IBM Application System/400

Sort
User's Guide and Reference

Version 3

SC09-1826-00

Note!

Before using this information and the product it supports, be sure to read the general information under “Notices” on page ix.

First Edition (September 1994)

This edition applies to the IBM AS/400 Operating System/400 licensed program (Program 5763-SS1), Version 3 Release 1, and to all subsequent releases and modifications until otherwise indicated in new editions. Make sure you are using the proper edition for the level of the product.

Order publications through your IBM representative or the IBM branch serving your locality. Publications are not stocked at the address given below.

A form for readers’ comments is provided at the back of this publication. If the form has been removed, you may address your comments to:

IBM Canada Ltd. Laboratory
Information Development
2G/345/1150/TOR
1150 Eglinton Avenue East,
North York, Ontario, Canada M3C1H7

You can also send your comments by facsimile (attention: RCF Coordinator), or you can send your comments electronically to IBM. See "Communicating your Comments to IBM" for a description of the methods. This page immediately precedes the Readers' Comment Form at the back of this publication.

When you send information to IBM, you grant IBM a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Contents

Notices ix
Trademarks and Service Marks ix

About This Manual xi
Who Should Use This Manual xi

Chapter 1. Introduction 1
General Description 1
How to Use Sort 2
Entering the Sort Specifications 3
Specifications Forms 3
Entering Specifications Interactively 4
Contents of the Source File 5
Types of Specifications 5
Sequencing the Specifications 6
The Purpose of Record and Field Specifications 6
Record Types 7
Selecting Control and Data Fields–Building the Output Record 7
Building the Control Field 8
The Data Description 8
Using FMTDTA to Process the Specifications 8
As a Line Command 9
From Screen Displays 9
What Happens when You Use the FMTDTA Command 9
Related Concepts 10
Files and Work Areas 10
Input Files 10
Output File 11
Work Area 12
Source File 12
Print File 12
Input and Output Record Format 12
Limitations on the Number of Records for Copy-Only Requests 13
Limitations on the Number of Records for Sort Requests 13
Tracking Any Problems 14
Message and Help Information 14
Displaying and Printing Messages 15
Correcting the Error 16
Control Language Command Help 16

Chapter 2. Sorting Records from a Single File 17
Examples 17
Example 1. Sorting a File with a Single Input Record Type 17
Entering the Specifications 18
Processing These Specifications 22
Example 2. Sorting Two Types of Records from the Input File 22
Entering the Specifications 23
Processing These Specifications 30
Example 3. Reformatting Records–1 30
Entering the Specifications 31

Processing These Specifications . . . 33
Example 4. Reformatting Records–2 . . . 33

Chapter 3. Sorting Records from Several Files . . . 35
Examples . . . 35
Example 1. Sorting Records from Five Input Files . . . 36
Entering the Specifications . . . 37
Processing These Specifications . . . 39

Chapter 4. Sorting Records from a Single File to Produce a Record Address File . . . 41
Automatic Updating . . . 42
Examples . . . 43
Example 1. Sorting an Inventory File by Item Class . . . 43
Entering the Specifications . . . 44
Processing These Specifications . . . 46

Chapter 5. Sorting Records from a File and Producing Totals . . . 47
Defining a Summary Set . . . 47
Handling Overflow . . . 48
Examples . . . 48
Example 1. Sorting a File and Summing Data in Certain Fields–One Record Type . . . 49
Entering the Specifications . . . 50
Processing These Specifications . . . 54
Example 2. Producing Summary Totals Based on Two Input Record Types 54
Entering the Specifications . . . 56
Processing These Specifications . . . 62
Example 3. Using an Overflow Field for Summary Data Overflow . . . 63
Entering the Specifications . . . 63
Processing These Specifications . . . 64
Example 4. Using an Overflow Indicator for Summary Data Overflow . . . 64
Entering the Specifications . . . 65
Processing These Specifications . . . 66

Chapter 6. Copying and Merging Records from Several Files into One . 67
Examples . . . 68
Example 1. Copying a File with No Reformatting . . . 68
Entering the Specifications . . . 68
Processing the Specifications . . . 70
Example 2. Copying/Merging Several Files to Produce a Reformatted Output File . . . 71
Entering the Specifications . . . 71
Processing These Specifications . . . 72

Chapter 7. The Format Data (FMTDTA) Command . . . 73
Accessing and Using the FMTDTA Command . . . 73
File Naming Conventions . . . 73
Syntax of the FMTDTA Command . . . 74
The FMTDTA Displays . . . 74
The Parameters of the FMTDTA Command . . . 75

Chapter 8. The Header Specification . . . 81
Summary of Column Entries . . . 82

Detailed Description of Each Column Entry 83
Columns 1 and 2 (Page) 83
Columns 3 through 5 (Statement Number) 83
Column 6 (Specification Identifier) 83
Columns 7 through 12 (Output File Type) 83
Columns 13 and 14 (Reserved) 83
Columns 15 through 17 (Maximum Control Field Length) 83
Copy-Only Request 83
Column 18 (Sort Sequence) 84
Columns 19 through 25 (Reserved) 84
Column 26 (Alternative Collating Sequence) 84
Column 27 (Reserved) 84
Column 28 (Include/Exclude Control Field in Output) 84
Columns 29 through 39 (Reserved) 85
Columns 40 through 72 (Comments) 85
Columns 73 through 80 (Program Identification) 85

Chapter 9. Record Specifications 87
Set of Record Specifications 87
Record Type 87
Summary of Column Entries 87
Detailed Description of Each Column Entry 89
Columns 1 and 2 (Page) and Columns 3 through 5 (Statement Number) . . 89
Column 6 (Specification Identifier) 89
Column 7 (Begin/Continue Record Definition) 89
Column 8 (Data Type) 91
Interpreting Alphanumeric Data 92
Interpreting Numeric Data 93
Interpreting the Sign of a Number 94
Columns 9 through 12 (Factor 1 Start Position) 95
Columns 13 through 16 (Factor 1 End Position) 95
Columns 17 and 18 (Comparison Operator) 95
Column 19 (Type of Data Contained in Factor 2) 96
Columns 20 through 23 (Factor 2 Start Position) 96
Columns 24 through 27 (Factor 2 End Position) 96
Columns 20 through 39 (Factor 2 Is a Constant) 96
Alphanumeric Constants (Column 8 Entry is C, Z, or D) 96
Numeric Constants (Column 8 Entry is P or U) 97
Columns 20 through 39 (Factor 2 Is a Keyword) 97
Columns 40 through 72 (Comments) 98

Chapter 10. Field Specifications 99
Order of Fields in the Output Record 99
Order of Field Specifications in the Source File 99
Summary of Column Entries 100
Detailed Description of Each Column Entry 101
Columns 1 and 2 (Page) and Columns 3 through 5 (Statement Number) 101
Column 6 (Specification Identifier) 101
Column 7 (Field Type) 101
Data Field (D in Column 7) 102
Normal and Opposite Control Fields (N or O in Column 7) 102
Forced Control Fields (F in Column 7) 102
Summary Data Field (S in Column 7) 104
Column 8 (Data Type) 104

Columns 7 and 8 (Details) . . . 107
Columns 9 through 12 (Field Location Start Position) . . . 107
Columns 13 through 16 (Field Location End Position) . . . 107
Column 17 (Conditionally Forced Character) . . . 108
Column 17 (Summary Overflow Indicator) . . . 108
Column 18 (Forced Character) . . . 108
Column 18 (Substitution Character) . . . 108
Column 19 (Continuation) . . . 108
Columns 20 through 22 (Overflow Field Length–Summary Sort) . . . 109
Column 20 (Alternative Collating Sequence by Field) . . . 109
Columns 23 through 39 (Reserved) . . . 109
Columns 40 through 72 (Comments) . . . 109

Chapter 11. The Comment Specification . . . 111
Summary of Column Entries . . . 111
Detailed Description of Each Column Entry . . . 111
Column 6 or 7 (Specification Identifier) . . . 111
Columns 8 through 72 (Comments) . . . 111

Appendix A. Information for System/36 Sort Utility Users Porting to the AS/400 System . . . 113
Running a System/36 Sort Utility Job on the AS/400 System . . . 113
Header Specification . . . 113
Record and Field Specifications . . . 113
System/36 Sort Utility Level Support . . . 113
Running a Sort Job in System/36 Sort Utility . . . 114

Appendix B. Information for System/38 Conversion Reformat Utility Users Porting to the AS/400 system . . . 115
Running a System/38 Conversion Reformat Utility Job in the AS/400 System 115
Header Specification . . . 115
Record and Field Specifications . . . 115
Running an AS/400 Sort Job in the System/38 environment or on a System/38 115

Appendix C. Standard and Alternative Collating Sequences . . . 117
Standard EBCDIC Collating Sequence . . . 117
Defining an Alternative Collating Sequence . . . 120
Specifying an Alternative Collating Sequence for the Entire Control Field . 121
Specifying an Alternative Collating Sequence on Parts of the Control Field 121
How to Code an ALTSEQ Specification . . . 122
Examples . . . 122
Example 1. Inserting a Special Character between Two Alphabetic Characters . . . 122
Example 2. Making Characters Equal . . . 123

Appendix D. Sorting Double-Byte Character Sets . . . 125
Coding Considerations for Double-Byte Character Sets . . . 125
Header Specifications . . . 126
Record Specifications . . . 127
Field Specifications . . . 128
Sorting Considerations for Japanese Double-Byte Character Sets . . . 129
Japanese Telephone Book Sorting Example . . . 131
Japanese Word Dictionary Sorting Example . . . 133
Kanji Character Dictionary Type Sorting Example . . . 133

Sorting Considerations for Simplified Chinese Double-Byte Character Sets . 135
Simplified Chinese Strokes/Radical Sorting Example 136
Sorting Considerations for Traditional Chinese Double-Byte Character Sets . 139
Traditional Chinese Strokes/Radical Sorting Example 140
Sorting Considerations for Korean Double-Byte Character Sets 142
Korean Pronunciation Sort Example 143

Bibliography 147

Glossary 149

Index 153

Notices

Any reference to an IBM* licensed program or other IBM product in this publication is not intended to state or imply that only IBM's program or other product may be used. Any functionally equivalent product, program, or service that does not infringe on any of IBM's intellectual property rights may be used instead of the IBM product, program, or service. Evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, is the user's responsibility.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, 208 Harbor Drive, Stamford, Connecticut, USA 06904-2501.

Changes or additions to the text are indicated by a vertical line (|) to the left of the addition or change.

This publication contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

Trademarks and Service Marks

The following terms, denoted by an asterisk (*), used in this publication, are trademarks of the IBM Corporation in the United States or other countries:

Application System/400	AS/400	IBM
Operating System/400	OS/400	400

About This Manual

This manual describes the Application System/400* (AS/400*) Sort utility, which is a function of the Operating System/400* (OS/400*) system. It contains a general description of the Sort utility, specific descriptions with examples, and detailed reference information. This book tells you how to use the Sort utility to select records from existing files, sort and reformat them, and then store the result. Topics covered also include copying records, merging records from several files, summing data in the selected records, and sorting double-byte character set (DBCS) data.

You may need to refer to other IBM manuals for more specific information about a particular topic. The *Publications Ordering*, SC41-3000 manual provides information on all the manuals in the AS/400 library.

For a list of related publications, see the “Bibliography” on page 147.

Who Should Use This Manual

This manual, the *Sort User's Guide and Reference*, SC09-1826, is written for programmers who wish to design and code Sort jobs, and for the system operators who may run these jobs.

Before you use this manual, you should be familiar with certain information:

- You should be familiar with your display station (also known as a work station) and its controls. Some elements of its display and certain keys on the keyboard are standard regardless of the software system currently running at the display station or the hardware system the display station is connected to. Some of these keys are:
  - Cursor movement keys
  - Command keys
  - Field exit keys
  - Insert and delete keys
  - The Error Reset key.

  This information is contained in the *System Operation for New Users*, SC41-3200.
- You should know how to operate your display station when it is connected to the IBM AS/400 system and running AS/400 software. This means knowing about the OS/400 system and the Control Language (CL) to perform the tasks of:
  - Sign on and sign off of the AS/400 system
  - Interact with displays
  - Use Help
  - Enter control commands
  - Call utilities
  - Respond to messages.

  To find out more about control language, refer to these IBM AS/400 publications:
  - *CL Programming*, SC41-3721
  - *CL Reference*, SC41-3722

If you are new to the AS/400 system, and have no knowledge of how it works, refer to the *System Operation for New Users*, SC41-3200.

Chapter 1. Introduction

This chapter contains a general description of the Sort utility and of what you should know when you use it. It covers:

- The functions of the Sort utility
- How to use the Sort utility
- How to enter the Sort specifications
- The contents of the source file in which you enter the specifications
- How to process the specifications
- Related concepts for using the Sort utility
- How to know if your sort program runs successfully.

The Sort utility provides a large range of sorting and reformatting possibilities; to copy and merge files; or to add information, sum data, and reference existing records in other files.

General Description

Use the Sort utility to sort and reformat records. You work with Control Language (CL) commands to enter and then process the appropriate instructions. With Sort, you can:

- ***Sort records from one file*** (called the input file) ***and store the result in a single physical output file***. This output file is called a physical file because it contains the actual records that were sorted. You can also divide the input records into fields, and reformat (select and rearrange) those fields in the output records.

  See Chapter 2, “Sorting Records from a Single File,” for examples.

- ***Sort records from several input files and store the result in a single physical output file***. You can also divide the input records into fields and reformat them in the output records.

  See Chapter 3, “Sorting Records from Several Files,” for examples.

- ***Sort records from a single input file and store the result in a single record address (RA) output file***. This output file does not contain the actual records that were sorted; rather, it contains a sorted list of addresses of the records in the original input file. This has the advantage of keeping the sorted result up to date for changes to previously existing records. However, if you add or delete records from the input file, the RA file does not have the addresses for the new records, and will still contain the addresses of the old records.

  See Chapter 4, “Sorting Records from a Single File to Produce a Record Address File,” for examples.

- ***Produce an output file containing totals***, rather than sorted or reformatted records.

  See Chapter 5, “Sorting Records from a File and Producing Totals,” for examples.

- ***Copy and merge*** records from several files into one file.

  See Chapter 6, “Copying and Merging Records from Several Files into One,” for examples.

How to Use Sort

Perform the following steps:

1. Fill in the appropriate Sort specifications manually on preprinted forms. (This step is optional. You can begin directly with step 2.)
2. Use the Source Entry Utility (SEU) to enter the specifications into a source file.
3. Use the Format Data (FMTDTA) command to process the specifications. You do this by entering the CL command, `FMTDTA`, followed by the appropriate options.

Figure 1 shows these steps and also gives the order of the specifications in the source file. Figure 2 on page 3 shows what happens when you use the `FMTDTA` command to process a source file.

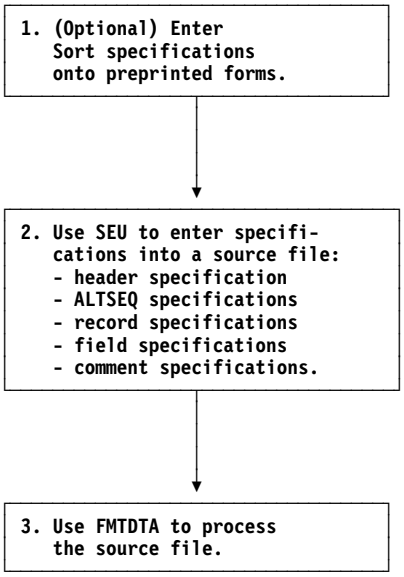


Figure 1. Steps in Using Sort

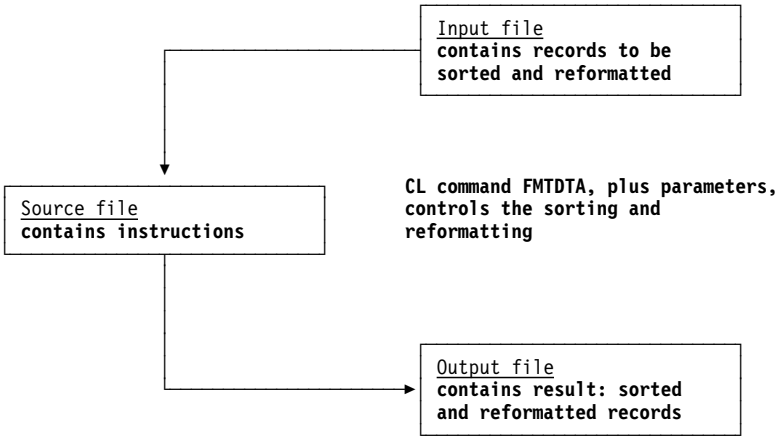


Figure 2. The Sort Process: What Happens in Step 3 of Figure 1-1

Entering the Sort Specifications

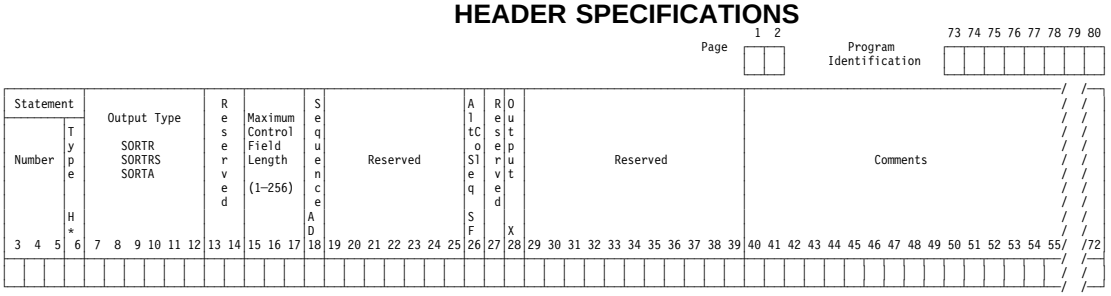
There are two ways to enter Sort specifications:

- Complete the appropriate specifications forms and submit them to be keyed into the system.
- Use the Source Entry Utility to enter the specifications directly.

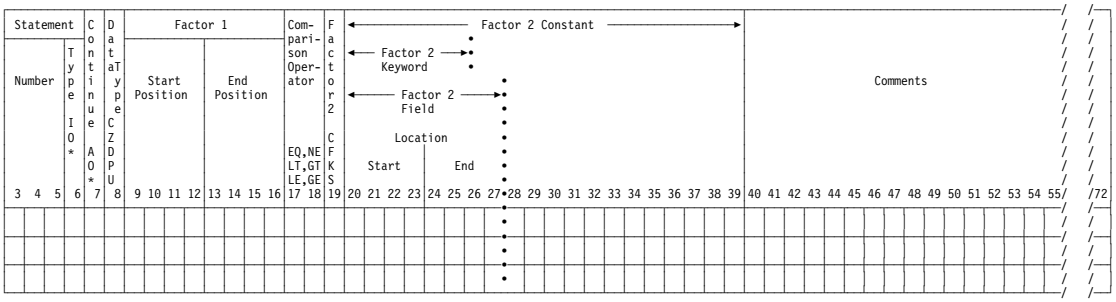
Specifications Forms

The three Sort specifications forms are shown in Figure 3 on page 4 and are followed by a description of how to complete them.

HEADER SPECIFICATIONS



RECORD SPECIFICATIONS



FIELD SPECIFICATIONS

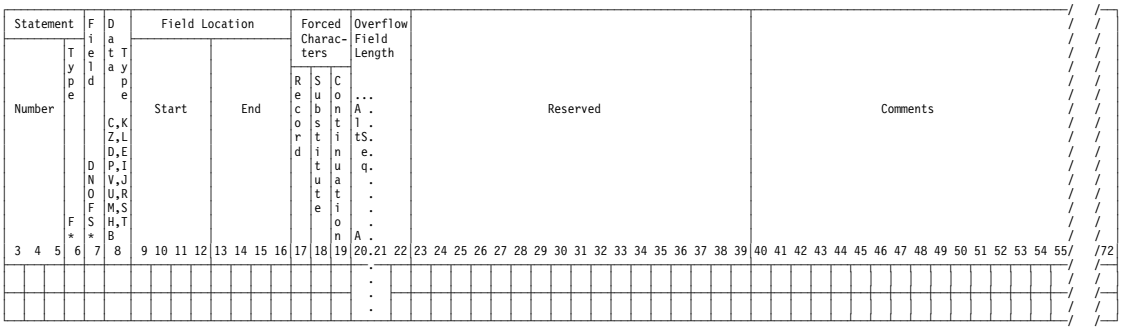


Figure 3. Sort Specification Forms

When completing the fields in these specifications, all entries must be right-justified in the spaces provided, unless otherwise indicated in the description. If you enter anything that is not right-justified, the system pads your entry with zeros. Entering a 1 that is not right-justified, for example, could be interpreted as a 10 or a 100, depending on where you place it in the field.

Entering Specifications Interactively

Use the Source Entry Utility (SEU) to enter source statements. Enter the specifications for a particular sort job into a member of a source file with SEU.

SEU is accessed through the CL command `STRSEU`. The *ADTS/400: Source Entry Utility*, SC09-1774 describes how to use the Source Entry Utility.

Contents of the Source File

The sort specifications you enter are stored in a member of a source file. The types of specifications and the rules regarding the sequence in which they must appear are described below.

Types of Specifications

There are five types of Sort specifications you can use to describe what you want sorted and how:

- The ***header specification*** describes general information such as output file type and collating sequence used. There must be only one for each sort job. Chapter 8, “The Header Specification,” describes all the possible entries.
- The ***ALTSEQ specifications*** are optional. If you want a collating sequence other than the standard one, use these specifications to describe your alternative collating sequence. Appendix C, “Standard and Alternative Collating Sequences” describes the standard EBCDIC collating sequences and how to code `ALTSEQ` specifications.
- The ***record specifications*** identify the input records you want to include or exclude from the sort. Chapter 9, “Record Specifications,” describes all the possible entries.
- The ***field specifications*** identify the control fields and the data fields on the input record that you want included in the output record. Chapter 10, “Field Specifications,” describes all the possible entries.
- The ***comment specification*** allows you to place comment entries throughout your sort program to make its purpose and process clearer for yourself and other users. There is no special form for a comment specification: entering an asterisk into column 6 or 7 of any of the specifications automatically identifies the specification as a comment. Chapter 11, “The Comment Specification,” provides details.

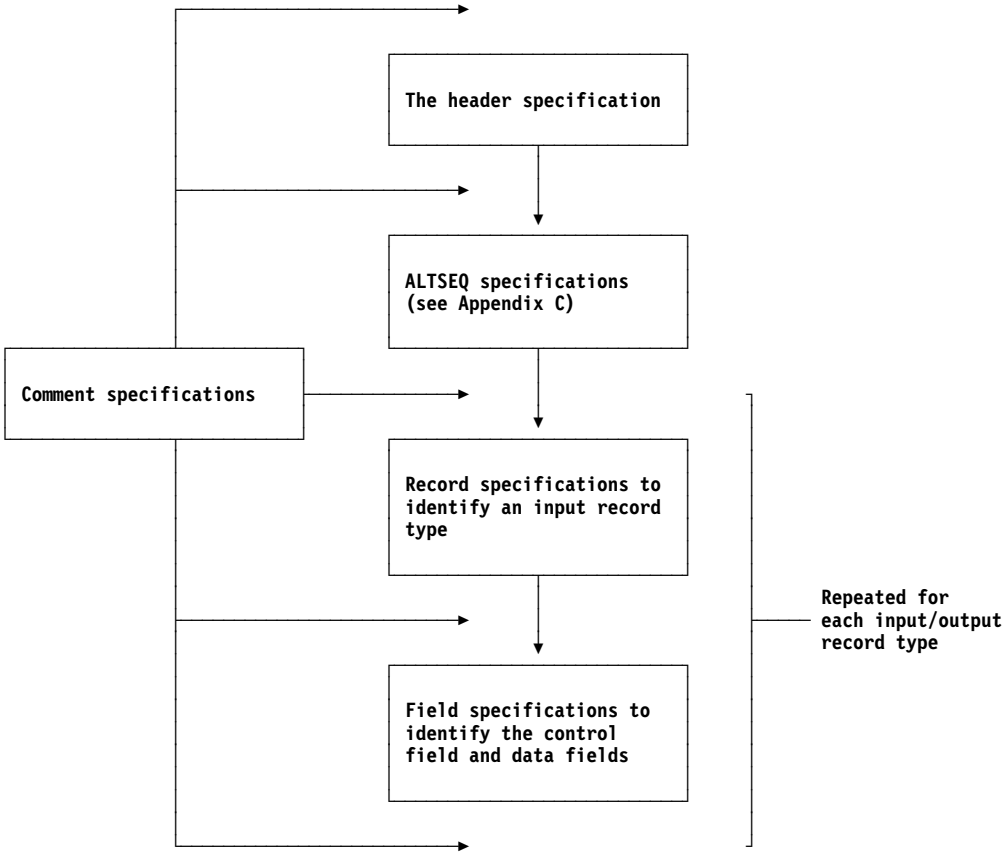


Figure 4. Order in Which You Should Store Sort Specifications in the Source File. You may insert comment specifications where they are most useful.

Sequencing the Specifications

The Sort utility orders the specifications by occurrence in the source member. This order, shown in Figure 4, is:

1. The header specification.
2. ALTSEQ specifications, if any. (These specifications are optional.)
3. One or more record specifications.
4. One or more field specifications for each record specification. Follow the record specification by the field specifications that pertain to it.

Repeat steps 3 and 4 for each input/output record type desired.

The Purpose of Record and Field Specifications

Together, record and field specifications determine the records to be sorted, the sorting order, and the information that will be written to the output file.

If all input records are to be sorted in the same way, no record specifications need to be entered: the field specifications are entered by themselves and apply to all input records. A sort program could thus consist of only two specifications: a

header specification, and a field specification to define the control field. When run, all input records would be sorted according to the data in the control field.

Record Types

Record specifications define which records are to be selected from an input file. At least one record specification (each grouping is called a ***set***) is needed to define a type of record.

Entering the first of a series of record specifications signals the start of a new record type. Any difference in the way an input record is to be selected or an output record is to be organized is considered a new record type, and requires a new set of record specifications. There are two main differences:

- **Input record type.** Differences in the criteria by which input records are to be selected. You enter a set of record specifications to identify each set of criteria.
- **Output record type.** Differences in the field specifications for an input record type. You must precede each set of field specifications by the appropriate record specifications, even if those record specifications are the same as the set previously entered.

Selecting Control and Data Fields–Building the Output Record

Each set of record specifications is followed by a series of field specifications. The field specifications select the fields from the input records and specify if and how they are to be changed. These input fields can either be data fields, or can become part of the control field. One or more field specifications may be needed for each field.

The order of the field specifications in the source file determines the order of the fields in the output records. For example, to move a field from the end of the input record to the beginning of the output record, place its field specification first in the source file, before any other. This way, you can select any portion of the input record, change it, and place its data anywhere in the output record.

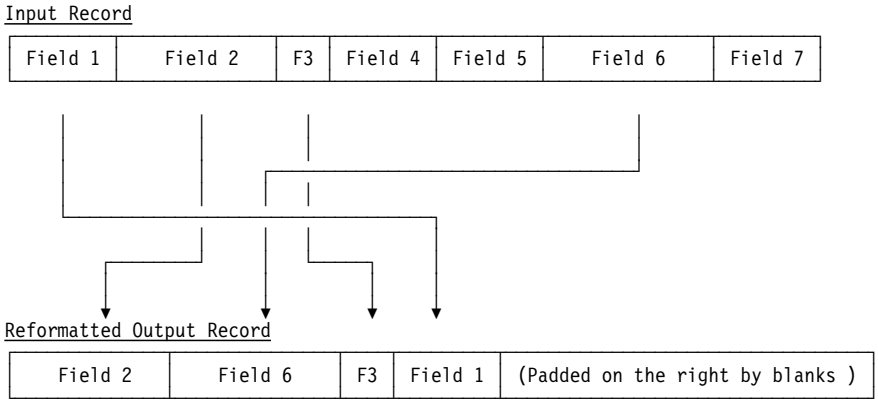


Figure 5. An Example of How Sort Builds an Output Record. The output record is padded on the right by blanks to make its length equal to the output file.

Building the Control Field

The control field is specified like any other field; however, its contents control the sorting. For example, to sort a series of input records in ascending order, Sort orders them by the contents of their control fields so that records whose control field contents begin with an `'A'` would precede those whose control field contents begin with a `'B'`.

Construct one control field for each sort job. For different input record types, however, you can combine different elements from the input record into the one control field. (The next section, "Using FMTDTA to Process the Specifications," describes how the Sort utility uses the control field to sort input records.)

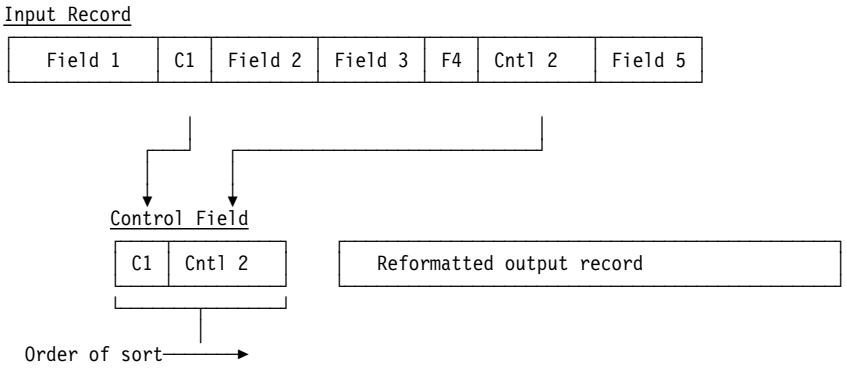


Figure 6. An Example of How Sort Builds a Control Field. The control field is then used as a key to sort the reformatted output record.

The Data Description

Field specifications allow you to completely restructure input records and build output records, and sort them in any way you choose. To enable you to do this, the Sort utility ignores the data description already associated with the files it processes. It treats both input and output files as unformatted strings of bytes; the length of the string is determined by the record length.

As part of the sort specifications, therefore, you must define the input fields and the output fields, and specify their start and end positions. Sort then divides the data in the files according to these specifications, and sorts and reformats the input records accordingly.

Ensure that the lengths of the fields you specify match the data in the file. For example, if positions 5 to 10 contain a date and positions 11 to 32 contain a name, the start and end positions of your fields will probably be 5, 10, 11, and 32.

Using FMTDTA to Process the Specifications

Once the specifications for a particular job have been entered in a source file, use the `FMTDTA` command and its parameters to run Sort. Refer to Chapter 7, "The Format Data (FMTDTA) Command," for a complete description of how to use this command.

Note: Running a sort job never changes your input files unless you specify the input file as the output file.

As a Line Command

Enter the command followed by the name of the file containing the records to be sorted (input file), and the name of the file to contain the sorted records (output file). For example:

```
FMTDTA INFILE((FILE1)) OUTFILE(FILE2)
```

where FILE1 is the input file, and FILE2 is the output file.

Because no source file is identified here, the system assumes the source specifications are stored in the first member of the default file, QFMTSRC, found in the library list. Because no libraries or members have been specified for input or output files, the system automatically defaults to the current library and the first member of each file. If you want to specify another library or member, enter the name of each in the following way:

```
FMTDTA INFILE((LIBRAR1/FILE1)) OUTFILE(FILE2 MEMB3)
          SRCFILE(LIBRAR2/FILE3)
```

Here, the input records are taken from the first member of file FILE1 in library LIBRAR1, and the output records are put into member MEMB3 of file FILE2 in the current library. The source specifications are taken from the first member in file FILE3 in library LIBRAR2.

To specify a member in the source file, add another parameter to the FMTDTA command. For example, to specify member MEM2 in the source file, enter:

```
FMTDTA INFILE((LIBRAR1/FILE1)) OUTFILE(FILE2 MEMB3)
          SRCFILE(LIBRAR2/FILE3) SRCMBR(MEM2)
```

From Screen Displays

To enter FMTDTA and its parameters from a prompt display, change the values on the FMTDTA screen that apply to your particular case and press Enter to run the command. Chapter 7, “The Format Data (FMTDTA) Command,” describes this method.

What Happens when You Use the FMTDTA Command

The following section provides a general description of processing:

1. ***Select records***. The FMTDTA command first processes the information in the header specification, and copies each input file into a work area in main storage. Only those records that are identified by the selection criteria on the record specifications will be copied.

   Be aware that once the records matching one record type are selected, they are no longer accessible for further selection. Thus, the number of input records available for processing decreases with the specification of each new record type. This method of selection can be regarded as a limitation, but it also means that, by arranging your record types carefully, you can maximize the sort process by entering a minimum number of specifications.

2. ***Reformat records***. The utility then restructures each input record according to the data field specifications for the record type. It takes sections of the input record, then changes and repositions them to create the output record.
3. ***Putting together the control field***. For each record type, you can specify one or more elements (from the input record) to make up the entire control field. The total length of all these elements must not exceed the length entered in columns 15 through 17 of the header specification.
4. ***Sort records***. The output records, stored in their own separate work area, are sorted according to the collating sequence specified and according to the contents of the control field.

   The control field is used as a key. Its size is determined by the entry in columns 15 through 17 of the header specification. The data that is placed in this control field during the reformatting process is determined by the control field specifications, and can be formatted differently for each record type.
5. ***Store records***. The result is stored in a physical file.

If no record or field specifications are entered, FMTDTA does no sorting or reformatting, and the process becomes one of copying and merging records from one or more input files into an output file.

The Sort utility does not require unique control fields. If duplicate control field values are found in the input records, those records will appear in a group in the output file, in the same order that they were found in the input file.

Related Concepts

The following concepts are useful in understanding how the Sort utility works.

Files and Work Areas

Sort can work with files on diskettes, disks, and databases. The following files and work areas are used:

- ***Up to 8 input files***. The input files can be any combination of physical, logical, or diskette unit files.
- ***One source file***. This file contains the specification statements.
- ***Work area***. A work area is automatically built and deleted by the Sort utility during the processing of sort requests. It is not needed or created for copy-only requests.
- ***One output file***. The output file is an existing physical file.
- ***One print file***. This file contains a listing of the source file with any errors and/or messages.

Input Files

The input files used by the Sort utility must follow these rules:

- Normal AS/400 security is in effect for all input files. The user using the utility must have use and read authority for the input files.
- The Sort utility defaults to a shared-read lock on the input file. If you want to restrict other users' access to the input file while the utility is running, you must override the locked state with the CL command Allocate Object (ALCOBJ). You

must issue this command before FMTDTA. The Deallocate Object (DLCOBJ) command can be used to return files to their original locked state.

- You must provide the name of the input file in the FMTDTA command, as described in Chapter 7, "The Format Data (FMTDTA) Command." The utility can be directed to a different file with a suitable override command.
- All input files are opened for sequential processing only. This means that for database files:
  - physical files or nonkeyed logical file records are read in arrival sequence
  - keyed logical file records are read in key sequence.
- Multiple input files are processed in the order specified in the FMTDTA command.
- All input record lengths should be equal. If they are different lengths, the shorter records may be padded on the right with data from previous longer records.
- If a device file is specified as the input file, the record length defined for the input records should not exceed the physical limitations of the device specified. If the record length does exceed the physical limitation of the specified device, undesirable results can occur.
- The record format information given in the specifications is not checked against the system data description associated with each input file, except for record length considerations.
- The Sort utility can process, as input files, both regular input files and other source files (those with the file type *SRC). Be aware that data management adds 12 bytes of sequence information and data to the front of every record in a source file. Thus, when using such a file as input, the actual data you may want to sort begins at byte 13.

Output File

The following rules apply to the file containing the output produced by the Sort utility:

- You must create the output file and member before running the Sort utility. The utility checks for the presence of this file and member; if neither exists, an error message is issued.

