

SC21-7658-4

File No. S34-33

IBM System/34

Sort

Reference Manual

Program Number 5726-UT1



SC21-7658-4

File No. S34-33

IBM System/34

Sort

Reference Manual

Program Number 5726-UT1

Fifth Edition (January 1982)

This is a major revision of, and obsoletes, SC21-7658-3 and Technical Newsletter SN21-8141. Changes or additions to the text and illustrations, except for the sequence specification form, are indicated by a vertical line to the left of the change or addition.

This edition applies to release 8, modification 0 of the 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.

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

It is possible that this material might contain reference to, or information about, IBM products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that IBM intends to announce such IBM products, programming, or services in your country. (For example, ideographic support is available only in Far East countries.)

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.

IBM may use and distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever. You may, of course, continue to use the information you supply.

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 other programmers and IBM field support personnel.

How This Manual is Organized

This manual has seven chapters, six 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	Loadable Sort Interface
F	Sort printed messages

System Requirements

Refer to the *IBM System/34 Planning Guide* 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

Coding Material

- *Sequence Specifications*, GX21-9089
- *Translation Table and Alternate Collating Sequence Coding Sheet*, GX21-9096

Related System/34 Publications

- *IBM System/34 Basic Assembler and Macro Processor Reference Manual*, SC21-7705
- *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 Installation and Modification Reference Manual: Program Products and Physical Setup*, SC21-7689
- *IBM System/34 Master Index*, SC21-7739
- *IBM System/34 Bibliography*, GH30-0231
- *IBM System/34 Programming Reference Summary*, GX21-7674
- *IBM System/34 Concepts and Design Guide*, SC21-7742

The *Publications Summary* in the *System/34 Introduction* 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 (Portion of Input Record to Use in Building/Sorting Work Records)	4-6
Sort Sequence Specifications	1-6	Columns 7-8	4-10
Sort Execution	1-7	Columns 9-16 (Field Location)	4-11
Sort Operation	1-7	Column 17 (Conditionally Forced Character)	4-12
Overview of Sequence Specifications	1-8	Column 18 (Substitute Character)	4-12
Order of Sequence Specifications	1-8	Column 19 (Specifying Replacements for More Than One Character)	4-14
Specification Columns to Consider for Addrout Sort (SORTA)	1-9	Examples of Using Forced Control Fields	4-14
Specification Columns to Consider for Tagalong Sort (SORTR)	1-10	Columns 20-22 (Overflow Field Length)	4-20
Specification Columns to Consider for Summary Tagalong Sort (SORTRS)	1-11	Summary Overflow Indicator Field	4-23
CHAPTER 2. HEADER SPECIFICATION	2-1	Column 20 (Alternate Collating Sequence by Field)	4-25
Header Specification Column Summary	2-2	Columns 23-39 (Reserved)	4-25
Column Descriptions	2-3	Columns 40-74 (Comments)	4-25
Columns 1-2 (Page Number) and 3-5 (Line Number)	2-3	CHAPTER 5. HOW TO RUN SORT	5-1
Column 6 (Line Type)	2-3	File Information	5-1
Columns 7-12 (Job)	2-3	File Characteristics	5-1
Column 12 (Equal Control Field Ordering)	2-3	File Programming Considerations	5-3
Columns 13-17 (Largest Sum [in bytes] of Control Field Lengths for Any Record Type)	2-4	Running Sort by Using the SORT Command	5-4
Column 18 (Ascending or Descending Sequence)	2-4	SORT Command Parameters	5-4
Column 26 (Collating Sequence)	2-4	SORT Command Keying	5-5
Column 27 (Print Option)	2-5	SORT Command Limitations	5-6
Column 28 (Output Option for Tagalong Sorts)	2-6	Running SORT by Using Your Own Sort Procedures	5-7
Columns 29-32 (Output Record Length [in bytes] for Tagalong Sorts)	2-6	File Information Needed When Using Your Own Sort Procedures	5-7
Column 34 (Non-Verify Option)	2-6	Source Member Information Needed When Using Your Own Sort Procedures	5-8
Column 36 (Null Output—Bypass Halt for Message SORT-7724)	2-6	Executing Sort Procedures Stored on Disk	5-8
Columns 40-74 (Job Description)	2-6	Entering Sort Procedures Through a Display Station Keyboard	5-10
CHAPTER 3. RECORD TYPE SPECIFICATIONS	3-1	CHAPTER 6. PERFORMANCE CONSIDERATIONS	6-1
Record Type Specification Column Summary	3-2	Number of Input Files	6-1
Column Descriptions	3-3	Number of Records to be Sorted	6-1
Columns 1-2 (Page Number) and 3-5 (Line Number)	3-3	Record Size	6-1
Column 6 (Line Type)	3-3	Number of Sequence Specifications	6-1
Column 7 (Continuation or Comments)	3-5	Alternate Collating Sequence	6-1
Column 8 (C/Z/D/P/U)	3-6	Disk Location of Files	6-2
Columns 9-16 (Factor 1 Field Location)	3-10	File Placement On a Single Disk System	6-2
Columns 17-18 (Relation)	3-11	File Placement On a Multiple Disk System	6-2
Column 19 (Field, Constant, or Keyword)	3-11	Automatic Work File Allocation	6-3
Columns 20-27 (Factor 2 Field Location)	3-11	Single Disk System	6-3
Columns 20-39 (Factor 2 Constant)	3-12	Multiple Disk System	6-3
Columns 20-39 (Factor 2 Keyword)	3-14	Order of Records in the Input File	6-3
Columns 40-74 (Comments)	3-14	Non-Verify Option	6-3
		Region Size	6-3
		System Environment in Which a Sort Job Executes	6-3
		Loadable Sort Interface	6-3
		Sample Timings	6-4

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
Sample Job 8: Reformatting Inventory File Using Equal Control Field Ordering	7-25
Header Specification	7-25
Record Type Specifications	7-26
Field Specifications	7-26
Sample Job 9: Merging Multiple Presorted Input Files Using Equal Control Field Ordering	7-27
Header Specification	7-28
Record Type Specifications	7-28
Field Specifications	7-28
 APPENDIX A. CALCULATING THE SIZES OF FILES FOR SORT.	 A-1
Input File	A-1
Output File	A-1
Work File	A-2
Work Record Length	A-2

APPENDIX B. COLLATING SEQUENCE	B-1
Standard Collating Sequence	B-3
Alternate Collating Sequence	B-3
Coding Rules	B-3
Input Order When ALTSEQ Statements Are Used	B-3
Programming Considerations	B-4
Effect of ALTSEQ Statements on Other Coding	B-4
Sample ALTSEQ Statements	B-4
Programming Aids	B-4
Translation Table and Alternate Collating Sequence	
Coding Sheet	B-5

APPENDIX C. HEADER, RECORD TYPE, AND FIELD SPECIFICATION COLUMN SUMMARIES	C-1
Header Specification Column Summary	C-1
Record Type Specification Column Summary	C-2
Field Specification Column Summary	C-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. LOADABLE SORT INTERFACE	E-1
Loadable Sort Parameter List	E-1

APPENDIX F. SORT PRINTED MESSAGES	F-1
--	------------

GLOSSARY	G-1
-----------------	------------

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

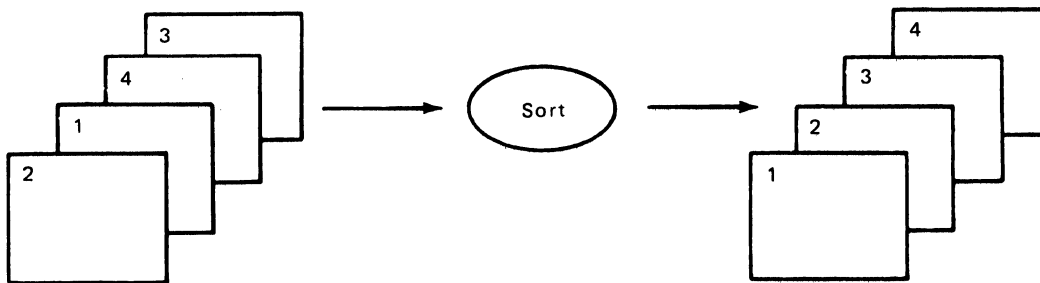
The sort program is part of the System/34 Utilities Program product. You can use the sort program to rearrange, drop, and reformat records in your files.

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

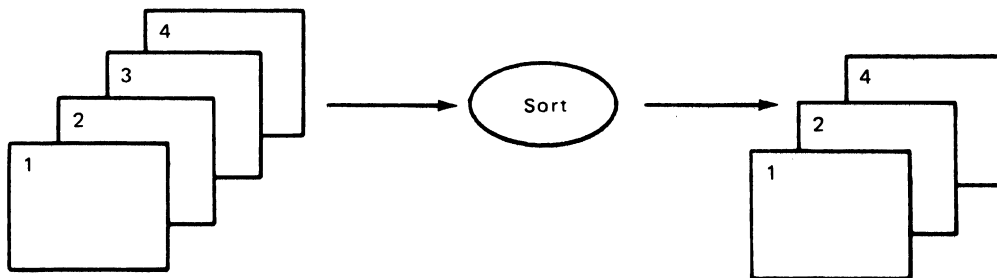
SORT FUNCTIONS

The three basic functions of sort are:

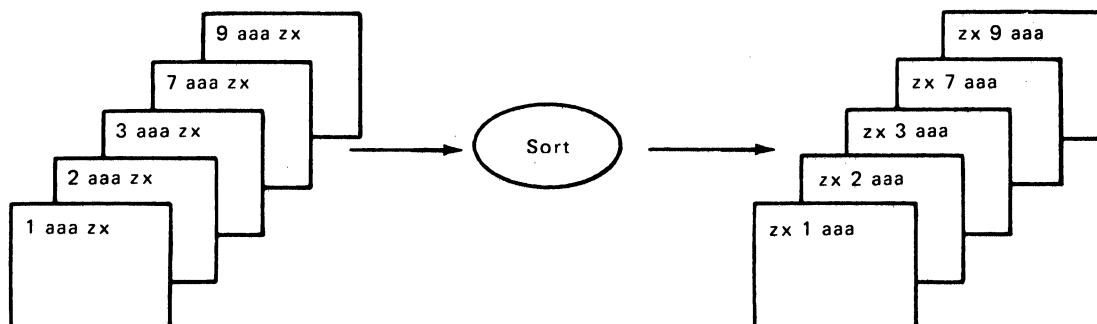
- To rearrange records in a file



- To drop records from a file



- To reformat records in a file



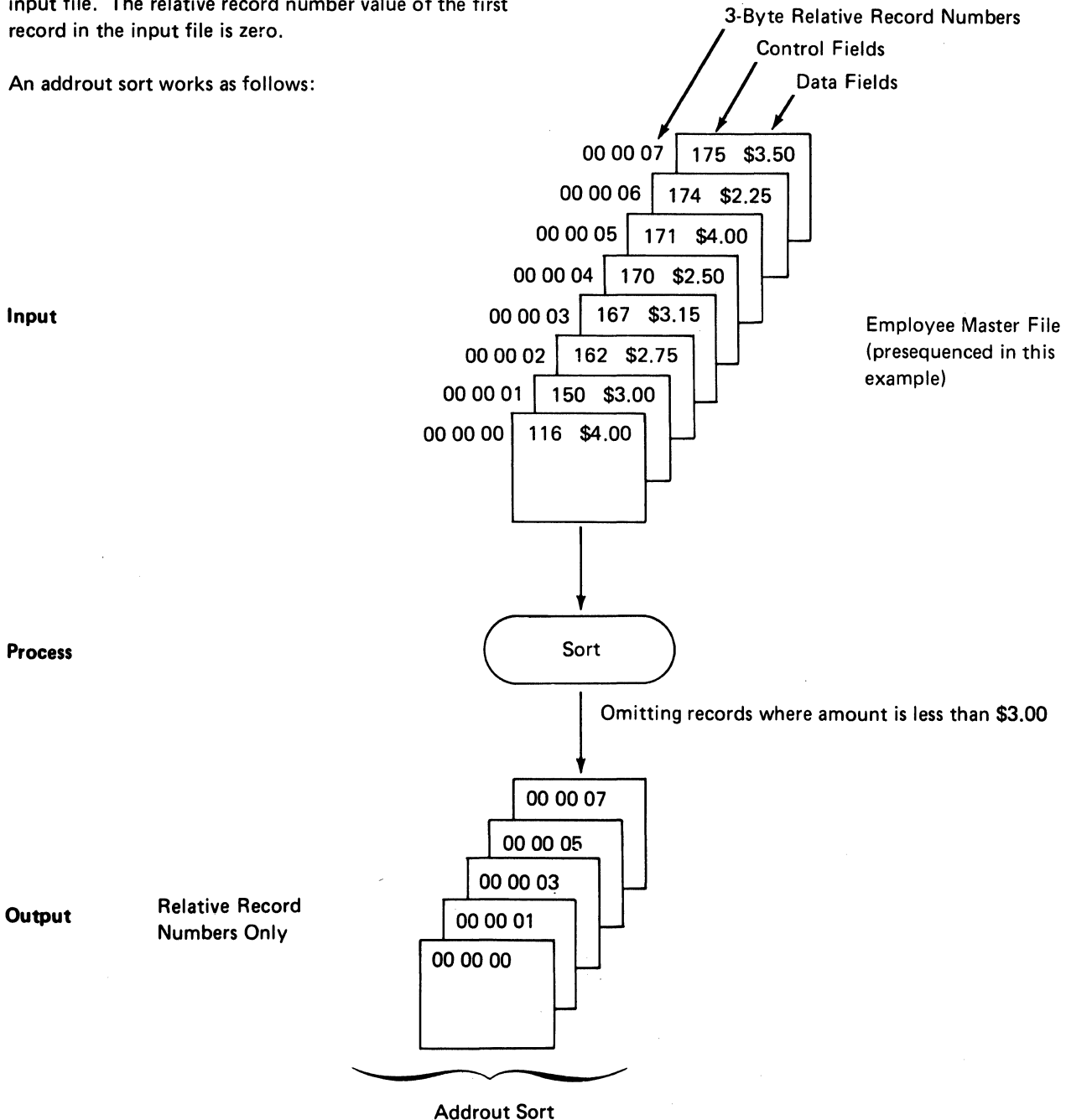
TYPES OF SORT: ADDRROUT, TAGALONG, AND SUMMARY TAGALONG

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. The relative record number value of the first record in the input file is zero.

An addrout sort works as follows:

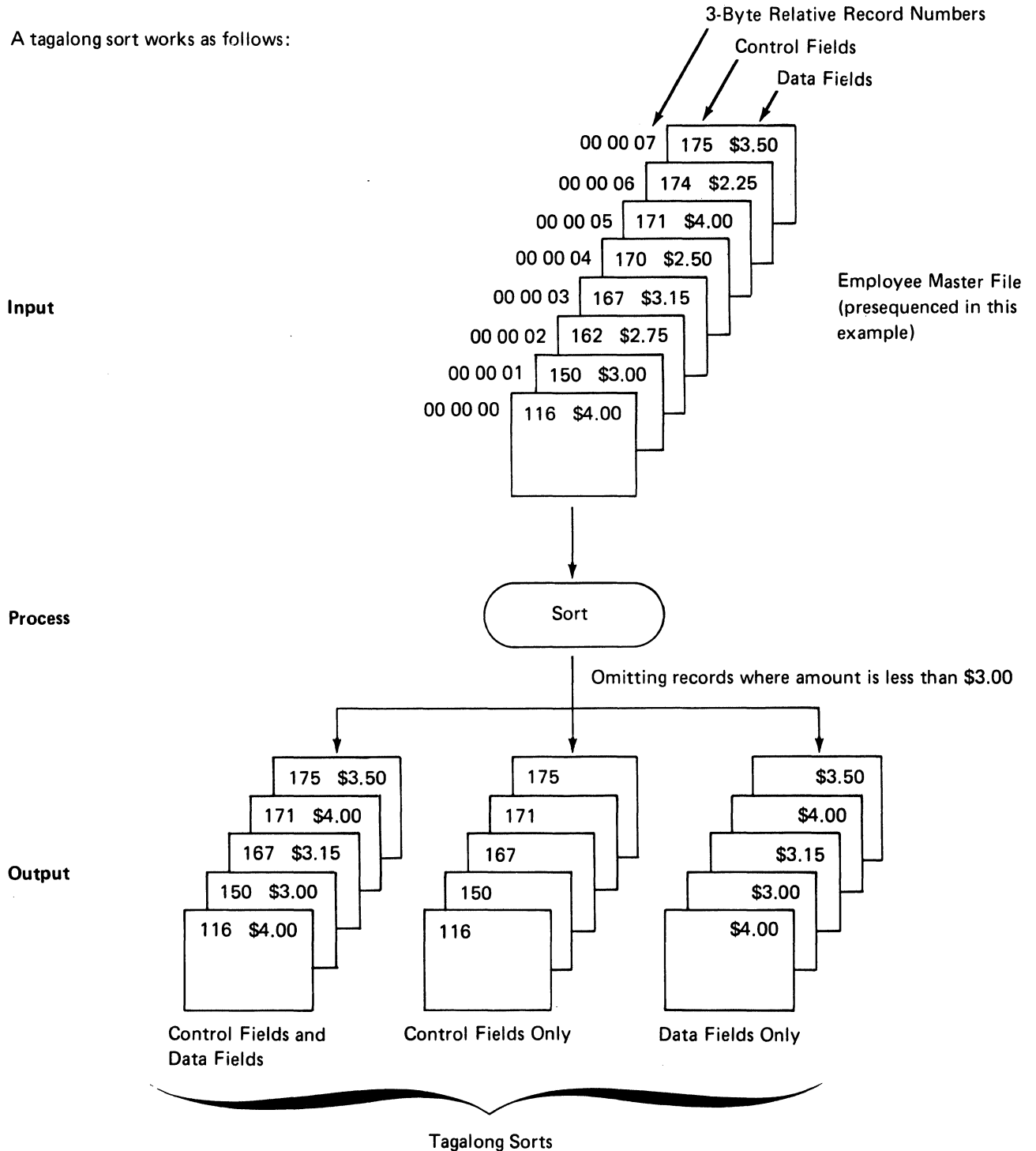


Tagalong Sort

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:



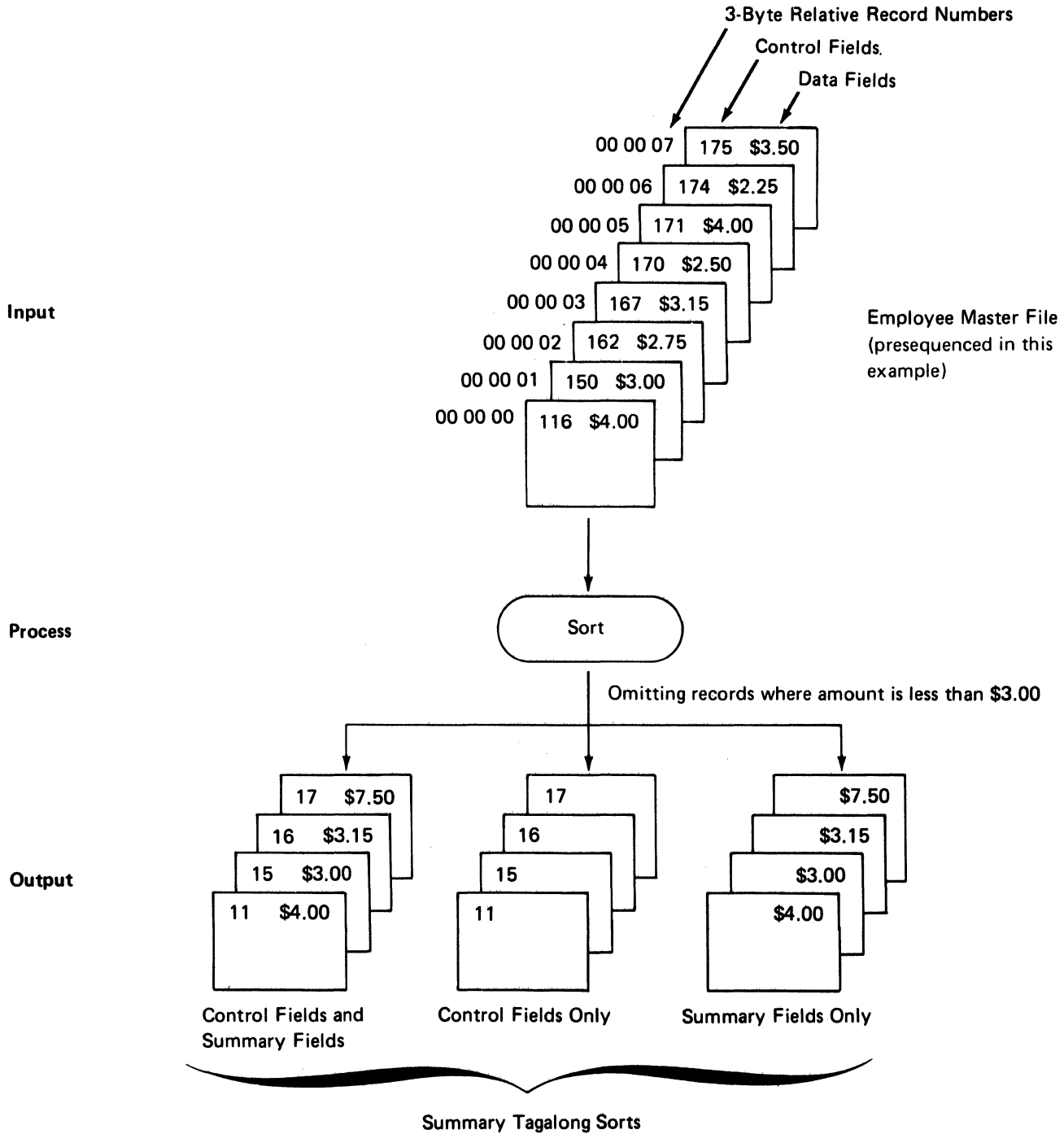
Summary Tagalong Sort

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

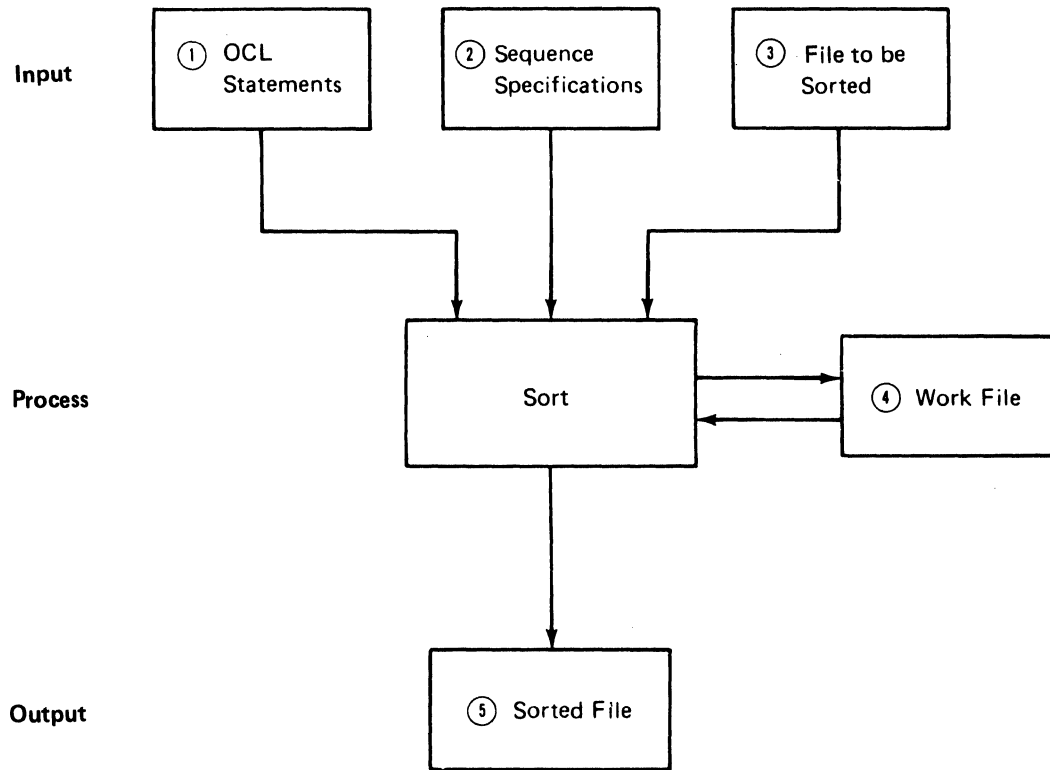
Summary sort output can also include data fields. However, you cannot determine which input record's data field(s) will appear in the summary record.

