specifications) 4-6
 - affect on your sorted file 4-6
 - combined with column 7 entry 4-9
 - used to select portions of input characters 4-6
 - what it tells sort 4-6
- calculating file size A-1
- calculating sum of control field lengths for a record type 2-2
- character overflow indicator field (column 17 field specifications) 4-11
- characters
 - in compare operations 3-5
 - representation within the computer 3-6
- collating sequence (column 26—header specification) 2-2
 - alternate B-3
 - defined F-1
 - specified for compare operations 2-2
 - standard B-3
 - table B-1, B-2
- command (see SORT command)
- comment line
 - described 4-6
 - field specifications 4-23
 - record type specifications 3-13
- comments (columns 40-72—field specifications) 4-20
- comments (columns 40-72—record type specifications) 3-13
- compare operations
 - affected by column 8 of record type specifications 3-5
 - affected by column 19 of record type specifications 3-10
 - affected by columns 17-18 of record type specifications 3-10
 - described 2-2
- conditional force
 - defined F-1
 - example of 4-4
 - using a force-all line 4-18
 - using normal or opposite and forced control fields 4-16
 - using only forced control fields 4-15
- conditionally forced character (column 17—field specifications) 4-11
 - examples 4-11
 - how to define 4-12
 - used to assign value to summary overflow field 4-11
 - what it tells sort 4-11
- constant
 - alphameric 3-11
 - as factor 2 (columns 20-39—record type specifications) 3-11
 - defined F-1
 - indicated by column 19 of record type specifications 3-10
 - length when used as factor 2 on record type specifications 3-10
 - position when used as factor 2 on record type specifications 3-10
 - positioning sign in numeric constants 3-11

- continuation or comment (column 7—record type specifications) 3-4
 - to describe different record types having identical field types 3-4
 - to further describe the same record type 3-4
 - to indicate comment line 3-4
 - to indicate start of include or omit set 3-4
- control and data fields
 - describing control field as both 2-3
- control fields
 - defined F-1
 - duplicate 4-4
 - forced 4-4
 - conditional 4-5
 - force-all 4-5
 - unconditional 4-5
 - how to drop 2-3
 - major 4-3
 - minor 4-3
 - normal 4-4
 - number of when two or more record types 4-4
 - opposite 4-4, 4-6
 - packed or zoned 4-6
 - specifying use by sort 4-3
 - to sequence information in sorted records 4-7
 - total lengths when two or more record types 4-4
 - types of 4-4
 - used as data fields 2-3
 - when to drop 2-3
- data
 - alphameric 3-6
 - defined F-1
 - numeric 3-7
 - representation within the computer 3-6
 - used in compare operations 3-5
- data and control fields
 - describing control fields as both 2-3
- data fields
 - description 4-5
 - describing on field specifications line 4-5
 - number of when two or more record types 4-5
 - total lengths when two or more record types 4-5
- data fields, summary (see summary data fields)
- defining
 - conditionally forced characters 4-12
 - force-all characters 4-12
 - unconditionally forced characters 4-12
- diagnostic messages (S, T, W) 2-3, E-1
- digit portion of a character 3-6
- display station keyboard, procedures entered through 5-7
- displayed messages 2-3
- dropping control fields 2-3
 - considerations for 2-3
 - how to do 2-3
 - when to do 2-3
- EBCDIC characters 3-6
- END statement
 - to indicate last sequence specification is entered 5-6, 5-7
- entering OCL statements 5-1
- entering sort procedures through a keyboard 5-7
- executing sort procedures stored on disk 5-6
- execution phase E-1
- factor 1 (columns 9-16—record type specifications) 3-9
 - changed by ALTSEQ statements B-4
 - coding rules 3-9
 - coding when only one character 3-9
 - describing more than one factor 1 field 3-9
 - how length affected by column 8 entry 3-9
 - how length affected if keywords are used 3-10
 - to identify included or omitted records 3-9
- factor 2 (columns 20-27 or 20-39, record type specifications)
 - changed by ALTSEQ statements B-4
 - coding the sign in packed and zoned numeric constants 3-12
 - how to position the keyword beginning in column 20 3-13
 - how to position the numeric constant in columns 20-39 3-11
 - length of factor 2 constant 3-11
 - length of factor 2 field 3-10
 - rules for coding 3-10
- factor 2 constant (columns 20-39—record type specifications) 3-11
- factor 2 field location (columns 20-27—record type specifications) 3-10
 - coding rules 3-10
 - coding when one character long 3-10
 - relationship to factor 1 field 3-10
- factor 2 keyword 3-10
- field
 - defined F-2
- field, constant, or keyword (column 19—record type specifications) 3-10
 - factor 1 field compared with another field, constant, or keyword 3-10
 - used to identify factor 2 field 3-10
- field specifications 4-1
 - as part of include set 3-3
 - column entries, explanation of 4-2, C-3
 - column summary 4-2, C-3
 - column(s)
 - 1-2 (page number) 4-3
 - 3-5 (line number) 4-3
 - 6 (line type) 4-3
 - 7 (field type or comments) 4-3
 - 8 (C/Z/D/P/U/V) 4-6
 - 9-12 (from) 4-10
 - 9-16 (field location) 4-10
 - 13-16 (to) 4-10
 - 17 (conditionally forced character) 4-11
 - 18 (substitute character) 4-11
 - 19 (specifying replacements for control field characters) 4-13
 - 20-22 (overflow field length) 4-19
 - 40-72 (comments) 4-20