  Note: The output file in the database can have either a keyed or an arrival sequence access path. Depending on the specified application, a keyed access path to the output file may be redundant, and can cause the Sort utility to run more slowly.
- The sum of the field lengths specified for each record in the output file should be consistent with the record length of the output file. If the reformatted record length is less than the output file record length, the file will contain left-justified data and will be padded on the right with blanks. If the reformatted record length is greater than the output file record length, the output record will be truncated (on the right) to the file record length.
- The data description of the output file, though not used by the Sort utility, becomes the data management view of the output data. The string of data constructed by the utility should correspond to this associated data description.
- Normal AS/400 security is enforced for the output file. You must have the proper authority to clear and write into the file and member.

- The Sort utility defaults to a shared-update lock on the output file. If you want to restrict other users' access to the output file while the utility is running, you must override the locked state with the ALCOBJ command. This command must be issued before the FMTDTA. The DLCOBJ command can be used to return the output file to its prior locked state.
- The names of the output file and member are specified by FMTDTA. The utility opens this file name and member, although you can use an override command to redirect the output to a different file.

  Note: The same physical database file and member name can be specified for both the output file and one of the input files. Here, the output data is written over the input data. The Sort utility clears the output file member so that it contains no records before output is written to the file (except for copy-only requests). If a major failure occurs between clearing the file member and writing the resequenced data records to the file member, your input data (and your sorted data) will be lost. Thus, make sure that this file member has adequate backup.

Work Area

Sort internally creates a work area to accommodate the input records and codes generated during the sorting process. The work area resides in a temporary work library, and is only accessible to the program that is being run. The area is automatically deleted at the end of the job.

Copying the input records into a work area and then working with only the copied records allows several programs to use the same input files at the same time. This is also the reason the input files are not altered by the Sort utility.

The work area is not created for copy-only requests.

Source File

The sort specifications are considered to be source input, and must therefore come from a source file (either a database or a diskette unit file). The name of the source file from which the sort specification set comes is specified as a parameter to the FMTDTA command. If the specified name is not a source file, an error condition occurs. You can use an override command to redirect the utility to another source file.

Print File

If the *PRT or *DUMP options of FMTDTA are selected, a print file can be identified to store information about the sort job. All the print data is directed to this file. The file that is specified must already exist before running Sort; otherwise, an error condition occurs.

If a print file is not specified, print output is sent to QSYSPRT, the system print file by default.

Input and Output Record Format

Any file that is used as input or output by the Sort utility can have a detailed record format description associated with it. This description specifies detailed field-level information, which is used by data management in performing field-level manipulations. This information is not used by the Sort utility, and does not affect the processing of the sort specifications.

The Sort utility receives a string of bytes as input from data management, and sends a string of bytes back to data management for placement in the output file. The field-level view used for processing by the Sort utility is defined by start and end positions in the specifications.

This field-level view is used only by the Sort utility, and does not have to match the one used by data management. However, inconsistencies can occur if the two views are not compatible; for example, if only half of a data field is selected by the Sort utility, only half of the original field will appear in the output record. Also, if a field that was originally defined to data management as a *character field* is specified by the Sort utility as a *numeric field*, the output may not be as you expected.

Limitations on the Number of Records for Copy-Only Requests

The Sort utility can copy an unlimited number of records, provided enough storage is available.

Although the Sort utility does generate tables during a copy-only request, the storage these tables occupy is insignificant in comparison to the amount of auxiliary storage available for the output file. To determine the approximate number of records you can copy, subtract 100 000 bytes from the amount of auxiliary storage you have available for your output file, and divide the remainder by the length of the records you are copying.

Limitations on the Number of Records for Sort Requests

The number of records that can be processed by a sort (as opposed to a copy-only) request is also restricted by the amount of auxiliary storage available. For sort requests, this amount must be sufficient to hold both the output file and the work area constructed during sorting.

1. ***Determine the length of the work records*** that occupy the work area for a sort request by adding together the following:
   - The length of the output record
   - The total length of the control field elements that the Sort utility uses
   - The value 4 for the 4-byte binary relative record number that the utility supplies for each work record.

   For sort requests that produce record address (RA) files as output, only the last two items in this list are included in the record that is transferred to the work area.
2. ***Determine the number of records in the work area.***

   The number of records in the work area is equal to the number of output records in the output file. Both are determined by the number of records selected (using either select or omit record specifications) from the input file. If no select or omit specifications exist, all the input records are present in the work area and in the output file. (For RA output, the actual data is not present; instead, there is a record address entry for each input record.)
3. ***Multiply the work record length by the number of work records*** in the work area to determine how much work area storage is needed for the work records.

4. ***To this, add two million bytes*** to determine the total work area space needed by the Sort utility.

   For example, if you have determined that there are 5 000 work records that are each 150 bytes long, the total storage required for the work records is 750 000 bytes. The work area would then require 2 750 000 bytes of storage (750 000 plus 2 million). If you determine that there are 30 000 work records that are each 500 bytes long, the storage required for the work records is 15 million bytes. The utility would then require 17 million bytes of storage.

5. ***Add the storage required for the output file*** to determine the total amount of storage required to resequence the specified number of input records.

Tracking Any Problems

When the processing of a sort job ends, the system sets a return code. If the job was successful, the return code is set to 0; if it was not successful, the code is set to 2.

Use this value to check the status of a sort job and to direct the system to perform certain tasks based on the result. To access the return code value, use the CL command RTVJOBA and its appropriate parameters. The *CL Reference* manual contains information about this command.

Message and Help Information

When you work with the Sort utility, the system responds to your commands by processing them and also by providing messages. These messages are displayed on your screen or sent to a printer.

Each message contains a minimum of three parts.

```
  5763SS1 V3R1M0  910329           IBM SORT MESSAGE(S)


     LINE
    00001  MSGID:  FMT1080  SEVERITY:  30  A
 B Message . . . . :   Maximum-control-field-length entry is greater
             than 256.
 C  Cause . . . . . :   Columns 13-17 of the header statement contains
             numeric entry greater than 256.
           Recovery  . . . :   Choose the correct entry.  Resubmit the
             request.

    00005  MSGID:  FMT6010  SEVERITY:  00
           Message . . . . :   1 error(s) found in source file  in library  mbr
           Recovery  . . . :   Correct any errors. Resubmit the request.

    0 errors found in source file QFMTSRC in library USELIB mbr SUMMARY.
    16 records read and 16 records selected from TESTS in library USELIB mbr
      TESTS.
    16 total records read, and 16 total records selected.
    1 records placed in output file TESTOUT in library USELIB mbr TESTSOUT.
    REQUEST SUCCESSFUL 12/25/91 08:40:40
=>
```

Figure 7. A Sample Screen Display of Sort Messages

A A number indicating the seriousness of the error.

Severity	Meaning
00	***Informational:*** The message is for your information only; no error has been detected and no reply is necessary. It suggests that a function is either in progress or reached a normal, successful completion.
10	***Warning:*** A potential error condition exists. A function may have been only partially completed; for example, some items in a list may have succeeded while others have failed. Or, the program may have taken a default to fill in missing input, and the default chosen may not have been valid. Thus, the results of the operation may be different from what you intended, and the program cannot determine if they are valid.
20	***Error:*** A definite error has been detected, but it is one for which automatic recovery procedures have probably been applied and for which processing has continued. A default may have been used to replace erroneous input. The results of the operation may not be valid.
30	***Severe error:*** The error detected is too serious for automatic recovery, and no defaults are possible. If the error was in source data, the entire input record was skipped. If it occurred while the program was running, it will lead to an *abnormal end of program or function* (severity code 40) condition. The results of the operation are not valid.
40	***Abnormal end of program or function:*** The running of the program has actually stopped, either because it was unable to handle invalid data or because the user has canceled it.

B The text you see online or on a listing, giving a brief (generally one-sentence) description of the problem.

C The text you see online when you press Help from the screen with the message displayed. It will be printed on your listing if you specify `*SECLVL` in your run-time options. (The default for this option is `*NOSECLVL`.) This text contains an expanded description of the message, and a section detailing the correct user response.

Displaying and Printing Messages

To display or print a particular message or messages, use the `DSPMSGD` command. This command is described in the *CL Reference* manual. The library name for your product is `QSYS`, and the file names are `QFMTMSG` and `QFMTTXT`.

Error messages fall into two basic categories:

- *Error messages,* to inform you of an error situation as the system was running your command
- *Informational messages,* to provide you with information about the processing and/or its result.

When a message is displayed, one or more of the following can occur:

- The message is displayed (called “first-level text”)

- A blinking cursor is repositioned at the input field in error
- The input field in error is highlighted in reverse image
- The keyboard is locked; that is, pressing any keyboard key has no result
- An alarm (beep) sounds.

Correcting the Error

If you know how to correct the error from the information provided in the error message, press the Error Reset key, enter the corrected information, and continue your work.

If the error message is not sufficient for you to decide how to handle the error, press the Help key. This displays additional information about the error, and describes how to correct it.

Note: Running a sort job can cause messages that are not part of the Sort utility to be displayed. In such cases, press Help to obtain further information, and then refer to the appropriate user's manual, if necessary.

Control Language Command Help

For more information about a command, use one of the following procedures:

- Type the command, with or without parameters, on any command line and press F4 (Prompt). The prompt display for that command is displayed. From here you can press Help for more information about the command and its parameters.
- Choose a menu selection that prompts you for parameter values before a command is run, and press Help on the prompt display.

Chapter 2. Sorting Records from a Single File

The Sort utility can sort records selected from a single file and produce a copy of the records that reflects the new order (see Figure 8). It can also reformat (select, change, or rearrange) fields within each record.

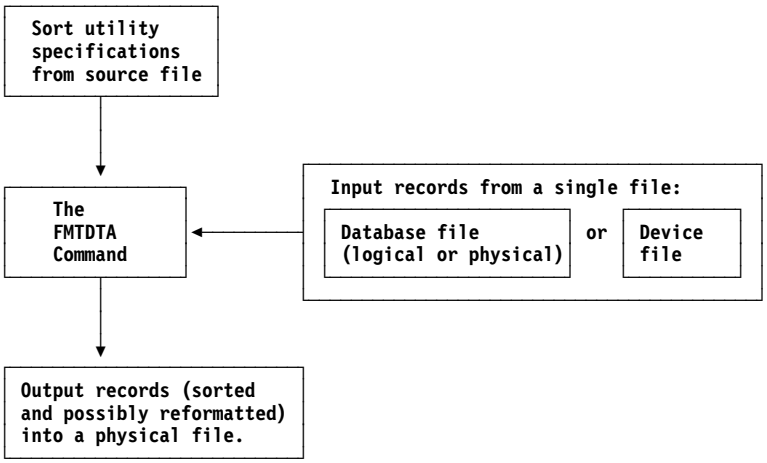


Figure 8. Sorting and Reformatting a Single File into One Physical File

The output is a physical file in the database, meaning that it contains a sorted and reformatted copy of the actual input records.

When sorting a single file to produce a physical output file, note the following:

- The output type on the header specification must be `SORTR`.
- The maximum control field length must be greater than zero.
- Field specifications must be entered to define the control field.
- You need not enter a separate field specification for each input field you want to appear in the output record. If there are to be no changes in their order or content, you can enter just one field specification that treats the entire input record as a single output field.
- The `INFILE` parameter of the `FMTDTA` command has only one value, because only one input file is being specified.

Examples

The following examples extract records from a single input file.

Example 1. Sorting a File with a Single Input Record Type

This example takes the records from a file (`INVENTRY`) and puts the sorted output records into a physical file (`NEWORD`). The fields that are identified by the sort specifications correspond to those that were originally defined for the records.

The Sort utility looks for only one type of input record. Records are selected by comparing the `INSTOCK` field (columns 69 through 74) with the `REORDER` field (columns 40 through 45) of the input record. If the `INSTOCK` field is less than the `REORDER` field, the record is selected for sorting and included in the output file.

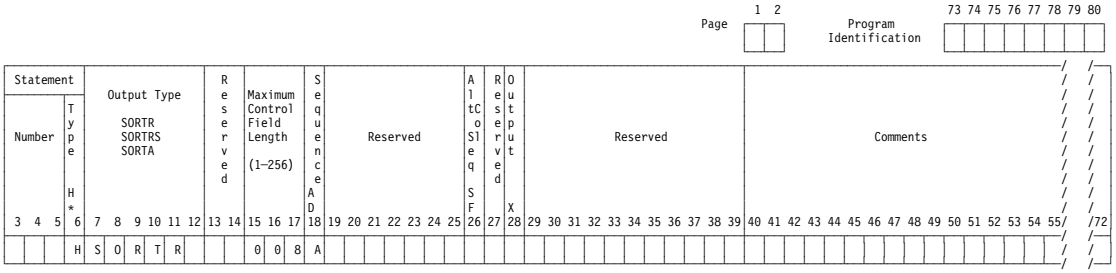
The records are sorted in ascending order based on the control field. The control field is composed of the input fields TYPENO (columns 1 and 2) and ITEMNO (columns 3 through 8).

The output records are to contain six rearranged data fields, one of which is the control field. Note that once the control field is used for sorting, if it is to be included in the output record it is treated as a data field.

Entering the Specifications

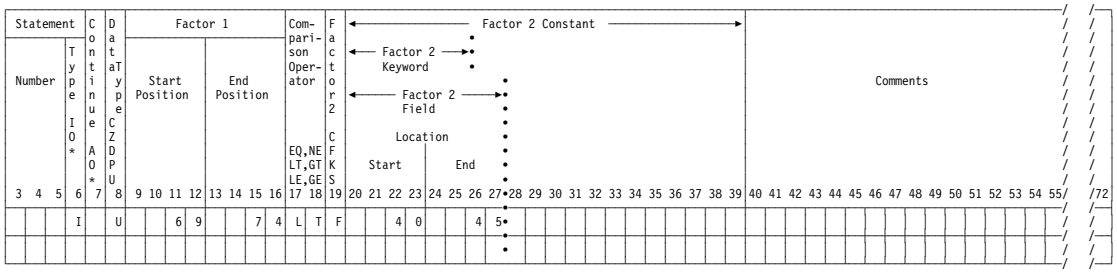
Enter the following header, record, and field specifications. (Only those columns that are specific to this example are described below.)

HEADER SPECIFICATIONS



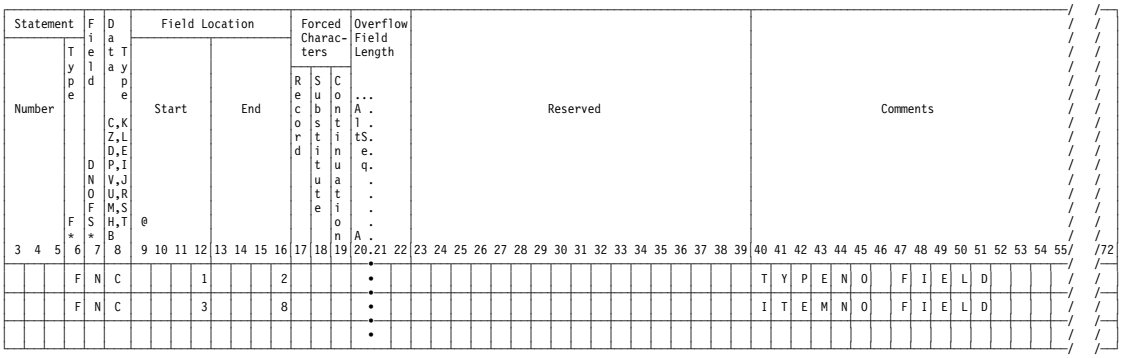
Columns	Entry	Explanation
7-12	SORTR	This is a regular sort; the output is to be a physical file.
15-17	8	The maximum control field length is 8 characters.
18	A	The output records are to be sorted in ascending order (A before B, 1 before 2, and so on). For the ordering of numbers relative to upper- and lowercase characters and special characters, refer to the collating sequences in Appendix C, “Standard and Alternative Collating Sequences.”
26	No entry	The Sort utility defaults to the standard collating sequence.
28	No entry	The default is to include the control field in the output record.

RECORD SPECIFICATIONS



Columns	Entry	Explanation
6	I	The input records that meet the conditions specified are to be included in the output.
7	No entry	The first of the set of record specifications that is to define this type of record.
8	U	The data in the Factor 1 and Factor 2 fields of each input record is to be interpreted as numeric, and in unpacked format. This means that each digit will be represented by 8 bits. The lower 4 bits (digit portion) identify the digit, while the higher 4 bits (zone portion) represent the sign (plus or minus) for the entire number. Note that by specifying this type of data, you are limiting the length of the Factor 1 and Factor 2 fields to 192 bits (l6 digits) or less.
9-12	69	The Factor 1 field begins in column 69 of the input record.
13-16	74	The Factor 1 field ends in column 74 of the input record. This selects the data from the INSTOCK field, corresponding to the original data definition for the input records.
17-18	LT	The data in the Factor 1 field must be less than the data in the Factor 2 field for the input record to be selected for sorting. The comparison is to be based on the collating sequence specified in the header specification, which here is the standard collating sequence.
19	F	The data in the Factor 2 field is to be the data taken from the input record field, as specified in columns 20 through 27.
20-23	40	The Factor 2 field begins in column 40 of the input record.
24-27	45	The Factor 2 field ends in column 45 of the input record. This selects the data from the REORDER field, corresponding to the original data definition for the input records.

FIELD SPECIFICATIONS



These two specifications establish the two elements of the control field. The contents of the first specification are:

Columns	Entry	Explanation
7	N	This element is to be treated as normal. This means that the unchanged data from the input record is to be used, and that the sort sequence for this element (or part of the control field) is that specified in the header specification (ascending).
8	C	The system looks at the data in this field as character data. Each group of 8 bits is assumed to represent a character, which can be a letter of the alphabet, a symbol, or a digit.
9-12	1	This element of the control field is to begin in column 1 of the input record.
13-16	2	The element is to end in column 2 of the input record. This selects the data from the TYPENO field, corresponding to the original data definition for the input records.
40-72	Comments	Identifies the name of the field selected.

The second field specification defines the second element of the control field, called ITEMNO in the input record. Sort regards both elements as a single string of control characters, and begins the sort with the character farthest to the left.

FIELD SPECIFICATIONS

Statement Number	Type	Field	Data Type	Field Location Start	Field Location End	Forced Characters Record	Forced Characters Substitute	Forced Characters Continuation	Overflow Field Length	Reserved	Comments
3 4 5	F* 6	DNOFS* 7	C,KZ,LD,EP,IV,JU,RM,SH,TB 8	9 10 11 12	13 14 15 16	17	18	19	...A.l.tS.e.q. . . . A. 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 / /72
	F	D	C	3 0	3 4				•		COST FIELD
	F	D	C	2 8	2 9				•		UNIT FIELD
	F	D	C	4 6	5 0				•		QTY FIELD
	F	D	C	8 6	9 1				•		DATE FIELD
	F	D	C	9 5	1 0 0				•		ORDERNO FIELD
									•		

These five specifications define five data fields to be taken from each input record. The contents of the first specification are:

Columns	Entry	Explanation
7	D	The field is to be treated as a data field, meaning that the unchanged data from the input record is to be transferred to the output record.
8	C	The system looks at the data in this field as character data. Each group of 8 bits is assumed to represent a character, which can be a letter of the alphabet, a symbol, or a digit.
9-12	30	This data field begins in column 30 of the input record.
13-16	34	This control field ends in column 34 of the input record. This selects the data from the COST field, corresponding to the original data definition for the input records.
40-72	Comments	Identifies the name of the field selected as a data field from the input record.

The field specification is followed by four more specifications, each defining a new data field (UNIT, QTY, DATE, and ORDERNO) corresponding to the original data description for the input records. Notice the order in which these fields are entered into the source file, represented by the number in columns 3 through 5. This is the order in which they will be put into the output record, **not** the order in which they occur in the input record.

Processing These Specifications

An example of the FMTDTA command used to process these statements is given below:

Specified on FMTDTA Command	Explanation
FMTDTA INFILE((PROD/INVENTRY MONDAY))+	The input records are to be taken from member MONDAY of database file INVENTRY in library PROD.
OUTFILE(PROD/NEWORD MONDAY)+	The output records are to be put into member MONDAY of file NEWORD in library PROD.

The specifications are automatically taken from the first member of file QFMTSRC *(default source file) in the library list.*

The OPTION parameter defaults to *CHK, *PR , *NODUMP, and *NOSECLVL.

The plus sign (+) shown for continuations is not required if the FMTDTA *command is entered from a display station.*

Example 2. Sorting Two Types of Records from the Input File

Example 2 also takes records from a single file and puts the sorted output records into a physical file; this time however, two types of input records are selected for sorting: those with IM in columns 79 through 80 of the input record, and those with IQ in columns 79 through 80. This distinction corresponds to two record types, ITEMASTER and QTRSALES, as originally defined by the record data description for the input file.

There is an additional condition placed on the second record type: only the QTRSALES records containing dates greater than or equal to October 7, 1989, are to be included in the sort.

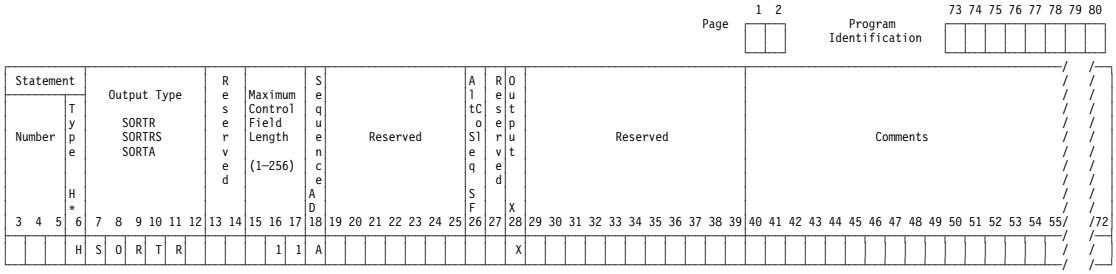
For ITEMNO records, the ITEMNO, DESC, SUPPLY, and SUPPADR fields are to be included in the output record. For QTRSALES records, the ITEMNO, QTRTOTAL, OUTLISTS, and DATE fields are to be included in the output record.

Both record types are to be sorted in ascending order in the output file based on the control field ITEMNO. QTRSALES records that have the same ITEMNO are arranged in descending order based on the control field QTRTOTAL. The two record types are merged in the output file. The first record will be the ITEMASTER record with the lowest ITEMNO, and following it will be the QTRSALES record with the same ITEMNO. If more than one QTRSALES record has the same ITEMNO, the records will be arranged in descending order based on the QRTOTAL field.

Entering the Specifications

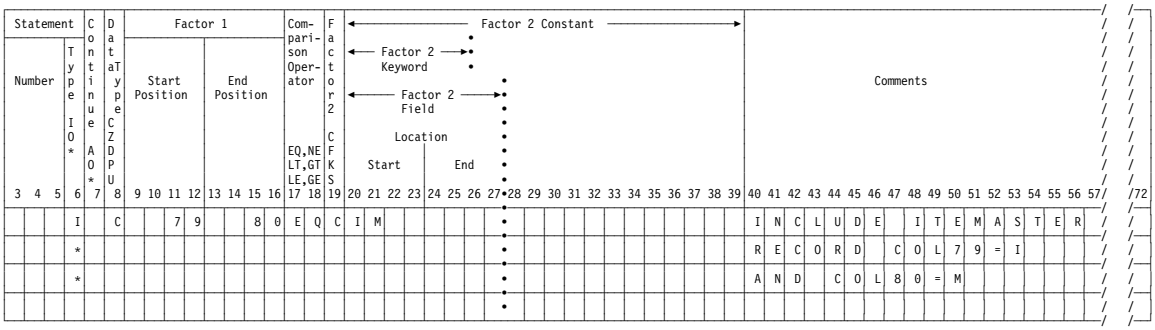
To sort this file, enter the following header, record, and field specifications. (Only those columns that are important to this example are described below.)

HEADER SPECIFICATIONS



Columns	Entry	Explanation
7-12	SORTR	This is a regular sort: the output is to be a physical file.
15-17	11	The maximum control field length is 11 characters. This is the sum of the two input fields, ITEMNO and QTRTOTAL, that are combined to form the control field.
18	A	The output records are to be sorted in ascending order (A before B, 1 before 2, and so on). For the ordering of numbers relative to upper- and lowercase characters and special characters, refer to the collating sequences in Appendix C, “Standard and Alternative Collating Sequences.”
26	No entry	The Sort utility defaults to the standard collating sequence.
28	X	The control fields are not to be included in the output record.

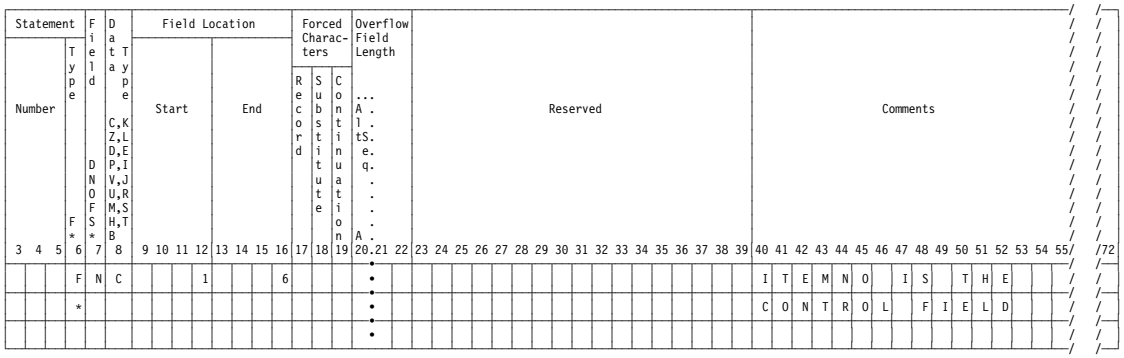
RECORD SPECIFICATIONS



Columns	Entry	Explanation
6	I	The input records that meet the conditions specified in this set of input record specifications are to be included in the output.
7	No entry	The first of the set of record specifications that is to define this type of record.
8	C	The data in the Factor 1 and Factor 2 fields of each input record is to be interpreted as character data. Each group of 8 bits represents a character.
9-12	79	The Factor 1 field begins in column 79 of the input record.
13-16	80	The Factor 1 field ends in column 80 of the input record. This selects the data from the ITEMASTER field, corresponding to the original data definition for the input records.
17-18	EQ	The data in the Factor 1 field must be equal to the data in the Factor 2 field for the input record to be selected for sorting.
19	C	The data in the Factor 2 field is to be the constant specified in columns 20 through 39 of this record specification.
20-39	IM	The constant value of the Factor 2 field is IM. The IM in columns 79 through 80 of the input record identifies this record as an ITEMASTER record.
40-72	Comments	Identifies the specified columns.

The record specification is followed by two comment specifications, identified by the asterisk in column 6.

FIELD SPECIFICATIONS



This specification establishes a control field for this record type:

Columns	Entry	Explanation
7	N	The control field is to be treated as a normal control field. This means that the unchanged data from the input record is to be used, and that the sort sequence is to be that specified in the header specification (ascending).
8	C	The system looks at the data in this control field as character data. Each group of 8 bits is assumed to represent a character, which can be a letter of the alphabet, a symbol, or a digit.
9-12	1	This control field begins in column 1 of the input record.
13-16	6	This control field ends in column 6 of the input record. This selects the data from the ITEMNO field, corresponding to the original data definition for the input records.
40-72	Comments	Identifies the name of the field selected as control field from the input record.

FIELD SPECIFICATIONS

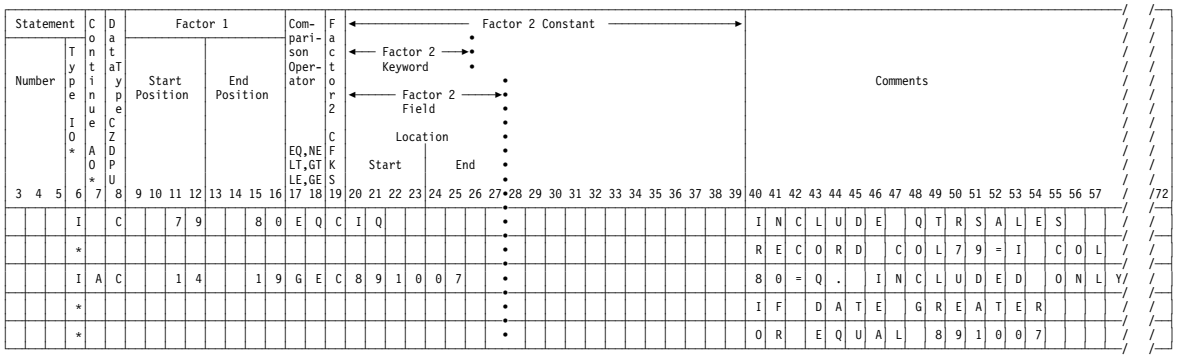
Statement Number	Statement Type	Field DNOFS*	Data Type C,KZ,LD,EP,IV,JU,RM,SH,TB	Field Location Start	Field Location End	Forced Characters Record	Forced Characters Substitute	Continuation	Overflow Field Length ...A.lt.e.q.....A.	Reserved	Comments
3 4 5		F* 6	7	9 10 11 12	13 14 15 16	17	18	19	20.21 22	23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39	40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 /72
		F	D C	1	6				•		ITEMNO FIELD
		F	D C	30	49				•		DESC FIELD
		F	D C	50	55				•		SUPPLY FIELD
		F	D C	56	78				•		SUPPADDR FIELD
									•		

These four specifications define four data fields to be taken from each input record of this type. The contents of the first specification are:

Columns	Entry	Explanation
7	D	The field is to be treated as a data field, meaning that the unchanged data from the input record is to be transferred to the output record.
8	C	The system looks at the data in this field as character data. Each group of 8 bits is assumed to represent a character, which can be a letter of the alphabet, a symbol, or a digit.
9-12	1	This data field begins in column 1 of the input record.
13-16	6	This control field ends in column 6 of the input record. This selects the data from the ITEMNO field, corresponding to the original data definition for the input records. This is also the same as the control field used for sorting. The data is used separately to control the sort and then respecified as data. This allows more flexibility. Control data can be altered before being used as a control; regular data can be repositioned in the output record.
40-72	Comments	Identifies the name of the field selected as a data field from the input record.

The field specification is followed by three more specifications, each defining a new data field (DESC, SUPPLY, and SUPPADR, respectively), corresponding to the original data description for the input records. Notice the order in which these fields are entered into the source file, represented by the sequence number in columns 3 through 5. This is the order in which they will be put into the output record.

RECORD SPECIFICATIONS



This set of record specifications defines a new record type. The record specifications are interspersed with comment specifications to help explain the input record selection criteria. The column entries for the first record specification in the set are:

Columns	Entry	Explanation
6	I	The input records that meet the conditions specified in this set of input record specifications are to be included in the output.
7	No entry	This specification is the first of the set of record specifications that is to define this type of record.
8	C	The data in the Factor 1 and Factor 2 fields of each input record is to be interpreted as character data. Each group of 8 bits represents a character.
9-12	79	The Factor 1 field begins in column 79 of the input record.
13-16	80	The Factor 1 field ends in column 80 of the input record.
17-18	EQ	The data in the Factor 1 field must be equal to the data in the Factor 2 field for the input record to be selected for sorting.
19	C	The data in the Factor 2 field is to be the constant specified in columns 20 through 39 of this record specification.
20-39	IQ	The constant value of the Factor 2 field is IQ. The IQ in columns 79 through 80 of the input record identifies this record as a QTRSALES record.
40-72	Comments	Identifies the specified columns.

This is followed by a comment specification and another record specification.

The second record specification in the set puts an additional condition on each input record before that record is selected for sorting. The significant column entries are:

Columns	Entry	Explanation
7	A	This specification adds another condition to the previous record specification.
8	C	The data in the Factor 1 and Factor 2 fields of each input record is to be interpreted as character data. Each group of 8 bits represents a character.
9-12	14	The Factor 1 field begins in column 14 of the input record.
13-16	19	The Factor 1 field ends in column 19 of the input record. This corresponds to the DATE field of the input record, as defined by the existing data description for the record.
17-18	GE	The data in the Factor 1 field must be greater than or equal to the data in the Factor 2 field for the input record to be selected for sorting.
19	C	The data in the Factor 2 field is to be the constant specified in columns 20 through 39 of this record specification.
20-39	891014	This is the constant value of the Factor 2 field, representing October 7, 1989.
40-72	Comments	Identifies the specified columns.

FIELD SPECIFICATIONS

Statement Number	File Type	Field *NOFS	Data Type C,KZ,LD,EP,IV,JU,RM,SH,TB	Field Location Start	Field Location End	Forced Characters Record	Forced Characters Substitute	Continuation	Overflow Field Length A	Reserved	Comments	
3 4 5	6	7	8	9 10 11 12	13 14 15 16	17	18	19	20.21 22	23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39	40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58	/72
	F	N	C		1 6				•		ITEMNO IS NORMAL	
	F	O	U	2 5	2 9				•		(ASCENDING)	
	*								•		QTRTOTAL IS OPPOSITE	

These two specifications establish the control field for this record type. The contents of the first specification define the first part of the control field, and are:

Columns	Entry	Explanation
7	N	The control field data is to be treated as a normal control field. This means that the unchanged data from the input record is to be used, and that the sort sequence is to be that specified in the header specification (ascending).
8	C	The system looks at the data as character data. Each group of 8 bits is assumed to represent a character, which can be a letter of the alphabet, a symbol, or a digit.
9-12	1	This control field data begins in column 1 of the input record.
13-16	6	This control field data ends in column 6 of the input record. This selects the data from the ITEMNO field, corresponding to the original data definition for the input records.
40-72	Comments	Identifies the name of the field selected as control field from the input record.

The second field specification defines another field (QTRTOTAL) from the input record, which will become the second part of the control field. Sort will regard both control fields as a single string of control characters, and will begin the sort with the character farthest to the left. The significant column entries are:

Columns	Entry	Explanation
7	0	The control field is to be treated as an opposite control field. This means that the input records are to be sorted so the data in the QTRTOTAL field ends in a sequence opposite to that specified in the header specification.
8	U	The data in this control field is to be interpreted as numeric data in unpacked format. This means that each digit is to be represented by 8 bits; the lower 4 bits identify the digit and the higher four (zone portion) represent the sign (plus or minus) of the entire number.
9-12	25	This control field data begins in column 25 of the input record.
13-16	29	This control field data ends in column 29 of the input record
40-72	Comments	Identifies the name of the field selected as control field from the input record.

FIELD SPECIFICATIONS

Statement Number	Type	Field FNOFS*	Data Type C,KZ,LD,EP,IV,JU,RM,SH,TB	Field Location Start	Field Location End	Forced Characters Record	Forced Characters Substitute	Forced Characters Continuation	Overflow Field Length ...A.lt.S.e.q.....A.	Reserved	Comments
3 4 5	6	7	8	9 10 11 12	13 14 15 16	17	18	19	20.21 22	23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39	40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 / /72
	F	D	C	1	6				•		ITEMNO FIELD
	F	D	C	25	29				•		QTRTOTAL FIELD
	F	D	C	11	13				•		OUTLISTS FIELD
	F	D	C	14	19				•		DATE FIELD
									•		
									•		

These four specifications define four data fields to be taken from each input record. The data fields are specified in the same way as the previous record type. Because the header specification omitted the control fields from the output record, the two control field elements are respecified here as the first two data fields of the output record.

Processing These Specifications

The FMTDTA command that could be used to process the specification is:

Specified on FMTDTA Command	Explanation
FMTDTA INFILE((PROD/SALEHIST))+	The input records are to be taken from the first member of file SALEHIST in library PROD.
OUTFILE(PROD/ITEMSALE)+	The output records are to be put into the first member of file ITEMSALE in library PROD.

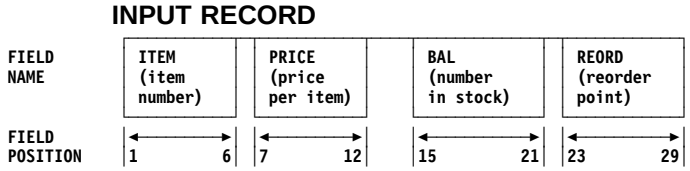
The specifications are automatically taken from the first member of file QFMTSRC *(default source file) in the library list.*

The OPTION parameter defaults to *CHK, *PRT, *NODUMP, and *NOSECLVL.