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.

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.

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.

Sort Sequence Specifications

You must always specify how you want a file sorted. You use sequence specifications to describe the type of sort you want, which records you want to sort, and how you want the records sorted. Sequence specifications can be stored as a source member or within a procedure member in a library; these specifications can also be entered through the display station keyboard. See Chapter 5 for a detailed description of how to supply sequence specifications for your sort job.

Sequence Specifications Sheet

The sequence specifications sheet contains three different kinds of specifications: header specification, record type specifications, and field specifications. The sample sequence specifications sheet shows the location and purpose of all three:

Line		Job	Card Match	CCP/Disk Sort Only	
Number	Type	Card: MATCH, SORT, SELECT, MERGE	Stacker Select	Record Length	Job Description
3	4	5	6	7	8
0	0	0	0	0	0

Line		Factor 1	Rel	Factor 2 (Field, Constant, or Keyword)	Comments
Number	Type	Location		Constant	Record Name
0	1				
0	2				
0	3				
0	4				
0	5				
0	6				

Line		Field	Overflow Field Length	Comments
Number	Type	Location	Reserved	Field Name
0	7			
0	8			
0	9			
1	0			
1	1			
1	2			
1	3			
1	4			

Here you specify the type of sort job you will run.

Here you specify which input file records you will sort.

Here you specify how you want the records sorted and formatted.

Sort Execution

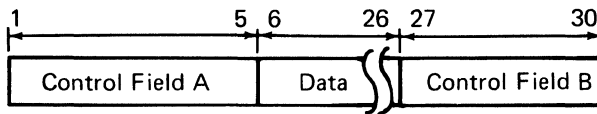
You can use the SORT command statement to execute certain sort jobs. See *Running Sort by Using the SORT Command* in Chapter 5 for a detailed description of how the SORT command works.

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:

- Sort reads a record from the input file.



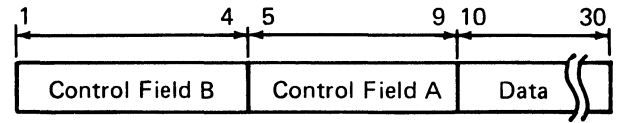
Note: Control fields are the fields you want the sort program to use to sort the records. See *Column 7* under *Column Descriptions* in Chapter 4 for a detailed discussion of control fields.

- 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.)
- 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:

- Put the contents of positions 27 through 30 in the input record into positions 1 through 4 of the work record.
- Put the contents of positions 1 through 5 in the input record into positions 5 through 9 of the work record.
- 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.)

- The sort program writes all of the input records specified by your record type specifications into the work file.
- 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.)
- 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	<ol style="list-style-type: none"> 1. Header line 2. Field lines 									
Some of the records in the file	All the same format	<ol style="list-style-type: none"> 1. Header line 2. Record type lines 3. Field lines 									
All or some of the records in the file	Several different formats	<table style="width: 100%; border: none;"> <tr> <td style="width: 60%; vertical-align: top;"> <ol style="list-style-type: none"> 1. Header line 2. Record type lines 3. Field lines </td> <td style="font-size: 2em; vertical-align: middle;">}</td> <td style="vertical-align: middle;">For first type of record format</td> </tr> <tr> <td style="vertical-align: top;"> <ol style="list-style-type: none"> 4. Record type lines 5. Field lines </td> <td style="font-size: 2em; vertical-align: middle;">}</td> <td style="vertical-align: middle;">For second type of record format</td> </tr> <tr> <td style="vertical-align: top;"> <ol style="list-style-type: none"> 6. Record type lines 7. Field lines </td> <td style="font-size: 2em; vertical-align: middle;">}</td> <td style="vertical-align: middle;">One set for each additional type of record format¹</td> </tr> </table>	<ol style="list-style-type: none"> 1. Header line 2. Record type lines 3. Field lines 	}	For first type of record format	<ol style="list-style-type: none"> 4. Record type lines 5. Field lines 	}	For second type of record format	<ol style="list-style-type: none"> 6. Record type lines 7. Field lines 	}	One set for each additional type of record format ¹
<ol style="list-style-type: none"> 1. Header line 2. Record type lines 3. Field lines 	}	For first type of record format									
<ol style="list-style-type: none"> 4. Record type lines 5. Field lines 	}	For second type of record format									
<ol style="list-style-type: none"> 6. Record type lines 7. Field lines 	}	One set for each additional type of record format ¹									

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.

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 Addrout Sort (SORTA)

The shaded columns are the ones you must consider for an addrout sort job.

Type of Sort

Control Field and Sequence Information

Page Sequence

IBM International Business Machines Corporation
SEQUENCE SPECIFICATIONS Header

GX21-9089 /UM050*
Printed in U.S.A.

Line	Job	Largest sum (in bytes) of Control Field Lengths for Any Record Type	Card Match	Stacker Select	Output Record Length	Check-point Option (C)	Non-Verify Option (N)	CCP/Disk Sort Only	Record Length	Job Description
Number	Type	Sequence (A/D/S)	S S P P S P	S S P P S P	Number (I/N)	All. Coll. Seq. (S/F)	Print Option	Output Option (X)	Record Length	
3	4	5	6	7	8	9	10	11	12	13
3	SORTA									

Page 2 Program Identification 75 76 77 78 79 80

Record Type

Line	Factor 1	Rel.	Factor 2 (Field, Constant, or Keyword)	Comments
Number	Location	EQ NE LT GT LE GE	Location	Record Name
3	4	5	6	7
3			Constant	Comparison of an Input Record Field and a Constant
4			Field	Comparison of Two Input Record Fields
5			Field	Comparison of an Input Record Field and the Program Date.

Field

Line	Location	Record Character	Substitute Character	Continuation	Alt. Seq. Field (A)	Overflow Field Length	Reserved	Field Name	Comments
Number	From	To							
3	4	5	6	7	8	9	10	11	12
0 8									Definition of Normal Control Fields
0 9									Definition of Opposite Control Fields
1 1									Definition of Forced Control Fields
1 2									
1 4									

■ Ideographic sort utility only.

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 /JUM050*
Printed in U.S.A.

Line	Job	Card Match	Sequence (A/D/S)	Stacker Select	Number (1/N)	Print Option	Output Record Length	CCP/Disk Sort Only	Record Length	Job Description
3	Card: MATCH, SORT, SELECT, MERGE	S	U	S	1	1				
4	Largest sum (in bytes) of Control Field Lengths for Any Record Type	M	S	M	1	1				
5	Disc: SORTR; E SORTA, SORTRS	P	U	P	1	1				
6	Tap: SORTT	S	O	S	1	1				
7		O								
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										

Page	1	2				
Program Identification	75	76	77	78	79	80

Record Type

Line	Factor 1	Rel.	Factor 2 (Field, Constant, or Keyword)	Comments
3		ED	Constant	
4		NE	Keyword	
5		LT		
6		GT		
7		LT		
8		GT		
9		LT		
10		GT		
11		LT		
12		GT		
13		LT		
14		GT		
15		LT		
16		GT		
17		LT		
18		GT		
19		LT		
20		GT		
21		LT		
22		GT		
23		LT		
24		GT		
25		LT		
26		GT		
27		LT		
28		GT		
29		LT		
30		GT		
31		LT		
32		GT		
33		LT		
34		GT		
35		LT		
36		GT		
37		LT		
38		GT		
39		LT		
40		GT		

Line	Record Character	Forced	Overflow Field Length	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				

Format for Sorted File

80	81	82	83	84	85	86	87	88	89	90	91	92	93	94
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

Comparison of an Input Record Field and a Constant

Field	Constant
-------	----------

Comparison of Two Input Record Fields

Field	Field
-------	-------

Comparison of an Input Record Field and the Program Date.

Field	Keyword
-------	---------

Definition of Normal Control Fields

0 8	F		
0 9	F		
1 0	F		
1 2	F		
1 4	F		

Definition of Opposite Control Fields

0 8	F		
0 9	F		
1 0	F		
1 2	F		
1 4	F		

Definition of Forced Control Fields

0 8	F		
0 9	F		
1 0	F		
1 2	F		
1 4	F		

Definition of Data Fields

0 8	F		
0 9	F		
1 0	F		
1 2	F		
1 4	F		

■ Ideographic sort utility only.

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 /UM060*
 Printed in U.S.A.
 Page 2
 Program Identification 75 76 77 78 79 80

Line Number	Job	Card Match	Stacker Select	Output Record Length	CCP/Disk Sort Only	Job Description
3	Card: MATCH, SORT, SELECT, MERGE	S	S	Output Record Length	Record Length	
4	Card: SORT, E	U	M	Output Record Length	Record Length	
5	Card: SORTA, SORTAS	P	P	Output Record Length	Record Length	
6	Card: SORTT	S	S	Output Record Length	Record Length	
7	Largest sum (in bytes) of Control Field Lengths for Any Record Type	M	M	Output Record Length	Record Length	
8		U	U	Output Record Length	Record Length	
9		P	P	Output Record Length	Record Length	
10		S	S	Output Record Length	Record Length	
11		M	M	Output Record Length	Record Length	
12		U	U	Output Record Length	Record Length	
13		P	P	Output Record Length	Record Length	
14		S	S	Output Record Length	Record Length	
15		M	M	Output Record Length	Record Length	
16		U	U	Output Record Length	Record Length	
17		P	P	Output Record Length	Record Length	
18		S	S	Output Record Length	Record Length	
19		M	M	Output Record Length	Record Length	
20		U	U	Output Record Length	Record Length	
21		P	P	Output Record Length	Record Length	
22		S	S	Output Record Length	Record Length	
23		M	M	Output Record Length	Record Length	
24		U	U	Output Record Length	Record Length	
25		P	P	Output Record Length	Record Length	
26		S	S	Output Record Length	Record Length	
27		M	M	Output Record Length	Record Length	
28		U	U	Output Record Length	Record Length	
29		P	P	Output Record Length	Record Length	
30		S	S	Output Record Length	Record Length	
31		M	M	Output Record Length	Record Length	
32		U	U	Output Record Length	Record Length	
33		P	P	Output Record Length	Record Length	
34		S	S	Output Record Length	Record Length	
35		M	M	Output Record Length	Record Length	
36		U	U	Output Record Length	Record Length	
37		P	P	Output Record Length	Record Length	
38		S	S	Output Record Length	Record Length	
39		M	M	Output Record Length	Record Length	
40		U	U	Output Record Length	Record Length	
41		P	P	Output Record Length	Record Length	
42		S	S	Output Record Length	Record Length	
43		M	M	Output Record Length	Record Length	
44		U	U	Output Record Length	Record Length	
45		P	P	Output Record Length	Record Length	
46		S	S	Output Record Length	Record Length	
47		M	M	Output Record Length	Record Length	
48		U	U	Output Record Length	Record Length	
49		P	P	Output Record Length	Record Length	
50		S	S	Output Record Length	Record Length	
51		M	M	Output Record Length	Record Length	
52		U	U	Output Record Length	Record Length	
53		P	P	Output Record Length	Record Length	
54		S	S	Output Record Length	Record Length	
55		M	M	Output Record Length	Record Length	
56		U	U	Output Record Length	Record Length	
57		P	P	Output Record Length	Record Length	
58		S	S	Output Record Length	Record Length	
59		M	M	Output Record Length	Record Length	
60		U	U	Output Record Length	Record Length	
61		P	P	Output Record Length	Record Length	
62		S	S	Output Record Length	Record Length	
63		M	M	Output Record Length	Record Length	
64		U	U	Output Record Length	Record Length	
65		P	P	Output Record Length	Record Length	
66		S	S	Output Record Length	Record Length	
67		M	M	Output Record Length	Record Length	
68		U	U	Output Record Length	Record Length	
69		P	P	Output Record Length	Record Length	
70		S	S	Output Record Length	Record Length	
71		M	M	Output Record Length	Record Length	
72		U	U	Output Record Length	Record Length	
73		P	P	Output Record Length	Record Length	
74		S	S	Output Record Length	Record Length	

Record Type

Line Number	Factor 1	Rel.	Factor 2 (Field, Constant, or Keyword)	Comments
3	Location	EQ	Constant	
4	Location	NE	Keyword	
5	Location	LT	Constant	
6	Location	GT	Keyword	
7	Location	LE	Constant	
8	Location	GE	Keyword	
9	Location	FC/K/B	Field	
10	Location	FC/K/B	Constant	
11	Location	FC/K/B	Field	
12	Location	FC/K/B	Field	
13	Location	FC/K/B	Field	
14	Location	FC/K/B	Field	
15	Location	FC/K/B	Field	
16	Location	FC/K/B	Field	
17	Location	FC/K/B	Field	
18	Location	FC/K/B	Field	
19	Location	FC/K/B	Field	
20	Location	FC/K/B	Field	
21	Location	FC/K/B	Field	
22	Location	FC/K/B	Field	
23	Location	FC/K/B	Field	
24	Location	FC/K/B	Field	
25	Location	FC/K/B	Field	
26	Location	FC/K/B	Field	
27	Location	FC/K/B	Field	
28	Location	FC/K/B	Field	
29	Location	FC/K/B	Field	
30	Location	FC/K/B	Field	
31	Location	FC/K/B	Field	
32	Location	FC/K/B	Field	
33	Location	FC/K/B	Field	
34	Location	FC/K/B	Field	
35	Location	FC/K/B	Field	
36	Location	FC/K/B	Field	
37	Location	FC/K/B	Field	
38	Location	FC/K/B	Field	
39	Location	FC/K/B	Field	
40	Location	FC/K/B	Field	
41	Location	FC/K/B	Field	



Field

Line Number	Record Character	Substitute Character	Continuation	Alt. Seq. Field (A)	Overflow Field Length	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						
73						
74						

Chapter 2. Header Specification

Only one header specification line is used for each sort job.
Possible column entries, summarized on the next 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.
	E (in column 12 only)	Maintain original sequence of equal control field records for addrout and tagalong sort jobs.
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 on the entire control field in compare operations. ALTSEQ statements will define the collating sequence to be used.
	F	Use an alternate collating sequence on specified control fields 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	1	Used only in System/34 ideographic sort jobs.
	2	Used only in System/34 ideographic sort jobs.
	Blank	Used only in System/34 ideographic sort jobs.
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.
40-74	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.

If page and line numbers are not entered, they will be created by default.

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.

Column 12 (Equal Control Field Ordering)

An E in column 12 tells the sort program that you want equal control field ordering done during a sort. Equal control field ordering can be used only with addrout sort (SORTA) and with tagalong sort (SORTR).

When column 18 of the header specification contains an A (ascending sequence), equal control field ordering causes records with identical control fields to be written to the output file in the same order they were read from the input file. When column 18 contains a D (descending sequence), equal control field ordering causes records with identical control fields to be written to the output file in the opposite order than they were read from the input file.

When using equal control field ordering:

- You must specify an E in column 12 of the header line.
- If you specify control fields, you must total the control field lengths for each record type, add 3 to the largest of these totals, and place that quantity in columns 13 through 17 of the header line. You must add 3 to the largest total of the control field lengths because the sort program forces a 3-byte relative record number into the last 3 bytes of the control field. Because the maximum control field length for any record type is 256, the control field lengths cannot exceed 253 when equal control field ordering is specified.
- Control fields defined by field specifications are not required; the sort program can be used just to reformat the records (*Sample Job 8* in Chapter 7). When control fields are not specified on field specifications, you must place a 3 in columns 13 through 17 of the header line to accommodate the 3-byte relative record number that the sort program forces into the last 3 bytes of the control field.
- To drop the control field information (the original control field data plus the 3-byte relative record number) from the output record after the records are sorted, code an X in column 28 on the header line for a tagalong sort job (SORTR). If you want to retain the original control field data in the output record, repeat the original control field data as a data field. If column 28 is blank and you code the original control field as a data field, the data in the control field will be specified twice in the output record.
- If you use two or more input files for a job, you can perform a merge by doing one of the following:
 - Add equal control field ordering to the sequence specifications that you originally used to sort the files.
 - If your original sequence specifications caused the record to be reformatted, you must specify new sequence specifications that include equal control field ordering (*Sample Job 9* in Chapter 7).

Note: The input files do not have to be presorted in order to perform a merge. The sort and merge can be performed simultaneously.

- The sequence in which the sort program merges the records with identical control fields into the output file depends on the entry in column 18 of the header specification.
 - When column 18 contains an A (ascending sequence) and the job contains only one input file, the records with identical control fields from the input file are placed into the output file in the same order they were read from the input file. When column 18 contains an A and the job contains multiple input files, the records with identical control fields in the lowest numbered input file are placed first in the output file; then the records with identical control fields in the next higher numbered input file are placed into the output file until all the records with identical control fields from all of the files have been placed into the output file. (For information on multiple input files and how they are processed, see Chapter 5.) Within the output file, the records with identical control fields from each input file are placed in the same order they were read from the input file.
 - When column 18 contains a D (descending sequence) and the job contains only one input file, the records with identical control fields from the input file are placed into the output file in the opposite order from which they were read from the input file. When column 18 contains a D and the job contains multiple input files, the records with identical control fields in the highest numbered input file are placed first in the output file; then the records with identical control fields in the next lower numbered input file are placed into the output file until all the records with identical control fields from all of the files have been placed into the output file. (For information on multiple input files and how they are processed, see Chapter 5.) Within the output file, the records with identical control fields from each input file are placed in the opposite order from which they were read from the input file.

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

To calculate this entry:

1. Add the lengths of the control fields (N, O, and F in column 7 of the field specifications) for each type of input record.

If column 12 contains an E, add 3 to the largest of the totals of the control field lengths. (For further information, see *Column 12 (Equal Control Field Ordering)* in this chapter.)
2. Enter the largest of the totals in columns 13 through 17. (The total, including the 3 added if column 12 contains an E, must not exceed 256. Therefore, with an E in column 12, the control field length for any record type cannot exceed 253.)

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 on the Entire Control Field

An S in column 26 tells the sort program you want to change the standard collating sequence on the entire control field. 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 on the entire control field.

Alternate Collating Sequence on Specified Control Fields

An F in column 26 tells the sort program you want to change the standard collating sequence for specified control fields. Only normal and opposite control fields can be changed. For any normal or opposite control field you want to change, place an A in column 20 of the control field line. You must also supply ALTSEQ statements immediately following the header specifications. Appendix B explains how to code ALTSEQ statements.