- field specifications (continued)
 - columns you must fill in 4-2, C-3
 - explanation of the column entries 4-2, C-3
 - what they describe for addroot sorts 4-3
 - what they describe for summary sorts 4-3
 - what they describe for tagalong sorts 4-3
 - what they tell sort 4-1
 - when using an include-all line 3-2
- field length, dependent on column 8 entry (field specifications) 4-11
- field location (columns 9-16—field specifications)
 - to identify location of field in input record 4-10
- field type or comments (column 7—field specifications) 4-3
- file allocation, automatic work 5-1
 - multiple disk system 6-2
 - single disk system 6-2
 - when using SORT command 5-4
- file characteristics
 - input file 5-1
 - output file 5-1
 - work file 5-1
- file information 5-1
 - running your own procedures 5-5
- file, input overlay considerations 5-2
- file label (see input file label; output file label)
- file location on disk
 - effect on sort time 6-1
- file placement on single disk system 6-2
 - effect on sort time 6-2
- file placement on multiple disk system 6-2
 - effect on sort time 6-2
- file programming considerations 5-2
- FILE statement parameter descriptions 5-5
- FILE statement when running your own procedures 5-5
- file types used with sort program 1-5
- force-all
 - defined F-2
 - explanation 4-5
 - result if not used 4-12
 - used after conditional force lines 4-5
- force-all character
 - example of 4-18
 - how to define 4-12
 - when to use 4-5
- forced character (column 18—field specifications)
 - conditionally forced 4-11
 - used to initialize summary overflow field 4-11
 - when and how character is forced 4-11
- forced control fields
 - conditional 4-5
 - defined F-2
 - examples 4-13
 - force-all 4-5
 - rules for using 4-6
 - types 4-4
 - unconditional 4-5
 - when to use 4-4
- forced data field 4-11
- format of records
 - affect on order of sequence specifications 1-8
 - definition 1-8
- from entry (columns 9-12—field specifications) 4-10
 - describing one character field 4-11
 - to indicate first position of field in input record 4-10
- from entry (columns 9-12—record type specifications) 3-10
 - to indicate first position of factor 1 in input record 3-10
- from entry (columns 20-23—record type specifications) 3-10
 - to indicate first position of factor 2 in input record 3-10
- generation phase E-1
- guide to using include and omit sets 3-4
- header specification
 - column entries, explanation of 2-1, C-1
 - column summary 2-1, C-1
 - column(s)
 - 1-2 (page number) 2-2
 - 3-5 (line number) 2-2
 - 6 (line type) 2-2
 - 7-12 (type of sort job) 2-2
 - 13-17 (largest sum [in bytes] of control field lengths for any record type) 2-2
 - 18 (ascending or descending sequence) 2-2
 - 26 (collating sequence) 2-2
 - 27 (print option) 2-3
 - 28 (output option, SORTR and SORTRS) 2-3
 - 29-32 (output record length for SORTR, SORTRS) 2-3
 - 34 (non-verify option) 2-4
 - 36 (null output—bypass halt for message SORT—7724) 2-4
 - 40-72 (job description) 2-4
 - columns you must fill in 2-1, C-1
 - explanation of column entries 2-1, C-1
 - what they tell sort 2-1
- how the sort program works 1-5, E-1
 - execution phase E-1
 - generation phase E-1
- implied include-all 1-8
 - defined F-2
 - example 1-8
- include line, to select records to sort 3-2
- include set
 - definition 3-3
 - mixing with omit sets 3-4
 - rules for using 3-3
 - types of 3-3
 - when to use 3-4
- include-all
 - defined F-2
 - implied 1-8
- include-all line
 - used to sort records not included or omitted 3-2