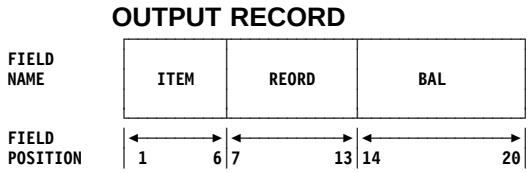
The plus sign (+) shown for continuations is not required if the FMTDTA *command is entered from a display station.*

Example 3. Reformatting Records–1

The order in which you select the input fields to be processed determines the order in which they will appear in the output records. Suppose you have an input record that looks like this:



You want your output record, however, to look like this:

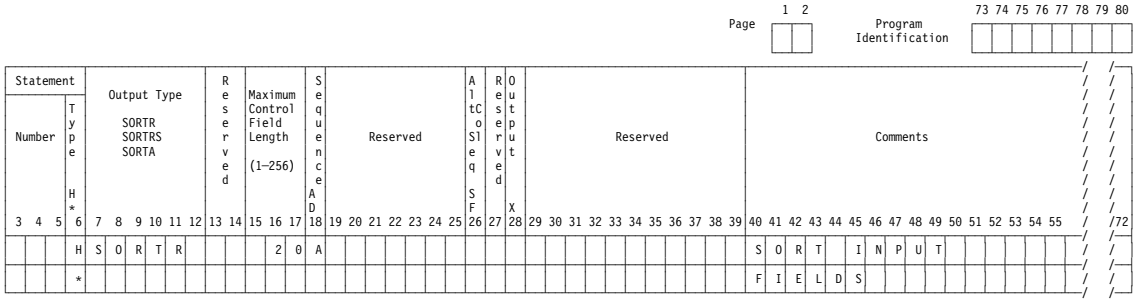


You also want to reorder all the records by item number.

Entering the Specifications

To sort this file, enter the following header, record, and field specifications. (Only those columns that are important to this example are described.)

HEADER SPECIFICATIONS

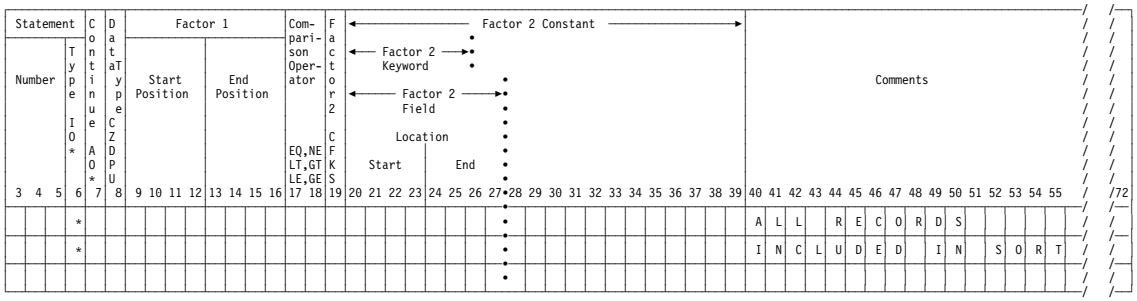


This is a standard header specification for a Sort job, similar to that entered for the previous examples. The significant column entry, here, is:

Columns	Entry	Explanation
28	No entry	The control field will be included in the output record.

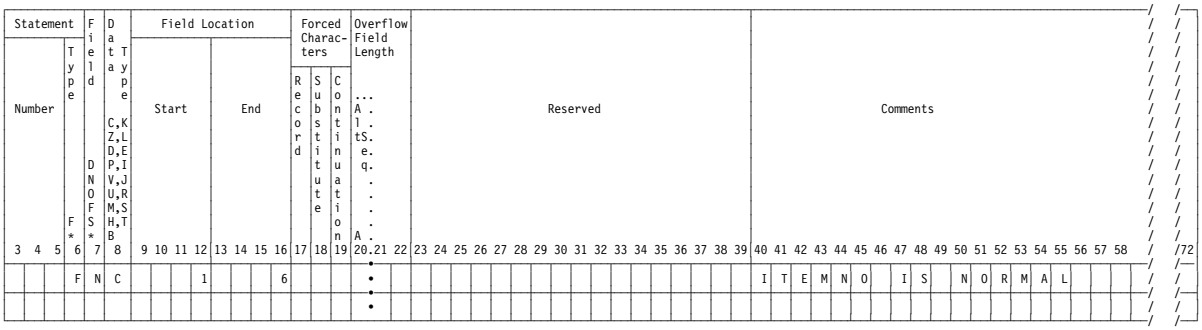
This means that the unchanged data from the input record is to be used, and that the sort sequence is specified in the header specification (ascending). A blank in column 28 will include the control field in the output record.

RECORD SPECIFICATIONS



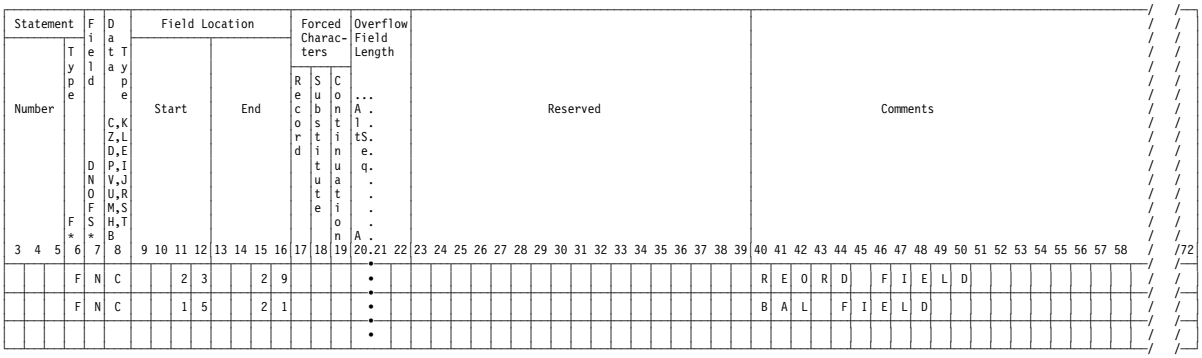
Because all input records are to be processed, no record specifications need to be entered, as indicated by the above comment specifications.

FIELD SPECIFICATIONS



This is the specification that defines the control field as ITEM from the input record. The control field is to be included in the output record and, because it is the first field specified, it will be the first field in the output record and will automatically occupy columns 1 to 6.

FIELD SPECIFICATIONS



The control field specification is then followed by two more field specifications, one identifying the field REORD, and the other identifying the field BAL from the input record.

Processing These Specifications

You can process these specifications by entering the FMTDTA command as described in "Example 2. Sorting Two Types of Records from the Input File" on page 22. Because the source records are no longer in the default file, you must also enter values for the SRCFILE and SRCMBR parameters of the command as shown below.

Specified on FMTDTA Command	Explanation
FMTDTA INFILE((PROD/SALEHIST))+	The input records are to be taken from the first member of file SALEHIST in library PROD.
OUTFILE(PROD/ITEMSALE)+	The output records are to be put into the first member of file ITEMSALE in library PROD.
SRCFILE(TEST/EXAMPLE)+ SRCMBR(EXAMPLE3)+	The reformat specifications are taken from member EXAMPLE3 in database file EXAMPLE, which is in library TEST.

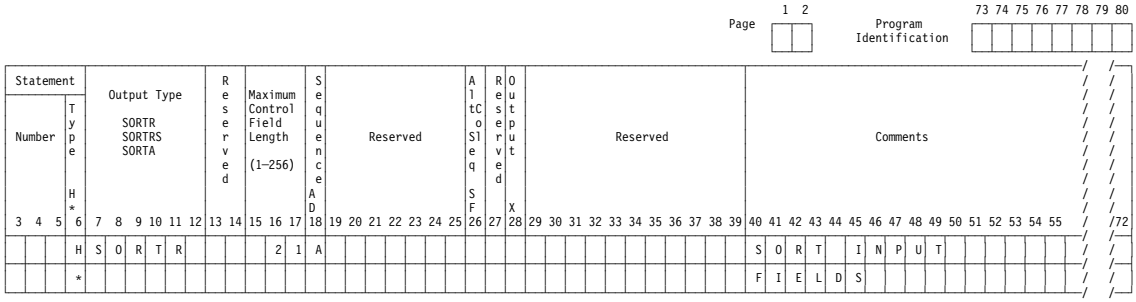
The OPTION parameters default to *CHK, *PRT, *NODUMP, and *NOSECLVL.

The plus sign (+) shown for continuations is not required if the FMTDTA *command is entered from a display station.*

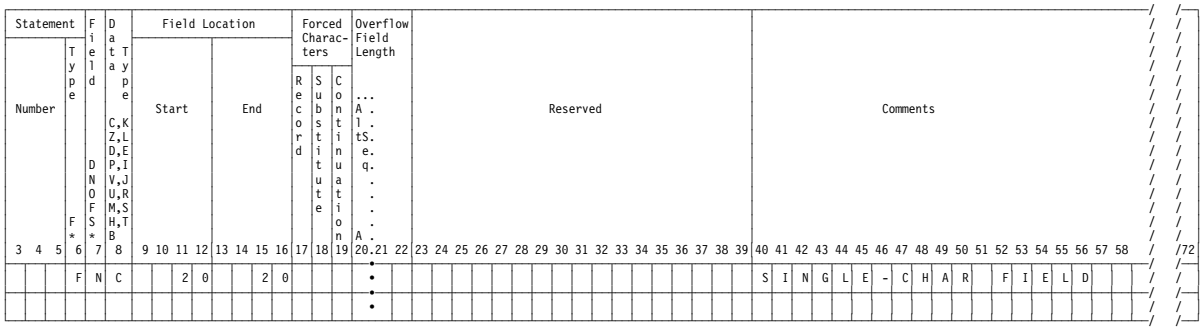
Example 4. Reformatting Records–2

In this example, you want the one-character field in position 20 of your input record to be in the first position of the output record. To do this, add the following field specification to the source file in the previous example:

HEADER SPECIFICATIONS



FIELD SPECIFICATIONS



Because of the sequence number, the control field specification (`ITEM` field) follows this data field specification. As a result, the control field is positioned in columns 2 through 7, after the one-character data field in the output record.

You then process the source file by using the `FMTDTA` command as in “Example 3. Reformatting Records–1” on page 30.

Chapter 3. Sorting Records from Several Files

The Sort utility can sort records selected from multiple physical, logical, or device files, and produce a copy of the records that reflects the new order. (See Figure 9.) It can also rearrange the fields within each record.

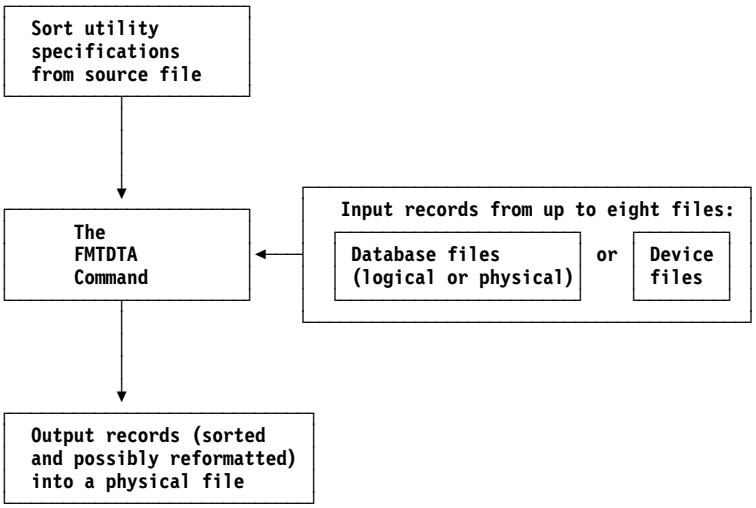


Figure 9. Sorting and Reformatting Several Files into One

Up to eight input files can be specified. The utility processes the files in the order in which they are entered in the FMTDTA command. The output is a physical file in the database, meaning that it contains a sorted and reformatted copy of the actual input records.

When sorting multiple input files to produce a physical output file, note the following:

- The output type on the header specification must be SORTR.
- The maximum control field length must be greater than zero.
- Field specifications must be entered to define the control field.
- You need not enter a separate field specification for each input field you wish to appear in the output record. If there are to be no changes in their order or content, you can treat the entire input record as one output field.
- The INFILE parameter of the FMTDTA command has several values, one for each input file being specified.

Examples

This section gives one example of how to use the Sort utility to sort records from several input files.

Example 1. Sorting Records from Five Input Files

This example processes five input transaction files, producing a single physical file as output. It also uses a ***forced character substitution*** to change part of the control field.

The control field is composed of two fields from the input record. The high order part comes from the ACCNO input field (columns 1 through 6), and the low order part a new character that is forced into the control field based on the value found in column 7. Letters in column 7 of the input records represent days of the week: M, T, W, R, and F. If column 7 contains an M, this character is a 1; if T, 2; if W, 3; if R, 4; and if F, 5. If it contains none of these, the new character will be 6. These modifications to the control field are made before the input records are sorted.

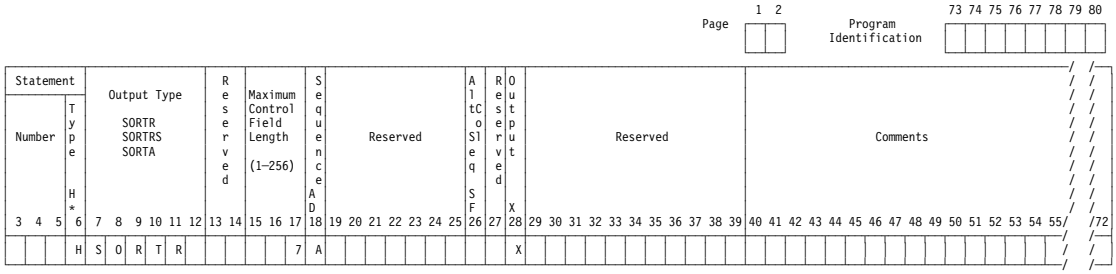
The original letters represent days of the week. One objective of the sort is to rearrange the input records so that Monday's records appear before Tuesday's, which appear before Wednesday's, and so on. If the letters were left in the control field, then the records would be sorted F-M-R-T-W. The change is for the convenience of this sorting only. It is invisible to you, since the input records remain unmodified, and the control fields are not included as control fields in the output records.

The output records are arranged in ascending order by ACCNO. If two records have the same ACCNO, they are arranged in ascending order by the character forced into the last column of the control field. All data fields will appear in the output records.

Entering the Specifications

To sort these files, enter the following header, record, and field specifications. (Only the columns important to this example are described.)

HEADER SPECIFICATIONS



Columns	Entry	Explanation
7-12	SORTR	This is a regular sort: the output is to be a physical file.
18	A	The output records are to be sorted in ascending order.
28	X	The control field is not to be included in the output record.

No record specifications are entered. Thus, there are no conditions imposed on the selection of input records. This means that all input records are to be processed.

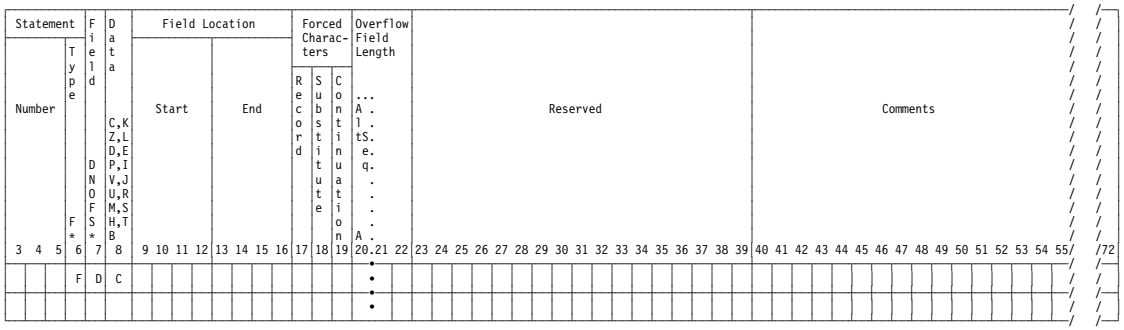
FIELD SPECIFICATIONS

Statement Number	Type	Field	Data Type	Field Location Start	Field Location End	Forced Characters Record	Forced Characters Substitute	Forced Characters Continuation	Overflow Field Length	Reserved	Comments
3 4 5	F* 6	DNOFS* 7	C,KZ,LD,EP,IV,JU,RM,SH,TB 8	9 10 11 12	13 14 15 16	17	18	19	...A.l.tS.e.q. . . . . A. 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
	F	N	C	1	6				•		ACCNO IS CTL FIELD
	F	F	C	7	7	M	1		•		IF MONDAY USE 1
	F	F	C	7	7	T	2	X	•		IF TUESDAY USE 2
	F	F	C	7	7	W	3	X	•		IF WEDNESDAY USE 3
	F	F	C	7	7	R	4	X	•		IF THURSDAY USE 4
	F	F	C	7	7	F	5	X	•		IF FRIDAY USE 5
	F	F	C				6	X	•		USE 6 FOR
	*								•		ANYTHING ELSE
	*								•		

These field statements define the source fields that are used to make up the control field. The first specification defines the input field that is to become the first part of the control field, and the next six all pertain to what is to form the last part of the control field. The significant column entries are:

Columns	Explanation
7	The first field specification is for a normal control field. The remaining six specifications contain an `F`, and describe a forced control field.
9-12	The start position of the field in the input record from which the data for the control field is taken. Data from columns 1 through 6 of the input record forms the first part of the control field; data from column 7 forms the last part.
13-16	The end position of the field in the input record from which the data for the control field is taken.
17	The value in this column identifies the character which, if found in the input record, is replaced.
18	This contains the value that is substituted.
19	An `X` in this column links the field specification to the previously entered field specification. Here, the last six specifications describe the same field.

FIELD SPECIFICATIONS



This field specification identifies the data fields to be included in the output records. No start or end positions are specified; therefore, all the input data fields are to be included in the output. The data in these fields is character data (column 8 contains a C).

Processing These Specifications

Following is an example of a FMTDTA command that could be used to process this specification. It is through this command that you state the five input files from which the input records are selected.

Specified on FMTDTA Command	Explanation
FMTDTA INFILE((PROD/TRANS MONDAY)+ (PROD/TRANS TUESDAY)+ (PROD/TRANS WEDNESDAY)+ (PROD/TRANS THURSDAY)+ (PROD/TRANS FRIDAY))+	Specifies the five input members from file TRANS in library PROD.
OUTFILE(PROD/TMERGED MONDAY)+	The output records are to be put into member MONDAY of file TMERGED in library PROD.

The specifications are automatically taken from the first member of file QFMTSRC *(default source file) in the library list.*

The OPTION parameter defaults to *CHK, *PRT, *NODUMP, and *NOSECLVL.

The plus sign (+) shown for continuations is not required if the FMTDTA command is entered from a display station.

Chapter 4. Sorting Records from a Single File to Produce a Record Address File

The Sort utility can sort (but not reformat) records selected from a single physical file and produce a Record Address (RA) file that reflects the new order. See Figure 10.

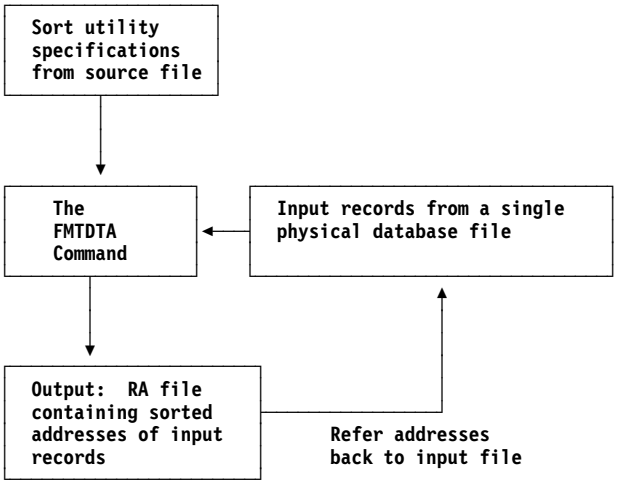


Figure 10. Sorting a Physical File into an RA File

The output is an RA file in the database. An RA file is a physical file that contains 4-byte binary relative record numbers rather than data. The output RA file has an arrival sequence access path. A relative record number of minus one (–1) indicates the end of an RA file.

Each record selected for processing produces a 4-byte, binary relative record number in the RA file.

When the relative record numbers are written to the output RA file, they are blocked up to the record size. Each output record represents a record in the input file. The position of the record in the output RA file is the position the input record has after the sort. When being printed or displayed, the RA file is read sequentially to process the records in the input file. As each relative record number is read, it can be used to access the original file directly.

Automatic Updating

An RA file is not a database object and does not have all the characteristics of a database file. It only reflects the characteristics of the input file at the time that it was created; it is not maintained dynamically as are database file access paths. If the input file is changed after the RA is created, you should be aware of the following:

- Records added to the input file after the RA file is created cannot be accessed by the RA file.
- If an RA file attempts to access a record that has been deleted from the input file, an exception condition will occur.
- Records that have been updated in the input file can still be accessed by the RA file, and will reflect the new data field values.

When you sort a single file to produce an RA output file, note the following:

- The output type on the header specification must be `SORTA`.
- The maximum control field length must be greater than zero.
- Column 28 of the header specification should contain an `X`. This excludes the control field from the output record.
- Field specifications must be entered to define the control field.
- No other field specifications should be entered. If you try to include data fields in the output records of an RA file, you can get inconsistent results.
- The `INFILE` parameter of the `FMTDTA` command has only one value, because only one input file can be specified for this type of sort.

Examples

This section gives one example of how to use the Sort utility to sort records and create an RA file.

Example 1. Sorting an Inventory File by Item Class

This example sorts the records taken from a file called ITMMSTR, which keeps track of the items and objects in a typical office. The items are grouped into classes, and each class is given a number. The records are to be sorted by item class.

The file will contain the addresses of the input records only; therefore, even if the original input file records are altered, any sorted output that is displayed or printed will contain the modifications and reflect the most up-to-date records.

Item class is a two-digit field in columns 41 through 42 of the input records, as shown in the following input file:

ITEM NUMBER	ITEM DESCRIPTION	ITEM TYPE	ITEM CLASS	WAREHOUSE STOCK LOCATION
20011230	Dbl pedestal desk lock	A	20	Frankfurt
30010010	Table desk-no center drawer	B	30	Rome
10012000	Swivel chair with arms	C	10	Toronto
70015120	5-drawer file with lock	D	70	New York
50011230	Storage cabinet with doors	E	50	Stockholm
40016210	Substitute drawer	F	40	Montreal
60013000	Overhead desk unit-2 shelves	G	60	Los Angeles
80012010	Chair-armless	H	80	Madrid

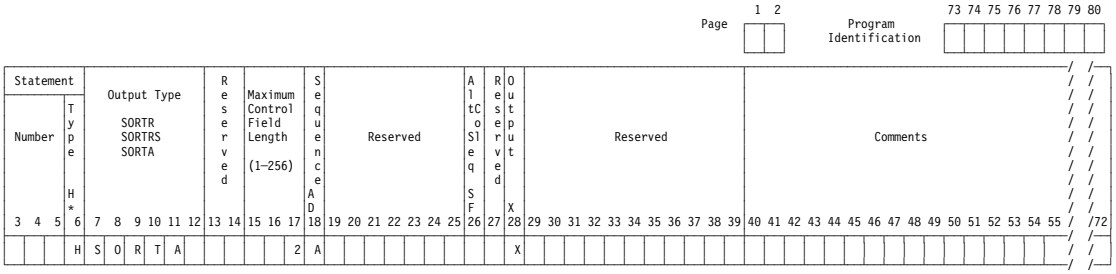
The records in the input file have the following format:

RECORD CODE	ITEM NUMBER	ITEM DESCRIPTION	ITEM TYPE	ITEM CLASS	WAREHOUSE STOCK LOCATION	
1	2 – 9	10 – 39	40	41 – 42	43 – 47	48 – 128

Entering the Specifications

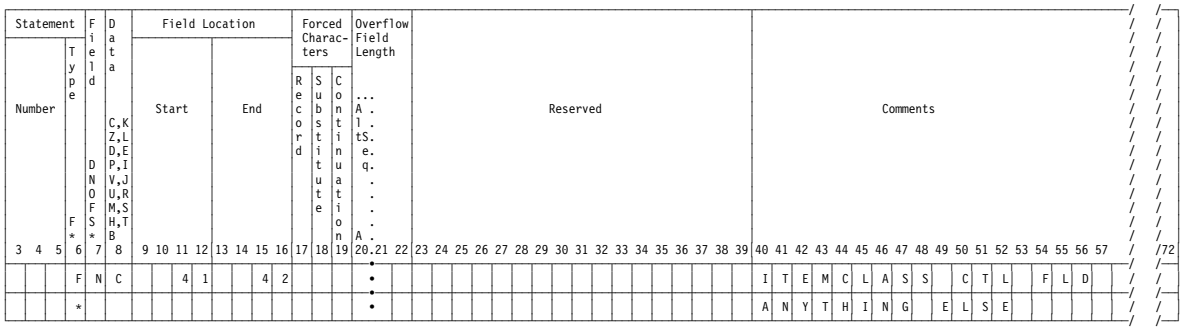
To sort this file, enter the following header, record, and field specifications. Only those columns that are important to this example are described.

HEADER SPECIFICATIONS



Columns	Entry	Explanation
7-12	SORTA	This is a Record Address (RA) sort: the output is to contain only record addresses.
15-17	2	The maximum control field length is 2 characters. The system will look at the entire control field and sort on the first two columns of that field only.
18	A	The output records are to be sorted in ascending order (A before B, 1 before 2, and so on). For the ordering of numbers relative to upper- and lowercase characters and special characters, refer to the collating sequences in Appendix C, “Standard and Alternative Collating Sequences.”
26	No entry	The Sort utility defaults to the standard collating sequence.
28	X	The output record is not to contain the control field data.

FIELD SPECIFICATIONS



This specification establishes the ITEMCLASS field as the control field. The contents of the specification are:

Columns	Entry	Explanation
7	N	The control field is to be treated as a normal control field. This means that the unchanged data from the input record is to be used, and that the sort sequence is specified in the header specification (ascending).
8	C	The system looks at the data in this control field as character data. Each group of 8 bits is assumed to represent a character, which can be a letter of the alphabet, a symbol, or a digit.
9-12	41	This control field is to begin in column 41 of the input record.
13-16	42	This control field is to end in column 42 of the input record. This selects the data from the TYPENO field, corresponding to the original data definition for the input records.
40-72	Comments	Identifies the name of the field selected as control field from the input record.

Processing These Specifications

Once the specifications for the RA sort job are stored in the source file, use the FMTDTA command to run the job. The INFILE parameter specifies a single database physical file. A FMTDTA command that could run this job is:

Specified on FMTDTA Command	Explanation
FMTDTA INFILE((LIBL1/ITMMSTR))+	The input records are to be taken from the first member of database file ITMMSTR in library LIBL1.
OUTFILE(LIBL2/RAMSTR)+	The output records are to be put into the first member of file RAMSTR in library LIBL2.

The specifications are automatically taken from the first member of file QFMTSRC *(default source file) in the library list.*

The OPTION parameter defaults to *CHK, *PRT, *NODUMP, and *NOSECLVL.

The plus sign (+) shown for continuations is not required if the FMTDTA *command is entered from a display station.*

Based on the actual input, the contents of the output file are:

```
/————————————————┐
/ RELATIVE RECORD │
/  NUMBERS        │
/                 │
/   00003         │
/   00001         │
/   00002         │
/   00006         │
/   00005         │
/   00007         │
/   00004         │
/   00008         │
/————————————————┘
```

Chapter 5. Sorting Records from a File and Producing Totals

In addition to sorting records from one or more input files, you can specify a ***summary sort***, whereby Sort adds together the contents of certain fields in the input records and produces a single output record containing the totals (see Figure 11). A maximum of 24 data fields can be summarized for each record type in a single job. These data fields are called ***summary data fields***, in order to distinguish them from regular data fields that are not summed.

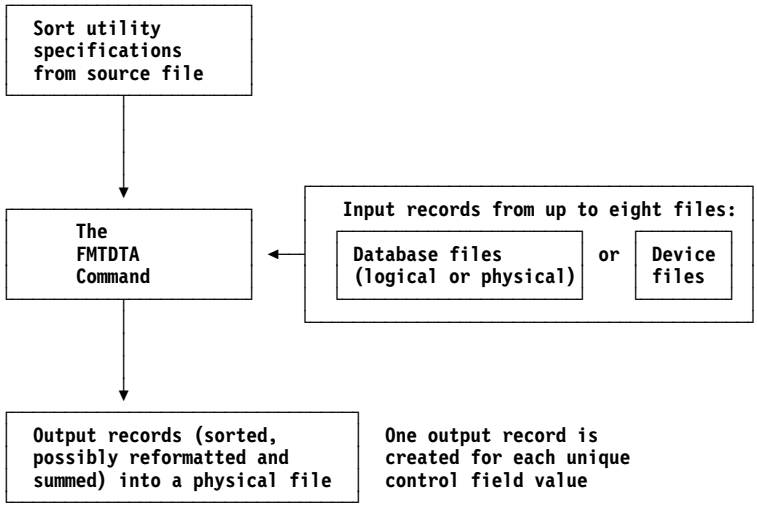


Figure 11. Sorting and Reformatting Records, and Producing Totals for Certain Data Fields

The output is a physical file in the database, meaning that it contains the actual output records. Note that the output file will contain only as many records as there are unique control field values; it is thus usually much smaller than the input file. The Sort utility sorts the output records according to the contents of the control field.

Defining a Summary Set

The following entries are required for a summary sort:

- The output file type in columns 7 through 12 of the header specification must contain SORTRS.
- The maximum control field length in columns 15 through 17 of the header specification must contain a numeric value greater than zero.
- At least one of each of the following field types must be defined:

  Control Field Enter one or more field specifications to define how the input records are to be grouped for summarizing. Column 7 will contain N, O, or V. A single summary record will be created for each set of input records with a unique control field value.

  Summary Data Field Enter one or more field specifications to define the data items that are to be summed. Column 7 must contain S. The corresponding summary data fields from all input records will be added together, and the total placed in the summary output record.

A summary output record may contain data fields that are not summary fields. These fields are defined using a field specification with D in column 7. The fields will contain data copied from the first input record processed for each unique control field value.

If two or more input record types are specified for a summary sort, each record type must have at least one summary data field defined. When specifying a summary sort with multiple record types, you should consider whether the control fields specified will allow records of different types to share an identical control field value, and if so, whether it is appropriate to add together the corresponding summary of data fields from the different record types.

Handling Overflow

If the totals are larger than the size allowed by the original summary data fields, the system puts only the lower-order digits into the summary field. The higher-order digits are lost, and the system gives no indication that this has happened. For example, if the total is 1 324 567, and the summary data field is only four characters long, the output record will contain only the last four digits, 4567, and you will not be aware that this is not the correct total.

To avoid this, you can do one of two things:

- You can designate an overflow indicator to appear on the output record in the case of an overflow.
- You can increase the size of the summary data field.

To use an overflow indicator you must enter an overflow indicator field specification containing S in column 7 and V in column 8. This will define a single character field in the next position of the output record to hold the overflow indicator. You may also enter a character in column 17 of the specification, which will be used to denote that overflow has occurred. If column 17 is left blank, an asterisk will be used instead. If a character is entered in column 18, it will be used to denote that overflow has not occurred.

The overflow indicator is placed in the output record if any of the summary data fields contained in the record have overflowed. Only one overflow indicator can be specified. If you specify more than one, the last indicator specified will be used.

To increase the size of the summary data field, enter the required size of the field in columns 20 through 22 of the field specification for the summary data field. The new size must not exceed the maximum size for the data type defined in column 8. The new, larger field is referred to as an overflow field, and replaces the summary data field in the output record.

You can specify an overflow field for any summary data field. You can designate an overflow indicator in addition to overflow fields.

Examples

Following are four examples that illustrate the summary sort feature.

Example 1. Sorting a File and Summing Data in Certain Fields–One Record Type

This example checks certain stock items for shortages. Here, the input file contains all inventory transactions. These are composed of three different record types according to the original data description: inventory records, receipt records, and adjustment records. The record code in column 2 (`I`, `R`, or `A`) of the input record identifies which of the three types the input record belongs to. Only the records with an `A` in column 2 (adjustment records) are to be selected. Figure 12 on page 50 shows part of the input file.

These adjustment records are sorted, and the number of items (`QUANTITY ORDERED` field) is totaled for each item (identified by the item number in the `ITEM NUMBER` field). The `ITEM NUMBER` field, therefore, is the control field, and the `QUANTITY ORDERED` field is the summary data field.

The output records are 14 columns long: the first 8 columns contain the control field data (the item number), and the last 6 columns contain the totals for each item. The output records are sorted so that the item numbers are in ascending order, according to the standard EBCDIC collating sequence for character data. There will be as many output records as there are different item numbers.

Because there is only one record type, all output records are sorted as one unit.

INVENTORY RECORDS

RECORD CODE	ITEM NUMBER	QUANTITY ORDERED	CUSTOMER NUMBER	INVOICE NUMBER	TRANSACTION DATE	SELLING PRICE
I	70015120	2	900	0111	880812	212
I	10012000	3	1100	0119	880812	295
I	70015120	4	1100	0113	880812	212
I	50011230	5	400	0662	880812	325
I	50011230	6	400	0221	880812	325
I	20011230	6	700	0568	880812	515
I	40016210	2	1000	2110	880816	42
I	40016210	5	1000	2110	880816	42
I	50011230	2	1100	3110	880816	325
I	10012000	3	400	7117	880816	295
.	.	.	.	.	.	.
.	.	.	.	.	.	.

DELETE CODE	RECORD CODE	ITEM NUMBER	QUANTITIY ORDERED	CUSTOMER NUMBER	INVOICE NUMBER	TRANSACTION DATE	SELLING PRICE
1	2	3 10	11 16	17 24	25 32	33 38	39 43

RECEIPT RECORDS

RECORD CODE	ITEM CLASS	ITEM NUMBER	SELLING PRICE	PURCHASE ORDER #	QUANTITY ORDERED	TRANSACTION DATE
R	20	20011230	85000	0568	10	880812
R	10	10012000	6295	4330	6	880816
R	40	40016210	542	2110	3	880816
R	50	50011230	28325	3110	2	880816
R	20	20011230	49515	0568	5	880816
.	.	.	.	.	.	.
.	.	.	.	.	.	.

DELETE CODE	RECORD CODE	ITEM CLASS	ITEM NUMBER	SELLING PRICE	PURCHASE ORDER NUMBER	QUANTITY ORDERED	TRANSACTION DATE
1	2	3	4 5	12 13	17 18	21 22	27 28 33

ADJUSTMENT RECORDS

RECORD CODE	ITEM NUMBER	ADJUSTMENT CODE	QUANTITY ORDERED	TRANSACTION DATE	WAREHOUSE
A	10012000	1	1	880816	2
A	20011230	1	6	880812	2
A	20011230	1	5	880816	3
A	40016210	2	4	880816	1
A	30010010	1	2	880816	1
.	.	.	.	.	.
.	.	.	.	.	.