Remember the following when specifying an alternate collating sequence on specified control fields:

- Record selection (including and omitting of records) and conditional forcing (force and force-all) are based on an input record that has not been changed by the alternate collating sequence.
- Any control field that has an A specified in column 20 should not be packed or zoned (P or U in column 8). Any control field that does not have an A specified in column 20 can be packed or zoned.

- Packed or zoned factors can be specified in include or omit record type specifications (P or U in column 8).
- If you specify an alternate collating sequence for a particular field, that field will be altered according to the alternate collating sequence whenever it is used again as a control field for that record type. (For an example, see *Column 20 (Alternate Collating Sequence by Field)* in Chapter 4.)

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 F 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, packed or zoned control fields, an alternate collating sequence, or equal control field ordering. In the first three cases, the sort program changes the control information (during the sorting process) so that it is meaningless. In the last case, the sort program forces a 3-byte relative record number into the last 3 bytes of the control field.

Using Fields Both as Control and Data Fields

If you are using opposite control fields, packed or zoned 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 40-74 (Job Description)

Columns 40 through 74 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.

Chapter 3. Record Type Specifications

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 the next 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.
	S	Used only in System/34 ideographic sort jobs.
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.
40-74	Any characters	Comments.
<div style="display: flex; align-items: center;"> <div style="width: 20px; height: 15px; background-color: #cccccc; border: 1px solid black; margin-right: 5px;"></div> Columns that must be filled in. </div>		

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.
	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.
	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.
	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).
Include only one record type (implied include-all)	H		Header line.
	F		No record type lines.
	F		Field line(s).
Include-all	H, F, or O		Header line, field line, or omit line.
	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.
	F		Field line(s).

} Record type lines

} Record type lines

} Record type lines

} Record type lines

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

} Record type lines

} Record type lines

} Record type lines

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

¹ For both factor 1 and factor 2 fields, which must be the same length.
² 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 on the entire control field (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.

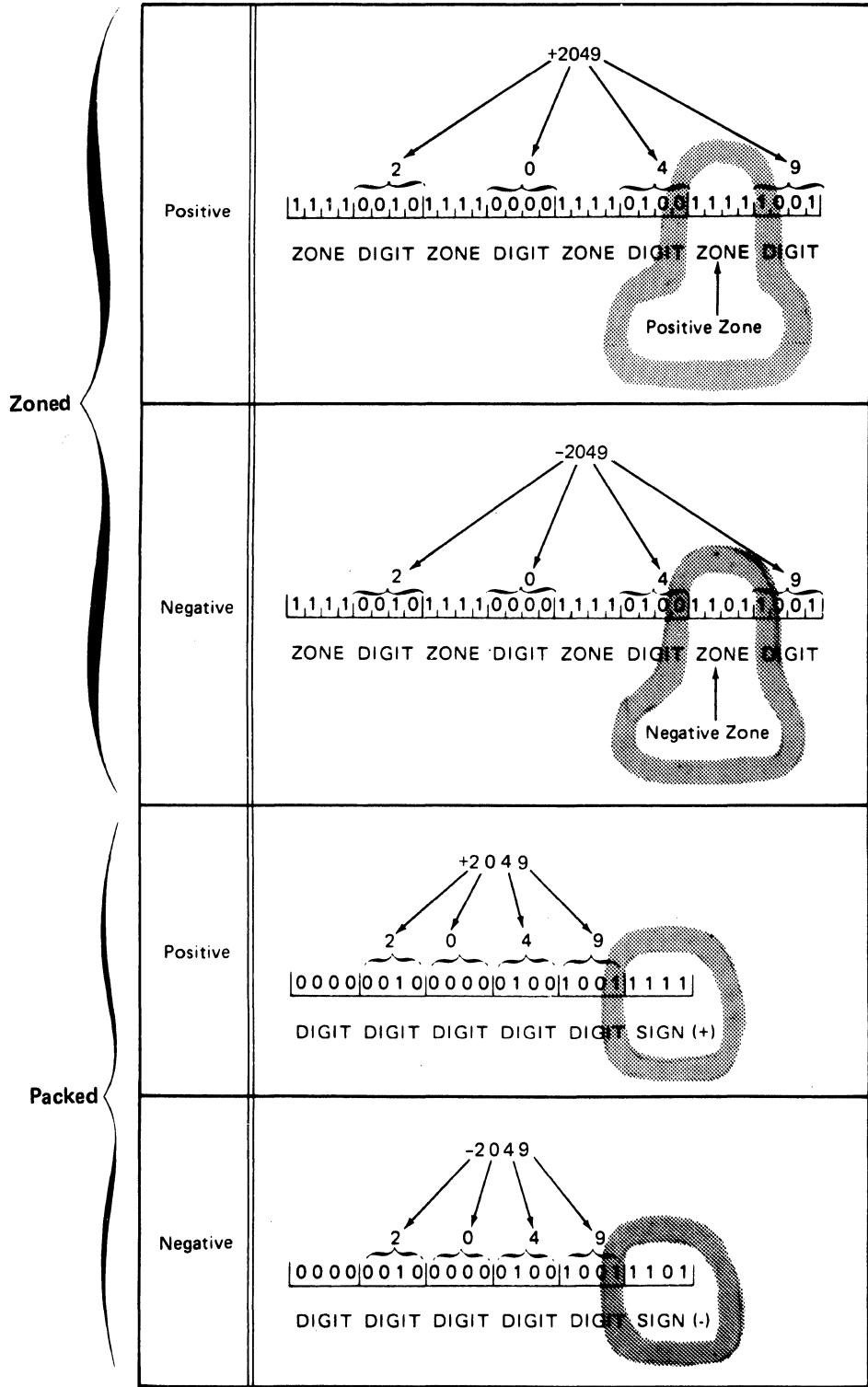
Zoned Digits		
Digit	Binary Form	
0	1111	0000
1	1111	0001
2	1111	0010
3	1111	0011
4	1111	0100
5	1111	0101
6	1111	0110
7	1111	0111
8	1111	1000
9	1111	1001

Zone Portion
Digit Portion

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

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 Number	Type (U/D)	Rel.	Factor 1		Factor 2 (Field, Constant, or Keyword)	
			From	To	From	To
0 1	I	C				
0 2						
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 on the entire control field (S in column 26 of the header line) 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. S is used only in System/34 ideographic sort jobs.

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.

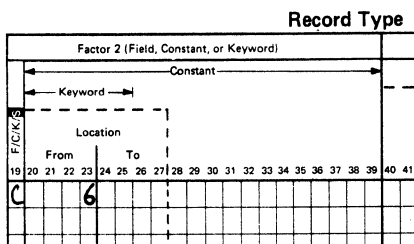
Coding Rules

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).

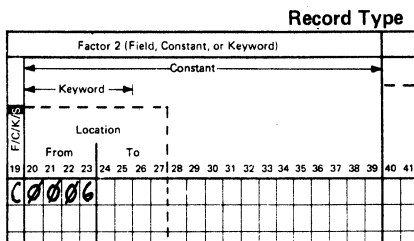
Columns 20-39 (Factor 2 Constant)

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 *bbb123* (with *b* 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.

Columns 20-39 (Factor 2 Keyword)

When factor 2 is a keyword (indicated by a K in column 19), 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
UDATE ^{2,3}	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

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

²If the UDATE 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.)

³If factor 2 is UDATE, record selection on and/or before or on and/or after a certain date (columns 17 and 18 contain LT, GT, LE, or GE) only functions with the international format (yymmdd). If the program date and the date contained in the input records are not in the international format, the keywords UYEAR, UMONTH, and UDAY, must be used to select the records.

Columns 40-74 (Comments)

Columns 40 through 74 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	Entries	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.
	E	Used only in System/34 ideographic sort jobs.
	I	Used only in System/34 ideographic sort jobs.
	J	Used only in System/34 ideographic sort jobs.
	R	Used only in System/34 ideographic sort jobs.
	S	Used only in System/34 ideographic sort jobs.
T	Used only in System/34 ideographic sort jobs.	
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.
	A (column 20 only)	Alternate collating sequence by field.
23-39	Blank	Reserved for system use.
40-74	Any characters	Comments.
<div style="display: flex; align-items: center; margin-bottom: 5px;"> <div style="width: 20px; height: 10px; background-color: #cccccc; border: 1px solid black; 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, black 2px, black 4px); border: 1px solid black; margin-right: 5px;"></div> Columns that must be filled in when forced control fields are being used. </div>		

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.)

Notes:

1. 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.
2. Sort does not check for the same type of control field across the record types. Sort generates the control field data as specified in the sort sequence specifications for each record type and performs a compare of the control fields to put them in sequence. If, for example, a packed field is defined as the first control field in one record type, and a character field is defined as the first control field in a second record type, the sorted sequence is unpredictable. This situation is not detected by the sort program.

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 unless equal control field ordering is specified. (For information on equal control field ordering, see *Column 12 (Equal Control Field Ordering)* in Chapter 2.)

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:

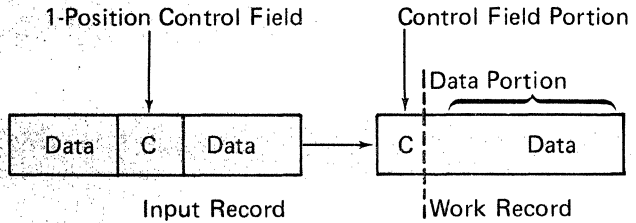
- Conditional
- Force-all
- Unconditional

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

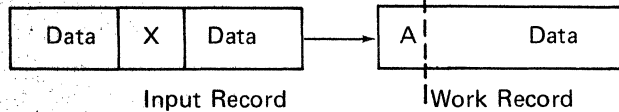
See *Column 17*, *Column 18*, and *Column 19* for information on how to fill out the field specifications when you are using forced control fields.

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.

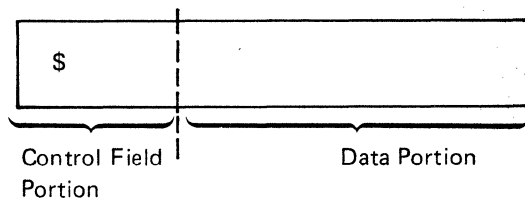


Force-all is a special type of conditional force. Force-all occurs only when a control field in an input record does *not* contain a particular entry. Suppose, for example, that you want to sort a file of records, each of which has a 1-position control field. If the character in the control field is *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.)

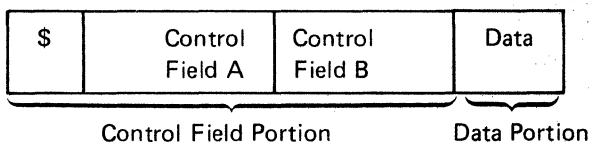
An unconditional force does not depend on entries in the input records. If, for example, you want to put a dollar sign (\$) in the first position of every output record, you would use an unconditional force. Your unconditional forced control field would tell the sort program to put a dollar sign in the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

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 (Portion of Input Record to Use in Building/Sorting Work Records)

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

The following entries are used only in System/34 ideographic sort jobs: E, I, J, R, S and T.

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

Include these field specifications to do the preceding sort.

International Business Machines Corporation

SEQUENCE SPECIFICATIONS

GX21 9089 AIM050
Printed in U.S.A.

Page 1 of 2 Program Identification 75 76 77 78 79 80

Line		Job		Card Match		Sequence (A/D/S)		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	U	M	P	U	S	P	Number (N)	Print Option	Output Option (X)	Output Record Length	Check-point Option (C)	Non-Verify Option (N)	Null Output (N)	Storage Size to Execute	Length Code (R/C)	Record Length	Job Description	
																						Stacker Select
01	H	SORT											X									

Record Type

Line		Factor 1		Rel.		Factor 2 (Field, Constant, or Keyword)		Comments						
Number	Type (I/O)	Continuation (A/D)	Stacker Select	Location		EQ	NE	LT	GT	LE	GE	F/C/K	Constant	Record Name
				From	To									
01	*													ALL RECORDS INCLUDED IN SORT

Field

Line		Type		Location		Forced		Overflow		Comments	
Number	Type	Type (I/O)	Type (I/O)	Location		Record Character	Substitute Character	Continuation	Alt. Seq. Field (A)	Field Length	Field Name
				From	To						
07	F	NC		1	2						THIS CONTROL FIELD IS DROPPED
08	F	OC		1	7						THIS CONTROL FIELD IS DROPPED
09	F	DC		1	2						CONTROL FIELD DESCRIBED AS DATA
10	F	DC		5	7						CONTROL FIELD DESCRIBED AS DATA

*Number of sheets per pad may vary slightly. Ideographic sort utility only.

Notice that the information occupies different positions in the input and output records. The record positions change when the control fields and data fields are moved from the input to the work and from the work to the output files. Since an opposite control field is specified, all of the control fields in the input record are dropped (columns 1 and 2 and 5 through 7). To save the control fields, specify them as data (see example).

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).

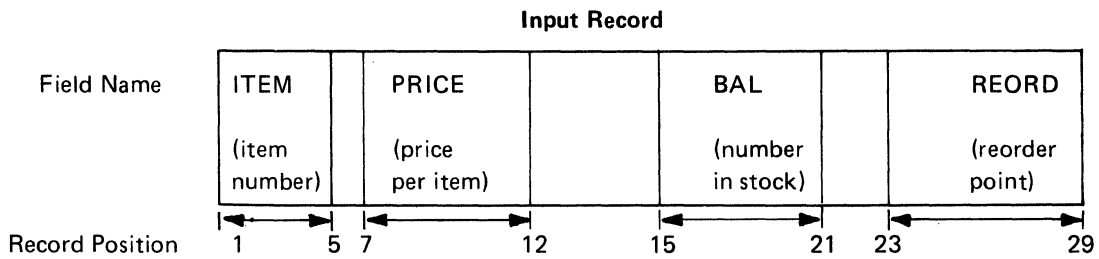
See *Column 7* and *Column 8* for detailed discussions of each entry.

Columns 9-16 (Field Location)

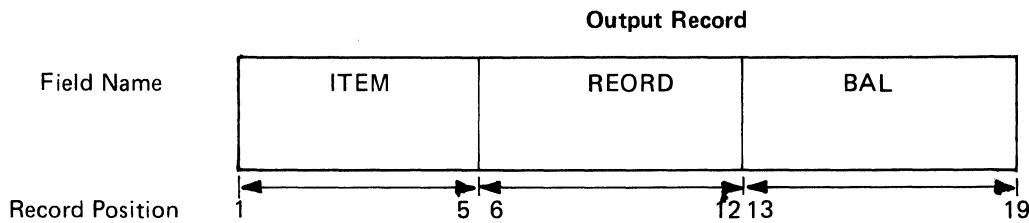
Columns 9 through 16 identify the input record positions that contain the record fields. Columns 9 through 12 (From) identify the starting position of a field; columns 13 through 16 (To) identify the position in which the field ends.

The order in which you describe the control fields in the field specifications determines their order in the sorted output records.

For example, suppose you have an input record that looks like this:



However, you want your sorted output record to look like this:



Assuming that you want to sort the records by item number, here is how you would fill out the field specifications:

Line Number	Type (NO/D/US?*) C/Z/D/P/U/M/E/T/P/H/S/T	Location		Record Character Substitute Character Continuation Alt. Seq. Field (A)	Overflow Field Length	Reserved	Field Name
		From	To				
07	F	NC	1	5			ITEM
08	F	DC	23	29			REORD
09	F	DC	15	21			BAL
10	F						
11	F						
12	F						
13	F						
14	F						

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 Number	Type	From	To	Location		Record Character	Substitute Character	Continuation	Alt. Seq. Field (A)	Overflow Field Length	Reserved
				From	To						
0 7	F			20	1						
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 a 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.

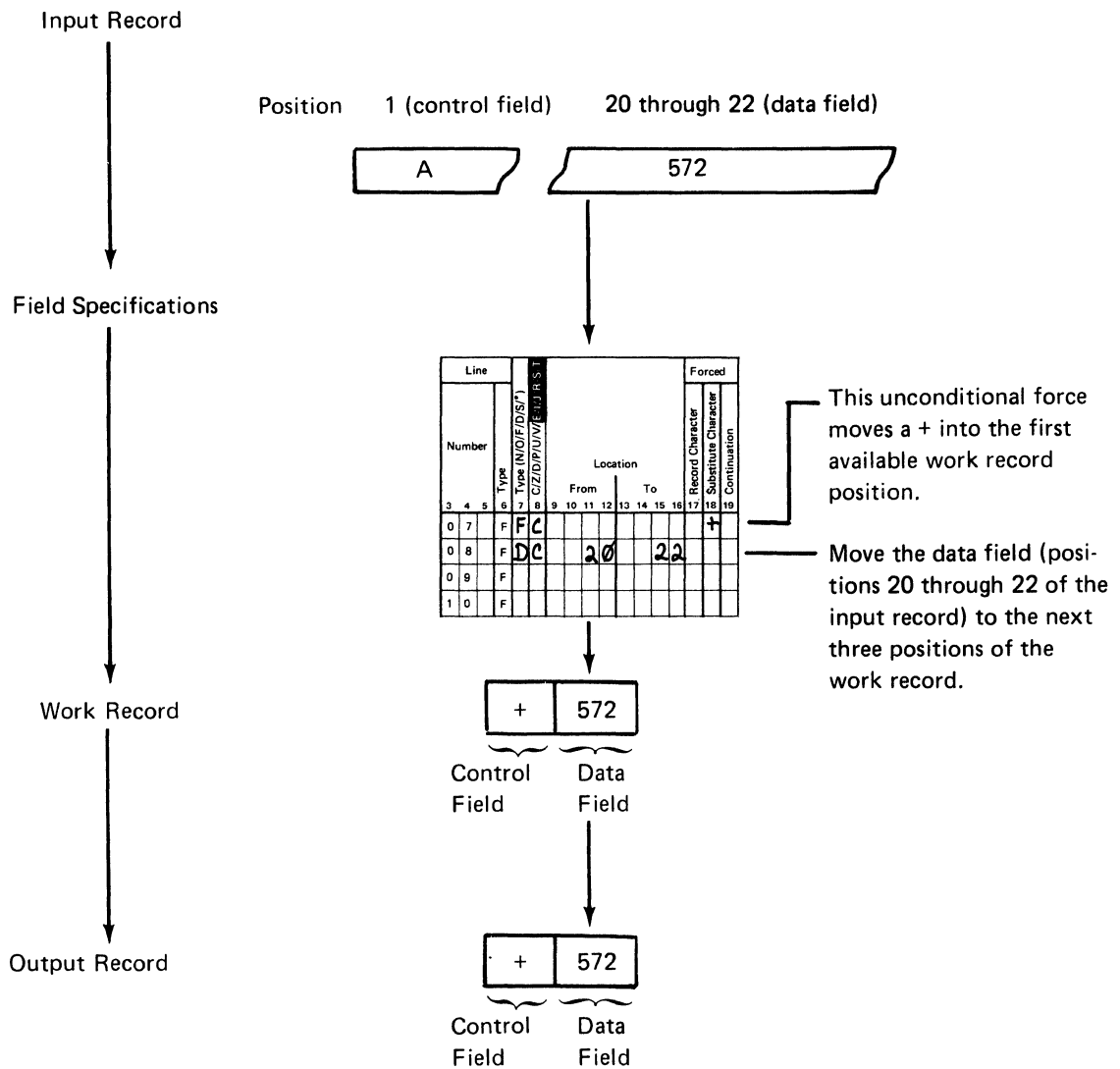
Examples of Using Forced Control Fields

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

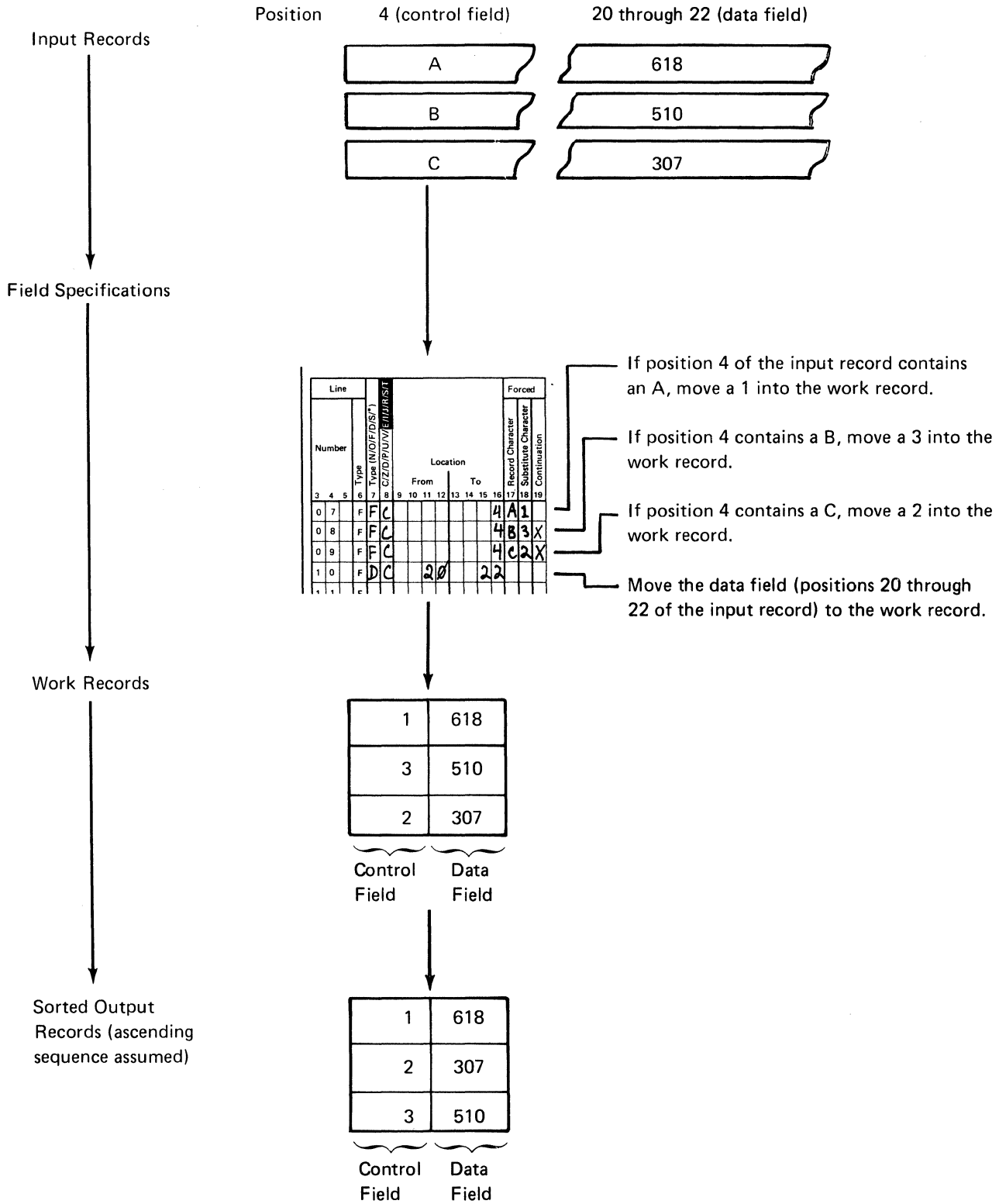
Unconditional Force

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.