- include sequences D-9
 - type 1 D-9
 - type 2 D-9
- initializing an overflow field 4-11
- input file
 - calculating storage tracks for A-1
 - characteristics 5-1
 - defined F-2
 - FILE statement required for 5-2
 - how its position on disk affects sort time 6-2
 - limitations when using the SORT command 5-4
 - overlay considerations 5-2
 - programming considerations 5-1, 5-2
 - running your own procedure 5-5
 - shared file; FILE statement needed for the work file 5-2
- input file label 5-3
- input needed for the sort program 1-5
- input sequence specifications, order when using ALTSEQ B-4
- integer
 - definition F-2
 - sorting D-1

- job (columns 7-12—header specification), to specify
 - type of sort 2-2
- job description (columns 40-72—header specification), to
 - supply comments 2-4
- job step
 - defined F-2
 - in multiple-step procedure 5-4

- keyboard, procedures entered through 5-7
- keyword
 - affect on format of program date 3-13
 - as factor 2 (column 19—record type specification) 3-10
 - defined F-2
 - indicated by column 19 of record type specification 3-10
 - length equals factor 1 on record type specifications 3-13
 - program date affect on factor 1 length 3-13

- label
 - defined F-2
- largest sum (in bytes) of control field lengths for any record
 - type (columns 13-17—header specification) 2-2
 - calculating entry 2-2
- length
 - factor 1 of record type specifications 3-9
 - factor 2 constant (columns 20-39—record type specifications) 3-11
 - factor 2 field location (columns 20-27—record type specifications) 3-10
 - maximum field length for columns 7-8 (field specifications) 4-6
- library
 - (see also system library and user library)
 - defined F-2

- line number (columns 3-5) with page number (columns 1-2) for
 - sequencing
 - field specifications 4-3
 - header specifications 2-2
 - record type specifications 3-2
 - line type (column 6)
 - to identify field line (field specifications) 4-3
 - to identify header line (header specification) 2-2
 - to identify include line (record type specifications) 3-2
 - to identify include-all line (record type specifications) 3-2
 - to identify omit line (record type specifications) 3-2

- messages
 - defined F-2
 - documentation for sort 2-3
 - issued E-1
 - sort printed E-1
 - sort programmer 2-3
 - types issued by the sort program 2-3
- multiple disk system
 - defined F-2
 - file placement on 6-2