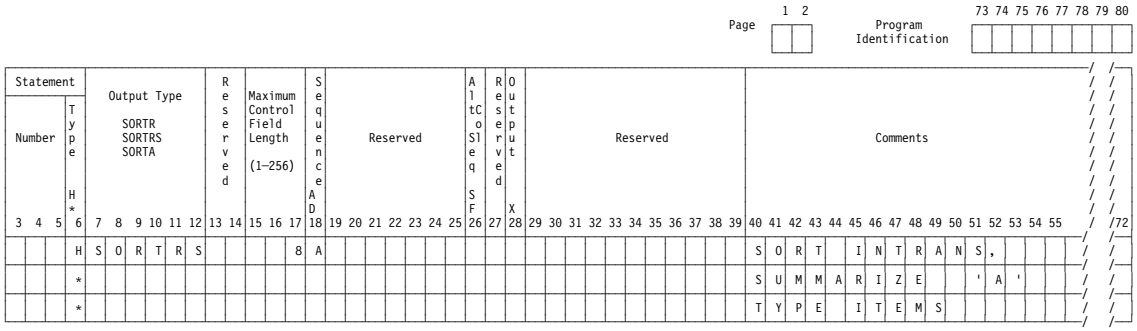
DELETE CODE	RECORD CODE	ITEM NUMBER	ADJUSTMENT CODE	QUANTITY ORDERED	TRANSACTION DATE	WAREHOUSE
1	2	3 10	11 12	17 18	23 24	25

Figure 12. The Input File Used in Example 1 Contains Three Different Record Types According to the Original Data Description for the File

Entering the Specifications

To sort this file, enter the following header, record, and field specifications. (Only those columns that are important to this example are described.)

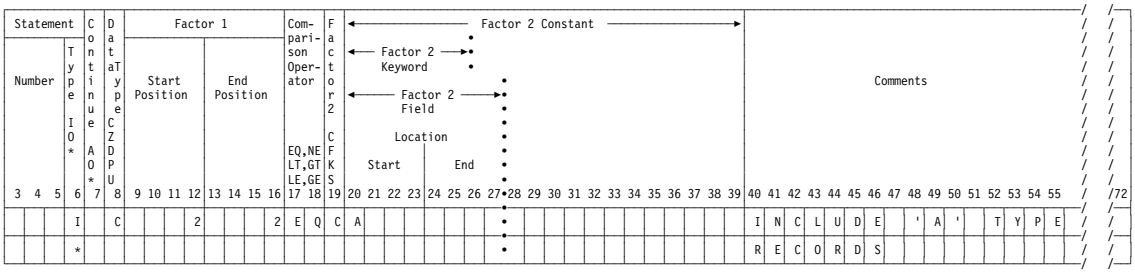
HEADER SPECIFICATIONS



The significant column entries are:

Columns	Entry	Explanation
12	S	Specifies the job as a summary sort. Data in specified summary data fields will be added together.
17	8	Specifies the length of the control field in the output record. In this example, item number, which spans eight columns, is the control field.
18	A	Specifies that the output records will be sorted in ascending order.
28	No Entry	Default. Specifies that the output records are to contain the control field.
48-72	Comments	Comments provide a record of your sort job.

RECORD SPECIFICATIONS

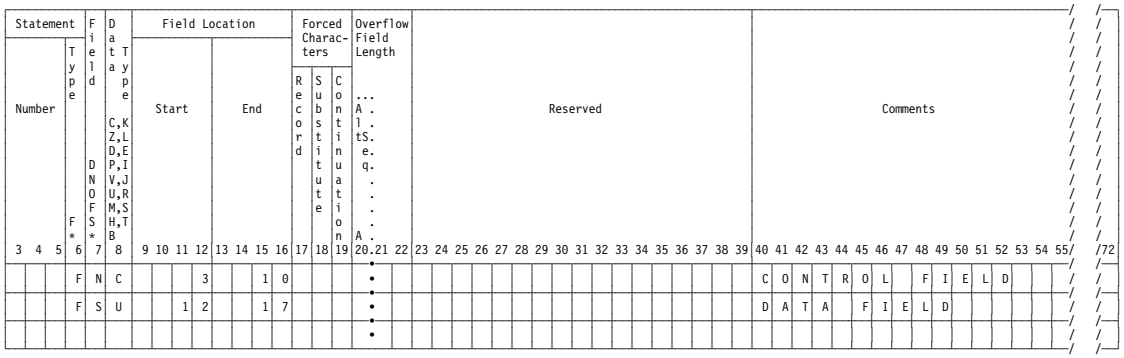


The significant column entries are:

Columns	Entry	Explanation
6	I	Specifies that the records described in this specification statement should be included in the sort (as opposed to omitted from the sort).
8	C	Identifies the data for Factor 1 and 2 as character data. The sort program will look at both the zone and digit portions of each byte when comparing Factor 1 with Factor 2.
9-12	No entry	The start position of the Factor 1 field is the same as the end position.
13-16	2	The end position of the Factor 1 field.
17-18	EQ	Specifies that the data in Factor 1 of each record must equal Factor 2 for the record to be selected for sorting.
19	C	Specifies that Factor 2 is to be a constant and equal to the value entered into columns 20 through 39 of this record specification.
20-39	A	This is the constant value for Factor 2.

Two field specifications are used. The first defines the control field; the second defines the data field to be summed (the summary data field).

FIELD SPECIFICATIONS



These are the two field specifications. The significant column entries for the control field specification are:

Columns	Entry	Explanation
7	N	The control field data is to be treated as a normal control field. This means that the unchanged data from the input record is to be used, and that the sort sequence is to be that specified in the header specification (ascending).
8	C	Specifies that Sort is to interpret the data in the control field as character data where both the zone and digit portions of each byte are compared. If, for example, only the digit portions were compared, data would be summed for those records whose control field digit portions matched; the control field values would not necessarily have to be unique, since the zone portions could differ.
9-12	3	Start position of the control field data from the input record.
13-16	10	End position of the control field data from the input record.

The significant column entries for the summary data field specification are:

Columns	Entry	Explanation
7	S	Identifies the field as a summary data field. The data in this field is the data to be summed.
8	U	The data in this field is to be interpreted as numeric data in unpacked format: each digit is represented by both the zone and digit portions of the byte, with the zone portion of the last byte determining the sign (positive or negative) for the entire number.
9-12	12	Start position of the data field in the input records.
13-16	17	End position of the data field in the input records.

Processing These Specifications

To process these specifications, enter the FMTDTA command with parameters as for any of the examples in the previous chapters. For example, you could enter:

Specified on FMTDTA Command	Explanation
`FMTDTA INFILE((Z/Y X))+`	The input records are to be taken from member X of database file Y in library Z.
`OUTFILE(SORTED)+`	The output records are to be put into the first member of file SORTED in the current library.

The specifications are automatically taken from the first member of file QFMTSRC *(default source file) in the library list.*

The OPTION parameter defaults to *CHK, *PRT, *NODUMP, and *NOSECLVL.

The plus sign (+) shown for continuations is not required if the FMTDTA *command is entered from a display station.*

Sort sends the output automatically to the device file QSYSPRT, which is the system printer.

The output file looks like the following:

```
ITEM NUMBER   QUANTITY ORDERED

  10012000              1
  20011230             11
  30010010              2
  40016210              4
```

```
Item        Quantity
Number      Ordered
1         8 9         14
```

Example 2. Producing Summary Totals Based on Two Input Record Types

In this example, you want to sum and sort the same file shown in Figure 12 on page 50 in the previous example. Two input record types are specified; inventory records (record code I), and receipt records (record code R). Both are reformatted in the same way: the control field (ITEM NUMBER) followed by the summary data field (SELLING PRICE).

Both types of input record are to be sorted by item number. If there are multiple occurrences of an item number, the selling prices are to be summed to produce a total selling price for each item number (total sales).

Figure 13, below, shows the two record types and the significant input fields for this file.

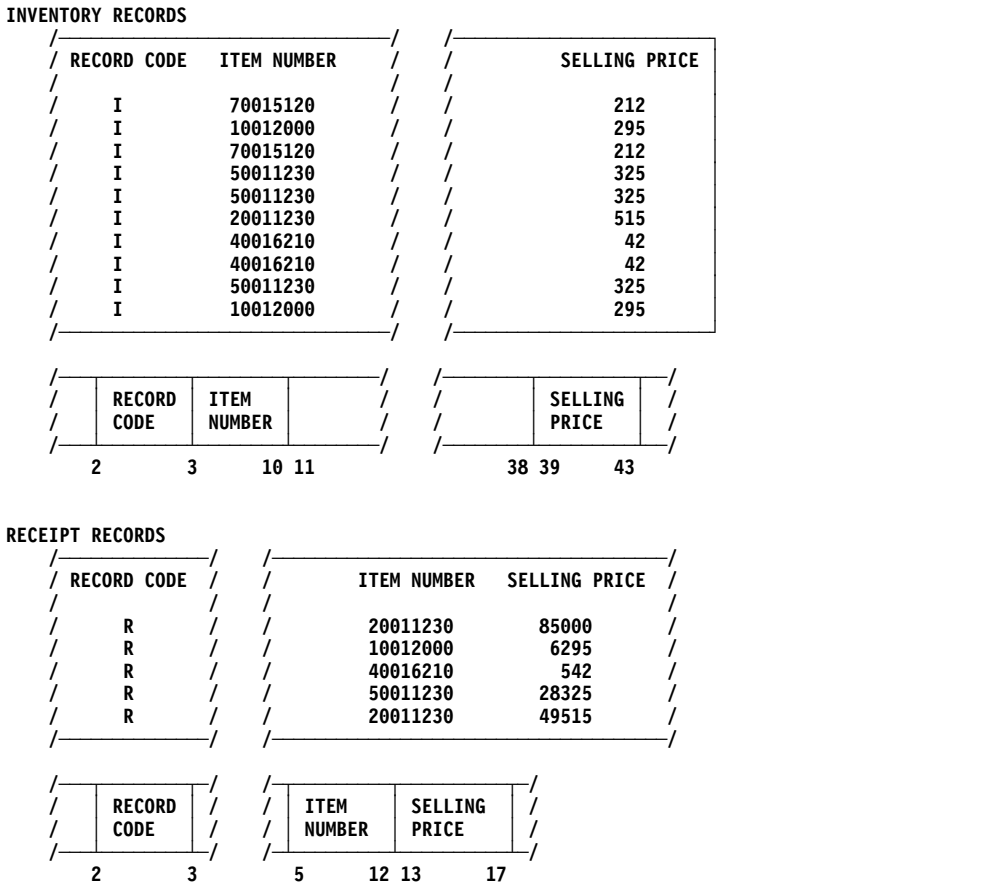
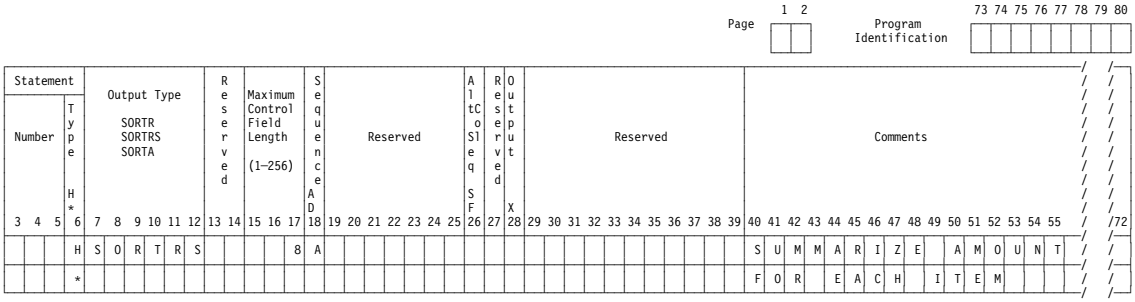


Figure 13. Example of the Two Different Record Types in This Input File

Entering the Specifications

Enter the following sort specifications. (Only those columns that are important to these examples are described.)

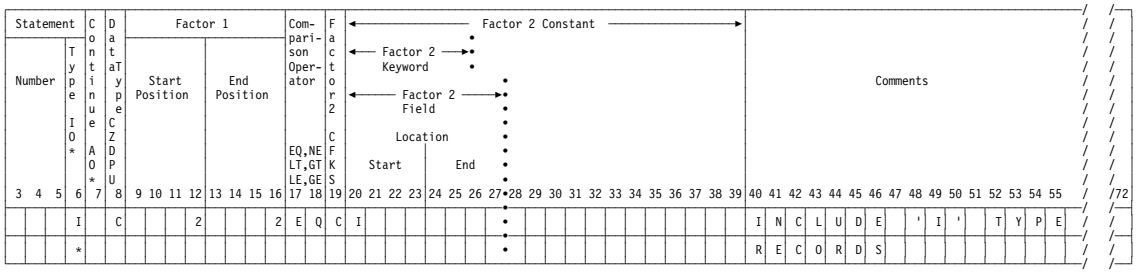
HEADER SPECIFICATIONS



The significant column entries are:

Columns	Entry	Explanation
12	S	Specifies the job as a summary sort. Data in specified summary data fields will be added together.
17	5	Specifies the length of the control field in the output record. In this example, item number, which spans five columns, is the control field.
18	A	Specifies that the output records are to be sorted in ascending order.
28	No entry	Specifies that the output records are to contain the control field as well as the summary data field.
40-72	Comments	Comments provide a record of your sort job.

RECORD SPECIFICATIONS

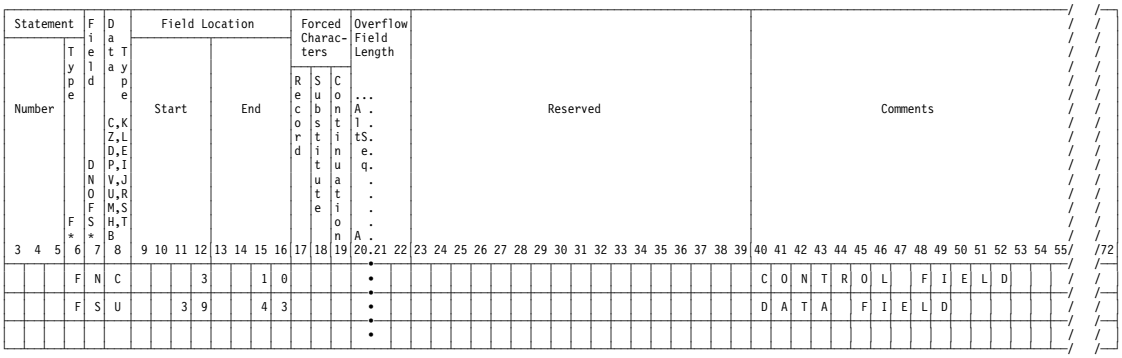


The significant column entries are:

Columns	Entry	Explanation
6	I	Specifies that the records described in this specification should be included in the summary sort (as opposed to being omitted from it).
8	C	Identifies the data for Factor 1 and 2 as character data. The system will look at both the zone and digit portions of each byte when comparing Factor 1 with Factor 2.
13-16	2	Start and end position of the Factor 1 field.
17-18	EQ	Specifies that the data in Factor 1 of each record must equal Factor 2.
19	C	Specifies that the data in the Factor 2 field is to be equal to the constant value entered into columns 20 through 39 of this record specification.
20-39	I	The constant value for Factor 2.

Because the input file has two different record types, and you intend to work with both, you must specify them as two separate sets of specifications. In the field specifications for the first set, a summary data field is defined after the control field.

FIELD SPECIFICATIONS



The significant column entries are:

Columns	Entry	Explanation
7	N	Identifies the field as a normal control field. The data in the control field is not to be forced (altered) before being used to control the sort.
8	C	Specifies that Sort is to interpret the data in the control field as character data, where both the zone and digit portions of each byte are compared.
9-12	3	The start position of the control field is the same as the end position.
13-16	10	The end position of the control field data in the input record.

The significant column entries for the summary data field specification are:

Columns	Entry	Explanation
7	S	Identifies the field as a summary data field. The data in this field is to be summed.
8	U	The data in this field is to be interpreted as numeric data in unpacked format: each digit is represented by both the zone and digit portions of the byte, with the zone portion of the last byte determining the sign (positive or negative) for the entire number.
9-12	39	The start position of the control field is the same as the end position.
13-16	43	The end position of the control field data in the input record.

Summary data fields are defined very much like normal output data fields:

- Enter F into column 6, to indicate a field selection specification.
- Enter S into column 7, to indicate a summary data field.
- Enter the data type of the field to be added into column 8.

  The same data types that are used for normal data fields are also used for summary data fields. This data type entry is important, because the method of adding and the interpretation of the result vary depending on the data type selected. In the following list, the word *number* refers to the entire sequence of digits: the number 4 526 is composed of the digits 4, 5, 2, and 6.

  P ***Packed data type.*** System considers each sequence of four bits as one digit, starting with the second sequence from the right. The four bits farthest to the right determine the sign of the number.

U ***Unpacked data type.*** System considers each sequence of eight bits (each byte) as one digit. The digit portions determine the digit, and are added together. The zone portion of the last byte determines the sign of the number. (The zone portions of the other bytes remain as `X'F'` to indicate to the system that these bytes are digits, as opposed to other characters.)

D ***Digit data type.*** System considers each byte as one digit, as for unpacked data, except that the zone portion of the last byte is not used to contain the sign of the number. The number has no sign; it is assumed to be positive.

C ***Character data type.*** System considers each byte as one character.

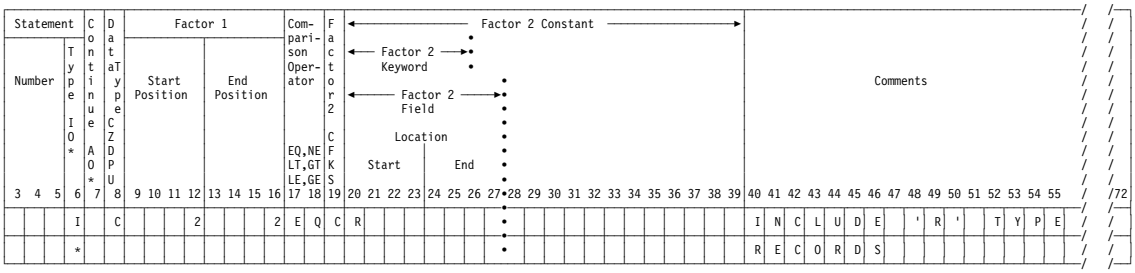
Z ***Zone data type.*** System considers each byte as one digit, as for unpacked data. Only the zone portions are added together.

Note: When using summary data fields, you must ensure that the data in the input file matches the data type given in the field specifications; otherwise, the results are unpredictable.

- Enter the start and end position of that part of the input record that is to become the summary data field. Like normal data fields, the starting position need not be specified for a field that is only one position long.

To allow the Sort utility to also take data from type R records, specify another record type by entering a second set of record and field specifications. You need the new field specifications because the start and end positions of the `ITEM NUMBER` and `SELLING PRICE` fields for the R input records are different from those for the I input records.

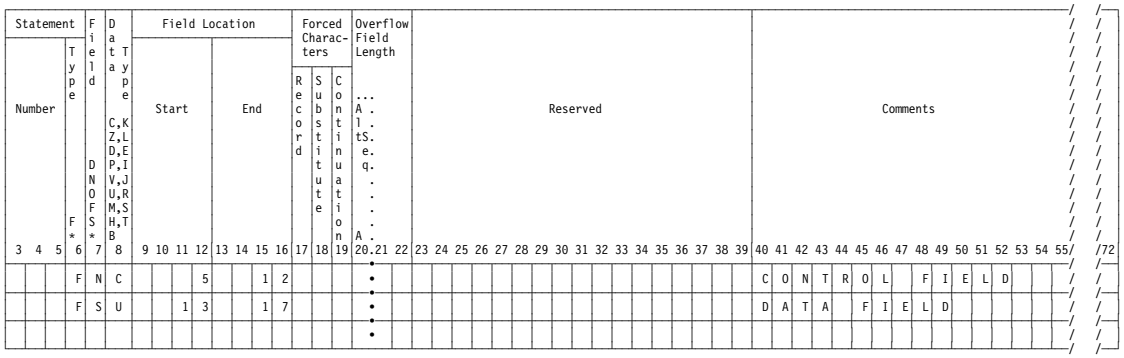
RECORD SPECIFICATIONS



The significant column entries are:

Columns	Entry	Explanation
6	I	Specifies that the records described in this specification are to be included in the sort.
8	C	Identifies the data for Factors 1 and 2 as character data. The system will look at both the zone and digit portions of each byte when comparing Factor 1 with Factor 2.
13-16	2	Start and end position of the Factor 1 field.
17-18	EQ	Specifies that the data in Factor 1 of each record must equal Factor 2.
19	C	Specifies that the value in Factor 2 is equal to the constant value entered into columns 20 through 39 of this record specification.
20-39	R	The constant value for Factor 2.

FIELD SPECIFICATIONS



The significant column entries are:

Columns	Entry	Explanation
7	N	Identifies the field as a normal control field. The control field data is not to be forced (altered) before being used to control the sort.
8	C	Specifies that Sort is to interpret the data in the control field as character data where both the zone and digit portions of each byte are compared.
9-12	5	Start position of the control field data in the input record.
13-16	12	End position of the control field data in the input record.

The significant column entries for the summary data field specification are:

Columns	Entry	Explanation
7	S	Identifies the field as a summary data field. The data in this field is to be summed.
8	U	The data in this field is to be interpreted as numeric data in unpacked format: each digit is represented by both the zone and digit portions of the byte, with the zone portion of the last byte determining the sign (positive or negative) for the entire number.
9-12	13	Start position of the data field in the input records.
13-16	17	End position of the data field in the input records.

Processing These Specifications

To process these specifications, enter the FMTDTA command with parameters as for any of the examples in the previous chapters. For example, you could enter:

Specified on FMTDTA Command	Explanation
`FMTDTA INFILE((PROD/INVENTRY MONDAY))+`	The input records are to be taken from member MONDAY of database file INVENTRY in library PROD.
`OUTFILE(PROD/NEWORD MONDAY)+`	The output records are to be put into member MONDAY of file NEWORD in library PROD.
`SRCFILE(TEST/EXAMPLE)+` `SRCMBR(EXAMPLE1)+`	The reformat specifications are taken from member EXAMPLE1 in database file EXAMPLE, which is in library TEST.

The OPTION parameter defaults to *CHK, *PRT, *NODUMP, and *NOSECLVL.

The plus sign (+) shown for continuations is not required if the FMTDTA *command is entered from a display station.*

The output file looks like the following:

```
     ITEM NUMBER    SELLING PRICE

       10012000         6885
       20011230        35030*
       40016210          626
       50011230        29300
       70015120          424
           .              .
           .              .

  * Overflow occurred
```

```
┌──────────┬──────────┐──/
│  Item    │ Selling  │  /
│  Number  │ Price    │ /
└──────────┴──────────┘──/
1          8 9        13
```

Example 3. Using an Overflow Field for Summary Data Overflow

This situation is the same as in "Example 2. Producing Summary Totals Based on Two Input Record Types" on page 54, except that here you will specify an overflow field to hold totals that may be more than five characters long.

Entering the Specifications

To define the overflow field, enter a length for this overflow field into the Overflow Field Length columns (columns 20 through 22) of the field specification. The length that you specify here must not be greater than the maximum length allowed for the input data type as specified in column 8 of the field specification (refer to Table 9 on page 104).

Note that, because of the addition of the overflow field, the output record length will be longer than the input record length.

Suppose you anticipate an overflow only for record type R of "Example 2. Producing Summary Totals Based on Two Input Record Types" on page 54. All header, record, and field specifications would therefore be the same as for that example, with the exception of a new field specification added to those for record type R. These specifications would then be as follows (only those columns that are important to this example are described):

FIELD SPECIFICATIONS

Statement Number	Type	Field	Data Type	Field Location Start	Field Location End	Forced Characters Record	Forced Characters Substitute	Forced Characters Continuation	Overflow Field Length	Reserved	Comments
3 4 5	F*	DNOFS* 7	C,KZ,LD,EP,IV,JU,RM,SH,TB 8	9 10 11 12	13 14 15 16	17	18	19	...Alt.Se.q. 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 /72
	F	N	C	5	12				10		CONTROL FIELD
	F	S	U	13	17				10		DATA FIELD

Figure 14. The Field Specifications for Record Type R from Example 2

The additional column entries for the summary data field specification are:

Columns	Entry	Explanation
20-22	10	Gives the length of the overflow field. By entering a value into this field, you automatically increase the size of this summary data field to 10 columns.

All other entries are the same as for Example 2.

Processing These Specifications

FMTDTA processes the job, as in "Example 2. Producing Summary Totals Based on Two Input Record Types" on page 54. The output file, however, looks like:

```
     ITEM NUMBER          SELLING PRICE

       10012000                6885
       20011230              135030
       40016210                 626
       50011230               29300
       70015120                 424
           .                     .
           .                     .
```

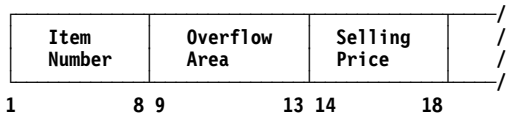


Figure 15. The Output File Produced in Example 3

Note that the larger summary data field can now accommodate the total value for item number 20011230.

Example 4. Using an Overflow Indicator for Summary Data Overflow

There may be times when you do not want to specify a summary overflow field length, or when you do not know how large a summary overflow field will be. In these cases, you can cause the utility to put a constant (the *overflow indicator*) into the output record each time an overflow occurs. Subsequent programs that read your sorted output file can then test for the presence of this character constant and act accordingly.

Entering the Specifications

You can specify the overflow indicator on any field specification for the job. If you specify different overflow characters on different field specifications, Sort uses the last one that was specified for each record type. For example, if you use the summary data field specification for record type R of the previous example, the field specifications would then become as follows (only those columns that are important to this example are described):

FIELD SPECIFICATIONS

Statement Number	Type	Field	Data Type	Field Location Start	Field Location End	Forced Characters Record	Forced Characters Substitute	Forced Characters Continuation	Overflow Field Length	Reserved	Comments
3 4 5	F*	DNOFS* 7	8 C,KZ,LD,EP,IV,JU,RM,SH,TB	9 10 11 12	13 14 15 16	17	18	19	...Al.tS.e.q.....A. 20.21	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
	F	N	C		5		1	2	•		CONTROL FIELD
	F	S	U	1	3		1	7	•		DATA FIELD
	F	S	V			?			-		PUT ? IN OUTPUT
	*								•		IF OVERFLOW

A new field specification is added to provide the overflow indicator. The significant column entries are:

Columns	Entry	Explanation
7	S	Identifies the field as a summary data field. The data in this field is to be summed.
8	V	Force a data character constant into the data field.
9-12	No Entry	Leave this part of the field specification blank.
13-16	No Entry	Leave this part of the field specification blank.
17	?	Provides the overflow indicator.

Note: If overflow occurs in any of the summary data fields, the character in column 17 will be written into the overflow indicator field.

If overflow occurs and no overflow character is identified (column 17 is blank) the system will set the overflow indicator to an asterisk (*).

Processing These Specifications

This job is processed like “Example 2. Producing Summary Totals Based on Two Input Record Types” on page 54. The output file contains the following entries:

```
     ITEM NUMBER    SELLING PRICE    OVERFLOW

       10012000          6885
       20011230         35030            ?
       40016210           626
       50011230         29300
       70015120           424
           .                .
           .                .
```

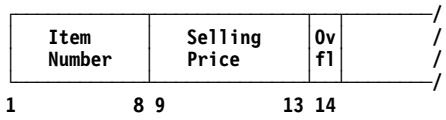


Figure 16. The Output Showing the Overflow Character

Notice that the system has expanded the output record by one column, and placed an asterisk into this column to indicate that the record with item number 20011230 has a summary data field with an overflow value.

Chapter 6. Copying and Merging Records from Several Files into One

If you do not specify any sorting of records, the Sort utility can be used to simply copy the data from one or more input files into a single physical file in the database. The data is placed in the output file in the same sequence as it is found in the input file; no reordering is done. See Figure 17.

If the output file already contains records, they are not cleared before the input records are copied. The input records are added to the end of the file.

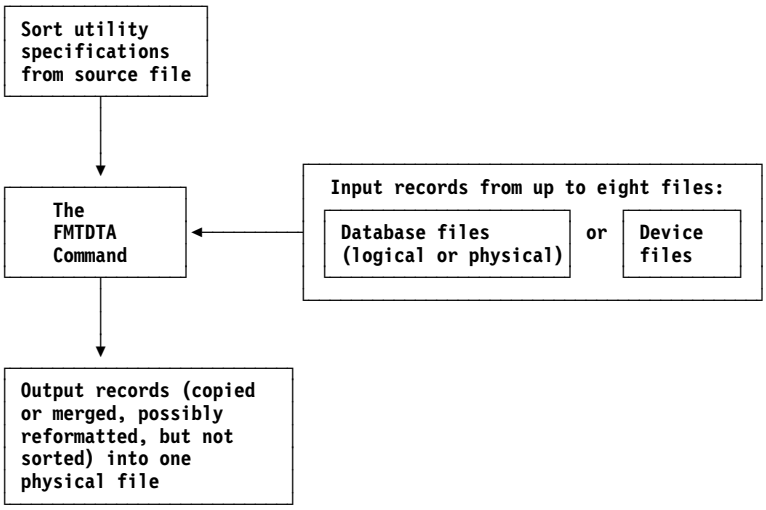


Figure 17. Copying or Merging One or More Files into One Physical File

The output is a physical file in the database, meaning that the output file contains a sorted and reformatted copy of the actual input records.

Copying records in this way has two aspects:

- Specifying Sort options with no reformatting of records
- Copying records and, at the same time, reformatting the output.

These two aspects follow the same rules as previously stated: for example, up to eight input files can be specified.

When copying and merging records from several files into one to produce a single physical output file, note the following:

- Enter `SORTR` as the output type on the header specification.
- Enter `0` as the maximum control field length, as there is to be no control field. This also implies that columns 15 to 39 of the header specification should be left blank, because the entries in these columns all pertain to the sorting process.
- Enter record specifications, if any conditions are placed on the input records selected for copying or merging.
- Enter field specifications, if the input record is reformatted in any way.

Examples

The following examples show the copying capabilities of Sort:

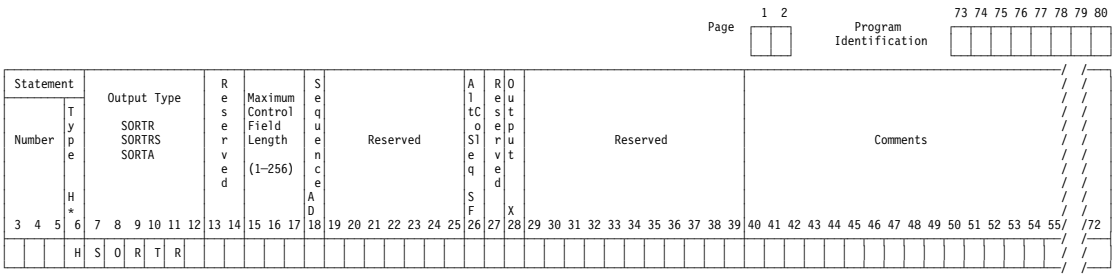
Example 1. Copying a File with No Reformatting

This example illustrates how to copy records from a file without sorting. Here records from three input files are to be merged into one output file. The header and field specifications do not specify any control field information. A number of "omit" record specifications (see "Column 6 (Specification Identifier)" on page 89) are present; records that meet the criteria of the specified comparisons are not to be included in the output file. Following the "omit" specifications is an "include-all" specification which specifies that all records not meeting the criteria of the specified "omit" comparisons are to be included in the output file. All data fields in the input records are included in the output records.

Entering the Specifications

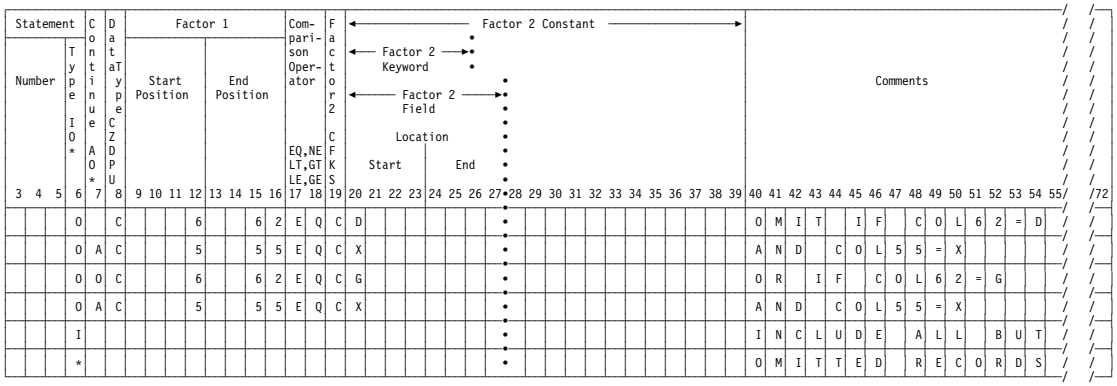
Complete the following specifications. (Only those columns that are important to this example are described.)

HEADER SPECIFICATIONS



This header specification states that a copy-only request is performed. No control field information is needed on a copy-only request because the records are copied in the order they are found. No sorting is done.

RECORD SPECIFICATIONS



These omit record specifications indicate which records are not included in the output file. The significant column entries are:

Columns	Entry	Explanation
6	O	The records that meet these conditions are omitted from the copy.
7	A or O	These records continue to specify the conditions that the record must meet in order to be omitted from the copy. An input record is omitted if position 62 contains a D and position 55 contains an X, or if position 62 contains a G and position 55 contains an X.
8	C	Character data is used for the comparisons.
15-16	62, 55	Identify which fields in the input record are to meet certain conditions.
17-18	EQ	The condition is that the value found in those fields must be equal to the Factor 2 field.
19	C	The value of the Factor 2 field is to be a constant.
20	D, X, G	The constant value is to be a D, X, or G, respectively.

The last record specification is an include-all statement, meaning that the only entry on that record specification is the sequence number and the I in column 6. This causes all the records not omitted by the omit specifications to be included in the copy.

FIELD SPECIFICATIONS

Statement Number	F Type	Field	Data Type	Field Location Start	Field Location End	Forced Characters Record	Forced Characters Substitute	Forced Characters Continuation	Overflow Field Length	Reserved	Comments
3 4 5	F* 6	DNOFS* 7	C,KZ,LD,EP,IV,JU,RM,SH,TB 8	9 10 11 12	13 14 15 16	17	18	19	...A.l.tS.e.q. . . . A. 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 / /72
	F	D	C						•		INCLUDE ALL INPUT
	*								•		FIELDS
									-		

This field specification states that all the input data fields are to be included in the output records. The data is character data.

Processing the Specifications

Below is an example of FMTDTA command that could be used to run this Sort job:

Specified on FMTDTA Command	Explanation
FMTDTA INFILE((PROD/DISKET1 MONDAY))+	Specifies input file in library PROD.
OUTFILE(PROD/TRANSM)+	The output file is TRANSM in library PROD. Since a member name is not specified, the output records are stored in the first member in the file.
SRCFILE(TEST/EXAMPLE)+ SRCMBR(EXAMPLE4)+	The reformat specifications are taken from member EXAMPLE4 in data file EXAMPLE, which is in library TEST.

The OPTION parameter defaults to *CHK, *PRT, *NODUMP, and *NOSECLVL.

The plus sign (+) shown for continuations is not required if the FMTDTA *command is entered from a display station.*

Example 2. Copying/Merging Several Files to Produce a Reformatted Output File

This example copies the same records from the same files as “Example 1. Copying a File with No Reformatting” on page 68. The fields in the output record, however, are to be rearranged.

Entering the Specifications

The header and record specifications are the same as for “Example 1. Copying a File with No Reformatting” on page 68.

The field specifications are as follows (only the column entries that are important to this example are described):

FIELD SPECIFICATIONS

Statement Number	Type	Field FNOFS*	Data Type C,KZ,LD,EP,IV,JU,RM,SH,TB	Field Location Start	Field Location End	Forced Characters Record	Forced Characters Substitute	Forced Characters Continuation	Overflow Field Length ...A.lt.Se.q.....A.	Reserved	Comments
3 4 5		6	7 8	9 10 11 12	13 14 15 16	17	18	19	20.21 22	23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39	40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 /72
		F	D C	7 2	1 2 8				•		F I E L D 7
		F	D C	6 6	7 1				•		F I E L D 6
		F	D C	5 2	6 5				•		F I E L D 5
		F	D C	4 8	5 1				•		F I E L D 4
		F	D C	2 2	4 7				•		F I E L D 3
		F	D C	3	2 1				•		F I E L D 2
		F	D C	1	2				•		F I E L D 1
									•		

Note that the entries in columns 9-16 reorder the fields on the output record in the reverse order from what they are in in the input record. Only the fields that are specified will appear in the output record.