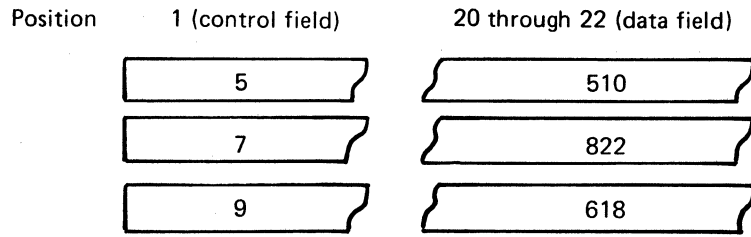
This example also shows how you can use column 19 to specify conditional replacements for more than one character. Because continuation lines are used, only one position in the work and output records is defined by the first three lines. If column 19 is left blank, each line would define a new position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

Input Records



Field Specifications

Line Number	Type	Type (N/O/F/D/S/*)	Location														Forced		
			From	To	Record Character	Substitute Character	Continuation												
07	F	N																	1
08	F	F																	197X
09	F	F																	179X
10	F	D								20					22				

Move the control field (in position 1 of the input record) to the work record, and use it to sort the records into the specified order (assume that ascending order is specified on the header line).

If position 1 of the input record contains a 9, replace it with a 7 in the work record.

If position 1 of the input record contains a 7, replace it with a 9 in the work record.

Move the data field (positions 20 through 22 of the input record) to the work record.

Work Records

5	510
9	822
7	618

Control Field Data Field

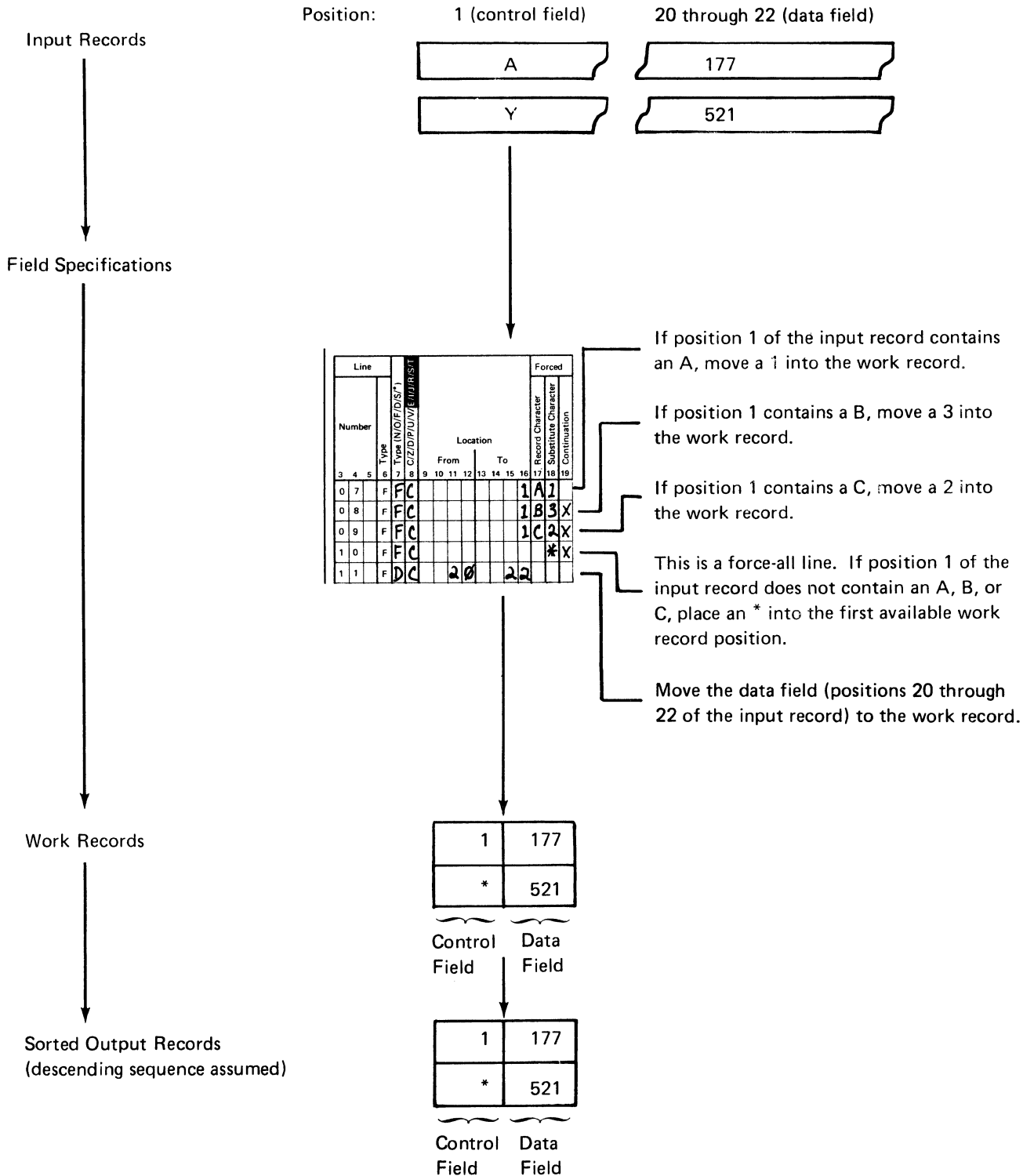
Sorted Output Records (ascending sequence assumed)

5	510
7	618
9	822

Control Field Data Field

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)

Use columns 20 through 22 with a summary tagalong sort (SORTRS) to eliminate the possibility of an overflow condition in a summary data field.

To eliminate the possibility of an overflow condition, increase the length of the field and place the entry for the new length in columns 20 through 22. The overflow field length entry should:

- Equal the length of the summary data field plus the anticipated overflow length
- Be right-adjusted to column 22
- Not exceed the maximum field length (For information about the maximum field length, see *Columns 9-16 (Field Location)* earlier in this chapter.)

Remember that the summary data in the output record is right-adjusted. Also, a maximum of 24 fields can be summarized for each record type in a sort job.

The following example shows records from an input file that a company uses to track orders. Each record in the input file contains the following data:

Input Record

Field Name	ITEM# (Item Number)	NUMORD (Number of Items Ordered)	PRICE (Cost per Item)	TOTAL (Number of Items Ordered X Cost per Item)
Record Position	1 ←→ 5	6 ←→ 9	10 ←→ 16	17 ←→ 21

Sample Records

Record Number	Item Number	Order Number	Price	Total
1	11111	10	50.00	500.00
2	11111	5	50.00	250.00
3	11111	12	50.00	600.00
4	22222	10	20.00	200.00
5	22222	35	20.00	700.00
.
.
.

The company tracks the daily total for each item sold. To obtain the daily amount for an item sold, the company programmer sorts on ITEM# (line 7 of the field specifications) and specifies TOTAL as a summary data field (line 8 of the field specifications).

See the following sequence specifications for coding information for this example.

If the data in the summary data field is in zoned decimal format (U in column 8 of line 8 in the example), and the programmer knows that the length of the output will exceed the length of the five-position summary data field by two positions, then columns 20 through 22 of the field specifications should contain a right-adjusted 7 (5+2=7). (For information about zoned decimal formats, see *Column 8* earlier in this chapter.)



SEQUENCE SPECIFICATIONS Header

Line	Job	Card Match	Stacker Select	Output Record Length	Check/print Option (C)	Non Verify Option (N)	Imagegraphic Sort (I)	Null Output (N)	Storage Size to Execute	CCP Disk Sort Only	Record Length	Job Description
3 4 5	Card: MATCH, SORT, SELECT, MERGE Disk: SORT, E, SORTA, SORTAS Tape: SORTS	Largest sum (in bytes) of Control Field Lengths for Any Record Type	S P P S P U M M U O O	Number (I/N) Alt. Call. Seq. (S/F) Print Option Output Option (X)								
0 0 0	SORTRS	5A		0							12	SUMMARY TAGALONG SORT

Record Type

Line	Type (IO)	Factor 1	Rel.	Factor 2 (Field, Constant, or Keyword)	Comments
3 4 5	Continuation (A/O) C/Z/D/P/U	Location From To	EQ NE LT GT LE GE	Constant Keyword Location From To	Record Name
0 1	IC		INEC		INCLUDE NONBLANKS IN COLUMN 1

Field

Line	Type (IN/OF/D/S/*)	Location	Forced	Overflow Field Length	Comments
3 4 5	C/Z/D/P/U/V/E/N/I/S/ST	From To	Record Character Substitute Character Continuation Alt. Seq. Field (A)	Reserved	Field Name
0 7	NC	1 5			SORT ON ITEM NUMBER
0 8	SU	17 21		7	SUMMARIZE; ALLOW FOR OVERFLOW

If the data is sorted according to the preceding sequence specifications and the overflow length is not specified, the summary data output records will look like this:

The TOTAL for ITEM# 11111 is \$1350.00; however, only \$350.00 is output.

11111135000
01

The TOTAL for ITEM# 22222 is \$900.00.

22222290000

If the overflow length is specified, the summary data output records will look like this:

The TOTAL for ITEM# 11111 is \$1350.00.

111110135000
Item Number Total Amount of Item Sold

The TOTAL for ITEM# 22222 is \$900.00.

222220090000
Item Number Total Amount of Item Sold

Because the length of the output field is less than the length of the actual output data, the data is truncated and 01 for item number 11111 is lost.

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

For example, if the summary data field in the preceding example was in packed decimal format (P in column 8 of line 8 of the field specifications) and the programmer knows that the length of the output will exceed the length of the packed summary field by two positions, then columns 20 through 22 of the field specifications should contain a 6 (eleven numbers plus the sign). The sequence specifications would be coded as follows:

Line		Job	Largest sum (in bytes) of Control Field Lengths for Any Record Type		Card Match		Stacker Select		Output Record Length		CCP Disk Sort Only		Job Description	
Number	Type		Sequence (A/D/S)	U	S	M	P	X	O	P	Number (I/N)	Print Option	Output Option (X)	Record Length
01	H	SORTRS	5A											SUMMARY TAGALONG SORT

Line	Number	Type (I/O)	Continuation (A/O/?)	Stacker Select	Factor 1	Rel.	Factor 2 (Field, Constant, or Keyword)	Comments
01	I	C			INEC		INCLUDE NON-BLANKS IN COLUMN 1	

Line	Number	Type (N/O/F/D/S/?)	Record Character	Substitute Character	Continuation	Alt. Seq. Field (N)	Overflow Field Length	Comments
07	F	NC	1	5				SORT ON ITEM NUMBER
08	F	SP	17	21		6		SUMMARIZE; ALLOW FOR OVERFLOW

*Number of sheets per pad may vary slightly. ■ Ideographic sort utility only.

Summary Overflow Indicator Field

Sometimes a summary data field may be too small to hold the sum of the corresponding data fields. If this is the case, you can specify a summary overflow indicator field. This summary overflow indicator field will contain the specified indicator. The indicator is placed in the output record of each field that produces an overflow. The summary overflow indicator field can be used alone or in conjunction with the entry in columns 20 through 22. See *Overflow Field Length* earlier in this chapter.

To specify a summary overflow indicator field you should specify:

- An S in column 7 and a V in column 8.
- A character in column 17 to which the overflow indicator field will be initialized if an overflow does occur in any summary data field. (If no entry is made in column 17, the initialization character defaults to an asterisk [*].)
- A character in column 18 to which the overflow indicator field will be initialized if an overflow does not occur in any summary data field. (If column 18 contains a blank, the overflow indicator field will be initialized to blank.)

Suppose the company programmer had the same input record and data previously described.

Input Record

Field Name	ITEM# (Item Number)	NUMORD (Number of Items Ordered)	PRICE (Cost per Item)	TOTAL (Number of Items Ordered X Cost per Item)
Record Position	1 ← 5	6 ← 9	10 ← 16	17 ← 21

Sample Records

Record Number	Item Number	Order Number	Price	Total
1	11111	10	50.00	500.00
2	11111	5	50.00	250.00
3	11111	12	50.00	600.00
4	22222	10	20.00	200.00
5	22222	35	20.00	700.00
.
.
.

Again, he uses ITEM# as the control field for the sort and specifies TOTAL as a summary data field; he also specifies a summary overflow indicator field. The following sequence specifications will sort the records on ITEM#, initialize the overflow field to a blank, and replace the blank with an ampersand (&) whenever a summary data field overflow occurs.



International Business Machines Corporation
SEQUENCE SPECIFICATIONS
Header

GX21-9089 UM050*
 Printed in U.S.A.

Page 1 2
 Program Identification 75 76 77 78 79 80

Line	Job	Card Match	Stacker Select	Print Option	Output Option (X)	Output Record Length	Check Point Option (C)	Non-Verify Option (N)	Ideographic Sort (I/S)	Null Output (N)	Storage Size to Execute	Length Code (R/C)	Record Length	Job Description
3	H	SORTRS	5A	0	11									SUMMARY TAGALONG SORT

Record Type

Line	Factor 1	Rel.	Factor 2 (Field, Constant, or Keyword)	Comments
0 1	I	C	INEC	INCLUDE NON-BLANKS IN COLUMN 1

Field

Line	Record Character	Substitute Character	Continuation	Alt. Seq. Field (A)	Overflow Field Length	Reserved	Field Name
0 7	F	NC	1	5			SORT ON ITEM NUMBER
0 8	F	SU	17	21			SUMMARIZE ITEM PRICE
0 9	F	SV		&			REPLACE BLANK WITH & ON OVERFLOW

*Number of sheets per pad may vary slightly.

■ Ideographic sort utility only.

Once the data is sorted, the output records will look like the following:

In this record, overflow occurred.

1 1 1 1 1 3 5 0 0 0 &

Item Number: 1 1 1 1 1
 Total Amount of Item Sold: 3 5 0 0 0
 Overflow occurred. Therefore, the total amount is incorrect.

In this record, overflow did not occur.

2 2 2 2 2 9 0 0 0 0 b

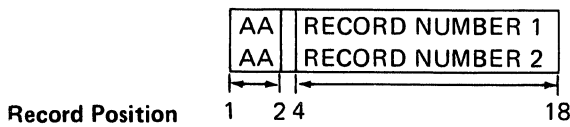
Item Number: 2 2 2 2 2
 Total Amount of Item Sold: 9 0 0 0 0
 Overflow did not occur. Therefore, the total amount (\$900.00) is correct.

Column 20 (Alternate Collating Sequence by Field)

Column 20 must contain an A for any normal or opposite control field that is to be altered by the alternate collating sequence when column 26 of the header line contains an F.

If you specify an alternate collating sequence for a particular field, that field will be altered according to the alternate collating sequence whenever it is used again as a control field for that record type. For example, suppose you have two input records that look like this:

Input Records



If the alternate collating sequence alters A's to equal B's, the following field specifications:

Line Number	Type	Type (M/O/F/D/S*)	C/Z/D/P/O/V	Location		Forced			Overflow Field Length
				From	To	Record Character	Substitute Character	Continuation	
0 7	F	N	C	1	2				
0 8	F	N	C	1	2			A	
0 9	F	N	C	1	2				
1 0	F	D	C	1	18				
1 1	F								
1 2	F								
1 3	F								
1 4	F								

Both of these control fields are altered according to the alternate collating sequence.

Columns 23-39 (Reserved)

These columns are not used. Leave them blank.

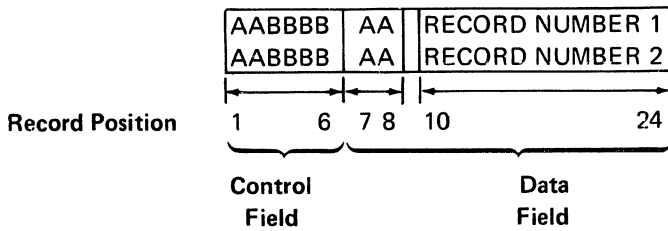
Columns 40-74 (Comments)

Columns 40 through 74 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.

Cause the following work records to be built:

Work Records



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 that are not delete-capable are set to blanks before the file is created. If the input file is a delete-capable file, all the input records are read by the sort program; however, the deleted records are not included in the sort.

Notes:

1. You cannot sort an offline multivolume file using the SORT command. You must supply your own sort procedure for the job. For an explanation of how to specify a sort procedure for a job, see *Running Sort By Using Your Own Sort Procedures* later in this chapter.
2. Unformatted sequential files created by FORTRAN do not have fixed length records. Therefore, they *should* not be used as input to the sort program.
3. When you supply OCL statements to load and run the #GSORT program, the sort program frees each input file as soon as all the records in the file have been read (a diskette remains allocated until the #KASRT program terminates). This allows other jobs to access the permanent file at the earliest opportunity. Therefore, the temporary input files that will be sorted should not be defined as RETAIN-S for scratch status.

If the file is defined as a scratch file, an error occurs that causes sort to terminate unsuccessfully, and one of the RETAIN-S input files has been read and freed by sort, then the file is no longer available and you cannot run the sort job again. To be able to run the sort job again, define the temporary input files as RETAIN-T for temporary status; then delete the files after the sort runs successfully.

Multiple Input Files

From one to eight files can be used as input to a sort job. The file names must be: INPUT *or* INPUT1 (but not both), INPUT2, INPUT3, INPUT4, INPUT5, INPUT6, INPUT7, and INPUT8.

The input files are processed INPUT (or INPUT1) serially to INPUT8 regardless of the order of the OCL FILE statements. The lowest numbered input file is processed fully first, then the next numbered input file is processed in its entirety, then the next until all the files have been individually processed in sequence. For a particular sort job, the input files do not have to be numbered serially; that is, the file names may be INPUT3, INPUT5, and INPUT8, or a particular job could have just INPUT8 as the input file.

If you have two or more input files, equal control field ordering (E in column 12 of the header specification) allows you to perform a merge. For information on how to perform a merge, see *Column 12 (Equal Control Field Ordering)* in Chapter 2 and *Sample Job 9* in Chapter 7.

Remember the following restrictions when using multiple input files:

- The SORT command cannot be used. Therefore, you must supply your own sort procedure for your job. For an explanation of how to specify a sort procedure for a job, see *Running Sort By Using Your Own Sort Procedures* later in this chapter.
- An addrout sort (SORTA in columns 7 through 12 of the header specification) is not allowed.
- The record lengths for all input files must be the same.

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. However, if the input file is an offline multivolume file, you must specify the work file area on disk. See *FILE Statement (for Disk Files)* in the *System Support Reference Manual* for more information about allocating the work file.

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.

The sort work file cannot be an offline multivolume file or an extendable disk file (EDF).

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 to 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.

The output file should not be a delete-capable file. If the output file is a delete-capable file and you attempt to write a deleted record (FF in the first byte of the record) to the output file, you will be required to cancel the sort job.

The output file cannot be an offline multivolume file if you use the SORT command. You must supply your own sort procedure for the job. For an explanation of how to specify a sort procedure for a job, see *Running Sort By Using Your Own Sort Procedures* later in this chapter.

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.

Offline Multivolume Files

An offline multivolume file is a sequential file that is processed as though it resides completely on disk, but it is, in fact, stored on one or more diskettes. Each diskette contains a portion of the offline multivolume file, called a file segment. These segments are processed on disk one at a time until the entire file is processed. Either the sort input file or the sort output file can be an offline multivolume file.

When the System/34 sorts offline multivolume files, you must supply your own sort procedure for the job. See *Running Your Own Sort Procedures* later in this chapter for an explanation of how to specify a sort procedure for a job. You cannot sort an offline multivolume file using the SORT command.

Also, you must specify two FILE statements with the same NAME parameter. The size specified on the FILE statement for an input disk file must be the same as the size specified when the file was created on the diskette.