- non-verify option (column 34—header specification) 2-4
 - timing considerations 2-4
- normal control field
 - changed by ALTSEQ statements B-4
 - defined F-2
 - how indicated 4-3
 - how sort uses 4-3
- number of records
 - affect on order of the sequence specifications 1-8
 - affect on sort time 6-1
- null output—bypass halt for message SORT—7724 2-4
- number of sequence specifications, affect on sort time 6-1
- numeric constants 3-11
- numeric data
 - data formats and column 8 entry combinations 3-7
 - defined by column 8 record type entry 3-7
 - definition of types 3-7
 - explanation of positive and negative (signed) numeric data 3-8
 - indicated by column 8 of record type specifications 3-7
 - packed numbers 3-7

- OCL statements
 - ways to enter 5-1
 - what they tell sort 5-5
 - what they tell about the input file 5-5
 - what they tell about the output file 5-5
 - what they tell about the work file 5-5
- omit line
 - used to omit records from your sort 3-2
 - when and how to use 3-2

- omit set
 - definition 3-3
 - mixing with include set 3-4
 - rules for using 3-3
 - types of 3-4
 - when to use 3-4
- opposite control field
 - changed by ALTSEQ statements B-4
 - column 8 field specification entry when sorting different record types 4-6
 - defined F-3
 - how indicated 4-3
 - how sort uses 4-6
 - when to use 4-6
- OR lines
 - definition 3-4
 - describing different record types 3-3
 - in include sets 3-3
 - in omit sets 3-4
- order of sequence specifications 1-8
 - affected by number and format of records 1-8
 - normal order 1-8
- organization of this manual iii
- out-of-sequence lines
 - how to insert in field specifications 4-3
 - how to insert in record type specifications 3-2
- output
 - addrout sort 1-2
 - summary tagalong sort 1-4
 - tagalong sort 1-3
- output file
 - calculating storage tracks for A-1
 - characteristics 5-1
 - defined F-3
 - how position on disk affects sort time 6-2
 - limitations when using the SORT command 5-4
 - programming considerations 5-1, 5-2
 - running your own procedure 5-5, 5-6
 - to overlay input file 5-2
- output file label 5-3
- output option for SORTR, SORTRS (column 28—header specification) 2-3
 - used to drop control fields in tagalong (SORTR) sorts 2-3
- output record length for SORTR, SORTRS (columns 29-32—header specification) 2-3
 - calculating when dropping control fields 2-3
 - calculating when not dropping control fields 2-3
- overflow field
 - defined F-3
- overflow field length entry (columns 20-22—field specifications) 4-19
- overflow indicator field
 - how to define (columns 7-8—field specifications) 4-12
 - how to initialize (column 18—field specifications) 4-11
 - when to use 4-19
- overlay of input file 5-2
- overview of the sort program 1-1
 - addrout sort 1-2
 - summary tagalong sort 1-4
 - tagalong sort 1-3
- packed numeric data
 - number of bits in each digit 3-7
 - position of sign code 3-8
 - possible positive and negative sign codes 3-7
- packed, zoned, opposite character, or opposite zone control fields 4-7
- page number (column 1-2)—with line number (columns 3-5) for sequencing (see line number)
- print option (column 27—header specification) 2-3
 - information sort can issue when using 2-3
- printed messages 2-3
- procedures
 - (see also sort procedures)
 - defined F-3
 - SORT command 5-3
- program date
 - as factor 2 (columns 20-39—record type specifications) 3-13
 - defined F-3
 - format affected by factor 2 keyword 3-13
 - indicated by column 19 of record type specifications 3-10
 - position when used as factor 2, record type specifications 3-11
 - possible keywords for 3-13
- program status messages (I) 2-3
- programmer messages for sort 2-3
- programming considerations
 - for using sort 5-1, 5-2
 - for using the SORT command 5-3, 5-4
- prompts
 - defined F-3
 - that appear when SORT is keyed 5-3, 5-4
- real number
 - definition F-3
 - sorting D-1
- record
 - defined F-3
- record format
 - effect on order of sequence specifications 1-8
 - definition 1-8
- record length (output)
 - calculating for tagalong sorts 2-3
 - defined F-3
- record order 6-3
 - effect on sort time 6-3
- record selection
 - binary values D-9
 - type 1 include sequence D-9
 - type 2 include sequence D-9
- record size
 - effect on sort time 6-1