Processing These Specifications

Assuming that the input records are in the same input files, the specifications are in the same source file, and the output file is to be the same output file as in “Example 1. Copying a File with No Reformatting” on page 68, enter the same FMTDTA command as in “Example 1. Copying a File with No Reformatting” on page 68 in order to process the specifications. See below.

Specified on FMTDTA Command	Explanation
FMTDTA INFILE((PROD/DISKET1 MONDAY)+ (PROD/DISKET2)+ (PROD/DISKET3))+	Specifies three input files, all in library PROD.
OUTFILE(PROD/TRANSM)+	The output file is TRANSM in library PROD. Since a member name is not specified, the output records are stored in the first member of the file.
SRCFILE(TEST/EXAMPLE)+ SRCMBR(EXAMPLE4)+	The reformat specifications are taken from member EXAMPLE4 in data file EXAMPLE, which is in library TEST.

The OPTION parameter defaults to *CHK, *PRT, *NODUMP, and *NOSECLVL.

The plus sign (+) shown for continuations is not required if the FMTDTA *command is entered from a display station.*

Chapter 7. The Format Data (FMTDTA) Command

The CL command Format Data (FMTDTA), when used with the appropriate parameters, processes a series of Sort specifications previously stored as a member in a library. This chapter describes the parameters and options of this command, and how to use it.

You **must** enter values for the first two parameters, `INFILE` and `OUTFILE`. The remaining parameters are optional. Default values are assumed for parameters that you omit.

Accessing and Using the FMTDTA Command

There are two ways of accessing and running the FMTDTA command:

- ***From a CL prompt display or a command menu***. To access the FMTDTA prompt displays, type `FMTDTA` on a command line and press F4. Alternatively, you can select FMTDTA from a command menu. Enter the appropriate parameters, and press Enter to process the sort job.
- ***As a line command***. If you are familiar with both the CL command language, and with the FMTDTA command and its parameters, you can type the FMTDTA command with its parameters on a command line, and press Enter to process the command.

File Naming Conventions

When identifying files and members in the system, you must use the appropriate convention. For example, `LIBRAR1/FILE1 MEM3` identifies member `MEM3` of file `FILE1` in library `LIBRAR1`. If you do not specify a member, the system defaults to the first member in a file. If a file contains no members, then the whole file is considered as one member.

Syntax of the FMTDTA Command

The syntax you use when entering the FMTDTA command as a line command is shown below.

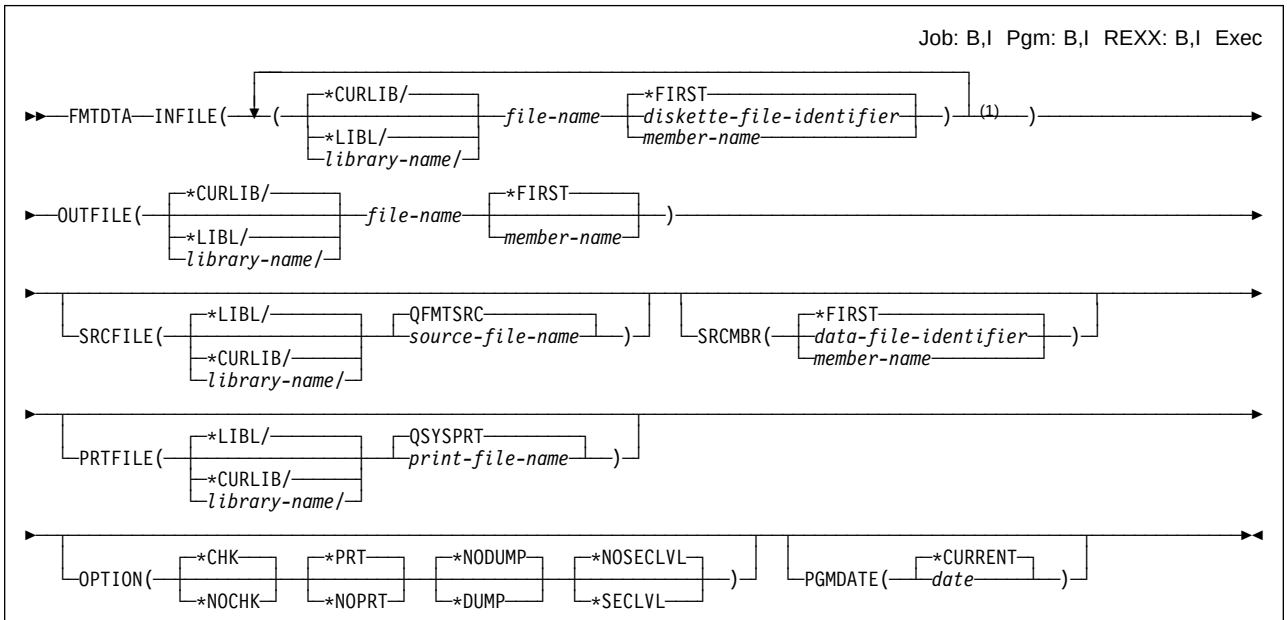


Figure 18. Syntax of the FMTDTA Command

The FMTDTA Displays

Type `FMTDTA` with no parameters and press F4 to pull up the first FMTDTA prompt display.

```
                          Format Data (FMTDTA)

Type choices, press Enter.
                                 _
Input file:
  File:  . . . . . . . . . . .   _________     Name
    Library: . . . . . . . . .     *CURLIB___  Name, *CURLIB, *LIBL
  Member:  . . . . . . . . . .   *FIRST____    Name, *FIRST
             + for more values _
Output file:
  File:  . . . . . . . . . . .   _________     Name
    Library: . . . . . . . . .     *CURLIB___  Name, *CURLIB, *LIBL
  Member:  . . . . . . . . . .   *FIRST____    Name, *FIRST
Source:  . . . . . . . . . . .   QFMTSRC___    Name
  Library: . . . . . . . . . .     *LIBL_____  Name, *CURLIB, *LIBL
Source:  . . . . . . . . . . .   *FIRST_____   Name, *FIRST



                                                                    Bottom
F3=Exit   F4=Prompt   F5=Refresh   F10=Additional parameters   F12=Cancel
F13=How to use this display   F24=More Keys
```

Figure 19. The First FMTDTA Screen

You must enter *at least* a value for Input File Name, and a value for Output File Name before you run the sort job. These names must already exist in the system. All other parameters have default values, which you can change if necessary. Press F10 to display additional parameters.

Additional parameters are found on this display.

```
                              Format Data (FMTDTA)

Type choices, press Enter.


                            Additional Parameters

Print file:  . . . . . .  .   QSYSPRT___     Name
  Library: . . . . . . . .     *LIBL_____    Name, *LIBL, *CURLIB
Options: . . . . . . . . .    _________      *CHK, *NOCHK, *PRT, *NOPRT...
               + for more values  _________
Program date:  . . . . . .  .  *CURRENT      Date, *CURRENT









                                                                      Bottom
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More Keys
```

Figure 20. The Second FMTDTA Display

When you have entered the option values you need, Press Enter to begin processing, or press F3 to exit without processing the command.

The Parameters of the FMTDTA Command

In the description of the parameters that follow, the defaults appear first. The parameters are presented in the sequence you would follow when entering the command on the command line. The parameter keyword is shown first, followed by the corresponding prompt that appears on the FMTDTA prompt display.

INFILE (*Input file*)

Specifies the names of the libraries, files, and members from which the input records are selected. If you type a plus sign (+) into this field, you can enter up to eight input files.

file_name

Enter the name of the file. This is a required parameter.

library_name

Enter any of the following values, or accept the default.

***CURLIB**

The current library. If you have not specified a current library, QGPL is used.

***LIBL**
The current library list. The system searches this list for the input file. The first file it finds with the specified file name is the one it searches for the input records.

library_name
The name of the particular library where the input file and member are to be found.

member_name
Enter any of the following values.

***FIRST**
The first member of the input file.

data_file_identifier
For diskette files, enter one data file identifier per diskette unit file name specified. If more than one diskette data file is being specified as the input file, the data file identifier must be entered for each.

member_name
For database files, enter the name of the member in which the input records are to be found.

OUTFILE (*Output file*)
Specifies the library, file, and member where the output (sorted and reformatted) records are stored.

file_name
Enter the name of the file. You must enter an option for this field.

library_name
Enter any of the following values, or accept the default.

***CURLIB**
The current library. If you have not specified a current library, QGPL is used.

***LIBL**
The current library list. The system will search this list for the output file. The first file it finds with the specified file name is the one into which it will put the output records.

library_name
The name of the particular library where the output file and member are to be found.

member_name
Enter any of the following values, or accept the default.

***FIRST**
The first member of the output file is to contain the output records.

member_name
Enter the name of the member into which the output records are to be put.

SRCFILE (*Source file name*)
Specifies the library and file where the specifications for this sort job are to be found. The possible values are:

QFMTSRC
An IBM-supplied source file.

source_file_name
Enter the name of a particular source file.

library_name
Enter any of the following values, or accept the default.

***LIBL**
The current library list. The system searches this list for the source file. The first file it finds with the specified file name is the one that it attempts to process for instructions.

***CURLIB**
The current library. If you have not specified a current library, QGPL is used.

library_name
The name of the particular library where the source file and member are to be found.

SRCMBR (*Source member name*)
Specifies the name of the member in which the source specifications are to be found. Enter any of the following values, or accept the default.

***FIRST**
Sort searches for the sort specifications in the first member of the source file.

data_file_identifier
For diskette files, enter the data file identifier for the diskette unit file name specified.

member_name
For database files, enter the name of the member in which the source specifications are to be found.

The following parameters are shown on the second FMTDTA display:

PRTFILE (*Print file name*)
Specifies the library and name of the printer device file to which the listing will be written. The possible values for the file name are:

QSYSPRT
The contents of this file are printed automatically by the system printer.

print_file_name
Enter the name of a particular printer device file.

library_name
The name of the library containing the print file. Possible values are:

***LIBL**

The current library list. The system searches this list for the printer device file. The first file it finds with the specified file name is the one that it uses as the printer device file.

**CURLIB*

The current library. If you have not specified a current library, QGPL is used.

library_name

The name of the particular library where the printer device file is to be found.

OPTION (*Options*)

Specifies the sequence-checking and print options to be used. If you specify more than one value for an option, the last value specified will be the value that is used. Enter a plus sign (+) into the field to the right of the **+ for more values** to enter more than one value. The possible options and their values are:

***CHK**

The Sort specifications are sequence-checked. The system verifies if the sequence numbers of the job specifications in the source file match their order in the source file. If not, a warning message is issued.

***NOCHK**

The Sort specifications are not sequence checked.

***PRT**

The Sort specifications for this job, as well as any error and warning messages issued by the system, are printed.

***NOPRT**

The specifications and messages are not printed.

***NODUMP**

The internal tables used for this job are not printed.

***DUMP**

The internal tables used for this job are printed. These tables can then help you isolate the cause of any particular problem.

***NOSECLVL**

Do not list second-level messages for this job.

***SECLVL**

List second-level messages for this job.

PGMDATE (*Program date*)

Specifies the date that is used with Factor 2 in the record specifications described in Chapter 9, “Record Specifications” The system can search for particular input records that contain the same, greater, or lesser value than this date. Possible values are:

***CURRENT**

The current date the job is being run.

***DATE**

Use the date as entered with the CL command QDHTFMT; if separators are used, use the date as entered with the QDATSEP command.

Chapter 8. The Header Specification

The header specification must be the first statement (excluding comments) in any set of Sort specifications. It provides the following information:

- The type of output file: a physical file, a summary data file, or a Record Address (RA) file
- The length of the control field
- The Sort order: ascending or descending
- Whether the standard or an alternative collating sequence is used
- Whether or not the control field is to be included in the output
- The name of the file in which this set of Sort specifications is to be stored.

HEADER SPECIFICATIONS

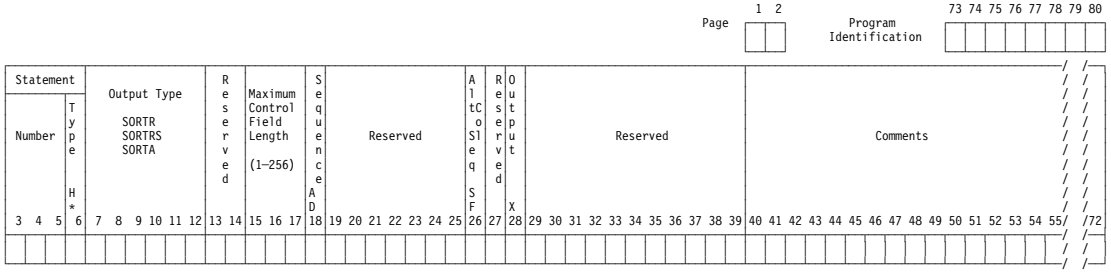


Figure 21. The Form for Entering Header Specifications

Summary of Column Entries

Table 1 summarizes the column entries for header specifications.

Table 1. Summary of Column Entries for the Header Specification

Columns	Entry	Explanation
1-2	00-99	Page number (no entry required).
3-5	000-999	Statement number (no entry required).
6	H	Identifies this specification as a header specification.
	*	Identifies this specification as a comment specification.
7-12	SORTR	Input records are to be sorted in a regular manner.
	SORTRS	Input records are to be sorted and summed.
	SORTA	Output file is to contain only the relative record numbers of the sorted records (record address (RA) sort).
13-14	Blank	Reserved.
15-17	Blank	No control field (copy-only request).
	1-256	The maximum control field length.
18	Blank	Output order is ascending by control field.
	A	Output order is ascending by control field.
	D	Output order is descending by control field.
19-25	Blank	Reserved.
26	Blank	Standard collating sequence is used.
	S	An alternative collating sequence is used for the entire control field.
	F	Use an alternative collating sequence for parts of the specified control field.
27	Blank	Reserved.
28	Blank	The control field is included in the output record.
	X	The control field is not included in the output record.
29-39	Blank	Reserved.
40-72	Any entry	Comments.
73-80	Any entry	Program Identification (treated as a comment by the system).

Detailed Description of Each Column Entry

Columns 1 and 2 (Page)
You can enter any number from 0 through 99 in columns 1 and 2 for page numbers. The pages should be numbered in ascending order.

Columns 3 through 5 (Statement Number)
These columns specify the statement number.

Column 6 (Specification Identifier)
This column must contain an H to identify this specification as a header specification. If it contains an *, the specification will be treated as a comment.

Columns 7 through 12 (Output File Type)
These columns identify the type of output to be produced. Possible entries are:

SORTR Output is to be a set of physical records sorted as specified (a physical file).

SORTRS Output is to be a set of physical records (a physical file). There is to be only one output record for each input record type; the output record contains the summed totals of given fields for all input records of that type.

SORTA Output is to be a list of addresses of the records that were sorted (a record address file). These addresses are pointers to the input file.

Note: You must put an X into column 28 in order to omit the control field from the output record. The presence of control field data in an RA file would give meaningless results.

Columns 13 and 14 (Reserved)
These columns must be blank.

Columns 15 through 17 (Maximum Control Field Length)
These columns contain a number from 0 to 256, which will be the length used for the control field in the output record. Shorter control fields will be left-justified in this space and padded on the right with zeros.

The number entered must be right-justified. For example, if you enter a 6 into column 16 only, the system would assume that you entered a 60. The number must be equal to or greater than the sum of the longest control field (as specified in the field specifications).

Control fields are composed of one or more fields taken from the input records. A different grouping can be specified for each input record type. The entry in this field must equal or exceed the length of the largest grouping.

Copy-Only Request
If columns 15 through 17 are left blank, no sorting or reformatting occurs. This is equivalent to specifying that records are to be copied or merged only, as described in Chapter 6, “Copying and Merging Records from Several Files into One.”

Column 18 (Sort Sequence)

The entry in this column specifies how the output records are to be sorted. Possible entries are:

Blank Ascending order

A Ascending order

D Descending order.

Note: For copy-only requests, this column must be left blank.

The sorting of input records is based on the contents of the control field for the record type. Input records are sorted according to this order unless this order is overridden by the field specifications for the particular control field.

Columns 19 through 25 (Reserved)

These columns must be blank.

Column 26 (Alternative Collating Sequence)

Column 26 defines whether the standard or alternative collating sequence is used. (See Appendix C, “Standard and Alternative Collating Sequences”). The possible entries are:

Blank Standard collating sequence

S Alternative collating sequence across the entire control field

F Alternative collating sequence for certain columns of the control field as specified by the field specifications for the control field.

Note: For copy-only requests, this column must be left blank.

If an alternative collating sequence is specified, you must enter the appropriate `ALTSEQ` specifications to define the alternative collating sequence. `ALTSEQ` statements, if used, must immediately follow the header specification.

Column 27 (Reserved)

This column must be blank.

Column 28 (Include/Exclude Control Field in Output)

This column specifies whether or not the control field is to appear in the output records. Possible entries are:

Blank The control field will appear as the first field in the output record

X The control field will not appear in the output record.

Notes:

1. Column 28 must contain an `X` for an RA sort.
2. The control field will appear in the output records in the same format it was in when used for processing the records. If, however, it was forced in any way (altered by the field specifications) or specified as being in packed format, the control field data will appear quite different from the source fields from which it was constructed.

Columns 29 through 39 (Reserved)
These columns must be blank.

Columns 40 through 72 (Comments)
These columns are available for comments.

Columns 73 through 80 (Program Identification)
Any name can be entered in these columns. This entry is treated as a comment.

Chapter 9. Record Specifications

Use record specifications to select the input records you want to process or to define which records you want to omit from processing. If all input records are to be processed, these specifications are not necessary.

Sort treats all input files as unformatted strings of bytes. You must divide the unformatted input file into the fields that are to be sorted or reformatted.

The record specification identifies the input records to be used (sorted or omitted) by identifying two fields, called Factor 1 and Factor 2. It states what the contents of these two fields should be, and what the relationship of these contents should be to each other in order for the input record to be processed. During processing, the Sort utility scans the input file for records whose contents match these conditions, and includes (or omits) them from the sort.

Set of Record Specifications

You can use one or more record specifications to establish all the conditions you want the input record to meet before being selected for sorting and reformatting.

Record Type

Any difference in either the conditions established for the selection of an input record, or the field specifications that define how these input records are to be processed, is considered as a difference in record type. Each new record type must be defined by its own set of record specifications.

RECORD SPECIFICATIONS

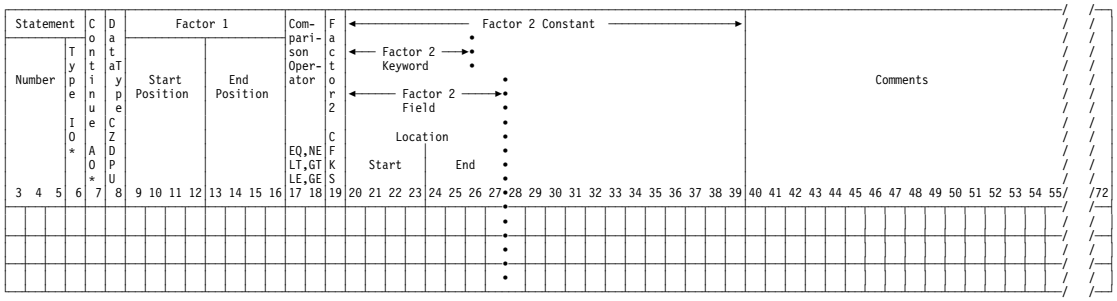


Figure 22. Form Used to Enter Record Specifications

Summary of Column Entries

Table 2 summarizes the column entries for the record statements.

Table 2 (Page 1 of 2). Summary of Column Entries for the Record Specification

Columns	Entry	Explanation
1-2	00-99	Page number (no entry required).
3-5	000-999	Statement number (no entry required).

Table 2 (Page 2 of 2). Summary of Column Entries for the Record Specification

Columns	Entry	Explanation
6	I	The record is to be included in the sort.
	O	The record is to be omitted from the sort.
	*	Identifies this specification as a comment specification.
7	Blank	This specification begins the definition of a new record type.
	A	This specification is in an AND relationship with the previous record specification.
	O	This specification is in an OR relationship with the previous record specification.
	*	Identifies this specification as a comment specification.
8	C	Factor 1 and Factor 2 must contain character data.
	Z	Factor 1 and Factor 2 must contain character data, and the zone portions of the binary representation of the characters are to be used for the comparison.
	D	Factor 1 and Factor 2 must contain character data, and the digit portions of the characters are to be used for the comparison.
	P	Factor 1 must contain signed, packed numeric data. If Factor 2 is defined, it must also contain signed, packed numeric data.
	U	Factor 1 must contain signed, zoned numeric data. If Factor 2 is defined, it must also contain signed, zoned numeric data.
9-12	1-9999	The start position of the Factor 1 field. This can be left blank.
13-16	1-9999	The end position of the Factor 1 field. If columns 9 through 12 are blank, this entry defines a Factor 1 field that is one character long in the position specified.
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	C	Factor 2 is a constant.
	F	Factor 2 is a field in the input record.
	K	Factor 2 is a keyword.
	S	Factor 2 is a constant, but it is shifted one character to the left before comparison.
20-23	1-9999	The start position of the Factor 2 field. This can be left blank if the Factor 2 field is one character long.
24-27	1-9999	The end position of the Factor 2 field. If columns 20 through 23 are blank, this entry defines a Factor 2 field that is one character long in the position specified.
20-39	Any entry	If columns 20 through 27 are not being used to define the start and end positions of the Factor 2 field, then columns 20 through 39 can be used to specify a constant or keyword against which Factor 1 is compared.
40-72	Any entry	Comments.

Detailed Description of Each Column Entry

Columns 1 and 2 (Page) and Columns 3 through 5 (Statement Number)

Together, these columns form a 5-digit sequence number.

Use the *CHK option of the FMTDTA command (see Chapter 7, "The Format Data (FMTDTA) Command") to check for out-of-sequence conditions in the Sort specification set.

Note: Alternative collating sequence (ALTSEQ) statements are not sequence-checked, but must immediately follow the header statement.

Column 6 (Specification Identifier)

This column identifies the specification as a record specification, and indicates whether the defined record is to be included in or omitted from the sort, or if it is to be treated as a comment. Possible entries are:

I The record that is being identified is to be included. These record specifications are also known as *include* specifications.

O The record that is being identified is to be omitted. These record specifications are also known as *omit* specifications.

***** The record specification is being treated as a comment.

If there are no conditions specified (meaning that columns 7 to 39 of the record specification are blank), Sort will use all the input records that were not used by previous include or omit record specification sets. Such a record specification, also known as an *include-all* specification, must be the last in a series of record specification sets.

Only one such blank record specification can be used per sort job.

You can also omit the blank record specification. If field specifications are entered on their own (that is, without being preceded by record specifications for a record type), Sort assumes the field specifications apply to all the remaining unprocessed input records. Again, such field specifications must be the last ones entered for the particular sort job.

Column 7 (Begin/Continue Record Definition)

Column 7 is used to either begin a record specification set, or to continue one by specifying new criteria to be met. The possible entries are:

Blank This specification is the first of a set of include or omit specifications for a particular input record type.

A This specification is part of the definition of a record type. It is in an AND relationship with the previous specification: the conditions in both this specification and the previous one (in this set) must be met for the record to be used (or omitted).

0 This specification is part of the definition of a record type. It is in an OR relationship with the previous specification: the conditions in either this specification or the previous one must be met in order for this type of record to be used (or omitted).

***** The record specification is treated as a comment.

The following tables show the combination of entries you can make in columns 6 and 7 for sets of include and omit specifications.

Table 3 show the entries for include sets.

Table 3. Include Sets

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

Note: Every include set must end with field statements. An include set can be followed by another include set or an omit set. Records not described in include sets will not be sorted.

Table 4 show the entries for include sets.

Table 4. Omit Sets

Type of Set	Column 6 Entry	Column 7 Entry	Explanation
Omit AND statements (one record type)	H or F		Header statement, or field statement (last statement of include set.
	O	␣	New record type indicated by a blank in column 7.
	O	A	Statements that describe the same record type (as the previous statement) have an A in column 7.
Omit OR statements (different record types)	H or F		Header statement or field statement (last statement of include set).
	O	␣	New record type indicated by a blank in column 7.
	O	O	Statements that describe a different record type (than the previous statement) have a O in column 7.
Omit AND and OR statement (different record types)	H or F		Header statement or field statement (last statement of include set).
	O	␣	New record type indicated by a blank in column 7.
	O	A	Statements that describe the same record type (as the previous statement) have an A in column 7.
	O	O	Statements that describe a different record type (than the previous statement) have an O in column 7.

Note: There are no field statements in omit sets. Each omit set must be followed by an include or an include-all set.

Column 8 (Data Type)

This column entry tells the Sort utility how to interpret data in the Factor 1 and Factor 2 fields. The fields are defined as either alphanumeric or numeric.

- ***Alphanumeric fields*** are composed of characters, and, in EBCDIC, each character byte is assumed to be composed of eight bits. The first four bits form the zone portion; the last four bits form the digit portion.
- ***Numeric fields*** are composed only of the digits 0 to 9 plus the sign of the number (+ or -). In unpacked format, each digit is composed of eight bits: the first four bits form the zone portion, and the last four bits form the digit portion. The zone portion of the rightmost byte represents the sign. (Unpacked format is also called zoned format, because it is the addition of a zone portion to each digit that “unpacks” the format.) In packed format, each digit is composed of only four bits. The rightmost four bits of the entire number represent the sign for the number, rather than a digit.

Because Sort reads the input record as a sequence of unformatted bytes (and bits), column 8 specifies how many bits are to be assigned to each character. For example, in packed format each digit is represented by four bits; in zoned format, by eight bits.

When the fields contain alphanumeric data, either the digit portion, the zone portion, or both portions of each character can be used. When the fields contain signed numeric data, you must specify whether the format is to be unpacked or packed. Depending on which type of field you specify, there are certain restrictions on the length of Factor 1 and Factor 2 fields. The possible entries are shown in Figure 23.

	Column 8 Entry	Compare Operations	Maximum Field Length *
Alpha-numeric Data	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 portion of the character	16 characters
Signed Numeric Data	P	Numeric data is packed.**	8 bytes, or 15 digits plus sign
	U**	Numeric data is unpacked (zoned) **	16 digits

* For both Factor 1 and Factor 2 fields

** Do not use a packed or zoned data type for Factor 1 field in an include or omit specification (P or U in column 8) if you specify an alternative collating sequence (S in column 26) in the header specification

Figure 23. Column 8 Restrictions on Length of Factor 1 and Factor 2 Fields

Note: If an alternative collating sequence is specified (`ALTSEQ` statements are present), column 8 cannot contain a `P` or a `U`. Only alphanumeric data can be compared.

Interpreting Alphanumeric Data

Each EBCDIC character has two parts: a zone portion and a digit portion. Some characters may have identical zone portions or identical digit positions, but no two characters can have an identical combination of zone and digit portions. See Table 5.

Table 5. Binary Representation of Selected AS/400 Characters

	Binary Representation	
AS/400 Character	**Zone Portion**	**Digit Portion**
*	0101	1100
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
0	1111	0000
K	1101	0010
?	0110	1111
P	1101	0111
blank	0100	0000 [1]
0	1111	0000 [1]
⋮		

[1] Note that the digit portion of a zero and a blank are exactly the same.

If you instruct the Sort utility to use only the digit portions of characters (by putting a `D` into column 8), characters with identical digit portions will look alike and compare as equal. Likewise, if you instruct it to use only the zone position of characters (by putting a `Z` into column 8), characters with identical zone positions will look alike and compare as equal. Thus the column 8 entry is critical to ensuring that the compare operations produce the results you intend.

For example, suppose you want only those records with a 2 in column 15 and a 2 in column 50. If you put a `D` into column 8, you will get the records you want, but you may also get many records you do *not* want (several characters have the same digit position as a `2`). To get *only* the desired records, with a `2` in column 15 and a `2` in column 50, you must put a `C` into column 8 instead. The `C` tells Sort to use both the zone and digit portions of characters in its compare operations, and no other character has the same combination of zone and digit portions as the ones you want.

Interpreting Numeric Data

If you specify that the Factor 1 and Factor 2 fields are to contain numeric data in packed format, Sort reads each four bits as a separate digit up to the last four bits in the field. These last four bits determine whether the number being read is positive or negative.

If you specify that these fields are to contain numeric data in zoned (unpacked) format, Sort reads the digit portion of each set of eight bytes up to and including the last byte. This determines the number in the field. Sort also reads the zone portion of the last byte. The value in that portion determines whether the number is to be positive or negative.

Table 5 shows the binary representation of the ten digits. Figure 24 shows some examples of packed and zoned numbers, and their signs.

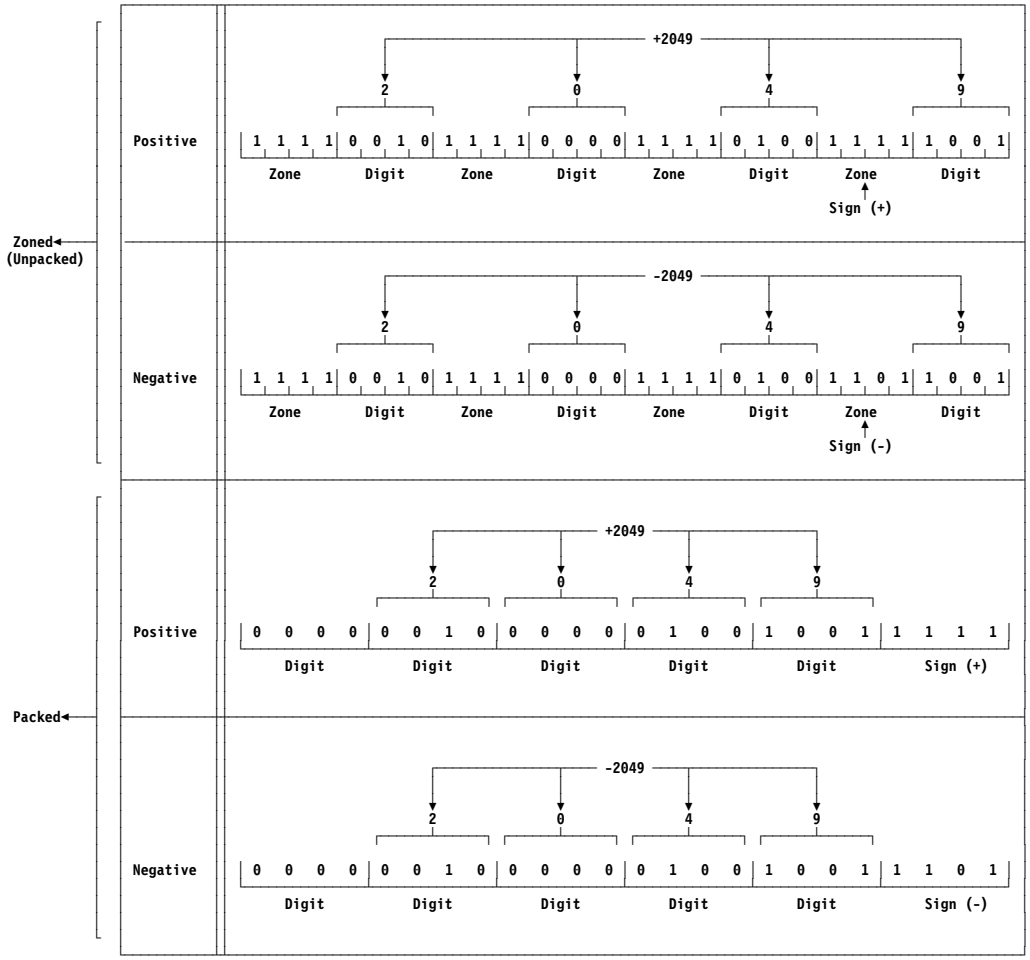


Figure 24. Sign Placement in Packed and Zoned (Unpacked) Numbers

Interpreting the Sign of a Number

Numbers can be either positive or negative. The sign of a number is indicated by a 4-bit binary code, as shown in Table 6.

Table 6. Binary Representation of Signs

Sign	Binary Code
+	1111 [1]
	1010
	1100
	1110
–	1101
	1011

1 Standard Form. The Sort Utility accepts all four forms of the plus sign. Before sorting the file, however, the program converts all forms of the plus sign to the standard form (1111). If you print a dump, the plus sign will always be expressed as a hexadecimal F.

If you specify alphanumeric data, digit portion only (a `D` in column 8), then Sort interprets each set of four bits as a digit, and automatically assumes that the resulting number is positive.

Columns 9 through 12 (Factor 1 Start Position)

The entry in columns 9 through 12 identifies where in the input record the Factor 1 field begins. This entry must be right-justified.

Note that the difference between this entry and that entered into columns 13 through 16 (Factor 1 End Position) determines the length of the Factor 1 field. This length can be from 1 to 256 characters, but is subject to some restrictions, as shown in Table 7.

Table 7. Restrictions on the Maximum Length of the Factor 1 Field

Restriction	Maximum Factor 1 Field Length
Factor 2 field is a constant.	20 characters
Factor 2 field is the keyword `UDATE`.	6 characters
Factor 2 field is the keyword `UMONTH`, `UDATE`, or `UYEAR`.	2 characters
Factor 1 field is alphanumeric, but only the zone portion of each character is to be used (column 8 contains a `Z`).	1 character
Factor 1 field is alphanumeric, but only the digit portion of the character is to be used (column 8 contains a `D`).	16 characters
Factor 1 field is numeric in packed format (column 8 contains a `P`).	8 bytes, equivalent to 15 digits and the sign
Factor 1 field is numeric in zoned format (column 8 contains a `U`).	16 bytes, equivalent to 16 digits and the sign (the sign is included in the last byte)

Note: To describe a one-character Factor 1 field, leave columns 9 through 12 blank and enter the number of the record position that contains the character into columns 13 through 16 (end position).

Columns 13 through 16 (Factor 1 End Position)

The entry in these columns identifies the end position of the Factor 1 field in the input record. The entry must be right-justified. If columns 9 through 12 are blank, an entry in columns 13 through 16 defines the location of a one-character field in the input record.

Columns 17 and 18 (Comparison Operator)

These columns specify how the contents of the Factor 1 and Factor 2 fields must compare in order for the input record to be used. The possible entries are:

EQ Factor 1 must be equal to Factor 2.

NE Factor 1 must not be equal to 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.

The collating sequence (described in Appendix C, “Standard and Alternative Collating Sequences” and specified in the header specification) determines whether one alphanumeric value is greater than or less than another. If an alternative collating sequence is specified on the entire control field, then both Factor 1 and Factor 2 are modified before the two are compared.

If only the zone portions of characters are to be compared (column 8 contains a Z), then EQ or NE are the only two entries you can make.

Column 19 (Type of Data Contained in Factor 2)

This column identifies whether Factor 2 is data, a constant, or a keyword. The possible entries are:

F Factor 2 is to contain data from the input record.

C Factor 2 is a constant, as specified in columns 20 through 39.

K Factor 2 is the keyword specified in columns 20 through 25.

S Factor 2 is a double-byte character set (DBCS) constant, requiring a shift one position to the left.

Columns 20 through 23 (Factor 2 Start Position)

If column 19 contains an F, the entry in these columns identifies the starting position of the Factor 2 field in the input record. The entry must be right-justified.

Columns 24 through 27 (Factor 2 End Position)

If column 19 contains an F, the entry in these columns identifies the end position of the Factor 2 field in the input record. The length of Factor 2 (the end position minus the start position plus 1) must equal that of Factor 1.

To describe fields that are only one-character long, leave columns 20 through 23 (start position) blank, and enter the number of the position on the input record that contains the character in columns 24 through 27 (end position).

Columns 20 through 39 (Factor 2 Is a Constant)

If column 19 contains a C, Factor 2 is a constant. Use columns 20 through 39 to enter the constant. The constant must be the same length as the Factor 1 field (unless packed or zoned numeric data is specified), and must be right-justified within the constant length.