Assume you want to sort an offline multivolume file called PAYMSTR, which contains 1000 records and has a record length of 540. The FILE statements in your sort procedure could look like this:

```
// LOAD #GSORT

// FILE NAME-INPUT,LABEL-PAYMSTR,UNIT-F1,
BLOCKS-96

// FILE NAME-INPUT,LABEL-PAYMSTR,UNIT-I1,
PACK-IBMIBM,LOCATION-M1.01,RETAIN-999

// FILE NAME-OUTPUT,LABEL-PAYOUT,
RECORDS-3000,RETAIN-J

// FILE NAME-OUTPUT,LABEL-PAYOUT,
RECORDS-1000,RETAIN-T,DISP-NEW
●
●
●

// RUN

// END
```

As shown in the preceding sort procedure, you must also specify a FILE statement for the sort work file on disk. The sort work file cannot be an offline multivolume file. See *Appendix A* for information on how to calculate file sizes.

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 ¹	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. ² (See Appendix A for information on how to calculate work file size.)
Sort output ³	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. ⁴	Output file can overlay the input file; that is, the input file can also be the output file. ⁵
Sort job running time	Usually shorter than SORTR or SORTRS.	Usually longer than SORTA.

¹You must specify the work file area on disk with a FILE statement if the input file is an offline multivolume file.

²If 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.)

³The output file should not be a delete-capable file. If the output file is a delete-capable file and you attempt to write a deleted record (FF in the first byte of the record) to the output file, you will be required to cancel the sort job.

⁴If your input file is overlaid, you must have a backup copy of your input file that you can access using your addrout sort output.

⁵The 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:

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

SORT PROCEDURE

Rearrange, drop and reformat records in a file.

```
Label Of File To Be Sorted ..... _  
Name Of Source Member Containing Sort Specifications .....  
Label Of The Output File .....  
Number Of Records To Be Placed In The Output File .....  
Name Of User Library Containing The Source Member ..... #LIBRARY  
Place Job On Input Job Queue (Y/N) ..... N
```

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 can be used as input to the sort program. Therefore, if more than one input file must be supplied to the sort program, the SORT command cannot be used.
- Only one file on the disk should have the input file label. If the SORT command is used and more than one file has 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.
- The input file cannot be an offline multivolume file or a job 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.
- The output file cannot be an offline multivolume file, an extendable disk file (EDF), or a delete-capable file.

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.

Note: If the SORT command is used to execute a sort job step from a procedure and the sort job step is being released (// ATTR RELEASE=YES), a // END statement must follow the SORT command statement in the procedure.

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 or INPUT1, INPUT2, . . . INPUT8. Use the LABEL keyword to give the sort program the label of your file if it is not INPUT or INPUT1, INPUT2, . . . INPUT8. For example:

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

where MASTER is the label of your input file.

Note: The input file can be an offline multivolume 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. If the input file is an offline multivolume file, you must also supply a FILE statement for the sort work file.

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- } { T }  
 { P }
```

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.

Notes:

1. The output file can be an offline multivolume file.
2. 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 offline multivolume files under the heading *Offline Multivolume Files* earlier in this manual and in the *Concepts and Design Guide*.)

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: Do not put the // END statement in the procedure unless the procedure contains the ATTR OCL statement releasing the requesting display station from the sort job step. See the *System Support Reference Manual* for more information about the ATTR OCL statement and the END control 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
```

Notes:

1. 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:
 - a. OCL statements following the last sort sequence specification statement will be diagnosed as sort sequence specification statements.
 - b. 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 executing procedure is finished.
2. For an offline multivolume file two FILE statements are required. (For information on offline multivolume FILE statements, refer to the *System Support Reference Manual*.) Remember that a sort work file cannot be an offline multivolume file or an extendable disk file (EDF).

Substitution Expressions

Substitution expressions allow you to replace information in the statements that are generated when a procedure is run. Examples of information that can be substituted are:

- Positional parameters on the procedure command that called the procedure
- Specified bytes in the display station local data area

Substitution expressions always begin and end with a question mark. A substitution expression begins any time a question mark is immediately followed by a number or by one of the following characters: C,D,F,L,M,R,S,U, or W. A question mark followed by any other character is not treated as a substitution expression. See the *System Support Reference* manual for detailed information about substitution expressions, substitution formats, format descriptions, and examples of the formats.

The following is an example using substitution expressions.

Example: Substitution from a Positional Parameter

Assume you have a BILLING procedure that is run twice a month:

- The procedure is run for the first time on the 15th day of the month, when customers whose names start with the characters A through M are billed.
- The procedure is run for the second time on the last day of the month, when customers whose names start with the characters N through Z are billed

All the records in a file named BILLING are sorted to group all customers to whom a bill will be sent. Assume the BILLING procedure has two parameters:

- The first initial of the first customers to be billed
- The first initial of the last customers to be billed

You would code the sort sequence specifications like this:

International Business Machines Corporation
SEQUENCE SPECIFICATIONS
Header

GX21-9089 'UM050*
Printed in U.S.A.

Page 1 2 Program Identification 75 76 77 78 79 80

Line Number	Job	Card Match	Stacker Select	Output Record Length	Job Description
000	H SORTA	20A		0	GENERATE ADDRROUT SORT

Line Number	Factor 1	Rel.	Factor 2 (Field, Constant, or Keyword)	Comments
01	I C 1	16	EC ? 1 ?	
02	I AC 1	1	LEC ? 2 ?	

Line Number	Field	Comments
07	F NC 1 20	SORT ON CUSTOMER NAME

*Number of sheets per pad may vary slightly. ■ Ideographic sort utility only.

After you code the sequence specifications, enter the following BILLING procedure into the system:

```
// LOAD #GSORT

// FILE NAME-INPUT,LABEL-BILLING

// FILE NAME-OUTPUT,LABEL-ADDRROUT,RETAIN-T,RECORDS-1000

// RUN

      HSORTA   20A       0           GENERATE ADDRROUT SORT

1 I  C      1  1GEC?1?

2 IAC      1  1LEC??

      FNC      1  20           SORT ON CUSTOMER NAME

// END
```

After you enter the sort procedure into the system, enter BILLING A,M to run the sort program for the 15th day of the month.

When the sort program runs:

- 1** This specification states that the sort program should include all customers whose first initial is greater than or equal to parameter 1. ?1? is replaced with the letter A.
- 2** This sequence specification tells the sort program to include the names of all customers whose first initial is less than or equal to parameter 2. ?? is replaced with the letter M.

When you use substitution expressions, comments should not be put on the same line with the substitution expressions. If you do put comments on the same line, data after the substitution may be shifted left or right depending on the number of characters replaced.

Note: Debug your sequence specifications carefully because the length of the positional parameter entry and the number of trailing blanks in the local data area can cause the sequence of the entries of the sequence specifications to change.

See the *System Support Reference Manual* for detailed information on substitution expressions.