- record type specifications
 - column entries, explanation of 3-1, C-2
 - column summary 3-1, C-2
 - column(s)
 - 1-2 (page number) 3-2
 - 3-5 (line number) 3-2
 - 6 (line type) 3-2
 - 7 (continuation or comments) 3-4
 - 8 (C/Z/D/P/U) 3-5
 - 9-16 (factor 1 field location) 3-9
 - 17-18 (relation) 3-10
 - 19 (field, constant, or keyword) 3-10
 - 20-25 (factor 2 keyword) 3-13
 - 20-27 (factor 2 field location) 3-10
 - 20-39 (factor 2 constant) 3-11
 - 40-72 (comments) 3-13
 - columns you must fill in 3-1, C-2
 - explanation of column entries 3-1, C-2
 - what they tell sort 3-1
 - when not required 3-1
- record types
 - defined F-3
 - described in include set 3-3
 - described in omit sets 3-4
- records
 - how number of affects sort time 6-1
 - how order of affects sort time 6-3
- region
 - defined F-3
- region size
 - effect on sort time 6-3
 - minimum 6-3
- relation (columns 17-18—record type specifications) 3-10
 - effect on omitting or including records 3-10
 - to specify factor 1 and factor 2 comparison 3-10
- relative record numbers, addrout sort output 1-2
- replacements for more than one character (column 19—field specifications)
 - when and how to indicate 4-13
- running time, sort job 5-2, 6-1
 - factors that affect 6-1
- sample sort jobs 7-1
 - SORTA
 - relative record addresses for inventory file 7-7 through 7-9
 - SORTR
 - stock items to reorder 7-3 through 7-6
 - transaction activity 7-9 through 7-16
 - summary history 7-22 through 7-24
 - using signed control fields 7-22
 - SORTRS
 - inventory adjustments 7-17 through 7-18
 - customer number issues 7-19 through 7-21
- select/build routine
 - defined 6-1
 - storage required for 6-1
- sequence
 - changing collating sequence of one character B-4
 - changing collating sequence of several characters B-4
 - standard and alternate collating B-3
 - sequence (column 18—header specification) 2-2
 - to indicate sorted record sequence 2-2
- sequence error message
 - field specifications 4-3
 - header specification 2-2
 - record type specifications 3-2
- sequence specifications
 - defined F-3
 - for a summary tagalong sort (SORTRS) 1-11
 - for a tagalong sort (SORTR) 1-10
 - for an addrout sort (SORTA) 1-9
 - grouping of 1-8
 - how number of affects sort time 6-1
 - inserting new ones in field lines 4-3
 - inserting new ones in record type lines 3-2
 - lines issued by the sort program 2-3
 - normal order of 1-8
 - recommended number for sort job 6-1
 - sequence numbers for 2-2
 - storage of 1-7
 - use of 1-7
 - using additional to decrease sort time 6-1
 - warning issued if not in order 2-2
 - wrong ones used 5-5
- sequence specifications sheet
 - location and purpose of specification types 1-6
 - types of specifications 1-6
- sequencing
 - on field specifications 4-3
 - on header specification 2-2
 - on record type specifications 3-2
 - using control fields 4-6
- set
 - include set 3-3
 - omit set 3-4
 - rules for using include and omit sets 3-3
 - types of include sets 3-3
 - types of omit sets 3-4
- sign code
 - packed and zoned numeric data 3-7
 - placement in packed and zoned numeric data 3-8
- signed constants 3-12
- signed numbers 3-7
 - how to code when used as factor 2 (columns 20-39—record type specifications) 3-11
 - possible sign codes 3-7
- single disk system
 - file placement on 6-2
- size of records, affect on sort time 6-1
- size of work file 5-2
- sort
 - types of 1-2
- SORT command
 - how keyed 5-3
 - limitations 5-4
 - parameters 5-3, 5-4
 - prompts for 5-3
 - sequence specifications placement when using 5-3
 - used to run sort 5-3
 - what it does 5-3
 - work file allocation when using 5-4
 - why used 5-3