For comparisons to packed numeric fields (column 8 contains P), the number of numeric digits in the Factor 2 constant must be equal to twice the Factor 1 length minus 1 (the sign occupies only one position in both the packed field and the zoned constant). For comparisons to zoned numeric fields (U in column 8), the number of numeric digits in the Factor 2 constant must equal the Factor 1 length.

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

Alphanumeric constants can be combinations of letters, digits, punctuation, and special symbols–anything that appears in the EBCDIC collating sequence shown in Appendix C, “Standard and Alternative Collating Sequences.” 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 or U)

Format: Numeric constants must be right-justified within the field length specified in Factor 1 (within twice the field length minus one position to contain the sign if Factor 1 is a packed number). For example, assume that Factor 1 defines a six-position zoned field in the input record, and that Factor 2 is the numeric constant `123`. To right-justify the constant within the six positions specified, you must put it in columns 23, 24, and 25. Leading zeros are not required, and blanks and zeros look the same for numeric constants. The constant cannot be completely blank, however. In the previous example, columns 20 through 25 could contain either `000123` or `XXX123` (with `X` representing a blank).

Signed Constants: Negative constants can be designated either by setting the zone portion of the character representing the least significant digit to the value of hexadecimal D, or by placing a separate minus sign in the following position. When the sign is designated by zone, the decimal digits 0 to 9 will be replaced by the characters], and J to R, respectively.***:*** A constant consisting of decimal digits only is taken as positive. Positive values may also be designated using a separate plus sign.

RECORD SPECIFICATIONS

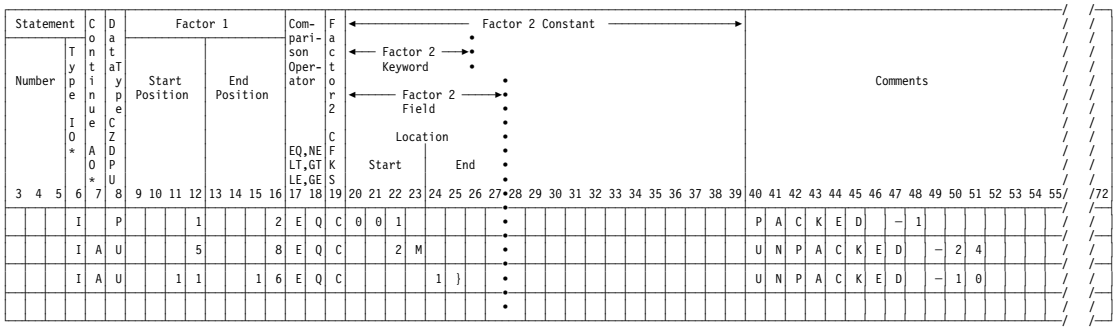


Figure 25. Examples of How to Code Negative Constants into Record Specifications

Columns 20 through 39 (Factor 2 Is a Keyword)

A `K` entered into column 19 identifies Factor 2 as a keyword that represents all or part of the program date. The keyword is entered into columns 20 through 39, starting with column 20. Unused columns should be left blank.

When Factor 2 is a keyword, Sort compares all or part of the program date with the Factor 1 field. The length of the Factor 1 field must be the same as the Factor 2 field. You can use the following keywords:

Keyword	Part of Program Date Compared	Factor 1 Field Length
`UDATE`	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

When Factor 2 is a keyword, you must indicate that both the zone and digit portions of characters must be compared to the Factor 1 field (by entering a C into column 8, to specify character data).

If UDATE is used, the program date must be in the same format as the date contained in the input records.

If Factor 2 is UDATE, then record selection on or before, or on or after, a certain date (determined by the comparison operator entered into columns 17 and 18) works only with the international date format (YYMMDD). If the program date and the input records date are not in the international date format, the keywords UYEAR, UMONTH, and UDAY, should be used to select the records.

The program date parameter of the FMTDTA command (see Chapter 7, “The Format Data (FMTDTA) Command”) provides the date, month, day, and year that will be used.

Columns 40 through 72 (Comments)

These columns are available for comments.

Chapter 10. Field Specifications

Field specifications define the fields in the input record. These can be control fields, data fields, and summary data fields.

Control fields are used to control the way in which the input record is sorted and reformatted, as described in Chapter 1, “Introduction.” *Data fields* identify those fields that contain data to be summed (see Chapter 5, “Sorting Records from a File and Producing Totals”), or fields that appear in the output records. *Summary data fields* are fields in the output record that contain the totals that are produced.

You can enter one or more field specifications for each field.

A series of field specifications must be entered for each record type. Record types are defined by record specifications, as described in Chapter 9, “Record Specifications.” Sort allows one series of field specifications to be entered without being preceded by the record specifications that define the record type. This can be done only once, and must be the last set of field-record specifications entered for the job. In this case, the system assumes that the record type is “all,” meaning that the field specifications pertain to all the remaining input records that have not yet been processed.

Order of Fields in the Output Record

The order of the fields in the output record will be the same as the order in which the fields are defined by the field specifications. It can be the same or different for each record type that is specified.

Order of Field Specifications in the Source File

The order of the field specifications relative to each other and to the header, record, and comment specifications, is determined by the specification sequence number entered into columns 1 through 5.

FIELD SPECIFICATIONS

Statement Number	Statement Typ	Field FNOFS*	Data Typ C,KZ,LD,EP,IV,JU,RM,SH,TB	Field Location Start	Field Location End	Forced Characters Record	Forced Characters Substitute	Forced Characters Continuation	Overflow Field Length Alt.Seq.A	Reserved	Comments
3 4 5	6	7	8	9 10 11 12	13 14 15 16	17	18	19	20.21 22	23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39	40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55/ /72

Figure 26. Form for Entering Field Specifications

Summary of Column Entries

Table 8 summarizes the column entries for field statements.

Table 8 (Page 1 of 2). Summary of Column Entries for the Field Specifications

Columns	Entry	Explanation
1-2	00-99	Page number (no entry required).
3-5	000-999	Statement number (no entry required).
6	F	Identifies this specification as a field specification.
7	D	The field is a data field.
	N	The field is a normal control field.
	O	The field is an opposite control field.
	F	The field is a forced control field.
	S	The field is a summary data field.
	*	Identifies this specification as a comment specification.
8	P	The field contains signed decimal data, in packed format.
	U	The field contains signed decimal data, in zoned format.
	C	The field contains character data.
	Z	Only the zone portion is used for sorting.
	D	Only the digit portion of each character is used for sorting.
	V	Force a single data character into the data field.
	F	Simplified Chinese DBCS control field using character-type, total strokes, radical sequence.
	G	Simplified Chinese DBCS control field using character-type, radical, strokes beyond radical sequence.
	N	Simplified Chinese DBCS control field using character-type, single-pronunciation, radical, total strokes, tie-breaker sequence.
	Q	Simplified Chinese DBCS control field using character-type, single-pronunciation, total strokes, radical, tie-breaker sequence.
	W	Simplified Chinese DBCS control field using character-type, single-pronunciation, tie-breaker, radical, total strokes sequence.
	X	Simplified Chinese DBCS control field using character-type, total strokes, radical, single-pronunciation, tie-breaker sequence.
	Y	Simplified Chinese DBCS control field using character-type.
	M	Traditional Chinese DBCS control field using character-type sequence.
	H	Traditional Chinese DBCS control field using character-type, stroke, radical sequence.
	B	Traditional Chinese DBCS control field using character-type, radical, strokes beyond radical sequence.
	K	Korean DBCS control field using primary Hangeul pronunciation of Hanja characters.
	L	Korean DBCS control field using secondary Hangeul pronunciation of Hanja characters that start Korean DBCS words, and primary Hangeul pronunciation of all other Hanja characters.
		Sei-On Katakana control field.
	E	Japanese DBCS control field using single-pronunciation, radical, stroke, tie-breaker
	I	sequence.
		Japanese DBCS control field using single-pronunciation, stroke, radical, tie-breaker
	J	sequence.
		Japanese DBCS control field using radical, stroke, tie-breaker
	R	sequence.
		Japanese DBCS control field using stroke, radical, tie-breaker
	S	sequence.
		Japanese DBCS control field using character-type sequence.
	T	

Table 8 (Page 2 of 2). Summary of Column Entries for the Field Specifications

Columns	Entry	Explanation
9-12	1-9999	The start position of the field. This entry can be left blank.
13-16	1-9999	The end position of the field. If columns 9 through 12 are blank, this entry defines a one-character field in the position specified in columns 13 through 16. If columns 9 through 12 and 13 through 16 are blank, the field is the entire input record.
17	Any character	The force character that is to be changed. Also used for summary overflow indicator fields; if so, this is the character to be used as the overflow indicator.
18	Any character	The force character that is to be substituted. Also used for summary overflow indicator fields; if so, this is the character that is to be put into the overflow indicator field if there is *no* overflow.
19	Blank Any character	This specification begins the definition of a field. This specification is a continuation of the previous field specification.
20-22	1-256 A (column 20 only)	Overflow field length – used for summary sort only. Alternative collating sequence by field.
23-39	Blank	Reserved
40-72	Any entry	Comments.

Detailed Description of Each Column Entry

Columns 1 and 2 (Page) and Columns 3 through 5 (Statement Number)

Together, these columns form a 5-digit sequence number.

Use the *CHK option of the FMTDTA command (see Chapter 7, "The Format Data (FMTDTA) Command") to check for out-of-sequence conditions in the Sort specification set.

Note: Alternative collating sequence (ALTSEQ) statements are not sequence-checked, but must immediately follow the header statement.

Column 6 (Specification Identifier)

This column must contain an F to identify this specification as a field specification. If it contains an * instead, the specification will be treated as a comment.

Column 7 (Field Type)

The entry here specifies the type of field. If the field is a *forced control field*, the way in which the control field is changed (forced) is further defined in columns 9 through 19. The possible entries into column 7 are:

D Data field.

N Normal control field. Input records (of the type described in the corresponding set of record specifications) are sorted so that the data in this field is in the sequence specified in column 18 of the header specification.

0 Opposite control field. Input records (of the type described in the corresponding set of record specifications) are sorted so that the data in this element of the control field is in the sequence opposite to that specified in column 18 of the header specification.

F Forced control field. Before any sorting is to be done, the contents of the control field must be altered according to the entries in columns 9 through 16, 17, 18, and 19.

S Data field to be summed. Used for summary sort only. The data found in this field in the input records (of the type described in the corresponding set of record specifications) is added together and the result placed in this field in the output records.

***** The record specification is treated as a comment.

If you are describing a control field, you can use the entry in column 8 to further specify which part of each character in the control is used for sorting or reformatting. Columns 8 through 19 allow you to control the sorting and reformatting more precisely. This can be useful for special types of sort, or to save space in memory and processing time during very long sorts.

Data Field (D in Column 7)

Data fields apply to copy and sort requests that produce physical output files (as opposed to RA output files). They are fields you want Sort to include in the output records. Control fields can also be specified as data fields, if you want them to appear in the output records. You can either globally request this in the header specification, or separately request it on a control field basis by respecifying this control field as a data field for this record type.

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 combined lengths of the data fields for each output record do not have to be the same for all record types. The Sort utility places blanks to the right of shorter output records so that all record lengths for all record types are equal to the output file record length.

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

Normal control fields sort according to the entry in column 18 of the header specification. Opposite control fields sort opposite to the entry found in column 18 of the header specification.

Forced Control Fields (F in Column 7)

You can change the contents of a control field (force them to be something else for the duration of the sort) before they are used to sort or reformat the input record.

There are three types of forced control fields:

- ***Conditional*** – depends on the data found in the input record.
- ***Conditional force-all*** – defines a value for records not meeting conditional force criteria.
- ***Unconditional*** – forces a specified character into a control field.

A control field that is forced can be only one character long.

The contents of columns 9 through 19 and the position of this specification relative to other field specifications in the same set, determine if the forcing is to be conditional, conditional force-all, or unconditional. See “Column 17 (Conditionally Forced Character)” on page 108, “Column 18 (Forced Character)” on page 108, and “Column 19 (Continuation)” on page 108 for information on how to fill out the field specifications when you are using forced control fields.

Conditional Forced Fields: A conditional force changes a character found in a control field element (the length of the control field element must be 1) to another character. Put the character you want to replace into column 17, and the character you want it replaced by into column 18. For example, if you want to replace all occurrences of `Z` in the first byte of the input record by a plus sign (+), complete the field specification as for a regular control field, and put a `1` in column 16, a `Z` in column 17, and a plus sign in column 18:

FIELD SPECIFICATIONS

Statement Number	Typ F*	Field DNOFS*	Data Typ C,KZ,LD,EP,IV,JU,RM,SH,TB	Field Location Start	Field Location End	Forced Characters Record	Forced Characters Substitute	Forced Characters Continuation	Overflow Field Length AlterSeq.. A.	Reserved	Comments
3 4 5	6	7	8	9 10 11 12	13 14 15 16	17	18	19	20.21 22	23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39	40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55/ /72
0 7	F	F	C			1	Z	+			

Sort does this replacement before using the control field to process the input record.

Conditional Force-All Fields: Force-all is a special type of conditional force. The force-all specification applies the change (as specified in column 18) to all those input records that have not previously had a conditional force applied to the control field. A force-all specification must be the last in a series of conditional force specifications. Note that you need one conditional force specification for each character you want to change in the control field. For example:

- 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` into it.

Enter a separate conditional force field specification for the first three cases, and a conditional force-all specification for the last case. A conditional force-all field is specified in the same way as an unconditional force field, except that it is related to the preceeding statements using a continuation character in column 19.

If you do not place a force-all statement after a conditional force line, and the Sort utility does not find the specified characters in the control field of the input record, then Sort will:

- Replace the control field character with `X'FF'` (if you specified ascending sequence in the header statement)
- Replace the control field character with `X'00'` (if you specified descending sequence in the header statement).

Note: If, however, the series of conditional force statements are entered as continuation specifications of a normal or opposite control field, any characters not conditionally forced will retain their original value.

Unconditional Forced Fields: An unconditional force places a defined character into a control field. To specify an unconditional force of a control field value, leave columns 9 through 17, and column 19 blank. Enter the required character in column 18.

Summary Data Field (S in Column 7)

A summary data field is one that is designated to hold selected accumulated totals. These totals can only be specified in summary sort (SORTRS) jobs. A maximum of 24 summary data fields can be defined for a record type.

In summary sort jobs, the data found in the field are added for all input records which have identical control field values. The same field in the output record is normally used to hold the totals, except in the case of an overflow.

In the case of a summary sort, only one output record is produced for each separate input record type specified. You can, in addition to specifying summary sort fields, also enter data field specifications. If you do so, however, the output record will contain the data field; the contents in that field will correspond to the contents of the data field found in the first input record (of that set) that was processed.

Column 8 (Data Type)

The column 8 entry here indicates what portion of the characters in the control field is used in sorting and reformatting the input records. Do not confuse it with the column 8 entry on the record specification, which helps select the input records for the job. This entry controls *how* those selected input records are to be sorted. You can also use it to specify the portions of data field characters that are to be transferred to the output record.

When you use this entry to change control field characters, the control field remains changed in the output records. Usually, this changed control field data is used only for the sorting and is otherwise meaningless; enter an X into column 28 of the header specification to exclude it from the output record. Or, use another field specification to respecify the original control field as a data field before entering the field specification that alters the contents of the control field.

Possible entries are shown in Table 9.

Table 9 (Page 1 of 3). Possible Column 8 Entries and Restrictions on Field Length

Col. 8 Entry	Character Portion Used	Maximum Field Length
C	Use both zone and digit portions of the characters.	999 characters (data) 256 characters (control field)
Z	Use only the zone portion of the character.	1 character
D	Use only the digit portion of the characters.	16 characters
U	The characters are interpreted as signed decimal numbers in zoned (unpacked) format. ⋆	16 digits
P	The characters are interpreted as signed decimal numbers in packed format.[1]	8 bytes or 15 digits and sign

Table 9 (Page 2 of 3). Possible Column 8 Entries and Restrictions on Field Length

Col. 8 Entry	Character Portion Used	Maximum Field Length
V	Force the replacement of the existing character with a new character.	1 character
F	Sort the Simplified Chinese DBCS control field into character-type, total strokes, radical sequence.	128 2-byte characters (=256 bytes)
G	Sort the Simplified Chinese DBCS control field into character-type, radical, strokes beyond radical sequence.	128 2-byte characters (=256 bytes)
N	Sort the Simplified Chinese DBCS control field into character-type, single-pronunciation, radical, total strokes, tie-breaker sequence.	128 2-byte characters (=256 bytes)
Q	Sort the Simplified Chinese DBCS control field into character-type, single-pronunciation, total strokes, radical, tie-breaker sequence.	128 2-byte characters (=256 bytes)
W	Sort the Simplified Chinese DBCS control field into character-type, single-pronunciation, tie-breaker, radical, total strokes sequence.	128 2-byte characters (=256 bytes)
X	Sort the Simplified Chinese DBCS control field into character-type, total strokes, radical, single-pronunciation, tie-breaker sequence.	128 2-byte characters (=256 bytes)
Y	Sort the Simplified Chinese DBCS control field into character-type sequence.	128 2-byte characters (=256 bytes)
M	Sort the Traditional Chinese control field into character-type sequence.	128 2-byte characters (=256 bytes)
H	Sort the Traditional Chinese control field into character-type, stroke, radical sequence.	128 2-byte characters (=256 bytes)
B	Sort the Traditional Chinese control field into character-type, radical, strokes beyond radical sequence.	128 2-byte characters (=256 bytes)
K	Use the primary Hangeul pronunciation of Hanja characters.	128 2-byte characters (=256 bytes)
L	Use the secondary Hangeul pronunciation of Hanja characters that start Korean DBCS words, and primary Hangeul pronunciation for all other Hanja characters.	128 2-byte characters (=256 bytes)
E	Sort a Katakana control field in SEI-ON sequence.	170 1-byte characters (= 170 bytes)
I	Sort the Japanese DBCS control field into single-pronunciation, stroke, tie-breaker sequence.	128 2-byte characters (=256 bytes)
J	Sort the Japanese DBCS control field into single-pronunciation, stroke, radical, tie-breaker sequence.	128 2-byte characters (=256 bytes)
R	Sort the Japanese DBCS control field into radical, stroke, tie-breaker sequence.	128 2-byte characters (=256 bytes)
S	Sort the Japanese DBCS control field into stroke, radical, tie-breaker sequence.	128 2-byte characters (=256 bytes)

Table 9 (Page 3 of 3). Possible Column 8 Entries and Restrictions on Field Length

Col. 8 Entry	Character Portion Used	Maximum Field Length
T	Sort the Japanese DBCS control field into character-type order.	128 2-byte characters (=256 bytes)

Note: ⋆ Note that -3 is less than 0, and +5 is greater than -6.

Suppose, for example, that your input records have a one-character control field that can be either *, 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 resequenced in ascending order using the digit position of the control field characters, enter `D` into column 8. The records will then be sorted in this order:

```
1
2
3
*
```

If you want the records resequenced in ascending order using both the zone and digit positions, enter `C` into column 8. The records will then be sorted in this order:

```
*
1
2
3
```

Suppose you place a `Z` into column 8 and want the records resequenced in ascending order. You can be sure that the records with an * control field will precede the records with a 1, 2, or 3 control field. Because 1, 2, and 3 have identical zone positions, records with any of these numbers as a control field will be in the order in which they were encountered in the input file.

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

Columns 7 and 8 (Details)

Table 10 shows the possible combinations for columns 7 and 8.

Table 10. Combinations of Column 7 and Column 8 Entries

Column 7	Column 8	Maximum Field Length
N or O	C	256 [1]
	Z	1 [1]
	D	16 [1]
	P	8 [1]
	U	16 [1]
F	C [2]	1
	Z [2]	1
	D [2]	1
D	C	99999
	Z	1
	D	16
	P	8
	U	16
	V	1
S	C	256
	Z	1
	D	16
	P	8
	U	16
	V	1

[1] *Cumulative maximum length for key fields.*

[2] *For an unconditional force and a force-all line, column 8 must contain a* C.

Columns 9 through 12 (Field Location Start Position)

These columns identify the starting position of a field. If columns 9 through 12 are blank, Sort assumes that the field length is equal to the record length.

To describe fields that are only one character long, leave columns 9 through 12 blank and enter the number of the record position that contains the character in columns 13 through 16.

Columns 13 through 16 (Field Location End Position)

These columns identify the position in which the field ends.

If columns 9 through 12 and 13 through 16 are blank, Sort assumes that the field length is equal to the record length.

To describe fields that are only one character long, leave columns 9 through 12 blank, and enter the number of the record position that contains the character in columns 13 through 16.

Column 17 (Conditionally Forced Character)

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

If a control field can contain any one of the several characters and you want to specify a replacement for each one, you must enter a field specification for each replacement, and use an entry in column 19 to relate the field specifications to each other.

See "Column 7 (Field Type)" on page 101 for a general discussion of forced control fields.

Make an entry in column 17 only when you want to see a conditional force. For example, use it to put an * instead of an A into a position on the output record. (See the descriptions for columns 7 and 18 in this chapter for more information about conditional force.)

Column 17 (Summary Overflow Indicator)

Enter the character you want Sort to put into the output record if overflow occurs. If nothing is specified, Sort assumes the asterisk (*) is the overflow indicator. This field should only be filled for SORTRS jobs.

In other words, if you specify a summary sort job, then in the case of an overflow (not enough room in the output record field for the accumulated total), the character you enter into column 17 will be the one that will be put into the overflow indicator field, a one-column field immediately to the right of the rightmost data field in the output record.

Column 18 (Forced Character)

The character here either replaces the control field character you specified in column 17, or, if there is no entry in column 17, is used to force the control field to take a particular value. Make an entry in column 18 only when you are using forced control fields. See "Column 7 (Field Type)" on page 101 for a general discussion of forced control fields.

You must use forced control characters only with one-character control fields.

Column 18 (Substitution Character)

When used in a summary sort job, the character you enter into this column is the one that is put into the overflow indicator field each time an overflow *does not* occur. You can use any character. If you do not enter a character, none is put into the output record.

Column 19 (Continuation)

Any entry put into this column links the field specification to the previously entered field specification. This is required when several field specifications are needed to define how a single field is processed (such as during a conditional force when the control field character is replaced by another character and several possibilities are to be specified).

If this column is left blank, the system assumes the field specification is for a new field.

Columns 20 through 22 (Overflow Field Length–Summary Sort)

The entry in these columns is a number from 1 to 256.

In summary sort jobs, the data fields that are summarized are used to contain the totals in the output record. It is possible, however, that these totals can require more space than the summary data field allows.

The length of this new overflow field should not exceed the maximum length for the output record.

The overflow field length should also not exceed the lengths specified in Table 10 on page 107 for each data type.

If the data to be summed is in packed format, each digit needs only 4 bits of memory for storage, plus another 4 bits to store the sign. Be aware that the number in columns 20 through 22 represents units of 8-bit bytes; that is, each byte represents two numbers in packed format, with the last byte representing only one digit and the sign (plus or minus) for the entire number. For example, if you specify 3 for the overflow field length, you are reserving 24 bits of storage: therefore 999 is the maximum number you can store in unpacked format, and +99999 is the maximum number you can store in packed format.

Column 20 (Alternative Collating Sequence by Field)

This column must contain an `A` for any normal or opposite control field that is to be altered by the alternative collating sequence when column 26 of the header specification contains an `F`.

If you specify an alternative collating sequence for a particular field, that field will be changed (according to the alternative collating sequence specified by `ALTSEQ`) whenever the field is used again as a control field for that record type.

Columns 23 through 39 (Reserved)

These columns must be left blank.

Columns 40 through 72 (Comments)

These columns are available for comments.

Chapter 11. The Comment Specification

One or more comment specifications can be used to document the job that is being done, so that you and other people will understand the sort process that is coded into the file. The comment specification can be inserted anywhere in the file, and can also precede the header specification.

RECORD SPECIFICATIONS

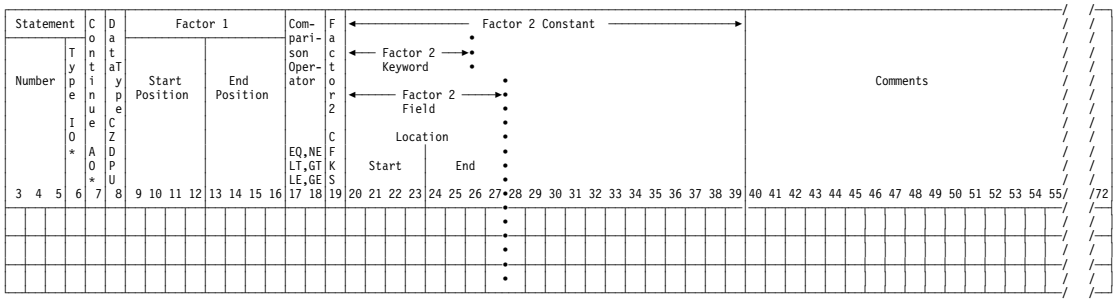


Figure 27. Comments Can Be Entered onto Header, Record, or Field Specification Forms

Summary of Column Entries

Table 1 summarizes the column entries for the comment specification.

Table 11. Summary of Column Entries for the Comment Specification

Columns	Entry	Explanation
6 or 7	*	Identifies this specification as a comment specification.
8-72	Any entry	Comments.

Detailed Description of Each Column Entry

Column 6 or 7 (Specification Identifier)

Column 6 or 7 must contain an * to identify this specification as a comment specification.

Columns 8 through 72 (Comments)

These columns are available for comments.

Appendix A. Information for System/36 Sort Utility Users Porting to the AS/400 System

This appendix describes those differences between the AS/400 system and System/36 that affect the way you enter Sort specifications.

Running a System/36 Sort Utility Job on the AS/400 System

Your System/36 specifications will continue to work on the AS/400 system. You should, however, be aware of the following items:

Header Specification

Print Option (Column 27)

Although Sort will accept all of the options available in the System/36 Sort Utility, assume the following:

0	Prints sort specifications, diagnostic messages, and program status message.
1, 2, or **3**	This column is syntax-checked, but no printout is generated when the job is run.

Note: No messages are displayed.

Ideographic Sort (Column 35) – Valid on DBCS systems only

Support for ideographic fields:

- One Ideographic Character (IGC) field type is used.
- Multiple IGC fields are used.

Null Output (Column 36)

Within the AS/400 system, the entry in this column is syntax-checked only.

Output Record Length (Columns 29 to 32)

When the output file is an externally described file, these columns are syntax-checked.

Record and Field Specifications

System/36 Sort Utility Level Support

When you select a summary sort on a file that contains nonnumeric data, or if a record without summary fields is defined, your results may differ from those on the System/36.

Also, when your summary sort contains more than one record format, and the corresponding summary fields are either different lengths or different data types, your results may differ from those on the System/36.

Notes:

1. The AS/400 Sort warning and error message are different from those on the System/36.
2. The AS/400 will not accept a minus sign in the last position of a numeric constant (denoting a negative value ending with zero). It should be replaced by the } symbol, or zero followed by a minus sign.

Running a Sort Job in System/36 Sort Utility

If you enter your coding specifications in AS/400 environment, but run them on the System/36 Sort Utility, make sure they conform to those for System/36.

You should also be aware of the following:

- The maximum length of the Factor 1 and Factor 2 fields in the record specification is 4KB (KB equals 1024 bytes).
- Because the System/36 system handles arithmetic statements differently from the AS/400 system, different results may occur when using nonnumeric data in summary sorts.
- On the AS/400 system, you can sort input files with different record lengths in the same sort job, but to run AS/400 sort jobs on the System/36, the input files must have the same record length.
- System/36, unlike the AS/400 system, does not convert invalid digits to zeros or invalid signs to positives (X'F'). Therefore, if you specify `Factor 1` as containing invalid packed numeric data, System/36 interprets the value differently from the AS/400 system. Your sort results will be different.

Appendix B. Information for System/38 Conversion Reformat Utility Users Porting to the AS/400 system

This appendix describes those differences between the AS/400 system and the System/38 Conversion Reformat Utility that affect the way that you enter Sort specifications.

Running a System/38 Conversion Reformat Utility Job in the AS/400 System

Your System/38 Conversion Reformat Utility specifications will continue to work on the AS/400 system. You should, however, be aware of the following:

Header Specification

Collating Sequence (Column 26): The Sort utility has the additional feature of being able to use an alternative collating sequence for specified parts of the control field. (Enter `F`.)

You must enter an `A` into column 20 of any field specification that defines a portion of the control field that is to be sorted with respect to the alternative collating sequence. If you specify an alternative collating sequence for any input field in this way, this sequence will apply to any other record types that also specify this input field as part of the control field.

Note: This enhancement cannot be used with the System/38 Conversion Reformat Utility.

Record and Field Specifications

The AS/400 will not accept a minus sign in the last position of a numeric constant (denoting a negative value ending with zero). It should be replaced by the } symbol, or zero followed by a minus sign.

Running an AS/400 Sort Job in the System/38 environment or on a System/38

You can run a Sort job in the System/38 environment of the AS/400 system, or on a System/38. If you do the latter, however, you should be aware of the following:

- The maximum control field length in the AS/400 system is 256 bytes; in System/38, it is 248.
- Although you can still use the terms `RAF` and `FILE`, AS/400 Sort uses `SORTA` in place of `RAF`, and `SORTR` in place of `FILE`.
- If you enter the coding specifications in the AS/400 system but run them in System/38 environment, make sure your coding specifications conform to those for System/38. In particular, the following AS/400 Sort features are not supported in System/38:
  - Summary Sort
  - DBCS Sorting
  - Selected Alternative Collating Sequences.

Appendix C. Standard and Alternative Collating Sequences

By default, the sorted output records are ordered according to the standard EBCDIC collating sequence. However, you can change this sequence for all or some characters by specifying an alternative collating sequence.

This appendix provides examples of the standard EBCDIC collating sequence, and explains how to define an alternative one.

Standard EBCDIC Collating Sequence

This collating sequence is an arrangement of data based on the EBCDIC character set. There are variations in the standard collating sequences, depending whether you are comparing:

- Both the zone and digit portions of characters
- Only the zone portions of characters
- Only the digit portions of characters.

Figure 28 on page 118 shows the standard collating sequence; Figure 29 on page 119 shows the collating sequence when you compare only the zone portion; and Figure 30 on page 120 shows the collating sequence when you compare only the digit portion.

Collating Sequence	Character	Hexadecimal Value
1	blank	40
2	¢	4A
3	.	4B
4	<	4C
5	(	4D
6	+	4E
7	|	4F
8	&	50
9	!	5A
10	$	5B
11	*	5C
12	)	5D
13	;	5E
14	¬	5F
15	-(minus)	60
16	/	61
17	](split bar)	6A
18	,	6B
19	%	6C
20	_(underscore)	6D
21	>	6E
22	?	6F
23	(grave)	79
24	:	7A
25	#	7B
26	@	7C
27	'	7D
28	=	7E
29	"	7F
30	a	81
31	b	82
32	c	83
33	d	84
34	e	85
35	f	86
36	g	87
37	h	88
38	i	89
39	j	91
40	k	92
41	l	93
42	m	94
43	n	95
44	o	96
45	p	97
46	q	98
47	r	99
48	(tilde)	A1

Collating Sequence	Character	Hexadecimal Value
49	s	A2
50	t	A3
51	u	A4
52	v	A5
53	w	A6
54	x	A7
55	y	A8
56	z	A9
57	{	C0
58	A	C1
59	B	C2
60	C	C3
61	D	C4
62	E	C5
63	F	C6
64	G	C7
65	H	C8
66	I	C9
67	}	D0
68	J	D1
69	K	D2
70	L	D3
71	M	D4
72	N	D5
73	O	D6
74	P	D7
75	Q	D8
76	R	D9
77	\	E0
78	S	E2
79	T	E3
89	U	E4
81	V	E5
82	W	E6
83	X	E7
84	Y	E8
85	Z	E9
86	0	F0
87	1	F1
88	2	F2
89	3	F3
90	4	F4
91	5	F5
92	6	F6
93	7	F7
94	8	F8
95	9	F9

Figure 28. Standard Collating Sequence

Only Zone Portion of Character Used			Only Zone Portion of Character Used			Only Zone Portion of Character Used		
Order in the Sequence*	Character	Corresponding Hexadecimal Number **	Order in the Sequence*	Character	Corresponding Hexadecimal Number **	Order in the Sequence*	Character	Corresponding Hexadecimal Number**
1 (lowest)	blank	40	con't	f	86	9	}	D0
	¢	4A		g	87		J	D1
	.	4B		h	88		K	D2
	<	4C		i	89		L	D3
	(	4D					M	D4
	+	4E					N	D5
	|	4F	6	j	91		O	D6
2	&	50		k	92		P	D7
	!	5A		l	93		Q	D8
	$	5B		m	94		R	D8
	*	5C		n	95			D9
	)	5D		o	96			
	;	5E		p	97			
	¬	5F		q	98	10	\	E0
				r	99		S	E2
3	- (minus)	60					T	E3
	/	61					U	E4
	](split bar)	6A	7	(tilde)	A1		V	E5
	,	6B		s	A2		W	E6
	%	6C		t	A3		X	E7
	_	6D		u	A4		Y	E8
	(underscore)			v	A5		Z	E9
	>	6E		w	A6			
	?	6F		x	A7			
				y	A8	11 (highest)	0	F0
4	(grave)	79		z	A9		1	F1
	:	7A					2	F2
	#	7B	8	{	C0		3	F3
	@	7C		A	C1		4	F4
	'	7D		B	C2		5	F5
	(apostrophe)			C	C3		6	F6
	=	7E		D	C4		7	F7
	"	7F		E	C5		8	F8
				F	C6		9	F9
5	a	81		G	C7			
	b	82		H	C8			
	c	83		I	C9			
	d	84						
	e	85						

*Characters sharing the same position in the sequence 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.

**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.

Figure 29. Standard EBCDIC Collating Sequence Used When You Compare Only the Zone Portion of Characters

Only Digit Portion of Character Used			Only Digit Portion of Character Used			Only Digit Portion of Character Used		
Order in the Sequence*	Character	Corresponding Hexadecimal Number **	Order in the Sequence*	Character	Corresponding Hexadecimal Number **	Order in the Sequence*	Character	Corresponding Hexadecimal Number**
1 (lowest)	blank	40		U	E4		z	A9
	&	50		4	F4		I	C9
	-(minus)	60	6	e	85		R	D9
	{	C0		n	95		Z	E9
	}	D0		v	A5		9	F9
	0	F0		E	C5	11	¢	4A
2	/	61		N	D5		!	5A
	a	81		V	E5		](split bar)	6A
	j	91		5	F5		:	7A
	(tilde)	A1		f	86	12	.	4B
	A	C1		o	96		$	5B
	J	D1		w	A6		,	6B
	\	E1		F	C6		#	7B
	1	F1	7	O	D6			
3	b	82		W	E6	13	<	4C
	k	92		6	F6		*	5C
	s	A2					%	6C
	B	C2	8	g	87		@	7C
	K	D2		p	97			
	S	E2		x	A7	14	(	4D
	2	F2		G	C7		)	5D
				P	D7		_	6D
4	c	83		X	E7		(underscore)	
	l	93		7	F7		'	7D
	t	A3					(apostrophe)	
	C	C3	9	h	88			
	L	D3		q	98	15	+	4E
	T	E3		y	A8		;	5E
	3	F3		H	C8		>	6E
				Q	D8		=	7E
5	d	84		Y	E8			
	m	94		8	F8	16	|	6F
	u	A4					¬	5F
	D	C4	10	(grave)	79		?	6F
	M	D4		i	89		"	7F
				r	99			

*Characters sharing the same position in the sequence 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.

**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.

Figure 30. Standard EBCDIC Collating Sequence Used When You Compare Only the Digit Portion of Characters

Defining an Alternative Collating Sequence

When you want the records in your output file to be sorted in an order different from the orders permitted by the standard collating sequences described in the first part of this appendix, define an alternative collating sequence.