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
```

Notes:

1. The END statement indicates to the sort program that the last sort sequence specification statement has been entered.
2. For an offline multivolume file two FILE statements are required. (For information on offline multivolume FILE statements, refer to *Offline Multivolume Files* earlier in this chapter and to the *Concepts and Design Guide*.) Remember that a sort work file cannot be an offline multivolume file or an extendable disk file (EDF).

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

- Number of input files
- 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
- Use of the loadable sort interface

The following text describes how each factor affects the time required to run a sort job. Sample timings are provided at the end of this chapter.

NUMBER OF INPUT FILES

The more input files you use in a sort job, the longer the sort job takes to run.

NUMBER OF RECORDS TO BE SORTED

The more records you want to sort, the longer the sort job takes to run. Therefore, for the most efficient sort job, 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).

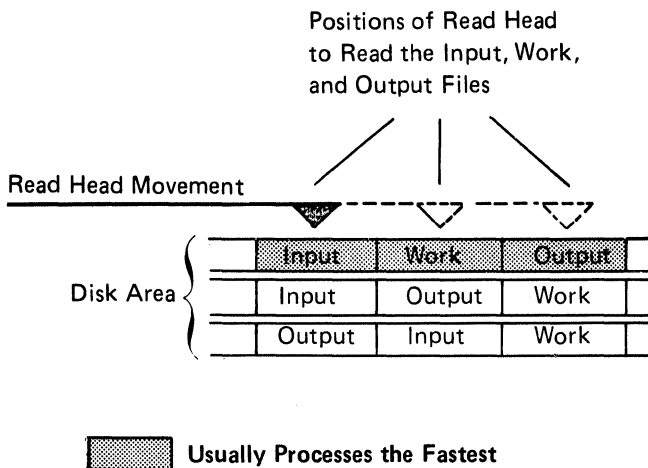
DISK LOCATION OF FILES

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:

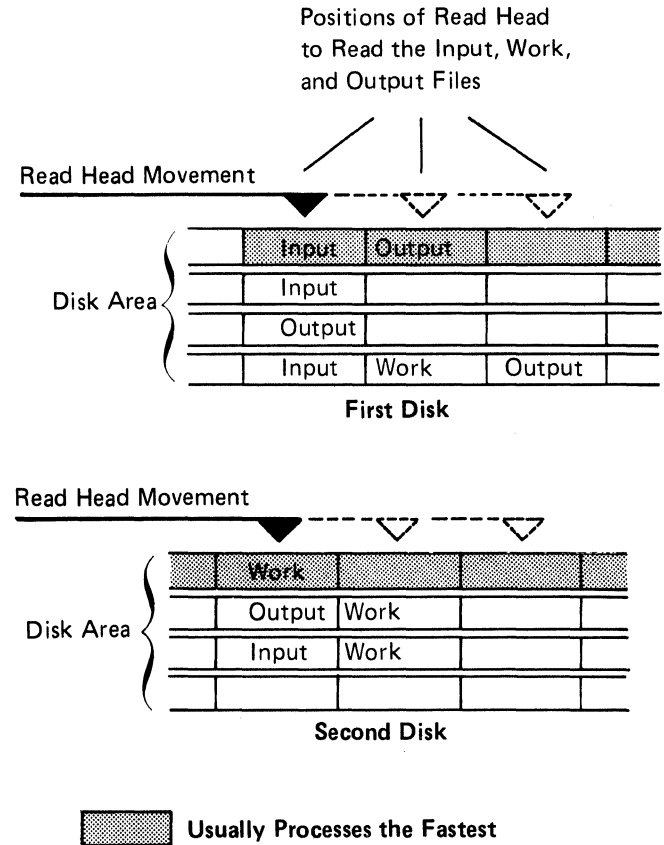
Methods of File Placement On a Single Disk System



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



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. See the *File Statement (for Disk Files)* in the *System Support Reference Manual* for more information about disk file placement.

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. The maximum region size for the sort program is 64K bytes. For information on how to increase the region size, see the *System Support Reference Manual*.

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.

LOADABLE SORT INTERFACE

The use of the loadable sort interface increases the time needed to run a sort job. For improved performance when using the loadable sort interface, increase the region size whenever you have enough main storage available to do so.

SAMPLE TIMINGS

In this section sample timings for the sort program are provided for your planning purposes.

The following paragraphs contain (1) information on the sequence specifications that were used to generate the sample timings and (2) the characteristics of the sort jobs that were used to generate the sample timings. If you deviate from any of these conditions, your sort run times may be longer.

The sequence specifications, which were read from a source member, contained the following key entries:

- Header*
- All of the jobs were tagalong jobs (SORTR in columns 7 through 12).
 - Alternate collating sequence was not specified (column 26 was blank).
 - Only displayed messages were issued (print option 3 in column 27).
 - Control fields were not dropped from the output records after the records were sorted (column 28 was blank).
 - The data written to the work file was verified (column 34 was either blank or contained a character other than an N).

Record Type Include and omit sets were not used.

Field Two field lines were specified.

The sort jobs had the following characteristics:

- The jobs were run on a *dedicated* system.
- Automatic work file allocation was used.
- Offline multivolume files were not specified.
- All input records were contained in a single input file.
- The jobs were logged to the history file (for information on the history file, see the *System Support Reference Manual*).

All timings in the following chart reflect the total sort run time (the time it takes for sort to get control plus generation time plus execution time).

Sample Timings (for sorting on a control field of 10 contiguous bytes)				
Number of Records	Region Size (in K)	Disk Capacity (in megabytes)	Time (in minutes and seconds)	
			Record Length for Input, Work, and Output Records (in bytes)	
			100	200
1000	14	27.1	:28	:40
		63.9	:27	:45
	24	27.1	:21	:32
		63.9	:19	:29
	48	27.1	:20	:24
		63.9	:16	:20
5000	14	27.1	2:02	3:34
		63.9	2:08	4
	24	27.1	1:19	2:24
		63.9	1:21	2:20
	48	27.1	1:06	1:37
		63.9	:58	1:25
10 000	14	27.1	3:58	8:47
		63.9	4:22	8:58
	24	27.1	2:47	4:55
		63.9	2:44	4:51
	48	27.1	2:04	3:33
		63.9	1:53	3:01
20 000	14	27.1	9:23	17:48
		63.9	9:34	18:35
	24	27.1	5:53	11:09
		63.9	5:52	10:28
	48	27.1	4:28	7:11
		63.9	3:59	6:11

Chapter 7. Sample Sort Jobs

Each of the nine 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 and job 8 use the following files and input records. (*Sample Job 7* and *Sample Job 9* use other files and input records, which are explained in the introduction to those jobs.)

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 Receipt records Adjustment records
		Shipments of items to customers Purchases of more items from suppliers (vendors) 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	Contents ¹	
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.

International Business Machines Corporation
SEQUENCE SPECIFICATIONS

GX21-9089 /UM050*
Printed in U.S.A.

Page 01 Program Identification 75 76 77 78 79 80

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	M	P	P	S	O	
00	H	SORTR	10A								

Line		Factor 1	Rel.	Factor 2 (Field, Constant, or Keyword)		Comments	
Number	Type (I/O)	Location	EQ NE LT GT LE GE	Constant	Keyword	Record Name	
01	I	69 74		40 45		AVAIL FLD LESS OR EQ TO REORD FLD	

Line		Location		Forced	Overflow	Comments	
Number	Type (N/O/F/D/S/*)	From	To	Record Character	Field Length	Reserved	
07	F	7	10			Total-10 Characters	
08	F	11	6			CLASS CLASS OF ITEM--JEWELRY, ETC	
09	F	11	24			ITEM STOCK ITEM NUMBER	
10	F	28	29			DESC DESCRIPTION OF ITEM	
11	F	30	34			UNIT UNIT OF MEASURE	
12	F	46	50			COST COST PER UNIT	
13	F	69	74			QTY REORD QUANTITY	
14	F	81	86			AVAIL UNITS AVAILABLE--BAL+ORDER	
						PDATE PURCHASE ORDER DATE	

■ Ideographic sort utility only.

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.

International Business Machines Corporation
SEQUENCE SPECIFICATIONS

GX21-9089 /UM050*
Printed in U.S.A.

Page 02 Program Identification 75 76 77 78 79 80

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	M	P	P	S	O	
07	F	DC	87 91								
08	F	DC	92 96								
09	F	DC	75 80								

Line		Location		Forced	Overflow	Comments	
Number	Type (N/O/F/D/S/*)	From	To	Record Character	Field Length	Reserved	
07	F	87	91			VEND1 FIRST VENDOR	
08	F	92	96			VEND2 SECOND VENDOR	
09	F	75	80			PORDER PURCHASE ORDER NUMBER	

■ Ideographic sort utility only.

Header Specifications

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 40-74

Columns 40 through 74 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 40-74

Columns 40 through 74 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	U	S	P	M	U	P	C	Record Length	
000	H	SORTA	10A										

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

Line		Field		Comments	
Number	Type (I/O)	Location	Record Character Substitutes Character Continuation	Reserved	Field Name
07	F	NC 7 10			CLASS
08	F	ND 1 6		Total 10 characters	CLASS OF ITEM - JEWELRY, ETC ITEM STOCK ITEM NUMBER
09	F				
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 addrout 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		Number (N)		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	M	P	P	U	O	P	Record Length	Record Length		
3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
000	H	SORTR										19			

Line		Factor 1		Factor 2 (Field, Constant, or Keyword)		Comments	
Number	Type (OO)	Location	Rel.	Location	Record Name		
3	4	5	6	7	8	9	10
01	I	D	7	10	LE	0130	INCLUDE CLASS FLDS 0126-0130
02	I	A	D	7	10	GE	0126

Line		Location		Forced		Overflow		Comments	
Number	Type (NO/E/S/F)	From	To	Record Character	Substitute Character	Continuation	Field Length	Reserved	Field Name
3	4	5	6	7	8	9	10	11	12
07	F	N	C	7	10			Total - 7 Characters	CLASS
08	F	O	D	97	99				TRANS
09	F	D	C	1	6				ITEM
10	F	D	C	97	99			Total - 19 Characters	TRANS
095	F	D	C	7	10				CLASS
12	F	D	C	106	111				ISSUES

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.

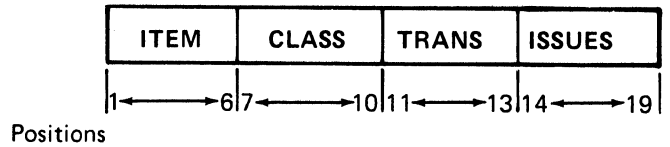
Column 7

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:



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.

		International Business Machines Corporation		GX21-9089 /UM050*		Printed in U.S.A.											
		SEQUENCE SPECIFICATIONS												Page 01		Program Identification	
		Header															
Line	Job	Card Match		Stacker Select		Output Record Length		CCP/Disk Sort Only		Record Length		Job Description					
Number	Type	Card: MATCH, SORT, SELECT, MERGE	Largest sum (in bytes) of Control Field Lengths for Any Record Type	S	M	P	U	Number (N)	Print Option	Output Option (X)	Check-point Option (C)	Non-Verify Option (N)	Null Output (N)	Storage Size to Execute	Length Code (R/C)	Record Length	Job Description
0 0	H	SORTA	13A														
Record Type																	
Line	Factor 1	Factor 2 (Field, Constant, or Keyword)		Comments													
Number	Type I/O	Location	Rel.	EQ	NE	LT	GT	LE	GE	Field Name	Record Name						
0 1	T	C	1EQCR								RECEIPT						
Field																	
Line	Record Character	Location		Forced	Overflow Field Length	Comments											
Number	Type (N/C/F/D/S/A)	From	To	Record Character	Substitute Character	Continuation	Alt. Seq. Field (A)										
0 7	F	C		1				Total—7 Characters	FORCED CONTROL FIELD								
0 8	F	ND	6	11				Total—20 Characters	ITEM NUMBER OF ITEM ORDERED								
0 9	F	DC	2	5			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										

■ Ideographic sort utility only.



International Business Machines Corporation
SEQUENCE SPECIFICATIONS

GX21-9089 /UM050*
Printed in U.S.A.

Page 1 2
Program Identification 75 76 77 78 79 80

Line	Job	SI	Card Match	3/F	X	11(O)	11(N)	CCP/Disk Sort Only
Header								
Line	Job	SI	Card Match	3/F	X	11(O)	11(N)	CCP/Disk Sort Only
Record Type								
Line	Type (I/O)	Continuation (A/O)*	Sticker Select	Factor 1	Rel.	Factor 2 (Field, Constant, or Keyword)	Comments	
Number				Location		Location	Record Name	
3				From To		From To		
0 1	I	C			EQ		ISSUE	
0 2					NE			
					LT			
					GT			
					LE			
					GE			

Line	Type	Record Character	Substitute Character	Continuation	Alt. Seq. Field (A)	Overflow Field Length	Comments
Number							Field Name
3							
0 7	F	FC				2	FORCED CONTROL FIELD
0 8	F	ND					ITEM STOCK ITEM NUMBER
0 9	F	ND	28	33			DATE DATE OF ORDER
1 0	F	DC	8	12			QTY NUMBER OF UNITS ORDERED
1 1	F	DC	13	19			CUST CUSTOMER NUMBER
1 2	F	DC	20	27			INV INVOICE NUMBER
1 3	F	DC		34			DIS DISCOUNT INFORMATION
1 4	F	DC	35	39			PRICE PRICE OF ITEM PER ITEM
Total-13 Characters							
Total-26 Characters							



International Business Machines Corporation
SEQUENCE SPECIFICATIONS

GX21-9089 /UM050*
Printed in U.S.A.

Page 1 2
Program Identification 75 76 77 78 79 80

Line	Job	SI	Card Match	3/F	X	11(O)	11(N)	CCP/Disk Sort Only
Header								
Line	Job	SI	Card Match	3/F	X	11(O)	11(N)	CCP/Disk Sort Only
Record Type								
Line	Type (I/O)	Continuation (A/O)*	Sticker Select	Factor 1	Rel.	Factor 2 (Field, Constant, or Keyword)	Comments	
Number				Location		Location	Record Name	
3				From To		From To		
0 1	I	C			EQ		ADJUST	
0 4					NE			
0 5					LT			
0 6					GT			
					LE			
					GE			

Line	Type	Record Character	Substitute Character	Continuation	Alt. Seq. Field (A)	Overflow Field Length	Comments
Number							Field Name
3							
0 7	F	FC				3	FORCED CONTROL FIELD
0 8	F	ND	2	7			ITEM STOCK ITEM NUMBER
0 9	F	DC		8			CODE ADJUSTMENT CODE
1 0	F	DC	9	13			QTY NUMBER OF UNITS
1 1	F	DC	14	17			OK AUTHORIZATION
1 2	F	DC	18	23			DATE DATE OF ADJUSTMENT
1 3	F						
1 4	F						
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:

- Columns 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.

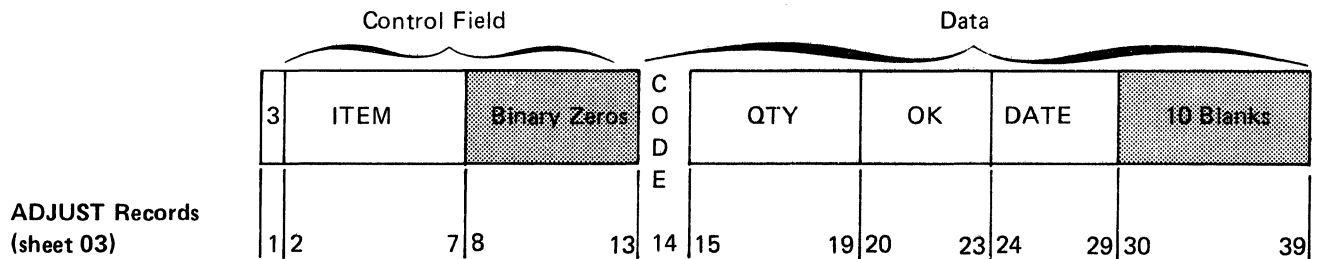
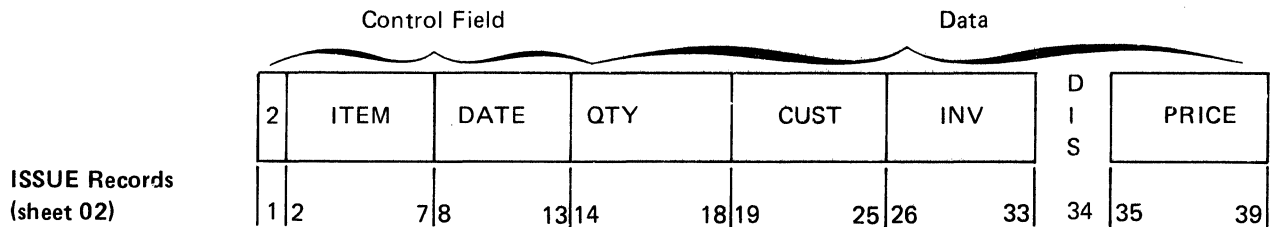
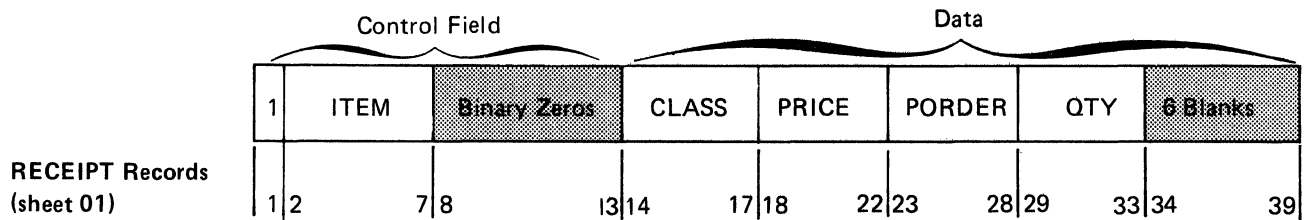
Column 6

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		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	M	P	U	P	U	Output Record Length	Record Length		
3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	0	H	SORTRS	6	A					6			

Line	Number	Type (I/O)	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	73	74
0	1	I	C		EQ	ADJUST																																																																	
0	2																																																																						
0	3																																																																						
0	4																																																																						
0	5																																																																						
0	6																																																																						

Line	Number	Type (N/D/F/D/S/)	Record Character	Substitute Character	Continuation	Alt. Seq. Field (A)	Overflow Field Length	Reserved	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	73	74
0	7	F	M	C			2	7	ITEM																																																														
0	8	F																																																																					
0	9	F																																																																					
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

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		Sequence (A/D/S)		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	O	P	Number (I/N)	Print Option	Output Record Length	Check-point Option (C)	Non-Verify Option (N)	Idiographic Sort (I/Z)	Storage Size to Exceed	Length Code (R/C)	Record Length	
00	H	SORTRS		7	A							23							

Line	Number	Type (I/O)	Continuation (A/O/P)	Factor 1	Rel.	Factor 2 (Field, Constant, or Keyword)	Comments
01	I	C		1	EQ	ISSUE	ISSUE

Line	Number	Type	Location	Forced	Overflow Field Length	Comments
07	F	NC	13 19			Total-7 Characters
08	F	DV				CUST
09	F	DV				#
10	F	SD	40 42			Total-16 Characters
11	F	DV				COM1
12	F	DV				#
13	F	SU	43 50			Total-16 Characters
14	F	SV		X		COST

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

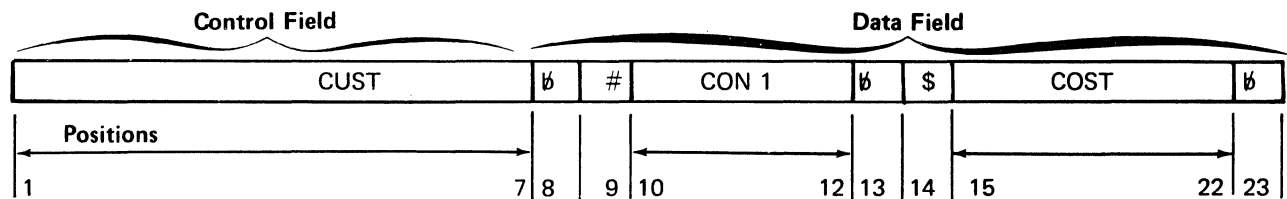
The X in line 14 defines the field as an overflow indicator field.

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:

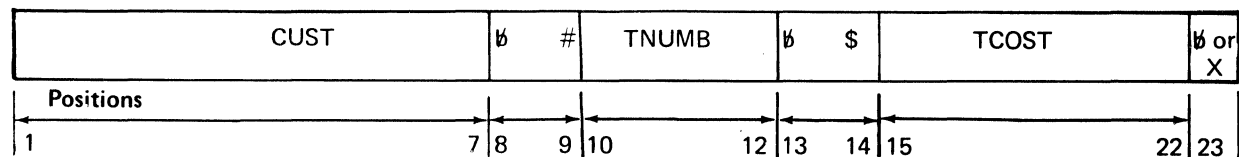
- 08 ␣ (blank)
- 09 #
- 11 ␣ (blank)
- 12 \$
- 14 ␣ (blank)

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: b #
10-12	TNUMB	Total number of issues to customer
13-14		Constants: b \$
15-22	TCOST	Total cost of all issues to customer
23	OFLOW	Overflow indicator: b – 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.



International Business Machines Corporation
SEQUENCE SPECIFICATIONS
Header

Line	Job	Card Match	Stacker Select	Output Record Length	CCP/Disk Sort Only	Record Length	Job Description
3	000H	SORTR	SUMPO	23			

Record Type

Line	Type (I/O)	Factor 1	Rel.	Factor 2 (Field, Constant, or Keyword)	Comments
01	O		R	INVALID SUMS	
02	F		R	INCLUDE-ALL POSITIVE TCOST	
03	#		R	INFORMATION	

Field

Line	Type (N/O/F/D/S/*)	Location	Forced	Overflow Field Length	Comments
07	F	15-21			NORMAL ZOWED
08	F	1-23			ENTIRE RECORD-DATA FIELD

■ Ideographic sort utility only.

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.

SAMPLE JOB 8: REFORMATTING INVENTORY FILE USING EQUAL CONTROL FIELD ORDERING

- The input for this job is the inventory file.
- The records are reformatted using equal control field ordering, but the records are not reordered.

Line	Job	Card Match	Stacker Select	Output Record Length	Job Description
010	H SORTRE			30	

Line	Factor 1	Rel	Factor 2 (Field, Constant, or Keyword)	Comments
01		EQ	Constant	
02		NE		
03		LT	Keyword	
04		GE		
05		LE	Location	
06		GE		

Line	Record Character	Substitute Character	Continuation	Overflow Field Length	Reserved	Field Name	Comments
07	D	C	1	6		ITEM	STOCK ITEM NUMBER
08	F	D	17	24		DESC	DESCRIPTION OF ITEM
09	F	D	30	34		COST	COST PER UNIT
10	F	D	35	39		PRICE	PRICE OF ITEM PER ITEM
11	F						
12	F						
13	F						
14	F						

Header Specification

Column 6

H identifies the line as a header line.

Columns 7-12

SORTR identifies the job as a tagalong sort.

Column 12

E indicates that equal control field ordering is to be used.

Columns 13-17

No control fields defined by field statements are specified for this job. A 3 is placed in columns 13 through 17 to allow for the 3-byte relative record number that the sort program is to use as the control field.

Column 18

The A stands for ascending order. The records will be placed into the output file in the same order they are contained in the inventory file.

Column 28

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

Columns 29-32

Because the control field is dropped from the output records, the length of the output records is the total length of the fields defined as data fields in the field specifications. The length is 30, which is the entry in 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 in column 6 of lines 07 through 10 indicates that these are field specification lines.

Columns 7, 8, and 9-16

Lines 07 through 10 indicate that the ITEM, DESC, COST, and PRICE fields are included as data fields in the output records.

SAMPLE JOB 9: MERGING MULTIPLE PRESORTED INPUT FILES USING EQUAL CONTROL FIELD ORDERING

- The input for this job is output files created by *Sample Job 6*.
- The records in the files are merged in ascending order using equal control field ordering.

Note: The input files do not have to be presorted in order to perform a merge. The sort and merge can be performed simultaneously.

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	M	P	U	O	P	Output Record Length	Record Length		
0 0 0 0	H	SORTRE	10A							23			

Line		Factor 1		Factor 2 (Field, Constant, or Keyword)		Comments	
Number	Type (IO)	From	To	From	To		
0 1							
0 2							
0 3							
0 4							
0 5							
0 6							

Line		Location		Record Character		Overflow Field Length		Comments	
Number	Type (N/D/S*)	From	To	Substitute Character	Continuation	Alt. Sub. Field (A)	Alt. Sub. Field (B)		
0 7	F	1	7					CUST	CUSTOMER NUMBER
0 8	F	2	23					ENTIRE	RECORD-DATA FIELD
0 9	F								
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.

Column 12

E indicates that equal control field ordering is to be used.

Columns 13-17

The control field consists of the 7-byte customer number plus the 3-byte relative record number. The length of the two control fields, 10, is the entry for columns 13 through 17.

Column 18

The A stands for ascending order. The A, along with the control field (CUST) and equal control field ordering, causes the merge to occur.

Column 28

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

Columns 29-32

Because the control field is dropped from the output records, the length of the output records is the total length of the fields defined as data fields in the field specifications. The length is 23, which is the entry in 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 in column 6 of lines 07 and 08 indicates that these are field specification lines.

Columns 7, 8, and 9-16

- Line 07 is the same control field, CUST (customer number), that was used by *Sample Job 6* to presequence the input files. This control field, along with equal control field ordering and the A in column 18 of the header specification, causes the merge to occur.
- 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)	¢	4A	1 (lowest)	blank	40
2	¢	4A		.	4B		&	50
3	.	4B		<	4C		- (minus)	60
4	<	4C		(4D		{	C0
5	(4D		+	4E		}	D0
6	+	4E	2		4F	2	\	E0
7		4F		!	5A		0	F0
8	&	50		\$	5B		/	61
9	!	5A		*	5C		a	81
10	\$	5B)	5D		j	91
11	*	5C		;	5E		~	A1
12)	5D			5F		A	C1
13	;	5E	3	/	61	J	D1	
14		5F		:	6A	1	F1	
15	- (minus)	60		,	6B	3	b	82
16	/	61		%	6C		k	92
17	:	6A		_	6D		s	A2
18	,	6B		(underscore)	6E		B	C2
19	%	6C		>			6F	K
20	_ (underscore)	6D	?	6F	S	E2		
21	>	6E	4	\	79	2	F2	
22	?	6F		:	7A	4	c	83
23	\	79		#	7B		l	93
24	:	7A		@	7C		t	A3
25	#	7B		'	7D		C	C3
26	@	7C		(apostrophe)	7E		L	D3
27	' (apostrophe)	7D		=			7F	T
28	=	7E	"	7F	3		F3	
29	"	7F	5	a	81	5	d	84
30	a	81		b	82		m	94
31	b	82		c	83		u	A4
32	c	83		d	84		D	C4
33	d	84		e	85		M	D4
34	e	85		f	86		U	E4
35	f	86		g	87		4	F4
36	g	87	h	88	6	e	85	
37	h	88	i	89		n	95	
38	i	89	6	j		91	v	A5
39	j	91		k		92	E	C5
40	k	92		l		93	N	D5
41	l	93		m		94	V	E5
42	m	94		n		95	5	F5
43	n	95		o	96	7	f	86
44	o	96		p	97		o	96
45	p	97	q	98	w		A6	
46	q	98	r	99	F		C6	
47	r	99			O		D6	
					W	E6		
					6	F6		

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 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		9	h	88	
56	z	A9		z	A9	q		98		
57	{	C0	&	50	y	A8				
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	10	\		79		
63	F	C6	E	C5		i		89		
64	G	C7	F	C6		r	99			
65	H	C8	G	C7		z	A9			
66	I	C9	H	C8		l	C9			
67	}	D0	I	C9		R	D9			
68	J	D1	- (minus)	60		Z	E9			
69	K	D2	}	D0		9	F9			
70	L	D3	J	D1		11	¢	4A		
71	M	D4	K	D2	!		5A			
72	N	D5	L	D3	!		6A			
73	O	D6	M	D4	:		7A			
74	P	D7	N	D5	12		.	4B		
75	Q	D8	O	D6			\$	5B		
76	R	D9	P	D7			,	6B		
77	\	E0	Q	D8			#	7B		
78	S	E2	R	D9			<	4C		
79	T	E3	\	E1		*	5C			
80	U	E4	S	E2		%	6C			
81	V	E5	T	E3		@	7C			
82	W	E6	U	E4		14	(4D		
83	X	E7	V	E5)		5D			
84	Y	E8	W	E6	_		6D			
85	Z	E9	X	E7	(underscore)		15	'	7D	
86	0	F0	Y	E8	(apostrophe)			16	+	4E
87	1	F1	Z	E9	15				;	5E
88	2	F2	blank	40					>	6E
89	3	F3	0	F0					=	7E
90	4	F4	1	F1					16	
91	5	F5	2	F2		┘				5F
92	6	F6	3	F3		?				6F
93	7	F7	4	F4		"				7F
94	8	F8	5	F5		11	(highest)			(highest)
95	9	F9	6	F6						
(highest)			7	F7						
			8	F8						
			9	F9						

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 for the entire control field, code an S in column 26 of the header specification and code ALTSEQ statements.

If you want to change the standard collating sequence for specified control fields, code an F in column 26 of the header specification, code ALTSEQ statements, and code an A in column 20 for each normal or opposite control field specification you want to change.

You can code as many ALTSEQ statements as you need. Each statement must begin with ALTSEQ and can contain a maximum of 96 positions.

Note: Do not use packed or zoned factors (P or U in column 8) in an include or omit record type specification if you specify an alternate collating sequence on the entire control field (S in column 26 of the header specification). However, you can use packed or zoned factors in include or omit record type specifications if you are using an alternate collating sequence on specified control fields (F in column 26 of the header specification.)

Coding Rules

1. Code ALTSEQ in the first six positions to tell the sort program that 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  | h | n | | | |
```

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  | | h | n | h | n |
```

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  | h | n | h | n | h | n |
```

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	nnnn	maximum of 96 positions
ALTSEQ	nnnn	maximum of 96 positions
ALTSEQ	nnnn	maximum of 96 positions
**		

¹ See 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

When an alternate collating sequence on the entire control field is used, ALTSEQ statements change:

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

When an alternate collating sequence on specified control fields is used, ALTSEQ statements only change specified normal and opposite control 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* 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

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	\	E0	
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	F0	
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.

Fourth Edition

Use this form only to document an alternate collating sequence or translation table for your program. The form could contain typographical errors.

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


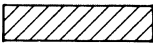
HEADER SPECIFICATION COLUMN SUMMARY

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.
	E (in column 12 only)	Maintain original sequence of equal control field records for addrout and tagalong sort jobs.
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 on the entire control field in compare operations. ALTSEQ statements will define the collating sequence to be used.
	F	Use an alternate collating sequence on specified control fields 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	1	Used only in System/34 ideographic sort jobs.
	2	Used only in System/34 ideographic sort jobs.
	Blank	Used only in System/34 ideographic sort jobs.
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.
40-74	Any characters	Comments.
<div style="display: flex; justify-content: center; gap: 20px; margin-top: 10px;"> <div style="text-align: center;"> <p>Columns that must be filled in for all sort jobs.</p> </div> <div style="text-align: center;"> <p>Columns that must be filled in for tagalong sort jobs.</p> </div> </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.
	S	Used only in System/34 ideographic sort jobs.
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.
40-74	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	Entries	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.
	E	Used only in System/34 ideographic sort jobs.
	I	Used only in System/34 ideographic sort jobs.
	J	Used only in System/34 ideographic sort jobs.
	R	Used only in System/34 ideographic sort jobs.
	S	Used only in System/34 ideographic sort jobs.
	T	Used only in System/34 ideographic sort jobs.
	9-12	1-4096
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.
	A (column 20 only)	Alternate collating sequence by field.
23-39	Blank	Reserved for system use.
40-74	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		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	O	Print Option	Output Option (X)	Record 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	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	74
0	0	H	SORTR								64																																																												

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

Line		Field		Comments	
Number	Type (NO/F/D/S/)	Location	Record Character	Overflow Field Length	Field Name
3	4	5	6	7	8
0	7	F	MC	1 4	CONTROL FIELD
0	8	F	DC	5 64	REST OF DATA
0	9	F			
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 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.



International Business Machines Corporation
SEQUENCE SPECIFICATIONS
 Header

GX21-9089 UM050*
 Printed in U.S.A.

Page 02 1 2
 Program Identification 75 76 77 78 79 80

Line	Job	Card Match	Stacker Select	Output Record Length	CCP Disk Sort Only	Record Length	Job Description
3	Card: MATCH SORT SELECT	Largest sum (in bytes) of Control Field Lengths for Any Record Type	S M P U O S	29	25 Check/Run Optimiz (C) 26 Run Verif Optimiz (V) 27 Null Output (N) 28 Storage Size to Execute 29 Length Control (C)	40	
4	DISK SORT TYPE SORTA SORTB SORTC SORTT						
5	Type						
6	0						
7	0						
8	0						
9	H						

Record Type

Line	Factor 1	Rel.	Factor 2 (Field Constant or Keyword)	Comments
3	EQ NE LT GT LE GE		Keyword Constant	Record Name
4	Location		Location	
5	From		From	
6	To		To	
7	0			
8	1			INCLUDE NEGATIVE INTEGERS IN SORT
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				
73				
74				

Field

Line	Location	Record Character	Substitute Character	Continuation	Alt. Sort Field (A)	Overflow Field Length	Comments
3	From						Reserved
4	To						Field Name
5	0						
6	7	F	C				FORCE 1 FOR NEGATIVE INTEGERS
7	8	F	NC	1	4		CONTROL FIELD
8	9	F	DC	1	64		DATA FIELD
9							
10							
11							
12							
13							
14							

■ Ideographic sort utility only.

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.

SAMPLE SORT JOB 3. SORTING ON A CONTROL FIELD THAT CONTAINS ONLY A POSITIVE OR NEGATIVE REAL NUMBER

Line		Job		Card Match		Stacker Select		Output Record Length		Record Length		Job Description	
Number	Type	Card: MATCH, SORT, SELECT, MERGE	Largest sum (in bytes) of Control Field Lengths for Any Record Type	S	U	M	P	S	O	U	Length	CCP Disk Sort Only	Program Identification
00	H	SORT	5A								64		

Line		Factor 1		Factor 2 (Field, Constant, or Keyword)		Comments	
Number	Type (H/O)	From	To	Rel.	Location	Record Name	
01	L			EQ	11-16	ALL POSITIVE REAL NUMBERS	

Line		Field		Comments		
Number	Type	From	To	Forced	Overflow Field Length	Field Name
07	F					FORCE 2 FOR POS. REAL NUMBERS
08	F	1	4			CONTROL FIELD
09	F	1	64			DATA FIELD



SEQUENCE SPECIFICATIONS Header

Line	Job	Card Match	Sequence (M/D/S)	Stacker Select	Number (L/N)	Print Option	Output Option (X)	Output Record Length	Record Length	Job Description
1	Card MATCH SORT SELECT MERGE									
2	Disk SORT/RE SORTA, SORTBS									
3	Table SORTT									

Record Type

Line	Factor 1	Rel.	Factor 2: Field	Constant or Keyword	Comments
3		EQ		Keyword	
4		NE		Constant	
5		LT			
6		GT			
7		LE			
8		GE			
0 1					INCLUDE NEGATIVE REAL NUMBERS
0 2					
0 3					
0 4					
0 5					
0 6					

Field

Line	Type	Location	Record Character	Substitute Character	Continuation	Overflow Field Length	Comments
0 7	F				1		FORCE L FOR NEG. REAL NUMBERS
0 8	F	1		4			** OPPOSITE SEQUENCE CONTROL FLD.
0 9	F	1		64			DATA FIELD
1 0	F						
1 1	F						
1 2	F						
1 3	F						
1 4	F						

■ Ideographic sort utility only.

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.