- sorting
 - integer (binary) D-1
 - specifications for positive integer D-1 through D-2
 - specifications for positive or negative integer D-3 through D-5
 - real number D-1
 - specifications for positive integer D-1 through D-2
 - specifications for positive or negative real number D-6 through D-8
- sort jobs
 - (see also sample sort jobs)
 - stages of 2-3
 - types of 1-2
- sort messages (see messages)
- sort printed messages E-1
 - issued E-1
- sort procedures
 - entering through a display station keyboard 5-7
 - executing when stored on disk 5-6
 - file information needed when using your own 5-5
 - running sort when using your own 5-5
 - source member information needed when using your own 5-6
- sort program
 - defined F-4
 - execution of 1-7
 - file types used 1-5
 - functions of 1-1
 - general description 1-1
 - how works 1-5
 - input needed 1-5
 - interaction with input, work, and output files 1-7
 - multiple copies of 1-1
 - operation 1-7
 - output considerations 5-5
 - procedures stored on disk, executing 5-6
 - programmer messages 2-3
 - sequence specifications 1-6
- sort time 6-1
 - affected by location of files 6-2
 - affected by number of records 6-1
 - affected by number of sequence specifications 6-1
 - affected by order of records in input file 6-3
 - affected by record size 6-1
 - affected by using alternate collating sequence 6-1
 - affected by using automatic work file allocation 6-3
- sort sequence specifications (see sequence specifications)
- SORTA (addroot sort)
 - (see also sample sort jobs)
 - contents of sorted records 1-2
 - function 1-2
 - input file overlay 5-2
 - output 1-2
 - running time 5-2
 - sequence specifications 1-9
 - using summary data fields 4-5
 - work file size 5-2
- sorted file, contents 1-5
- SORTR (tagalong sort)
 - (see also sample sort jobs)
 - calculating output record length 2-3
 - contents of sorted records 1-3
 - function 1-3
 - input file overlay 5-2
 - output 1-3
 - running time 5-2
 - sequence specifications 1-10
 - using summary data fields 4-5
 - work file size 5-2
- SORTRS (summary tagalong sort)
 - (see also sample sort jobs)
 - calculating output record length 2-3
 - contents of sorted records 1-4
 - function 1-4
 - input file overlay 5-2
 - output 1-4
 - running time 5-2
 - sequence specifications 1-11
 - using overflow indicator fields 4-12
 - using summary data fields 4-5
 - work file size 5-2
- source member
 - creating and updating 5-6
 - defined F-4
 - information needed
 - when running your own procedures 5-6
 - when using the SORT command 5-3
- SOURCE statement, running your own procedures 5-6
- standard collating sequence
 - specified on header line (column 26) 2-2
 - table B-1, B-2
 - when digit portion is used B-1, B-2
 - when zone and digit portions are used B-1, B-2
 - when zone portion is used B-1, B-2
- substitute character (column 18—field specifications)
 - conditionally forced 4-11
 - used to initialize summary overflow fields 4-11
 - when and how character is forced 4-11
- summary data fields
 - defined F-4
 - how overflow indicated 4-12
 - how to define 4-5
 - maximum number defined 4-5
 - used by each type of sort 4-5
- summary overflow field
 - how to define 4-12
 - how to initialize 4-11
- summary sort (SORTRS) (see SORTRS)
- summary sort overflow 4-11
- summary tagalong sort (SORTRS)
 - (see also SORTRS)
 - defined F-4
- system data
 - alphanumeric 3-6
 - how characters represented within the computer 3-6
 - how signed numeric constants are represented on record type specifications 3-7
 - how to interpret 3-5
 - in compare operations 3-5
 - numeric data 3-7
 - zone and digit portions of characters 3-6