To define an alternative collating sequence:

1. ***Header specification:*** Specify one of the following in column 26 (the ALTCOLLSEQ column) of the header specification:
   - An S to show that the alternative collating sequence applies to the entire control field
   - An F to show that the alternative collating sequence applies to only parts of the control field.
2. ***Field specifications:*** If only parts of the control field are involved, enter an A in column 20 of those field specifications that define the input fields that are to become part of the control field and to which the alternative collating sequence is to apply.

Specifying an Alternative Collating Sequence for the Entire Control Field

To do this:

1. Enter S in column 26 of the header specification.
2. Code the required ALTSEQ specifications to immediately follow the header specification.

Sort will apply these changes to the entire job. The appropriate characters are changed in all input records before any processing is done: this means that the selection of input records, as stated on the record specifications for the job, is based on the changed characters as specified in the alternative collating sequence.

Forced characters and any other constants and comparisons will also be changed and controlled by the alternative collating sequence.

Note: Do not use packed or zoned Factor 1 or Factor 2 fields in an include or omit record specification (P or U in column 8) if you specify an alternative collating sequence on the entire control field. You can only do so when you are using an alternative collating sequence only on specified parts of the control field.

Specifying an Alternative Collating Sequence on Parts of the Control Field

To specify an alternative collating sequence for normal and opposite control fields:

1. Enter F in column 26 of the header specification.
2. Code the required ALTSEQ specifications immediately following the header specification.
3. Enter an A in column 20 of those field specifications that define an input field that is part of the control field, and to which the alternative collating sequence applies.

Be aware that:

- Record selection (including or omitting records) and conditional force (replacing a single character or all characters) are based on an input record that has not been changed by the alternative collating sequence.
- Any control field specification that has an A in column 20 must not be packed or zoned. (Do not code a P or U in column 8.)
- If you specify an alternative collating sequence for a particular input field, this sequence will be applied to any other input record type that uses the same input field as part of its control field.
- The alternative collating sequence will apply only to the specified normal and opposite control fields that were indicated in this way.

ALTSEQ specifications never change data fields in records or forced control field characters. Characters defined for an alternative collating sequence could be interpreted as other data fields.

How to Code an ALTSEQ Specification

For each character whose collating sequence you want to change from the standard EBCDIC, enter the original hexadecimal value followed by the new hexadecimal value into the ALTSEQ specification. The system then sorts the character according to its new position, relative to the other characters. You can code as many changes into one ALTSEQ specification as will fit.

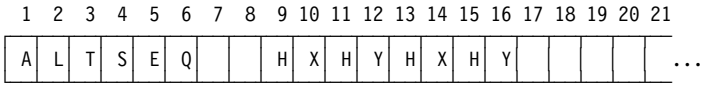


Figure 31. How You Code an ALTSEQ Specification

Use the following steps to code ALTSEQ specifications. You can code them onto any Sort specifications form.

1. Code ALTSEQ into columns 1-6 to tell the Sort utility that you want to change the standard collating sequence.
2. Leave columns 7-8 blank.
3. Enter the hexadecimal equivalent (shown as X'HX' in the figure) of the character you are taking out of its normal sequence into the next two columns.
4. Enter the hexadecimal equivalent (shown as X'HY' in the figure) of the value that the character specified is to assume into the next two columns.
5. Enter as many pairs of original and new hexadecimal values (each occupying four columns) as required. Leave no spaces between sets of hexadecimal numbers.
6. When you reach the end of one specification (column 96), you can continue entering more pairs of hexadecimal values in a new ALTSEQ specification. (Repeat steps 1 to 5.)
7. Do not code comments on an ALTSEQ line.

Note: 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. The system does not know which character should precede the other, and therefore, does not sort these two characters. If you do not want the two characters to be equal, you must also move the character that normally occupies that position. (See the examples later in this appendix.)

Examples

The following examples describe two situations that involve using an alternative collating sequence.

Example 1. Inserting a Special Character between Two Alphabetic Characters

You can alter the normal collating sequence several of ways. For example, you can insert a character between two existing characters, take a character out of the sequence, or change characters (put A where Z is and Z where A is). Regardless of how you alter the sequence, you must specify every character that is to be changed by the alteration. For example, if you want a dollar sign ($) to be posi-

tioned in the collating sequence between A and B, change the normal sequence as follows:

Normal Sequence	Altered Sequence
A	A
B	$
C	B
D	C
E	D
F	E
G	F
H	G
I	H
	I

Note: For this sequence, assign the dollar sign ($) the same value as B, B the same value as C; continue this process until the I has the same value as an unprintable character. This produces the desired results, because there are no printable characters between I and J. (See Figure 28 on page 118.)

The ALTSEQ specification used to enter this alternative collating sequence is:

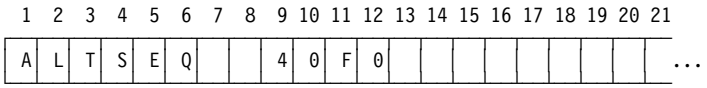
```
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45
A L T S E Q     5 B C 2 C 2 C 3 C 3 C 4 C 4 C 5 C 5 C 6 C 6 C 7 C 7 C 8 C 8 C 9 C 9 C A ...
```

Columns	Entry	Explanation
1-6	ALTSEQ	Identifies the specification as an ALTSEQ specification.
7-8	Blanks	
9-12	5BC2	The $ is given the same value as the B.
13-16	C2C3	The B is given the same value as the C.
17-20	C3C4	The C is given the same value as the D.
21-24	C4C5	The D is given the same value as the E.
25-28	C5C6	The E is given the same value as the F.
29-32	C6C7	The F is given the same value as the G.
33-36	C7C8	The G is given the same value as the H.
37-40	C8C9	The H is given the same value as the I.
41-44	C9CA	The I is given the same value as an unprintable character.

Example 2. Making Characters Equal

If you want one character to be considered the same as another, both must hold the same position in the collating sequence. For example, you may want a blank to be considered a zero. Therefore, you need to define an alternative collating sequence, in which the blank is the same as the zero, because it holds the same position in the sequence.

The corresponding ALTSEQ specification would look like this:



Columns	Entry	Explanation
1-6	ALTSEQ	Identifies the specification as an ALTSEQ specification.
7-8	Blanks	
9-12	40F0	The blank is given the same value as the zero.

Whenever a blank is read and used in a comparison, it is considered to be a zero. Thus, if you were comparing the numbers 0036 and bb36 (where b = blank), both would equal 0036.

Note: Use care when using D (digit) or U (unpacked) fields with an alternative sequence. The sign in the number is stored as part of the number, and could therefore be translated into a character that could be sorted; the record could be positioned at an unexpected place in the file.

Appendix D. Sorting Double-Byte Character Sets

This appendix describes the requirements for sorting double-byte character sets (DBCS) in the following languages:

- Japanese
- Simplified Chinese
- Traditional Chinese
- Korean.

The *ADTS/400: Character Generator Utility*, SC09-1769 contains additional information about creating and maintaining user-defined double-byte character sets, and updating the DBCS Sort Tables.

Before you can sort double-byte characters, you must have the following:

- The DBCS version of the AS/400 system installed on your system.
- A DBCS-capable display station (if you want to display or enter double-byte characters in your source file).
- A printer capable of printing characters in your language, if you are going to print double-byte characters.
- Sort tables
  - DBCS master sort table

    The DBCS master sort table for each language contains the sort information to specify for each user-defined DBCS character in that language. Every time you define a new character or make changes to an existing character, you must update the sort information for that character in this table.
  - DBCS active sort table

    The DBCS active sort table for each language contains the active collating sequence of the DBCS characters in that language. Every time you update the DBCS master sort table, you should update this table as well.

  Note: For Korean DBCS characters, there is a DBCS active sort table but no DBCS master sort table. The active sort table is used to convert Hanja characters to their equivalent Hangeul characters.

The *Data Management*, SC41-3710 contains more information.

Coding Considerations for Double-Byte Character Sets

Although all the columns on the header, record, and field specifications are available for sorting double-byte character sets, certain combinations may not be valid or may not produce the desired results. This section provides you with necessary details unique to coding your specifications to avoid run-time errors.

HEADER SPECIFICATIONS

Page 1 2 Program Identification 73 74 75 76 77 78 79 80

Statement Number	Type H*	Output Type SORTR SORTRS SORTA	Reserved	Maximum Control Field Length (1–256)	Sequence AD	Reserved	Alt Coll Seq SF	Reserved	Output X	Reserved	Comments
3 4 5	6	7 8 9 10 11 12	13 14	15 16 17	18	19 20 21 22 23 24 25	26	27	28	29 30 31 32 33 34 35 36 37 38 39	40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55/ /72

Header Specifications

The following coding considerations must be noted to successfully run your program. For further details regarding coding your header specifications, see Chapter 8, “The Header Specification.”

- Columns 15 through 17 (Maximum Control Field Length)

  Shift-in (SI) and shift-out (SO) control characters are present in all records and do not need to be included as part of a DBCS control field. The maximum control field length must not exceed 256 bytes. The control field is the number of bytes, not the number of characters.

  Note: When you convert a one-byte EBCDIC Katakana field to the Japanese (Sei-On) Katakana control field, you must calculate the value in columns 15 through 17. To do this, divide the EBCDIC Katakana field length by 4 and round the quotient, if there is one, up to the nearest whole number. Multiply this value by 2, and add the result to the original Katakana field length. The maximum control field length is 256 bytes.

- Column 26 (Alternative Collating Sequence)

  An `S` or `F` entry in this column can produce undesirable results, because the double-byte alternative collating sequence is not supported.

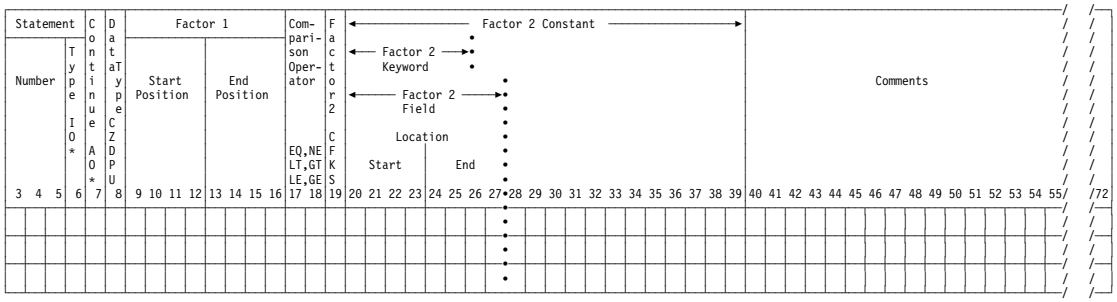
- Column 28 (Include or Exclude Control Field in Output)

  When DBCS control fields are used, they are changed in building the work record; therefore, if you must keep the original data, repeat the information as a data field. A blank in column 28 means to keep the control field; an `X` means to drop the control field.

- Column 35 (DBCS Sort Option)

  This column is valid **only** in the System/36 environment and applies **only** to Japanese DBCS sorts. Entries `1` and `2` will **only** be supported for System/36 compatibility; otherwise, you must leave this column blank.

RECORD SPECIFICATIONS



Record Specifications

For further details regarding coding your record specifications, see Chapter 9, “Record Specifications.”

- Column 8 (Data Type)

  For users of double-byte characters, the only valid entry is `C`.

  Note: Although `Z` and `D` are not invalid entries and no error message will be issued if they are coded, undesirable results may occur when you sort double-byte character sets.

- Columns 17 through 18 (Comparison Operator)

  Although all entries are valid for Hangeul characters, only `EQ` and `NE` are meaningful when Factor 1 and Factor 2 contain all other DBCS characters.

- Column 19 (Type of Data Contained in Factor 2)

  Enter `C` to allow a DBCS data type field in the input record (Factor 1) to be compared to a DBCS constant that will be represented, for example, by `SOKKKKSI` (DBCS constant).

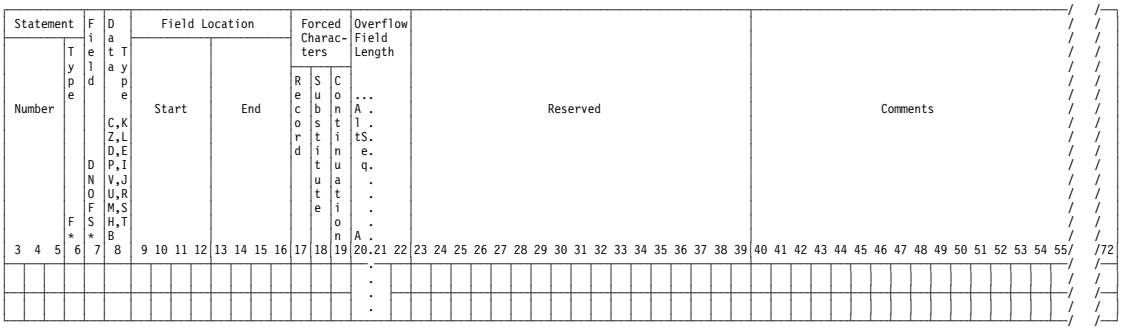
  Enter `S` to allow a DBCS constant to be shifted one byte to the left. A blank will be placed on the right.

  Factor 2 as a constant may contain shift-in or shift-out characters that are not defined as part of Factor 1. If Factor 2 is a DBCS constant, then it must be delimited by shift-out (`SO`) and shift-in (`SI`) on the record specification.

  If shift-in is not included in the Factor 2 comparison, the length of Factor 2 must be specified as the length of Factor 1 minus 1.

  Note: For packed data, Factor 2 must be the same size as the Factor 1 field.

FIELD SPECIFICATIONS



Field Specifications

For further details regarding coding your field specifications, see Chapter 10, “Field Specifications.”

- Column 8 (Data Type)

  This field describes the structure of the data in the field. You can show the data being defined, for example, one-byte alphanumeric Katakana or two-byte DBCS.

  A DBCS control field with a `C` in this column is sorted into IBM-code (EBCDIC) order, with a maximum length of 128 two-byte characters (256 bytes).

  A DBCS control field with a `K` or `L` in this column is sorted into the following sequence:

  – blanks
  – special symbols
  – a through z
  – A through Z
  – Greek letters
  – Japanese Katakana symbols
  – 0 through 9
  – Roman numerals
  – DBCS (in IBM-code sequence).

  If column 8 contains a `K`, each Hanja character will sort to the same position as the primary Hangeul pronunciation of that character.

  If column 8 contains an `L`, Hanja characters that start a DBCS word will sort to the same position as the secondary Hangeul pronunciation of that character. All other Hanja characters will sort to the same position as the primary Hangeul pronunciation.

- Columns 9 through 16 (Location of Field in the Input Record)

  The length of the control field must be a multiple of two, and its length must not exceed 256 bytes.

  Note: DBCS characters for which no entries in the active collating sequence exist will have a DBCS control field entry of `X'FFFF'` if ascending sequence is specified on the header specification. This will cause records with undefined DBCS characters to be placed behind those with defined DBCS characters.

- Columns 17 through 19 (Forced Character, Field, or Summary Overflow Indicator)

  `A`, `N`, and `K` (alphanumeric and Katakana) are valid, but DBCS characters are not supported.

Sorting Considerations for Japanese Double-Byte Character Sets

Japanese double-byte characters can be sorted into five different character sequences. Also, the alphanumeric Katakana fields can be sorted into the Sei-On Katakana sequence. These five sequences are:

1. Radical/stroke sequence/tie-breaker/DBCS code sequence
2. Stroke/strokes beyond radical/radical sequence/tie-breaker/ DBCS code sequence
3. Single-pronunciation/radical sequence/stroke sequence/tie-breaker/ DBCS code sequence
4. Single-pronunication/strokes beyond radical/radical sequence/ tie-breaker/DBCS code sequence
5. Character-type sequence.

Figure 32. Japanese Character Sequences

Radical: Characters can be ordered into their respective radicals. Within each radical set, the characters are ordered by strokes. Radical numbers can be user-defined. The *ADTS/400: Character Generator Utility*, SC09-1769 contains a listing of radical numbers.

Strokes: Characters can be ordered by the number of strokes used to write the character, by the total number of strokes, or by the number of strokes excluding the radical.

Single-pronunciation: Characters can be ordered by the phonetic sound they have when pronounced. Pronunciations can be either user-defined or defined by IBM, and are represented by Katakana.

Character-type: Characters can be ordered into one of the following character-type sequences:

- Special characters
- Alphabet
- Russian
- Greek
- Katakana
- Hiragana
- Arabic
- Roman
- Kanji.

Tie-breaker: The tie-breaker is one or more Katakana characters used to represent any alternative pronunciation of IBM-supplied characters only. In DBCS sorting, tie-breaker is a value (based on the sequence position of phonetic characters) that determines the output order of DBCS characters that otherwise compare equal.

DBCS Code Sequence: If all the sort criteria are the same for any two DBCS characters in sequences 1 through 4 in Figure 32 on page 129, then the characters will be ordered based on their internal hexadecimal representations.

Sei-On Katakana Sequence: Characters are a combination of single-pronunciation Kanji characters. Also, the equivalent pronunciations of voiced sound, semi-voiced sound, long sound, and Katakana small sound are used.

The following example shows the pronunciation and sort sequence:

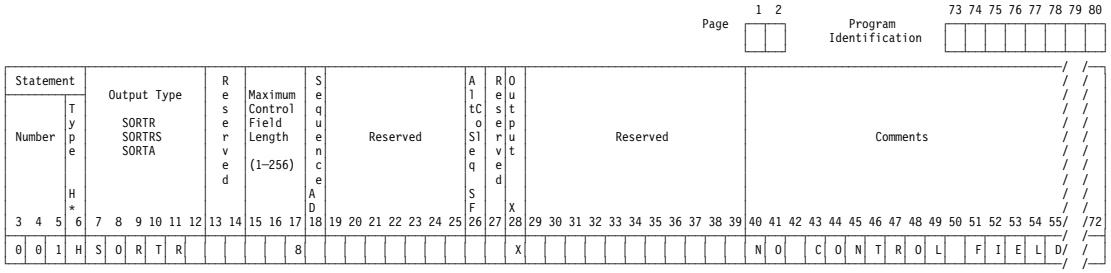
Meaning	Combination of Single-pronunciation and Kanji	Sei-on Pronunciation	Sort Sequence (ascending)
rule	kiyaku	kiyaku	2
customer	kyaku	kiyaku	3
reverse	gyaku	kiyaku	4
for conscience sake	kiyasume	kiyasume	5
rejection	kyakuka	kiyakuka	6
move backward	gyakukou	kyakukou	7
scream with laughter	kyah	kiyaa	1

Japanese Telephone Book Sorting Example

The following example illustrates sorting a person's name by telephone book type sort while the person's title is manager. The DBCS sort program requires one control field to sort into telephone book sequence. The example shows the field specification entries that generate the control field when the field in the input records contains four DBCS characters. The input layout file is as follows:

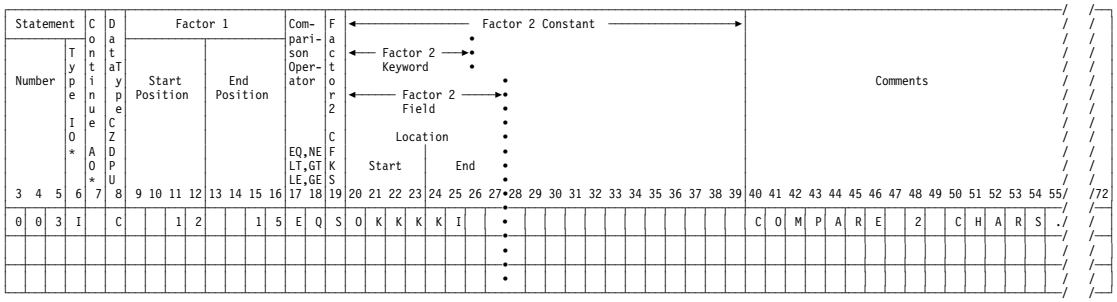
From	To	Explanation
1	10	Kanji field with So and Si person's name.
11	16	Kanji field with So and Si title.
17	64	Other information.

HEADER SPECIFICATIONS



Columns	Entry	Explanation
7-12	SORTR	This is a regular sort: the output is a physical file.
17	8	The maximum control field length is 8 characters.
28	X	The control field does not appear in the output record.
40-72	Comments	Identifies that the control field is dropped.

RECORD SPECIFICATIONS



Columns	Entry	Explanation
6	I	Specifies that the records described in this specification are to be included in the sort.
8	C	Identifies the data for Factor 1 and Factor 2 as character data. The system will look at both the zone and the digit portions of each byte when comparing Factor 1 with Factor 2.
9-12	12	Factor 1 field begins in column 12 of the input record.
13-16	15	Factor 1 field ends in column 15 of the input record.
17-18	EQ	The data in the Factor 1 field must equal the data in the Factor 2 constant for the input record to be selected for sorting.
19	S	The data in the Factor 2 field is constant, and is shifted one column to the left before comparison.
20	0	Represents the shift-out character.
21-24	KKKK	The two-byte characters to be compared.
25	I	Represents the shift-in character.
40-72	Comments	Identifies that two double-byte characters are being compared.

FIELD SPECIFICATIONS

Statement			Field	Data Type	Field Location		Forced Characters			Overflow Field Length	Reserved	Comments
Number		Type	DNOFS**	C,KZ,LD,EP,IV,JU,RM,SH,TB	Start	End	Record	Substitute	Continuation	...A.tS.e.q.		
3 4 5		F* 6	7	8	9 10 11 12	13 14 15 16	17	18	19	A. 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 /72
0	0	3 F	N	J		2				.		S R T S E Q U E N C E
										.		
										.		

Columns	Entry	Explanation
6	F	Identifies that this is a field specification.
7	N	Identifies the field as a normal control field. The control field data is not forced (modified) before it is used to control the sort.
8	J	The data field contains a Japanese DBCS control field using single-pronunciation, stroke, radical, tie-breaker sequence.
9-12	2	The data field begins in column 2 of the input record.
13-16	9	The control field ends in column 9 of the input record.
40-72	Comments	Identifies the sort as a telephone book type.

Japanese Word Dictionary Sorting Example

The following example illustrates the coding required on the field specification to sort in Japanese word dictionary type order. The program requires two control fields. The example shows the specification entries that generate the control fields when four DBCS characters are contained in the input record.

FIELD SPECIFICATIONS

Statement Number	Type	Field	Data Type	Field Location Start	Field Location End	Forced Characters Record	Forced Characters Substitute	Forced Characters Continuation	Overflow Field Length	Reserved	Comments
	F	DNOFS	C,KZ,LD,EP,IV,JU,RM,SH,TB						...A.l.tS.e.q.		
	*	*							A.		
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 / /72
0 0 4	F	N	E	1 1	2 6				.		SEI-ON KATAKANA
0 0 5	F	N	S	2	9				.		SRT SEQUENCE

Columns	Entry	Explanation
6	F	Identifies that this is a field specification.
7	N	Identifies the field as a normal control field. The control field data is not forced (modified) before being used to control the sort.
8	E	The data field contains a Sei-On Katakana control field.
9-12	11	The data field begins in column 11 of the input record.
13-16	26	The control field ends in column 26 of the input record.
40-72	Comments	Identifies the pronunciation of the word.

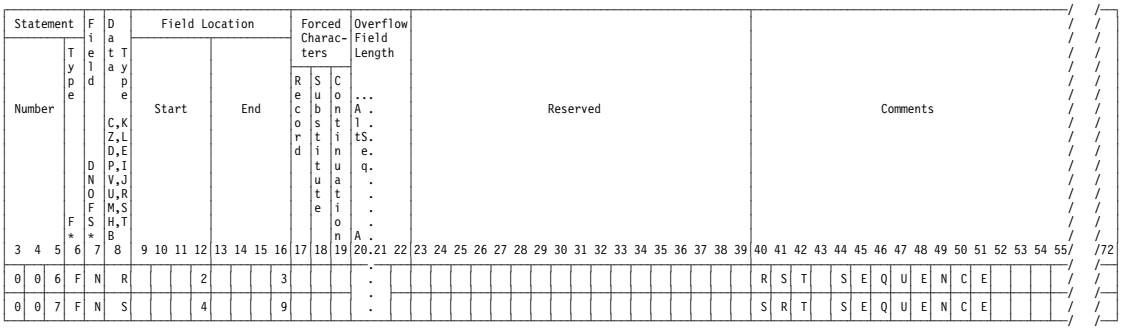
The following describes the column entries that differ on the second control field.

Columns	Entry	Explanation
8	S	The data field contains a Japanese DBCS control field using stroke, radical, tie-breaker sequence.
9-12	2	The data field begins in column 2 of the input record.
13-16	9	The control field ends in column 9 of the input record.
40-72	Comments	Identifies the stroke, radical, tie-breaker sequence.

Kanji Character Dictionary Type Sorting Example

The following example illustrates the coding required on the field specification to sort into Kanji character dictionary type order. The program requires two control fields. The example shows the specification entries that generate the control fields when four DBCS characters are contained in the input record.

FIELD SPECIFICATIONS



Columns	Entry	Explanation
6	F	Identifies that this is a field specification.
7	N	Identifies the field as a normal control field. The control field data is not forced (modified) before being used to control the sort.
8	R	The data field contains a Japanese DBCS control field using radical, stroke, tie-breaker sequence.
9-12	2	The data field begins in column 2 of the input record.
13-16	3	The control field ends in column 3 of the input record.
40-72	Comments	Identifies the radical, stroke, tie-breaker sequence.

The following describes the column entries that differ on the second control field.

Columns	Entry	Explanation
8	S	The data field contains a Japanese DBCS control field using stroke, radical, tie-breaker sequence.
9-12	4	The data field begins in column 4 of the input record.
13-16	9	The control field ends in column 9 of the input record.
40-72	Comments	Identifies the stroke, radical, tie-breaker sequence.

Sorting Considerations for Simplified Chinese Double-Byte Character Sets

Simplified Chinese double-byte characters can be sorted into the following seven different sequences:

1. Character-type/total strokes/radical/DBCS code sequence
2. Character-type/radical/strokes beyond radical/DBCS code sequence
3. Character-type/single-pronunciation/radical/total strokes/tie-breaker/ DBCS code sequence
4. Character-type/single-pronunciation/total strokes/radical/tie-breaker/ DBCS code sequence
5. Character-type/single-pronunciation/tie-breaker/radical/total strokes/ DBCS code sequence
6. Character-type/total strokes/radical/single-pronunciation/tie-breaker/ DBCS code sequence
7. Character-type.

Figure 33. Simplified Chinese Character Sequences

Character-Type: Different character-types can be ordered by the following sequences:

- User-defined special characters
- General character, Ordinal, and Numeric
- Latin alphabet
- Japanese Kana
- Greek alphabet
- Russian alphabet
- Chinese phonetic symbol
- Chinese phonetic-annotated letter
- Chinese characters.

Strokes: Characters can be ordered by the number of strokes used to draw the character or by the total number of strokes.

Radical: Characters can be ordered into their respective radicals. Within each radical set, the characters are ordered by strokes. Radical numbers can be user-defined. The *ADTS/400: Character Generator Utility*, SC09-1769 contains a listing of radical numbers.

Strokes beyond radical: Characters can be ordered by the number of strokes used to draw the character or by the number of strokes excluding the radical.

Single-pronunciation: Characters can be ordered by the phonetic sound they have when pronounced. Pronunciations are represented by tone concatenating Pinyin. There are five tones: light tone (without tone mark), high and level tone, rising tone, falling-rising tone, and falling tone. When you sort a Simplified Chinese character by single-pronunciation sequence, it is sorted by Pinyin and then by tone sequence.

Tie-breaker: The tie-breaker is the tone and Pinyin used to represent any alternative pronunciation of IBM-supplied characters only. In DBCS sorting, tie-breaker is a value (based on the sequence position of phonetic characters) that determines the output order of DBCS characters that otherwise compare equal.

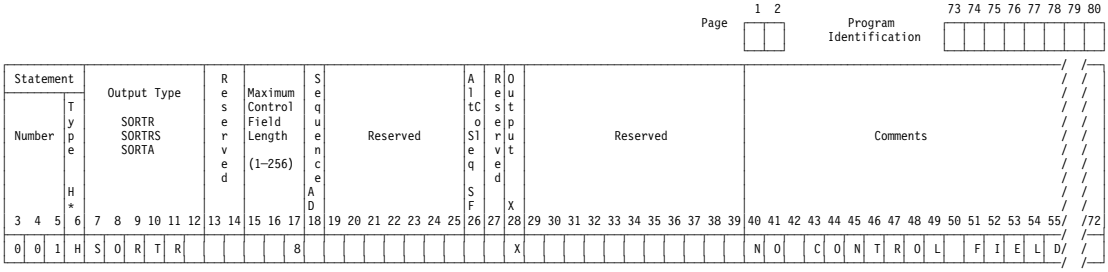
DBCS Code Sequence: If all the sort criteria are the same for any two DBCS characters in sequences 1 to 6 in Figure 33 on page 135, the characters will be ordered based on their internal hexadecimal representation.

Simplified Chinese Strokes/Radical Sorting Example

The following example illustrates sorting a person's name by Strokes/Radical sequence for employees working in the Purchasing Department. The DBCS sort program requires one control field to sort into this sequence. This example shows the specifications required to generate the control field if the field in the input records contains four DBCS characters. The input layout file is as follows:

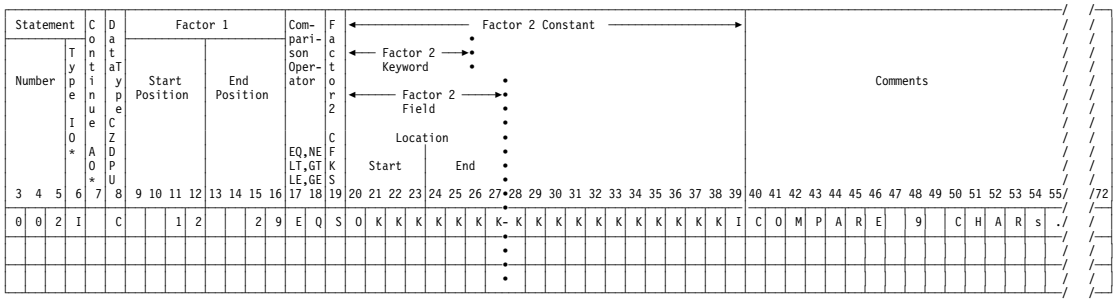
From	To	Explanation
1	10	Simplified Chinese field with SO and SI person's name.
11	30	Simplified Chinese field with SO and SI department name.
31	64	Other information.

HEADER SPECIFICATIONS



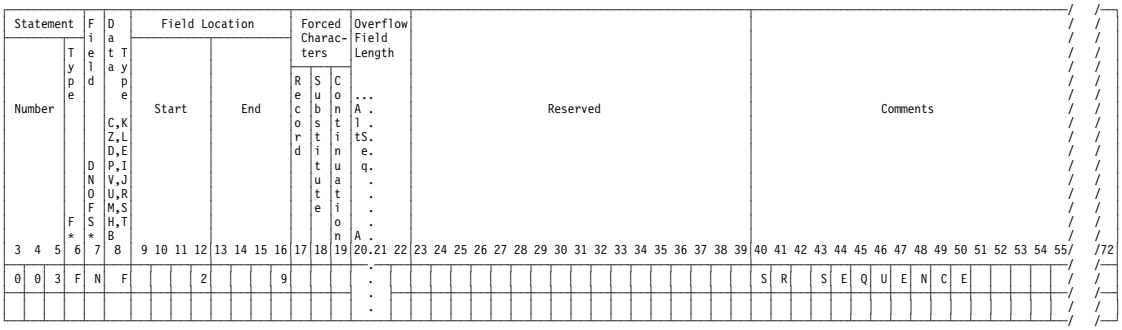
Columns	Entry	Explanation
6	H	Identifies that this is a Header Specification.
7-12	SORTR	This is a regular sort: the output is to be a physical file.
17	8	The maximum control field length is 8 characters.
28	X	The control field will not appear in the output record.
40-72	Comments	Identifies that the control field will be dropped.

RECORD SPECIFICATIONS



Columns	Entry	Explanation
6	I	Specifies that the records described in this specification are to be included in the sort.
8	C	Identifies the data for Factor 1 and Factor 2 as character data. The system will look at both the zone and the digit portions of each byte when comparing Factor 1 with Factor 2.
9-12	12	Factor 1 field begins in column 12 of the input record.
13-16	29	Factor 1 field ends in column 29 of the input record.
17-18	EQ	The data in the Factor 1 field must equal the data in the Factor 2 constant for the input record to be selected for sorting.
19	S	The data in the Factor 2 field is constant, and is shifted one column to the left before comparison.
20	O	Represents the shift-out character.
21-38	KKKKKKKKKKKKKKKKKK	The 9 two-byte characters (Purchasing Department) to be compared.
25	I	Represents the shift-in character.
40-72	Comments	Identifies that two Kanji characters are being compared.

FIELD SPECIFICATIONS



Columns	Entry	Explanation
6	F	Identifies that this is a field specification.
7	N	Identifies the field as a normal control field. The control field data is not forced (modified) before it is used to control the sort.
8	F	The data field contains a DBCS control field using character-type, strokes, radical sequence.
9-12	2	The data field begins in column 2 of the input record.
13-16	9	The control field ends in column 9 of the input record.
40-72	Comments	Identifies the sort as a Strokes/Radical sequence.

Sorting Considerations for Traditional Chinese Double-Byte Character Sets

Traditional Chinese double-byte characters can be sorted into the following three different sequences:

1. Character-type/strokes/radical/DBCS code sequence
2. Character-type/radical/strokes beyond radical/DBCS code sequence
3. Character-type sequence.

Figure 34. Traditional Chinese Character Sequences

Character-Type: Characters can be ordered into one of the following character-type sequences:

- Special characters
- Alphabet
- Greek
- Katakana
- Roman.

Strokes: Characters can be ordered by the number of strokes used to draw the character or by the total number of strokes.

Radical: Characters can be ordered into their respective radicals. Within each radical set, the characters are ordered by strokes. Radical numbers can be user defined. For a listing of radical numbers, refer to the *ADTS/400: Character Generator Utility*, SC09-1769.

Strokes beyond radical: Characters can be ordered by the number of strokes used to draw the character or by the number of strokes excluding the radical.

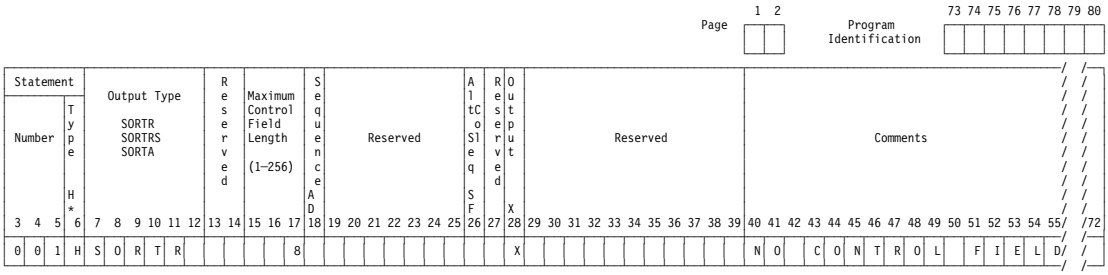
DBCS Code Sequence: If all of the sort criteria are the same for any two DBCS characters in sequences 1 and 2 in Figure 34, the characters will be ordered based on their internal hexadecimal representations.