Record Type

Line		Type (I/O)	Continuation (A/O/?)	C/Z/D/P/U	Stucker Select	Factor 1		Rel.	Factor 2: Field, Constant, or Keyword		Comments
Number	Location					From	To		From	To	
0 1	I O					2EQC@					DIGIT = C
0 2	I AP					2LEC2					AND ≤ HEX 2F
0 3	I AP					2GTCL					AND > HEX 1F
0 4											
0 5											
0 6											

	Hexadecimal Value		Binary Value		Decimal Value
Field to be selected	2C	=	0010 1100	=	44
Character @	7C	=	0111 1100	=	124
Packed character 2	2F	=	0010 1111	=	47
Packed character 1	1F	=	0001 1111	=	31

The record type specifications shown will cause selection of all records containing (in column 2) a value between 31 (hex 1F) and 47 (hex 2F) whose digit portion is equal to C.

Note: The 16 hexadecimal values whose zone portions are B cannot be accessed with these types of entries. Therefore, there can be no record selection based on these values.

Sample Sort Job 4 Description

The purpose of this sample job is to select records from an inventory file whose reorder point is 7800. The reorder point is a 2-byte quantity in columns 44 and 45 of the inventory records. The quantity is stored in binary (7800 = 0001 1110 0111 1000), and the hexadecimal representation of this number is hex 1E78. (Use the *Programming Reference Summary* to help with binary and hexadecimal conversions.)

IBM

International Business Machines Corporation
SEQUENCE SPECIFICATIONS
 Header

GX21-9089 UM050*
 Printed in U.S.A.

Page 011 Program Identification 75 76 77 78 79 80

Line	Job	Card Match	Stacker Select	Output Record Length	Record Length	Job Description
1	SORTER	6A		111		

Line	Type	Factor 1	Rel	Factor 2	Comments
0 1	ID	44EQC+		DIGIT = E **	
0 2	IAP	44LEC1		AND < = HEX 1F ** TYPE 2	
0 3	IAP	44GTC0		AND > HEX 0F **	
0 4	*****			*** AND ***	
0 5	IAZ	45EQC@		ZONE = 7 ** TYPE 1	
0 6	IAD	45EQC8		DIGIT = 8 **	

Line	Type	Location	Forced	Overflow Field Length	Comments
0 7	ND	1 6			SEQUENCE BY ITEM
0 8	DC	7 111			REST OF DATA
1 0	F				
1 1	F				
1 2	F				
1 3	F				
1 4	F				

■ Ideographic sort utility only.

	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.

Appendix E. Loadable Sort Interface

The loadable sort interface allows you to load the sort program from an assembler program. The assembler macros, \$SORT and \$SRT, can be used to generate the call to the sort program and the loadable sort parameter list. For information on \$SORT and \$SRT, refer to the *Basic Assembler and Macro Processor Reference Manual*. For information on the loadable sort parameter list, see *Loadable Sort Parameter List* in this appendix. When the loadable sort interface is used, the sort sequence specifications must be either in a source member or in the loadable sort parameter list.

Remember the following when using the loadable sort interface:

- All files used during the sort job must have been defined by FILE statements and must have been closed before the sort program is called. (See the *System Support Reference Manual* for an explanation of how to specify FILE statements.)
- If one (or more) of the input files used by the sort program is an offline multivolume file, the work file must be defined by a FILE statement. (See Chapter 5 for information on how to specify a FILE statement for a work file.)
- If a sort job does not complete successfully, control is returned to the assembler program with bit 3 of the indicator byte on in the loadable sort parameter list (see *Loadable Sort Parameter List* in this appendix).

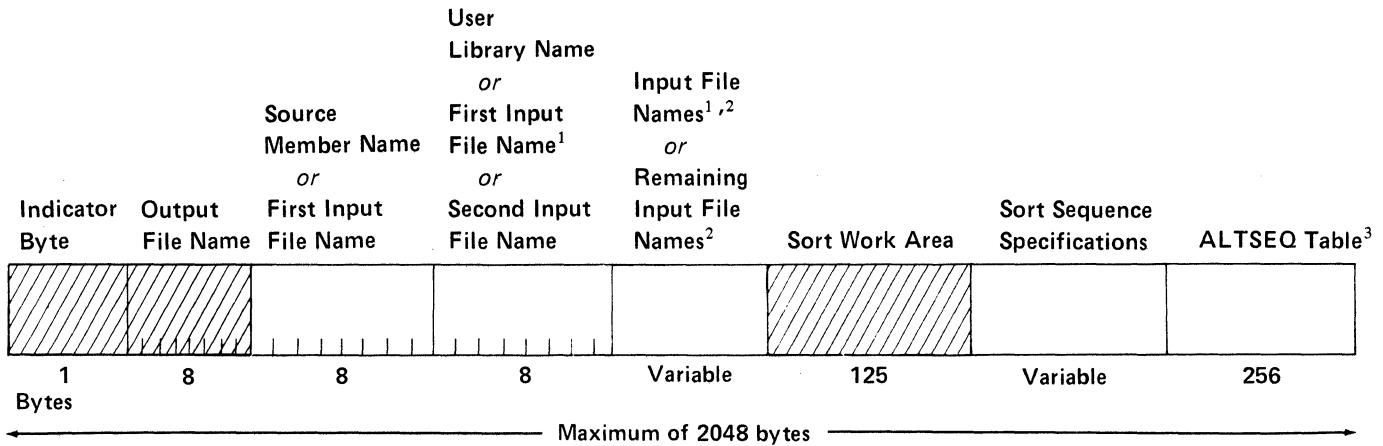
LOADABLE SORT PARAMETER LIST


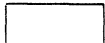
The size (in bytes) of the loadable sort parameter list varies depending upon:

- Whether the sort sequence specifications are in a source member or in the parameter list
- The number of input file names being passed to the sort program
- The number of sort sequence specifications, if they are in the parameter list
- Whether or not an ALTSEQ table is in the parameter list

The parameter list can contain a maximum of 2048 bytes.

An overview of the loadable sort parameter list follows.



 = Required
 = Optional

¹ At least one input file name must be passed to the sort program.
² Each input file name (up to a maximum of eight) must be 8 bytes long.
³ The ALTSEQ table must be in bytes 1793 through 2048.

An explanation of the contents of the parameter list follows.

The indicator byte, the first byte of the parameter list, contains indicators in the first four bits and the number of input files in the last four bits:

Indicator Byte

Bit 0 *On*: Sort sequence specifications are in a source member.
 Off: Sort sequence specifications are in the parameter list.

If bit 0 is off, bit 1 also must be off.

Bit 1 *On*: Source member is in a user library.
 Off: The source member containing the sort sequence specifications is in #LIBRARY, or the sort sequence specifications are in the parameter list.

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.

Bit 2 *On:* An ALTSEQ table is in bytes 1793 through 2048 (the last 256 bytes) of the parameter list.
Off: An ALTSEQ table is not in bytes 1793 through 2048 (the last 256 bytes) of the parameter list.

Bit 3 *On:* The sort job did not complete successfully.
Off: The sort job was successful.

 This bit must be off when calling the sort program.

Bits 4-7 Specifies the number of input file names (maximum of eight) in the parameter list. For example, 0010 in the last four bits indicates that two input file names are being passed to the sort program.

Output file name Must be in bytes 2 through 9. The name must begin in byte 2; if the name is not 8 bytes long, it must be padded on the right with blanks.

Source member name Must be in bytes 10 through 17 if the sort sequence specifications are in a source member (bit 0 of the indicator byte is on). The name must begin in byte 10; if the name is not 8 bytes long, it must be padded on the right with blanks.

User library name Must be in bytes 18 through 25 if the source member named in bytes 10 through 17 is in a user library (bit 1 of the indicator byte is on). The name must begin in byte 18; if the name is not 8 bytes long, it must be padded on the right with blanks.

 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.

Input file names A maximum of eight input file names can be specified in the parameter list. The placement of the first file name in the parameter list depends on the status of bits 0 and 1 of the indicator byte.

 If bits 0 and 1 of the indicator byte are both off, the first file name must be specified in bytes 10 through 17 of the parameter list.

 If bit 0 is on and bit 1 of the indicator byte is off, the first input file name must be specified in bytes 18 through 25 of the parameter list.

 If bits 0 and 1 of the indicator byte are both on, the first input file name must be specified in bytes 26 through 33 of the parameter list.

 Additional input file names must directly follow the 8 bytes of the previous input file name. An input file name must begin in the leftmost byte of the 8 bytes; if the file name is less than 8 bytes long, it must be padded on the right with blanks.

Sort work area Must immediately follow the 8 bytes of the last input file name. This 125-byte area is used as a work area when giving control to and returning control from the sort program.

Sort sequence specifications

If bit 0 of the indicator byte is off, the sort sequence specifications, corresponding to columns 6 through 39 on the sequence specifications form, must immediately follow the 125-byte sort work area. The header specification must come first, followed by the record type and field specifications. Each sort sequence specification must be 34 bytes long.

If an ALTSEQ table is specified (bit 2 of the indicator byte is on), do not specify the 34-byte sort sequence specifications beyond byte 1792 of the parameter list. If an ALTSEQ table is not specified (bit 2 of the indicator byte is off), do not specify the 34-byte sort sequence specifications beyond byte 2048 of the parameter list.

You must specify a // END statement immediately following the last 34-byte sort sequence specification unless the last sort sequence specification ends beyond byte 1758 (when an ALTSEQ table is specified) or beyond byte 2014 (when no ALTSEQ table is specified). In these cases, the sort program assumes a // END statement.

ALTSEQ table

Must be in bytes 1793 through 2048 (the last 256 bytes) of the parameter list if bit 2 of the indicator byte is on.

The table must contain 256 entries, each 1 byte long. The 1-byte entries correspond to the entries in the ENTRY column of the *Translation Table and Alternate Collating Sequence Coding Sheet*. When you want to alter the normal collating sequence in the table, enter an alternate collating sequence entry instead of the normal collating sequence entry. (See Appendix B for information on the collating sequence and an example of the *Translation Table and Alternate Collating Sequence Coding Sheet*.)

Note: The use of the loadable sort interface increases the time needed to run a sort job. For improved performance when using the loadable sort interface, increase the region size when you have enough main storage available to do so. (See the *System Support Reference Manual* for information on how to increase the region size.)

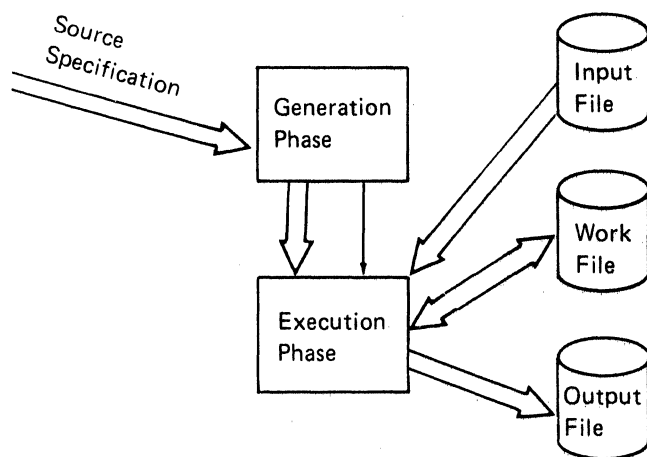
Appendix F. Sort Printed Messages

This appendix describes sort programmer printed messages. (Sort displayed messages are described in the *Displayed Messages Guide*.) Sort programmer messages are activated (or suppressed) by the print option in the header statement. If activated, these messages are issued by 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 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-7000 through SORT-7099

These messages are sort displayed messages and are described in the *Displayed Messages Guide*.

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; then 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-7106 S BOTH INPUT AND INPUT1 FILE STATEMENTS PRESENT

File statements contain the file names, INPUT and INPUT1. One, but not both, of the file names can be used for a job. Eliminate one of the FILE statements or change one of the file names to INPUT2, INPUT3, INPUT4, INPUT5, INPUT6, INPUT7, or INPUT8; then resubmit the job.

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 a blank, an S, or an F. An S entry is assumed. If alternate collating sequence is not to be used, leave column 26 of the header line blank; then resubmit the job. If alternate collating sequence on the entire control field is correct, place an S in column 26; then resubmit the job. If alternate collating sequence by field is correct, place an F in column 26 and place an A in the normal and opposite control field lines to be altered; then resubmit the job.

**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-7163 T TOO MANY INPUT
RECORDS IN THE INPUT
FILES**

More than 16 777 215 records are in the input files. (Multiple input files can contain a maximum of 16 777 215 records.)

If the job has two or more input files and some of the input records will be omitted from the job, you may be able to complete the job by doing the following: eliminate one or more of the input files from the job (so that there are fewer than 16 777 215 records) and use the output file from the job as an input file for a second job along with the input file(s) that was eliminated from the first job.

**SORT-7164 S SORTA INVALID WITH
MULTIPLE INPUT FILES**

An addrout sort (SORTA in columns 7 through 12 of the header specification) cannot be specified when multiple input files are used for a job. Modify your OCL statements to perform an addrout sort on only one input file, or merge the input files into one file, before performing an addrout sort.

**SORT-7165 T INPUT RECORD LENGTHS
INCONSISTENT**

Multiple input files are being used as input to the sort program, and the record lengths for all of the input files are not the same. The record lengths for all of the input files must be the same.

**SORT-7176 S INVALID ALTERNATE
SEQUENCE STATEMENT**

Column 26 of the header line specifies that an alternate sequence is to be used or an alternate collating sequence table was passed to the job in the loadable sort parameter list, 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. Do one of the following: correct the incorrect statement, delete the incorrect statement 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 or increase the region size.

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-RELA-
TIONSHIP 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 packed or zoned factors (P or U in column 8), but the header line specifies an alternate collating sequence, or an alternate collating sequence table was passed to the job in the loadable sort parameter list. The sort program changes factor 1 and factor 2 as the ALTSEQ specifications indicate when alternate collating sequence on the entire control field is specified. This change can affect the units position (and the sign) of a zoned 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 on the entire control field (S in column 26 of the header specification).

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-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-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-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-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-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-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-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 indi-
cates 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-7700 through SORT-7799

These messages are sort displayed messages and are
described in the *Displayed Messages Guide*.

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.

**SORT-7934 A CANNOT WRITE DELETED
RECORD TO DELETE-
CAPABLE FILE**

The sort output file is a delete-capable file, and an attempt is being made to write a deleted record (hex FF in the first byte of the record) to the file. To correct this error do one of the following:

- Change the sort output file so that it is not a delete-capable file
- Supply new sequence specification statements so that hex FF is not placed in the first byte of any of the output records

Sort displayed message SORT-7734 is issued after this message.

addroot file: A record address disk file produced by the sort program. An addroot 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.

addroot 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*.

delete-capable file: A file that can contain records that are logically deleted, though no physical compression occurred when the records were deleted.

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.

extendable disk file (EDF): A shared or nonshared file that can be increased by a specified value whenever additional space is needed.

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).

ideographic: Consisting of both graphics and pictograms and often other types of symbols.

Ideographic Generator/Sort Program Product: The program product (program number 5726-IG1) that consists of the character generator utility and the ideographic sort utility.

ideographic sort utility: A part of the Ideographic Generator/Sort Program Product that is used to 1) arrange records (or their relative record numbers) in a predetermined sequence according to data contained in one or more specific fields within the records and 2) rebuild the tables which contain the predetermined sequence if the user wants to alter the predetermined sequence.

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 UDATE).

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.

multivolume file: A diskette file that resides on more than one diskette. See also *offline multivolume file*.

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.

offline multivolume file: A multivolume file that is processed in segments by the system. Each segment is processed before the next segment is copied to or from the disk.

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.

- * indicates comments on field lines 4-6
- * indicates comments on record type lines 3-5

- action messages (A)
 - issued by the sort program 2-5
- addrout (address out) sort
 - (see also SORTA)
 - definition G-1
 - example 1-2
- alphanumeric constants
 - as factor 2 (columns 20-39-record type specifications) 3-12
 - length and position 3-12
- alphanumeric data
 - effect of column 8 record type entry** 3-7
- alternate collating sequence
 - effect on factor 1 and factor 2 comparison** 3-11
 - effect on sort time** 6-1
 - ALTSEQ statement use B-3
 - definition G-1
 - entire control field 2-5
 - indicated on header line (column 26) 2-4
 - restrictions on using column 8 record type entries B-3
 - specified control fields 2-5
 - storage required 6-1
- ALTSEQ statements
 - coding B-3
 - definition G-1
 - examples B-4
 - loadable sort parameter list E-2, E-4
 - placement in sequence specifications B-3
 - programming considerations B-4
 - what they can and cannot change B-4
- AND lines
 - described 3-5
 - describing a record type 3-4
 - describing additional factor 1 fields 3-10
 - include sets 3-4
 - omit sets 3-5
- ascending or descending sequence
 - specified in column 18 of header specification 2-4
 - with equal control field ordering (E in column 12 of header specification) 2-3
- automatic work file allocation 5-2, 6-3
 - effect on sort time** 5-6, 6-3
 - multiple disk system 6-3
 - single disk system 6-3
- backup copy
 - access by addrout sort (SORTA) output 5-3
 - binary sorting D-1
 - definition G-1
 - overlying a file 5-3
- blank records
 - omitting 5-1
 - test for 5-1
- bypass halt for message SORT-7724 2-6

- C/Z/D/P/U (column 8-record type specifications) 3-6
 - effect on factor 1 length (columns 9-16)** 3-10
 - alphanumeric or numeric data indicator 3-6
 - compare characters indicator** 3-6
 - indicate records you want to sort** 3-6
 - packed or zoned data indicator 3-6
 - restrictions when using alternate collating sequence 3-6
 - signed and unsigned data 3-8
 - when factor 1 and factor 2 contain alphanumeric data 3-7
- C/Z/D/P/U/V (column 8-field specifications) 4-6
 - effect on sorted file** 4-6
 - combined with column 7 entry 4-10
 - input characters selector 4-6
 - what it tells sort 4-6
- calculating file size A-1
- calculating sum of control field lengths 2-4
- character overflow indicator field (column 17-field specifications) 4-12
- character replacement 4-12
- character substitute 4-12
- characters
 - compare operations 3-6
 - representation within the computer 3-7
- collating sequence (column 26-header specification) 2-4
 - alternate
 - definition G-1
 - entire control field 2-5
 - specified control fields 2-5
 - specified for compare operations 2-4
 - standard B-3, 2-5

- collating sequence (column 26–header specification) (continued)
 - table B-1, B-2
- command (see SORT command)
- comment line
 - field specification
 - columns used (40-74) 4-25
 - description 4-6
 - storage used 6-1
 - record type specification
 - columns used (40-74) 3-14
 - description 3-14
- comments (columns 40-74–field specifications) 4-25
- comments (columns 40-74–record type specifications) 3-14
- compare operations
 - column 19 of record type specifications effect 3-11
 - column 8 of record type specifications effect 3-6
 - columns 17-18 of record type specifications effect 3-11
 - description 2-4
- comparisons using program date 3-14
- conditional force
 - definition G-1
 - example 4-4
 - using a force-all line 4-19
 - using normal or opposite and forced control fields 4-17
 - using only forced control fields 4-16
- conditionally forced character (column 17–field specifications) 4-12
 - assigning value to summary overflow field 4-12
 - examples 4-12
 - how to define 4-13
 - what it tells sort 4-12
- constant
 - alphameric 3-12
 - as factor 2 (columns 20-39–record type specifications) 3-12
 - definition G-1
 - indicated by column 19 of record type specifications 3-11
 - length when used as factor 2 3-12
 - numeric 3-12
 - position when used as factor 2 3-12
 - positioning sign in numeric constants 3-12
- continuation or comment (column 7–record type specifications) 3-5
 - comment line indicator 3-5
 - describing different record types with identical control fields 3-5
 - include or omit set indicator 3-5

- control and data fields
 - describing control field as both 2-6
 - use 2-6
- control fields
 - definition G-1
 - dropping 2-6
 - duplicate 4-4
 - equal ordering 2-3
 - forced 4-4
 - conditional 4-4
 - force-all 4-5
 - unconditional 4-5
 - including original data in output record 4-7
 - major 4-3
 - minor 4-3
 - normal 4-4
 - number of when two or more record types 4-4
 - opposite 4-4, 4-7
 - packed or zoned 4-7
 - 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 4-3
 - used as data fields 2-6
- data
 - alphameric 3-7
 - definition G-1
 - numeric 3-8
 - representation within the computer 3-7
 - used in compare operations 3-6
- data and control fields
 - describing control fields as both 2-6
- data fields
 - describing on field specifications line 4-5
 - description 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)
- date comparison 3-14
- defining
 - conditionally forced characters 4-13
 - force-all characters 4-13
 - unconditionally forced characters 4-13
- delete-capable file
 - as input file 5-1
 - definition G-1
 - restriction as output file 5-2

diagnostic messages (S, T, W) F-1, 2-5
digit portion of a character 3-7
display station keyboard, entering
procedures through 5-12
displayed messages 2-5
dropping control fields 2-6