- system environment
 - affect on sort job running time 6-3
- system library
 - defined F-4
 - searched if source member not found on disk 5-6

- tagalong sort (SORTR)
 - (see also SORTR)
 - defined F-4
- time required to sort a file 5-2, 6-1
 - affected by alternate collating sequence 6-1
 - affected by automatic work file allocation 6-3
 - affected by file location on disk 6-2
 - affected by multiple disk system 6-2
 - affected by multiprogramming mode 6-2
 - affected by non-verify option 6-3
 - affected by number of input records 6-1
 - affected by number of sequence specifications 6-1
 - affected by order of records in the input file 6-3
 - affected by region size 6-3
 - affected by single disk system 6-2
 - affected by size of records 6-1
 - affected by system environment 6-3
- to entry (columns 13-16—record type specifications) 3-9
 - to identify last position of factor 1 field in the input record 3-9
- to entry (columns 24-27—record type specifications) 3-10
 - to identify last position of factor 2 field in input record 3-10
- translation table and alternate collating sequence coding sheet B-5
- types of sort 1-2

- unconditional force
 - defined F-4
 - example 4-14
 - explanation 4-5
- unconditionally forced character, how to define 4-14
- user library
 - action taken if omitted 5-3
 - defined F-4
 - when required 5-3
- using duplicate control fields 4-4

- warning
 - issued when sequence numbers not in order 2-2
- work file
 - automatic allocation 5-1
 - calculating size when programming A-1
 - calculating storage tracks for A-1
 - defined F-4
 - FILE statement needed when input file shared 5-4
 - how position on disk affects sort time 6-2
 - how used 1-5
 - programming considerations 5-1, 5-2
 - running your own procedure 5-5
 - SORT command affect on 5-5
 - work record, how used by sort 1-5
- work record length
 - affected by type of output A-1
 - affected by type of sort A-1

- zone, portion of system character 3-6
- zoned numeric data 3-7
 - number of bits in each digit 3-7
 - numbers 3-7
 - position of the sign code 3-8
 - possible positive and negative sign codes 3-7

READER'S COMMENT FORM

Please use this form only to identify publication errors or request changes to publications. Technical questions about IBM systems, changes in IBM programming support, requests for additional publications, etc, should be directed to your IBM representative or to the IBM branch office nearest your location.

Error in publication (typographical, illustration, and so on). **No reply.**

Page Number *Error*

Inaccurate or misleading information in this publication. Please tell us about it by using this postage-paid form. We will correct or clarify the publication, or tell you why a change is not being made, provided you include your name and address.

Page Number *Comment*

Note: All comments and suggestions become the property of IBM.

● No postage necessary if mailed in the U.S.A.

Check if reply is requested.

Name _____

Address _____

Fold

Fold

FIRST CLASS
PERMIT NO. 40
ARMONK, N. Y.

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES

POSTAGE WILL BE PAID BY . . .

IBM Corporation
General Systems Division
Development Laboratory
Publications, Dept. 245
Rochester, Minnesota 55901



Fold

Fold



International Business Machines Corporation

**General Systems Division
4111 Northside Parkway N.W.
P.O. Box 2150
Atlanta, Georgia 30301
(U.S.A. only)**

**General Business Group/International
44 South Broadway
White Plains, New York 10601
U.S.A.
(International)**



International Business Machines Corporation

General Systems Division
4111 Northside Parkway N.W.
P.O. Box 2150
Atlanta, Georgia 30301
(U.S.A. only)

General Business Group/International
44 South Broadway
White Plains, New York 10601
U.S.A.
(International)

IBM System/34 Sort Reference Manual (File No. S34-33) Printed in USA SC21-7658-1

SC21-7658-1