EBCDIC characters 3-7
EDF (extendable disk file) (see extendable
disk file (EDF))
END statement
to indicate last sequence
specification 5-8, 5-12
to indicate last sequence specification
in loadable sort parameter list E-4
entering OCL statements 5-1, 5-7
entering sort procedures through a
keyboard 5-1, 5-10
equal control field ordering (column
12-header specification)
combining records (merge) 2-3
example 7-27
effect of column 18 (ascending or
descending sequence of header
specification) 2-3
specifying 2-3
to reformat records 2-3
example 7-25
executing sort procedures stored on
disk 5-8
execution phase F-1
definition G-2
extendable disk file (EDF)
definition G-2
work file limitation 5-2

factor 1 (columns 9-16-record type
specifications) 3-10
changed by ALTSEQ statements B-4
coding as one character 3-10
coding rules 3-10
column 8 entry affect 3-10
keyword lengths 3-10
length 3-10
multiple fields 3-10
record identifier 3-10
factor 2 (columns 20-27 or 20-39, record
type specifications) 3-11, 3-12, 3-14
alphameric constants 3-12
ALTSEQ statements affect B-4
character constants 3-12
coding rules 3-11
constants 3-11

factor 2 (columns 20-27 or 20-39, record
type specifications) (continued),
ending position 3-11
keywords 3-14
length 3-11, 3-12
location 3-11
negative numbers 3-13
numeric constants 3-12
one character entry 3-11
packed numbers 3-12
program date comparison 3-14
relationship to factor 1 field 3-11
signed constants 3-13
signed numbers 3-12
starting position 3-11
zoned numbers 3-13
field
definition G-2
field length, column 8 entry affect (field
specifications) 4-12
field location (columns 9-16-field
specifications) 4-11
field specifications 4-1
as part of include set 3-4
column entries summary C-3, 4-2
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
7-8 entry combinations 4-10
8 (Portion of Input Record to Use in
Building/Sorting Work Records) 4-6
9-12 (from) 4-11
9-16 (field location) 4-11
13-16 (to) 4-11
17 (conditionally forced
character) 4-12
18 (substitute character) 4-12
19 (specifying replacements for control
field characters) 4-14
20 (alternate collating sequence by
field) 4-25
20-22 (overflow field length) 4-20
23-39 (reserved) 4-25
40-74 (comments) 4-25
for sort types 4-3
what they tell sort 4-1
field type or comments (column 7-field
specifications) 4-3
field, constant, or keyword (column
19-record type specifications) 3-11
factor 1 field comparison 3-11
factor 2 field identifier 3-11
file
allocation of work file 5-2
automatic allocation 5-2
effect of disk location on sort
time 6-2

- file (continued)
 - input file 5-1
 - input overlay considerations 5-3
 - labels 5-4
 - location on disk 6-2
 - output file 5-2
 - placement on single/multiple disk system 6-2
 - procedure requirements 5-7
 - programming considerations 5-3
 - size, calculating A-1
 - SORT command considerations 5-6
 - status 5-1
 - types used with sort 1-5
 - work file 5-2
- FILE statement
 - parameter descriptions 5-7
 - procedure requirements 5-7
- force-all
 - definition G-2
 - explanation 4-5
 - result if not used 4-13
 - use after conditional force lines 4-5
- force-all character
 - defining 4-13
 - example 4-19
 - using 4-5
- forced character (column 18-field specifications)
 - conditionally forced 4-12
 - summary overflow field initialization 4-12
 - when, how character forced 4-12
- forced control fields
 - conditional 4-4
 - definition G-2
 - examples 4-14
 - force-all 4-5
 - rules for using 4-7
 - types 4-4
 - unconditional 4-5
 - when to use 4-4
- forced data field 4-6, 4-12
- format of records
 - definition 1-8
 - effect on sequence specifications 1-8
- from entry (columns 20-23-record type specifications) 3-11
 - factor 2 position indicator 3-11
- from entry (columns 9-12-field specifications) 4-11
 - describing one character field 4-12
 - input record field indicator 4-11
- from entry (columns 9-12-record type specifications) 3-10
 - factor 1 indicator 3-10

- generation phase F-1
 - definition G-2
- guide to using include and omit sets 3-5
- header specification
 - column entries, explanation of C-1, 2-1
 - column summary C-1, 2-2
 - column(s)
 - 1-2 (page number) 2-3
 - 3-5 (line number) 2-3
 - 6 (line type) 2-3
 - 7-12 (type of sort job) 2-3
 - 12 (equal control field ordering) 2-3
 - 13-17 (largest sum [in bytes] of control field lengths for any record type) 2-4
 - 18 (ascending or descending sequence) 2-4
 - 26 (collating sequence) 2-4
 - 27 (print option) 2-5
 - 28 (output option, SORTR and SORTRS) 2-6
 - 29-32 (output record length for SORTR, SORTRS) 2-6
 - 34 (non-verify option) 2-6
 - 36 (null output-bypass halt for message SORT-7724) 2-6
 - 40-74 (job description) 2-6
 - columns you must fill in C-1, 2-2
 - explanation of column entries C-1, 2-2
- how sort works F-1, 1-5
 - execution phase F-1
 - generation phase F-1
- identifying records to be sorted 3-3
- implied include-all 1-8
 - definition G-2
 - example 1-8
 - include 3-3
- include-all 3-3
 - limitations 3-3
 - purpose 3-3
 - storage used 6-1
- include sequences D-9
 - example D-9
 - type 1 D-9
 - type 2 D-9
- include set
 - definition 3-4
 - field specifications as part 3-4
 - mixing with omit sets 3-5
 - rules for using 3-5
 - types 3-4

include set (continued)
 when to use 3-5
 indicator byte E-2
 initializing overflow field 4-12
 input file
 calculating storage tracks A-1
 characteristics 5-1
 definition G-2
 delete-capable 5-1
 FILE statement required 5-3
 how its position on disk affects sort
 time 6-2
 limitations when using the SORT
 command 5-6
 multiple 5-1
 offline multivolume file
 invalid with the SORT command 5-1, 5-6
 valid with your own sort
 procedures 5-1, 5-7
 overlay considerations 5-3
 programming considerations 5-1, 5-3
 running your own procedure 5-7
 shared file; FILE statement needed for
 the work file 5-3
 input file label 5-4
 input file name(s) in loadable sort
 parameter list E-3
 input needed for the sort program 1-5
 input sequence specifications, order when
 using ALTSEQ B-3
 integer
 definition G-2
 sorting D-1
 interface, loadable sort E-1

job (columns 7-12—header specification),
 to specify type of sort 2-3
 job description (columns 40-74—header
 specification), to supply comments 2-6
 job step
 definition G-2
 in multiple-step procedure 5-6

keyboard, procedures entered through 5-12
 keyword

effect on program date format 3-14
 as factor 2 (columns 20-39—record type
 specifications) 3-14
 definition G-2
 indicated by column 19 of record type
 specifications 3-11

keyword (continued)

length equals factor 1 on record type
 specifications 3-14
program date effect on factor 1
 length 3-14
 relationship to column 8 of record type
 specifications 3-14

label 5-4

definition G-2

largest sum (in bytes) of control field
 lengths for any record type (columns
 13-17—header specification) 2-4
 calculating entry 2-4

length

factor 1 of record type
 specifications 3-10
 factor 2 constant (columns 20-39—record
 type specifications) 3-12
 factor 2 field location (columns
 20-27—record type specifications) 3-11
 maximum for columns 7-8 (field
 specifications) 4-10

library

(see also system library and user
 library)
 defined G-3

line number (columns 3-5) with page number
 (columns 1-2) for sequencing
 field specifications 4-3
 header specification 2-3
 record type specifications 3-3

line type (column 6)

to identify field line (field
 specifications) 4-3
 to identify header line (header
 specification) 2-3
 to identify include lines (record type
 specifications) 3-3
 to identify omit line (record type
 specifications) 3-3

loadable sort interface

characteristics E-1
 loadable sort parameter list E-1
 storage required 6-3

loadable sort parameter list

detailed description E-1
 overview E-2
 size (in bytes) E-1

- merge
 - example 7-27
 - how to perform 2-3
 - when to perform 2-3
- messages
 - definition G-3
 - displayed F-1
 - documentation for sort 2-5
 - issued F-1
 - printed F-1, 2-5
 - types issued by sort 2-5
- multiple disk system
 - definition G-3
 - file placement 6-2
- multiple input files
 - characteristics 5-1
 - merging 2-3
 - storage required 6-1
- multivolume file
 - (see also offline multivolume file)
 - definition G-3

- non-verify option (column 34—header specification) 2-6
 - timing considerations 2-6, 6-3
- normal control field
 - changed by ALTSEQ statements B-4
 - definition G-3
 - how indicated 4-3
 - how sort uses 4-3
- null output—bypass half for message SORT-7724 2-6
- number of input files 5-1
- number of records
 - effect on order of sequence specifications 1-8
 - effect on sort time 6-1
- numeric constants 3-12
- numeric data
 - defined by column 8 record type entry 3-8
 - definition of types 3-8
 - formats and column 8 entry combinations 3-8
 - packed numbers 3-6
 - positive and negative (signed) numeric data 3-8
 - record type specifications entry 3-6, 3-8

- OCL statements
 - examples 5-7, 5-12
 - executing stored statements 5-8
 - file requirements 5-7
 - sort procedure requirements 5-7
 - storing 5-8
 - ways to enter 5-1, 5-8, 5-12
 - what they tell sort 5-7
- offline multivolume file
 - definition G-3
 - FILE statements example 5-2
 - FILE statements required 5-2
 - SORT command, using 5-1, 5-2, 5-6
 - sort procedures, using 5-1, 5-2, 5-7
 - work file allocation 5-1
- omit line
 - how to use 3-3
 - storage used 6-1
- omit set
 - definition 3-4
 - mixing with include set 3-5
 - rules for using 3-4
 - types 3-5
 - when to use 3-5
- opposite control field
 - changed by ALTSEQ statements B-4
 - definition G-3
 - how indicated 4-3
 - how sort uses 4-4, 4-7
 - sorting different record types 4-7
 - when to use 4-4, 4-7
- OR lines
 - describing different record types 3-4
 - description 3-5
 - include sets 3-4
 - omit sets 3-5
- order of sequence specifications
 - effect of record number and format 1-8
 - normal order 1-8
- organization of this manual iii
- out-of-sequence lines
 - inserting in field specifications 4-3
 - inserting in record type specifications 3-3
- output
 - addroot sort 1-2
 - file 5-1
 - summary tagalong sort 1-4
 - tagalong sort 1-3
- output file
 - calculating record lengths 2-6
 - calculating storage A-1
 - characteristics 5-2
 - definition G-3
 - disk position effect on sort time 6-2
 - in loadable sort parameter list E-2, E-3
 - label 5-4
 - limitations with SORT command 5-6

output file (continued)
 offline multivolume file
 SORT command, using with 5-2, 5-6
 sort procedures, using with 5-2, 5-7
 overlay input file 5-3
 programming considerations 5-2, 5-3
 record lengths for tagalong sorts 2-6
 restriction as delete-capable file 5-2
 running your own procedure 5-7, 5-8
 output option for tagalong sort (column
 28-header specification) 2-6
 dropping control fields 2-6
 overflow field
 definition G-3
 overflow field length entry (columns
 20-22-field specifications) 4-20
 overflow indicator field
 how to define (columns 7-8-field
 specifications) 4-13
 initializing (column 18-field
 specifications) 4-12
 when to use 4-20
 overlay of input file 5-3
 overview of the sort program 1-1
 addrout sort 1-2
 summary tagalong sort 1-4
 tagalong sort 1-3

packed character 3-8, 4-7
 packed control field 4-7
 packed numeric data
 number of bits in each digit 3-8
 position of sign code 3-9
 positive and negative sign codes 3-8
 page number (columns 1-2)-with line number
 (columns 3-5)
 for sequencing 2-3, 3-3, 4-3
 parameter list, loadable sort E-1
 positive and negative sign codes 3-8
 print option (column 27-header
 specification) 2-5
 printed messages 2-5
 procedures
 definition G-3
 SORT command 5-4
 substituting values into sequence
 specifications 5-9
 writing your own 5-7
 program date
 as factor 2 3-14
 column 19 entry of record type
 specifications 3-11
 comparing with other fields 3-11, 3-14
 definition G-3
 format affected by factor 2
 keyword 3-14

program date (continued)
 keyword 3-14
 position when used as factor 2 3-14
 program status messages (I) 2-5
 programming considerations
 sort 5-1, 5-3
 SORT command 5-4
 prompts
 definition G-3
 SORT command 5-5

real number
 definition G-3
 sorting D-1
 record
 definition G-4
 record format
 definition 1-8
 effect on order of sequence
 specifications 1-8
 record length (output)
 calculating for tagalong sorts 2-6
 definition G-4
 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 summary C-2, 3-2
 column(s)
 1-2 (page number) 3-3
 3-5 (line number) 3-3
 6 (line type) 3-3
 7 (continuation or comments) 3-5
 8 (C/Z/D/P/U) 3-6
 9-16 (factor 1 field location) 3-10
 17-18 (relation) 3-11
 19 (field, constant, or keyword) 3-11
 20-27 (factor 2 field location) 3-11
 20-39 (factor 2 constant) 3-12
 20-39 (factor 2 keyword) 3-14
 40-74 (comments) 3-14
 columns you must fill in C-2, 3-2
 what they tell sort 3-1
 when not required 3-1
 record types
 definition G-4
 described in include set 3-4
 described in omit sets 3-5
 record, reformatting
 equal control field ordering 2-3
 example 7-25

records

- how number of affects sort time 6-1
- how order of affects sort time 6-3
- using equal control field ordering 2-3
 - example 7-25

region

- definition G-4

region size

- effect on sort time 6-3
- minimum 6-3

relation (columns 17-18—record type specifications) 3-11

- comparing factor 1 and factor 2 3-11
- omitting or including records 3-11

relative record numbers, addrout sort

- output 1-2

replacing characters (column 19—field specifications)

- when and how to indicate 4-14

run time, sort job 6-1

- factors that affect 6-1

sample sort jobs 7-1

SORTA

- relative record addresses for inventory file 7-7

SORTR

- stock items to reorder 7-3
- summary history 7-22
- transaction activity 7-9
- using signed control fields 7-22

SORTRS

- customer number issues 7-19, 7-20, 7-21
- inventory adjustments 7-17, 7-18

sample timings for sort program 6-4

- characteristics of sort jobs 6-4
- sequence specifications used to generate 6-4

select/build routine

- definition 6-1
- storage required 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) 2-4

- sorted record sequence 2-4

sequence error message

- field specifications 4-3
- header specification 2-3
- record type specifications 3-3

sequence specifications

- addrout sort (SORTR) 1-9
- decreasing sort time 6-1
- definition G-4
- grouping 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 3-3
- lines issued by the sort program 2-5
- loadable sort parameter list E-2, E-4
- normal order 1-8
- number for sort job 6-1
- sequence numbers 2-3
- storage 1-6
- summary tagalong sort (SORTRS) 1-11
- tagalong sort (SORTR) 1-10
- use 1-6
- warning issued if not in order 2-3
- wrong ones used 5-8

sequence specifications sheet

- location of types 1-6
- purpose of specification 1-6
- types of specifications 1-6

sequencing

- on field specifications 4-3
- on header specification 2-3
- on record type specifications 3-3
- using control fields 4-7

set

- include set 3-4
- omit set 3-5
- rules for using include and omit sets 3-4
- types of include sets 3-4
- types of omit sets 3-5

sign code

- packed and zoned numeric data 3-8
- placement in packed and zoned numeric data 3-9

signed constants 3-13

- signed numbers 3-8
 - how to code as factor 2 3-12
- sign codes 3-8

single disk system

- file placement 6-2
- size of loadable sort parameter list E-1
- size of records, effect on sort time 6-1
- size of work file 5-3

sort

- types 1-2

SORT command

- how keyed 5-5
- limitations 5-1, 5-6
- parameters 5-5
- prompts 5-5
- sequence specifications placement 5-6
- used to run sort 5-4
- what it does 5-4
- why used 5-4

SORT command (continued)
 work file allocation 5-6
 sort interface, loadable E-1
 sort jobs
 (see also sample sort jobs)
 stages 2-5
 types 1-2
 sort messages F-1, G-3, 2-5
 sort parameter list, loadable E-1
 sort printed messages F-1
 sort procedures
 entering through a display station
 keyboard 5-12
 executing when stored on disk 5-8
 file information needed 5-7
 running sort when using your own 5-7
 source member information 5-8
 sort program
 definition G-4
 description 1-1
 execution 1-7
 file types used 1-5
 functions 1-1
 how works 1-5
 input needed 1-5
 interaction with files 1-7
 messages issued 2-5
 multiple copies 1-1
 operation 1-7
 output considerations 5-7
 procedures stored on disk,
 executing 5-8
 sequence specifications 1-6
 timings 6-4
 sort sequence specification (see sequence specifications)
 sort time 6-1
 factors that affect 6-1, 6-3
 sort work area in loadable sort parameter list E-2, E-3
 SORTA (addrout sort)
 (see also sample sort jobs)
 contents of sorted records 1-2
 function 1-2
 input file overlay 5-3
 output 1-2
 run time 5-3
 sequence specifications 1-9
 using summary data fields 4-5
 work file size 5-3
 sorted file, contents 1-5
 sorting
 integer (binary) D-1
 specifications for positive integer D-1, D-2
 specifications for positive or negative integer D-3, D-4, D-5
 real number D-1
 specifications for positive integer D-1, D-2
 specifications for positive or negative real number D-6, D-7, D-8
 sorting (continued)
 real number (continued)
 specifications for positive or negative real number D-6, D-7, D-8
 SORTR (tagalong sort)
 (see also sample sort jobs)
 calculating output record length 2-6
 contents of sorted records 1-3
 function 1-3
 input file overlay 5-3
 output 1-3
 run time 5-3
 sequence specifications 1-10
 using summary data fields 4-5
 work file size A-1, 5-3
 SORTRS (summary tagalong sort)
 (see also sample sort jobs)
 calculating output record length 2-6
 contents of sorted records 1-4
 function 1-4
 input file overlay 5-3
 output 1-4
 run time 5-3
 sequence specifications 1-11
 using overflow indicator fields 4-13
 using summary data fields 4-5
 work file size A-1, 5-3
 source member
 creating and updating 5-8
 definition G-4
 information needed
 when running your own procedures 5-8
 when using the SORT command 5-4
 loadable sort parameter list E-2, E-3
 SOURCE statement for procedures 5-8
 standard collating sequence
 specified on header line (column 26) 2-4
 table B-1, B-2
 using zone and digit portions B-1, B-2
 substitute character (column 18-field specifications)
 conditionally forced 4-12
 initializing summary overflow fields 4-12
 when and how character is forced 4-12
 substitution expressions
 considerations 5-9
 example 5-9
 with comments 5-11
 summary data fields
 defining 4-5
 definition G-4
 maximum number 4-5
 overflow indication 4-13
 used by each type of sort 4-5
 summary overflow field
 defining 4-13
 initializing 4-12
 summary sort (SORTRS) (see SORTRS)

- summary sort overflow 4-12
- summary tagalong sort (SORTRS)
 - (see also SORTRS)
 - definition G-4
- system data
 - alphanumeric 3-7
 - compare operations 3-6
 - how characters represented within
 - computer 3-7
 - interpreting 3-6
 - numeric data 3-8
 - signed numeric constants 3-8
 - zone and digit portions of characters 3-7
- system environment
 - effect on sort job run time 6-3
- system library
 - definition G-4
 - searched if source member not found on disk 5-8

- tagalong sort (SORTR)
 - (see also SORTR)
 - definition G-4
- time required to sort a file 5-3, 6-1, 6-4
 - factors that affect 6-1, 6-3
- to entry (columns 13-16-record type specifications) 3-10
 - factor 1 field identifier 3-10
- to entry (columns 24-27-record type specifications) 3-11
 - factor 2 field identifier 3-11
- translation table and alternate collating sequence coding sheet sample B-5
- types of sort 1-2

- unconditional force
 - definition G-4
 - example 4-15
 - explanation 4-5
- unconditionally forced character,
 - defining 4-13
- user library
 - action taken if omitted 5-4
 - definition G-4
 - loadable sort parameter list E-2, E-3
 - when required 5-4
- using duplicate control fields 4-4

- warning
 - issued when sequence numbers not in order 2-3
- work area, sort
 - in loadable sort parameter list E-2, E-3
- work file
 - automatic allocation 5-2
 - calculating size A-2
 - calculating storage A-2
 - definition G-4
 - disk positions effect on sort time 6-2
 - extendable disk file restriction 5-1
 - FILE statement for shared input file 5-7
 - how used 1-5
 - limitations 5-2
 - offline multivolume file restriction 5-1
 - programming considerations 5-1, 5-3
 - running your own procedure 5-7
 - SORT command effect 5-7**
- work record length
 - output type effect A-2
 - sort type effect A-2
- work record, how used by sort 1-5

- zone, portion of system character 3-7
- zoned numeric data 3-8
 - compare operations 3-7
 - number of bits in each digit 3-8
 - position of the sign code 3-9
 - positive and negative sign codes 3-8

Please use this form only to identify publication errors or to request changes in publications. Direct any requests for additional publications, technical questions about IBM systems, changes in IBM programming support, and so on, to your IBM representative or to your nearest IBM branch office.

If your comment does not need a reply (for example, pointing out a typing error) check this box and do not include your name and address below. If your comment is applicable, we will include it in the next revision of the manual.

If you would like a reply, check this box. Be sure to print your name and address below.

Page number(s):

Comment(s):

Please contact your nearest IBM branch office to request additional publications.

Name _____

Company or
Organization _____

Address _____

IBM may use and distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever. You may, of course, continue to use the information you supply.

No postage necessary if mailed in the U.S.A.

City

State

Zip Code

Cut Along Line

Fold and tape

Please do not staple

Fold and tape



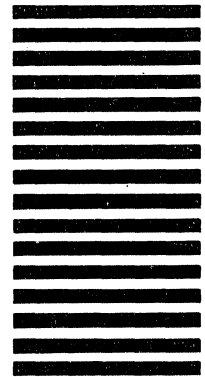
NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

POSTAGE WILL BE PAID BY ADDRESSEE:

IBM CORPORATION
General Systems Division
Development Laboratory
Publications, Dept. 245
Rochester, Minnesota 55901



Fold and tape

Please do not staple

Fold and tape