Traditional Chinese Strokes/Radical Sorting Example

The following example illustrates sorting a person's name by Strokes/Radical sequence for employees working in the Purchasing Department. The DBCS sort program requires one control field to sort into this sequence. This example shows the specifications required to generate the control field if the field in the input records contains four DBCS characters. The input layout file is as follows:

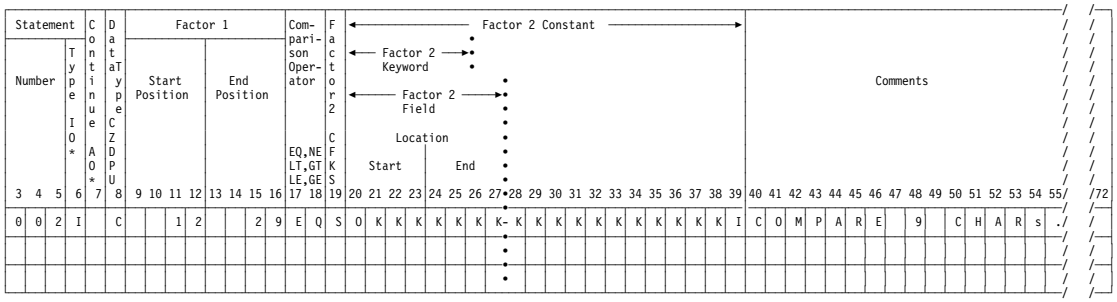
From	To	Explanation
1	10	Traditional Chinese field with SO and SI person's name.
11	30	Traditional Chinese field with SO and SI department name.
31	64	Other information.

HEADER SPECIFICATIONS



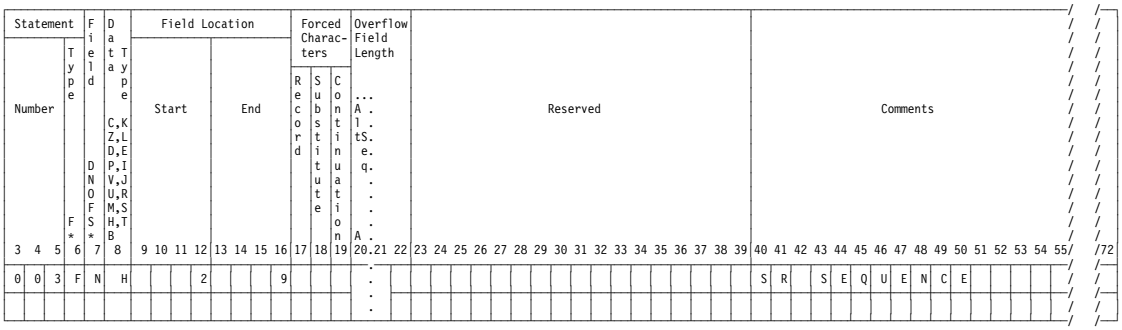
Columns	Entry	Explanation
6	H	Identifies that this is a header specification.
7-12	SORTR	This is a regular sort: the output is a physical file.
17	8	The maximum control field length is 8 characters.
28	X	The control field does not appear in the output record.
40-72	Comments	Identifies that the control field is dropped.

RECORD SPECIFICATIONS



Columns	Entry	Explanation
6	I	Specifies that the records described in this specification are to be included in the sort.
8	C	Identifies the data for Factor 1 and Factor 2 as character data. The system will look at both the zone and the digit portions of each byte when comparing Factor 1 with Factor 2.
9-12	12	Factor 1 field begins in column 12 of the input record.
13-16	29	Factor 1 field ends in column 29 of the input record.
17-18	EQ	The data in the Factor 1 field must equal the data in the Factor 2 constant for the input record to be selected for sorting.
19	S	The data in the Factor 2 field is constant, and is shifted one column to the left before comparison.
20	0	Represents the shift-out character.
21-38	KKKKKKKKKKKKKKKKKK	The 9 two-byte characters (Purchasing Department) to be compared.
25	I	Represents the shift-in character.
40-72	Comments	Identifies that two Kanji characters are being compared.

FIELD SPECIFICATIONS



Columns	Entry	Explanation
6	F	Identifies that this is a field specification.
7	N	Identifies the field as a normal control field. The control field data is not forced (modified) before it is used to control the sort.
8	H	The data field contains a DBCS control field using character-type, strokes, radical sequence.
9-12	2	The data field begins in column 2 of the input record.
13-16	9	The control field ends in column 9 of the input record.
40-72	Comments	Identifies the sort as a Strokes/Radical sequence.

Sorting Considerations for Korean Double-Byte Character Sets

The Korean double-byte character set consists of two distinct types of characters: Hangeul and Hanja. Hangeul characters are sorted according to their DBCS codes.

For Korean DBCS characters, there is a DBCS active sort table but no DBCS master sort table. The *Data Management* contains more information about these tables.

The Korean DBCS can be sorted according the following sequence:

- Pronunciation.

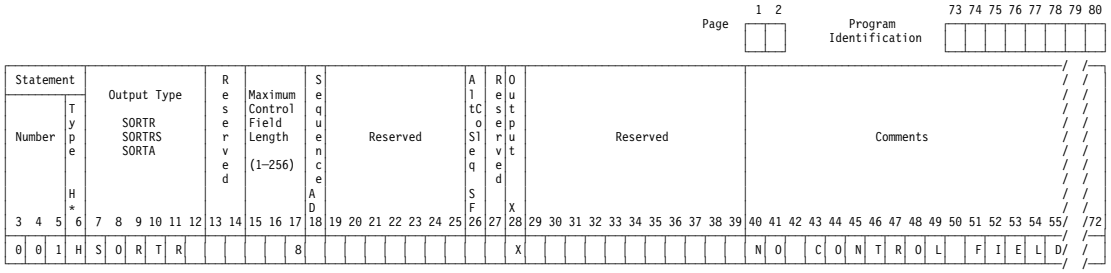
Pronunciation: Characters can be ordered by the phonetic sound they have when pronounced. Pronunciations can be either user-defined or defined by IBM. All Hanja characters are equivalent in pronunciation to some Hangeul character; however, a second pronunciation may be necessary if the character is at the start of a word.

Korean Pronunciation Sort Example

The following example illustrates sorting women's names. There are two fields: name and sex. The input layout file is as follows:

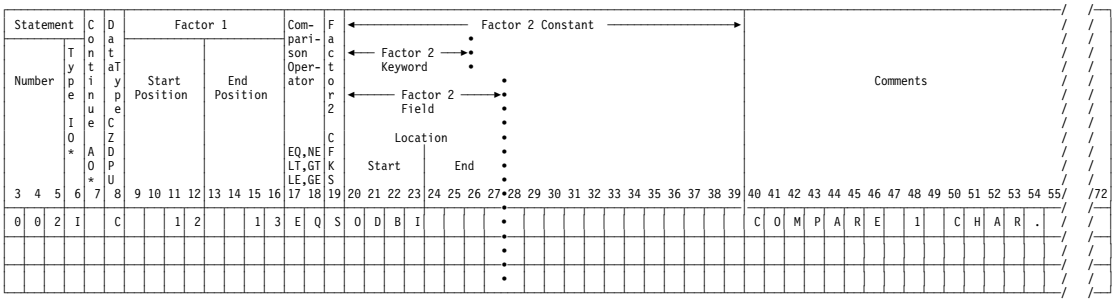
From	To	Explanation
1	10	Hangeul/Hanja field with shift-out and shift-in, person's name.
11	14	Hangeul/Hanja field with shift-out and shift-in, sex.
17	72	Other information.

HEADER SPECIFICATIONS



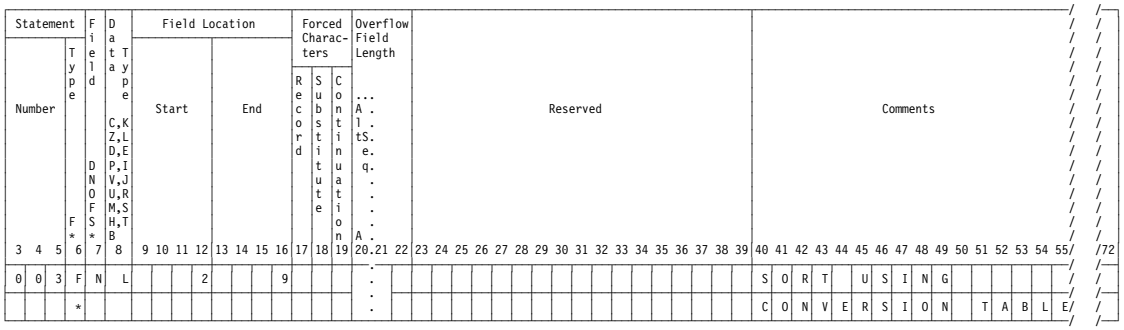
Columns	Entry	Explanation
7-12	SORTR	This is a regular sort: the output is a physical file.
17	8	The maximum control field length is 8 characters.
28	X	The control field does not appear in the output record.
40-72	Comments	Identifies that the control field is dropped.

RECORD SPECIFICATIONS



Columns	Entry	Explanation
6	I	Specifies that the records described in this specification are to be included in the sort.
8	C	Identifies the data for Factor 1 and Factor 2 as character data. The system will look at both the zone and the digit portions of each byte when comparing Factor 1 with Factor 2.
9-12	12	Factor 1 field begins in column 12 of the input record.
13-16	13	Factor 1 field ends in column 15 of the input record.
17-18	EQ	The data in the Factor 1 field must equal the data in the Factor 2 constant for the input record to be selected for sorting.
19	S	The data in the Factor 2 field is constant, and is shifted one column to the left before comparison.
20	0	Represents the shift-out character.
21-23	DB	The DBCS character to be compared.
25	I	Represents the shift-in character.
40-72	Comments	Compare 1 character.

FIELD SPECIFICATIONS



Columns	Entry	Explanation
6	F	Identifies that this is a field specification.
7	N	Identifies the field as a normal control field. The control field data is not forced (modified) before it is used to control the sort.
8	L	The Hanja characters that start the DBCS word will sort to the same position as the secondary Hangeul pronunciation of the characters.
9-12	2	The data field begins in column 2 of the input record.
13-16	9	The control field ends in column 9 of the input record.
40-72	Comments	Sort using conversion tables.

Bibliography

The manuals below are listed with their full title and base order number, and with the shortened version of the title. When these manuals are referred to in the text, the shortened version of the title is used.

- *ADTS/400: Character Generator Utility*, SC09-1769
  Short title: *ADTS/400: Character Generator Utility*
- *ADTS/400: Programming Development Manager*, SC09-1771
  Short title: *ADTS/400: Programming Development Manager*
- *ADTS/400: Screen Design Aid*, SC09-1768
  Short title: *ADTS/400: Screen Design Aid*
- *ADTS/400: Screen Design Aid for the System/36 Environment*, SC09-1893
  Short title: *ADTS/400: Screen Design Aid for the System/36 Environment*
- *ADTS/400: Source Entry Utility*, SC09-1774
  Short title: *ADTS/400: Source Entry Utility*
- *Backup and Recovery – Advanced*, SC41-3305
  Short title: *Backup and Recovery – Advanced*
- *Data Management*, SC41-3710
  Short title: *Data Management*
- *System Operation for New Users*, SC41-3200
  Short title: *System Operation for New Users*
- *CL Programming*, SC41-3721
  Short title: *CL Programming*
- *CL Reference*, SC41-3722
  Short title: *CL Reference*
- *Programming Reference Summary*, SX41-3720
  Short title: *Programming Reference Summary*
- *Publications Ordering*, SC41-3000
  Short title: *Publications Ordering*

Glossary

active sort table. A system-supplied DBCS sort table that contains the collating sequences for all defined double-byte characters in a double-byte character set. These tables are maintained by the character generator utility function of the Application Programming Tools licensed program.

alternative collating sequence. A user-defined collating sequence that replaces the standard EBCDIC collating sequence.

ascending key sequence. The arrangement of data in order from the lowest value of the key field to the highest value of the key field. Contrast with *descending key sequence*.

binary. (1) Pertaining to a system of numbers to the base two; the binary digits are 0 and 1. (2) Involving a choice of two conditions, such as on-off or yes-no.

Character Generator Utility (CGU). A function of the Application Programming Tools licensed program that is used to define and maintain user-defined double-byte characters and related sort information.

collating sequence. The sequence in which characters are ordered within the computer for sorting, combining, or comparing.

conditional force. The replacement of control field characters before the records are sorted. This is done if the control field in the input record contains a particular entry.

control field. A field that identifies a record's relationship to other records (such as a part number in an inventory record). Control fields determine the order of records in the sorted file.

DBCS. See *double-byte character set (DBCS)*.

DBCS code. The hexadecimal code, 2 bytes in length, that identifies a double-byte character.

DBCS font file. A system-supplied file that holds the 24x24 character images of one of the following groups of commonly used characters: 1) Japanese non-Kanji and basic-Kanji, 2) Korean non-Hangeul/non-Hanja, Hangeul, and a subset of Hanja, 3) Traditional Chinese non-Chinese and a subset of Traditional primary Chinese characters, or 4) all IBM-defined Simplified Chinese characters.

DBCS font table. A system-supplied table that holds either 24x24 or 32x32 character images of a double-byte character set. Japanese 24x24 font table holds Japanese extended Kanji and user-defined characters. Korean 24x24 font table holds a subset of Hanja and user-defined characters. Traditional Chinese 24x24 font table holds a subset of primary Traditional Chinese, all secondary Chinese, and user-defined characters. Simplified Chinese 24x24 font table holds user-defined characters. A 32x32 DBCS font table holds character images of a complete double-byte character set, including its user-defined characters.

DBCS number. The decimal value, two through five digits in length, that identifies a double-byte character.

DBCS sort table. A system-supplied object that contains sequencing information to sort double-byte characters. See also *master sort table* and *active sort table*.

descending key sequence. The arrangement of data in order from the highest value of the key field to the lowest value of the key field. Contrast with *ascending key sequence*.

Double-Byte Character Set (DBCS). A set of characters in which each character is represented by 2 bytes. Languages such as Japanese, Chinese, and Korean, which contain more symbols than can be represented by 256 code points, require double-byte character sets. Because each character requires 2 bytes, typing, displaying, and printing DBCS characters requires hardware and supporting programs that are DBCS-capable. There are four double-byte character sets supported by the system: Japanese, Korean, Simplified Chinese, and Traditional Chinese.

force-all. A specification that tests if the control field in the input record contains a particular entry. If it does not, the control field character is replaced before the record is sorted.

forced character substitution. The substitution of one character of the control field by another for sorting purposes.

forced control field. A one-position control field that results from replacing one character with another, or from forcing a character into a control field position.

Hangeul. A written language of Korea. Each Hangeul character is composed of two to six Jamo characters.

Hanja. Chinese characters used in Korean written language.

Hiragana. A native Japanese character set used mainly to express the pronunciation of native Japanese words. Contrast with *Katakana*.

include set. Sequence specifications that identify one or more record types to be sorted.

Jamo. Elements of Korean written language. Korean alphabet.

Japanese basic-Kanji character set. A subset of Japanese DBCS, consisting of commonly used Kanji characters. There are 3 226 Kanji characters in this set.

Japanese extended-Kanji character set. A subset of Japanese DBCS, consisting of less commonly used Kanji characters. There are 3 487 characters in this set.

Japanese double-byte character set. An IBM-defined double-byte character set for Japanese consisting of the Japanese non-Kanji set, basic Kanji set, extended Kanji set, and up to 4 370 user-definable characters.

Japanese non-Kanji character set. A subset of the Japanese DBCS, consisting of non-Kanji characters like Greek, Russian, Roman numeric, alphanumeric and related symbols, Katakana, Hiragana, and special symbols. There are 550 characters in this set.

Kanji. Chinese characters used in Japanese written language.

Katakana. A native Japanese character set that is used to write foreign words phonetically in Japanese. Contrast with *Hiragana*.

Korean double-byte character set. An IBM-defined double-byte character set for Korean, consisting of Korean non-Hangeul/non-Hanja set, Hangeul set, Hanja set and up to 1 880 user-definable characters.

Korean Hangeul character set. A subset of the Korean DBCS, consisting of 2 369 Hangeul characters and 52 Jamo characters.

Korean Hanja character set. A subset of the Korean DBCS, consisting of 4 500 Hanja characters.

Korean non-Hangeul/non-Hanja character set. A subset of the Korean DBCS, consisting of non-Hangeul/non-Hanja characters, such as Greek, Russian, Roman numeric, alphanumeric and related symbols, Katakana, Hiragana, and special symbols. There are 398 characters in this set.

master sort table. A system-supplied table that contains sort information required for sorting double-byte characters. This table is maintained by the character generator utility function of the Application Development Tools licensed program.

normal control field. A control field that is sorted in the sequence specified in the header specification.

opposite control field. A control field that is sorted in the opposite sequence of that specified in the header specification.

output file. A file resulting from the resequencing or copy request.

overflow indicator. A specified character that is to be placed at the end of an output record in the case of an overflow during a summary sort.

overflow field. A field that allows for field expansion.

overflow indicator field. The one-character field into which the *overflow character* is placed.

packed decimal format. A format in which each byte (except the rightmost byte) within a field represents two numeric digits. The rightmost byte contains one digit and the sign. For example, the decimal value +123 is represented as 0001 0010 0011 1111. Contrast with *zoned decimal format*.

record address file. A physical file that contains 4-byte binary relative record numbers rather than data.

set. A grouping of one or more record specifications.

shift-in character. A control character (hex 0F) that indicates the end of a string of double-byte characters. Contrast with *shift-out character*.

shift-out character. A control character (hex 0E) that indicates the start of a string of double-byte characters. Contrast with *shift-in character*.

Simplified Chinese. The Chinese character set that has been simplified by reducing the number of strokes in common characters and deleting complicated variants. Simplified Chinese characters are used primarily in the People's Republic of China.

Simplified Chinese double-byte character set. An IBM-defined double-byte character set for Simplified Chinese. It consists of Simplified Chinese non-Chinese set, primary set, secondary set, and up to 1 880 user-definable characters.

Simplified Chinese non-Chinese character set. A subset of the Simplified Chinese DBCS, consisting of non-Chinese characters, such as Latin alphabet, Greek, Russian, Roman numeric, alphanumeric and related symbols, Katakana, Hiragana, Japanese, special symbols, and Chinese phonetic symbols. There are 712 characters in this set.

Simplified Chinese primary character set. A subset of the Simplified Chinese DBCS, consisting of commonly used Chinese characters. There are 3 755 characters in this set.

Simplified Chinese secondary character set. A subset of the Simplified Chinese DBCS, consisting of less commonly used Chinese characters. There are 3 008 characters in this set.

sort sequence specifications. Source statements that specify the sequence of a sort.

specification sheets. Forms on which a program is coded and described.

summary data field. A data field designated for accumulated totals.

Traditional Chinese. The Chinese character set expressed in traditional form. Traditional Chinese characters are used in Taiwan, Hong Kong, and some other parts of the world.

Traditional Chinese double-byte character set. An IBM-defined double-byte character set for Traditional Chinese, consisting of the traditional Chinese non-Chinese set, primary set, secondary set, and up to 2 632 user-definable characters.

Traditional Chinese non-Chinese character set. A subset of the Traditional Chinese DBCS, consisting of non-Chinese characters, such as Greek, Russian, Roman numeric, alphanumeric and related symbols, Katakana, Hiragana, special symbols, and Chinese phonetic symbols. There are 675 characters in this set.

Traditional Chinese primary character set. A subset of the Traditional Chinese DBCS, consisting of commonly used Chinese characters. There are 5 401 characters in this set.

Traditional Chinese secondary character set. A subset of the Traditional Chinese DBCS, consisting of less commonly used Chinese characters. There are 7 652 characters in this set.

unconditional force. A specification that always results in a character being forced into the control field before the records are sorted.

work area. An area in storage reserved for temporary storage of data that is being resequenced.

work file. A file that is used for temporary storage of data being processed.

work record. A record built by the Sort program for later processing.

zoned decimal format. A format for representing numbers in which the digit is contained in bits 4 through 7 and the sign is contained in bits 0 through 3 of the rightmost byte; bits 0 through 3 of all other bytes contain 1s (hexadecimal F). 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.

Index

Special Characters

*, as overflow character 48
*CHK option, FMTDTA command 78
*CURLIB 75
*CURRENT option 79
*DATE option 79
*DUMP option 78
*FIRST, member name 76
*LIBL 76
*NOCHK option, FMTDTA command 78
*NODUMP option 78
*NOPRT option 78
*NOSECLVL option 78
*PRT option, FMTDTA command 78
*SECLVL 15
  option of FMTDTA command 78
*SRC filetype 11

A

access restrictions 10, 12
accumulated totals
  example of 48
  overflow 108
active sort table
  brief description 125
  coding considerations
    field specifications 128
    header specifications 126
    record specifications 127
  definition 149
adding data field, summary sort 102
adding records, RA files 42
address file, record, definition 150
alarm 16
alphanumeric
  constants, Factor 2 field 96
  data, interpretation of 92
  fields 91
ALTCOLLSEQ column, header specification 120
alternative collating sequence
  advantage over System/38 Conversion Reformat Utility 115
  by field 109
  definition 149
  for individual fields 84
  header specification 84, 120
alternative collating sequence *(continued)*
  how to specify 120
  introduction 117
  limitation of System/38 Conversion Reformat Utility 115
  specifying on header specification 82, 111
ALTSEQ specifications
  as a type of sort specification 5
  for entire control field 121
  for part of control field 121
  how to code 122
  how to define 120
AND specifications
  continuation of preceding statement 89
  field specifications 38
  record specification 28, 89
arrival sequence 41
ascending key sequence, definition 149
attributes of output file 11
authority 11
automatic updating in RA files 42
auxiliary storage
  output file 13
  sort requests 13

B

begin/continue record definition, record specification 89
bibliography 147
binary representation
  definition 149
  of characters 93
  of signs 94
blank record specifications 89
blanks
  padding the output record 7
blinking cursor 16
building the control field 8
building the output record 7
byte strings 8

C

CGU (Character Generator Utility), definition 149
changing
  contents of control field 102
  input records 8

changing *(continued)*
records, RA files 42
the collating sequence 117
Character Generator Utility (CGU), definition 149
character-type
Japanese 130
Korean 142
Simplified Chinese 135
Traditional Chinese 139
characters
binary representation of 93
composition of
digit portion 91
zone portion 91
example, sorting character data 21
forced 108
identical digit portions 93
identical zone portions 92
insertion into sort sequence 122
special 117
substitution
field specification 108
forced, definition 149
forced, example 36
checking
sequence 101
status of job 14
syntax, System/36 Sort Utility 113
Chinese double-byte character sets
Simplified Chinese
character-type sequence 135
code sequence 136
radical sequence 135
single-pronunciation sequence 135
sorting considerations 135
strokes beyond radical sequence 135
strokes sequence 135
tie-breaker sequence 136
Traditional Chinese
character sequences 139
character-type sequences 139
code sequence 139
radical sequence 139
sorting considerations 139
strokes beyond radical sequence 139
strokes sequence 139
CL commands 4
ALCOBJ 10, 12
DLCOBJ 10, 12
DSPMSGD (display message) 15
CL commands *(continued)*
FMTDTA (Format Data)
See FMTDTA (Format Data) CL command language 73
STRSEU 4
coding negative constants into record specifications 97
collating sequence
alternative
by field 109
definition 149
order of sort records 10
using ALTSEQ specifications to describe 5
definition 149
determines comparison value 96
standard and alternative 117
System/38 Conversion Reformat Utility 115
column entries
ALTSEQ specification 122
comment specification 111
field specification 100
header specification 82
record specifications 87
command language, CL
See CL commands
command line, entering FMTDTA command from 9
comment specification 111
comments
entering 5
field specification 109
header specification 85
location of comment specifications 111
record specification 98
comparing Factor 1 and Factor 2 fields 95
comparison operators
description 96
EQ 96
GE 96
GT 96
LE 96
LT 96
NE 96
record specification 95
comparison, using the collating sequence 96
conditional force
conditionally forced character
description 108
field specification 108
definition 149

conditional force *(continued)*
force-all
control field 102
description 103
field 103
forced control field 102
forced field 103
constant
Factor 2 96
format of 97
signed, Factor 2 field 97
constant value for Factor 2 96
constant value for Factor 2, record specification 96
continuation 108
continue/begin record definition, record specification 89
control field
alternative collating sequence 120
building of 8, 23
changing content of 102
characters, changing 104
composition of 83
definition 149
duplicate values 10
field specifications 102
forced
conditional 103
definition 149
field specification 102
normal/opposite 104
unconditional 104
length 83
maximum length in System/38 Conversion Reformat Utility 115
maximum length of 83
normal
definition 150
field specification 101, 102
omission of 83
opposite
definition 150
field specification 102
output option 106
output option with opposite 106
packed 106
specifying data fields as 102
summary sort 48
zoned 106
control field value, unique for summary sort 48
control language
See CL commands
conventions, file naming 73
copy-only request 68, 83
copying
merging 67, 83
overview 67
records 10
with reformatting 67, 71
without reformatting 68
correcting errors 16
creation of output record 10
current library 75
current library list 76
cursor, blinking 16

D

data description
field descriptions override 8
output file 11
record format not checked against 11
data field
entry on field specification 101
field specifications 102
summary sort 48
summary, definition 151
summing 102
data file identifier 77
data management 11, 12, 13
data management view 11
data type
alphanumeric 92
character 92, 106
description 104
digit 93
numeric, signed 92
record specification 91
signed numeric 92
specifying 104
database file 10, 12, 76
date format, international 98
DBCS
See double-byte character set
decimal format
packed, definition 150
zoned, definition 151
default source file
overriding 12
QFMTSRC 9

defaults
normal control fields 102
using all input records 89
deleting records, RA files 42
device file
as input file 11
diskette 12
multiple input 41
printer 77
QSYSPRT 77
single input 41
diagnostic message, System/36 Sort Utility 113
digit 91
digit portion
EBCDIC 117
of character byte 91
diskette files 76, 77
diskette unit file 12
diskettes 10
disks 10
display parameters 77
displaying messages 15
documenting the job 111
double-byte character set
character-type 135, 139
character-type sequence 130
code sequence
Japanese 130
Simplified Chinese 136
Traditional Chinese 139
code, definition 149
definition 149
font file, definition 149
number, definition 149
pronunciation 142
radical 129, 135, 139
Sei-On Katakana sequence 130
single-pronunciation 135
single-pronunciation sequence 129
sorting 125
strokes 129, 135, 139
strokes beyond radical 135, 139
tie-breaker 136
tie-breaker sequence 130
DSPMSGD, CL command 15

E

EBCDIC collating sequence
description 117
digit portion comparison 120
EBCDIC collating sequence *(continued)*
standard collating sequence 118
zone portion comparison 119
entering 4
comments 5
specifications 3
error
Error Reset key 16
messages 78
missing file 11
exception condition, RA files 42
exclude input records 87

F

Factor 1
end position, record specification 95
field length 92, 95
field, record specification 87
fields, alternative collating sequence 121
length restrictions 95
start position, record specification 95
Factor 2
1-character long 96
alphanumeric constants 96
as a keyword, record specification 97
comparisons to packed numeric fields 96
comparisons to zoned numeric fields 96
constant 96
end position, record specification 96
field length 92, 96
field, record specification 87
fields, alternative collating sequence 121
length considerations 96
length restrictions 96
numeric constants 97
signed constants 97
start position, record specification 96
UDATE keyword 97
UDAY keyword 97
UMONTH keyword 97
UYEAR keyword 97
field length
Factor 1, restrictions on 95
Factor 1/Factor 2 92
restrictions
on field statement 107
specifying
Factor 1 95
Factor 2 96
on field statement 107

field location in input record, field specification 107
field specifications
  column 8 entries 101, 106
  description 5
  entering 99—101
field-level
  information 12
  manipulations 12
  view 13
field, control, definition 149
fields
  alphanumeric 91
  alternative collating sequence 84
  data
    entry on field specification 101
    field specifications 102
  Factor 1 and Factor 2 87
  Factor 1, length 95
  input, start/end positions 8
  numeric
    description 91
    packed format 91
    zoned format 91
  order in output record 7
  order of 99
  overflow 48
  reformatting 1
  right justification 4
  selecting and rearranging 1
  summary data 47
files
  *SRC filetype 11
  database 10
  differences from System/38 Conversion Reformat Utility 115
  file naming conventions 73
  filetype source 11
  input same as output 12
  keyed logical 11
  keyword replaced by SORTR 115
  names
    QFMTMSG 15
    QFMTSRC 9
    QFMTTXT 15
    sort message file 15
  nonkeyed logical 11
  physical 1
  printer 10
  program identification in header specification 85
files *(continued)*
  QFMTMSG 15
  QFMTSRC 9
  QFMTTXT 15
  QSYSPRT 12
  source 5
  source, order of field specifications within 99
  writing to 11
FMTDTA (Format Data) CL command 9
  *CURRENT option 79
  *DATE option 79
  *DUMP option 78
  *NOCHK option 78
  *NODUMP option 78
  *NOPRT option 78
  *NOSECLVL option 78
  *PRT option 78
  *SECLVL option 78
  access and use 73
  as a line command 9
  command line, entering from 9
  first display 74
  input file name parameter 75
  input library name parameter 75
  input member name parameter 75
  options parameter 78
  options, *CHK 78
  output file parameter 76
  parameters explained 75
  print file name parameter 77
  program date parameter 78
  prompt displays 74
  purpose of 73
  running the sort job 75
  screen displays 9
  second display 75
  source file name parameter 77
  syntax of 74
force character, field specification 108
force-all statement 102
force-all, definition 149
force, conditional, definition 149
force, unconditional, definition 151
forced character substitution, definition 149
forced character, field specification 108
forced control field 149
  conditional 103
    force-all 103
  field specifications 102
  normal/opposite 104

forced control field *(continued)*
unconditional 104
forced field, unconditional 104
form
field specification, sample 99
format
date, international 98
input and output record 12
unpacked, numeric fields 91
Format Data (FMTDTA) CL command
See FMTDTA (Format Data) CL command
format description 12
format of numeric fields
packed 91
zoned 91
format, unpacked 19

G

graphic symbol, printable 123

H

Hangeul, definition 149, 150
Hanja, definition 149, 150
header specification 5, 81
alternative collating sequence 84, 120
column entries 82
column summary 82, 111
comments 85
differences from System/36 Sort Utility 113
file-name 85
include/exclude control field 84
maximum control field length 83
output file type 83
page 83
sort sequence 84
SORTA entry 83
SORTR entry 83
SORTRS entry 83
specification identifier 83
statement number 83
Help key 16
Hiragama, definition 149

I

identifier
comment specification 111
specification, field specification 101
include (record) specifications 89
include input records 87
include set, definition 150
include-all specifications 68
include/exclude control field, header specification 84
including input records 89
indicator field, overflow, definition 150
indicator, summary overflow, field specification 108
input fields, start/end positions 8
input file name 75
input files 10
device file 11
processing 11
same as output file 12
security 10
sequential processing of 11
single 17
used by several programs simultaneously 12
input library name 75
input member name 75
input record
accessibility 9
changing 8
format 12
include/exclude 87
including 89
input field, location of 107
length 11
omitting 89
type 7
inserting characters into sort sequence 122
interactive entry of specifications 4
internal tables 78
international date format 98
interpreting alphanumeric data 92
interpreting numeric data 93
interpreting the sign of a number 94

J

Jamo, definition 150
Japanese character set, Kanji, definition 150
Japanese double-byte character set, definition 150
Japanese double-byte character sets
character sequences 129
character-type 130
DBCS code sequence 130
radical 129
Sei-On Katakana sequence 130

Japanese double-byte character sets *(continued)*
character sequences *(continued)*
single-pronunciation 129
strokes 129
tie-breaker 130
character-type 130
DBCS code sequence 130
radical sequence 129
Sei-On Katakana sequence 130
single-pronunciation 129
strokes 129
tie-breaker 130
Japanese non-Kanji character set, definition 150
job, summary sort 104
justification, entering fields 4

K

Kanji character set, basic, definition 150
Kanji character set, extended, definition 150
Katakana, definition 150
key sequence 41
key sequence, ascending, definition 149
keyboard, locked 16
keyed access path 11
keyed logical file 11
keying specifications 3
keyword
Factor 2, record specification 97
UDATE 95
UDATE, Factor 2 field 97
UDAY, Factor 2 field 97
UMONTH 95
UMONTH, Factor 2 field 97
UYEAR 95
UYEAR, Factor 2 field 97
value for Factor 2 96
Korean character set,
non-Hangeul/non-Hanja 150
Korean double-byte character set, definition 150
Korean double-byte character sets
pronunciation 142
sort tables 125

L

length
exceeding maximum for output record 109
Factor 1 and Factor 2 fields 92
Factor 1/Factor 2, System/36 Sort Utility 114
input record 11
length *(continued)*
input records 8
of control field 83
of Factor 1 field 95
of work records 13
overflow field, field specification 109
print file output record 12
libraries 4
library
current 75, 76
library with sort message files 15
temporary work 12
limitations
copy-only requests 13
length of forced control field 102
sort requests 13
line command 9
linking field specifications 38
location of field in input record, field
specification 107
locked keyboard 16
logical file
multiple input 35
single input 17, 67

M

making characters equal in sort sequence 123
master sort table 125
master sort table, definition 150
maximum control field length
header specification 83
summary sort 48
maximum length for output record,
exceeding 109
maximum length of Factor 1/Factor 2, System/36
Sort Utility 114
member 4
merging records 10
merging/copying
to produce a reformatted output file 71
messages 14, 78
error 15
first-level text 15
informational 15
second-level text 16
System/36 Sort Utility 113
modifying records, RA files 42

N

name
  file 85
names of output file and member 12
naming
  conventions 73
  file 73
negative constants, coding of 97
nonkeyed logical file 11
normal control field 102
  field specification 101, 102
normal control field, definition 150
number 91
  interpreting the sign of 94
  sequence, field specification 101
  signed 94
number of records in work area 13
number of records, copy-only requests 13
numeric constants, Factor 2 field 97
numeric data
  interpreting 93
  signed 92
numeric fields 91
  packed format 91
  zoned format 91

O

omit (record) specifications 89
omit specifications 68
omitting input records 89
opposite control field 102, 150
  field specification 102
  output option 106
opposite control field, definition 150
opposite sorting 29
optional attributes of output file 11
optional specifications 6
OPTIONS
  *CHK 78
  *DUMP 78
  *NOCHK 78
  *NOPRT 78
  *NOSECLVL 78
  *SECLVL 78
  keyword value
options parameter 78
OR a record specification 90
OR extension 108
  to forced field 108
order of field specifications in source file 99
order of fields, 99
order of specifications 2, 78
ordering specifications 6
output file 11, 150
  header specification 83
  physical 17, 35, 67
  record address 41
  record length 11
  same as input file 12
  storage 13
  type 83
output file parameter
  FMTDTA command 76
output record
  building of 7
  creation 10
  exceeding maximum length for 109
  format 12
  length, print file 12
  order of fields 7
  padding with blanks 7, 102
  type 7
output, redirecting 12
overflow
  field 48
  field length 109
  handling overflow 48
  indicator 48, 64
  indicator, field specification 108
overflow field, definition 150
overflow indicator field definition 150
overflow indicator, definition 150
overriding defaults 11
  source file 12

P

packed
  control field 106
  numeric fields 91
   zoned numbers, sign placement 94
packed decimal format, definition 150
padding
  output records with blanks 102
  padding with zeros 4
  with blanks 7
parameters, FMTDTA command 8, 75
physical file
  as output 17, 35, 67
  multiple input 35

physical file *(continued)*
single input 17
with physical output file 17, 35, 67
with record address file output 41
positioning record specifications 89, 101
positioning specifications in source file 83
print file 12
description 12
name parameter, FMTDTA command 77
number of 10
specifying on FMTDTA command 12
print option, System/36 Sort Utility 113
printer 10
device file 77
file QSYSPRT 77
files 10
system 12
printing error/warning messages 78
printing messages 15
problems 14
processing 10
a Sort job 3
speed 11
time, saving 102
producing totals 47
program date parameter 78
program date, record specifications 98
program status message, System/36 Sort Utility 113
pronunciation, Korean 142

Q

QFMTMSG file 15
QFMTSRC file 9, 77
QFMTTXT file 15
QSYSPRT file 12, 77

R

RA file 1
adding records 42
automatic updating 42
changes to input file 1
changing records 42
deleting records 42
including data fields 42
omission of control field 83
record length 13
sorting 41
radical sequence, Japanese 129
radical, Simplified Chinese 135
radical, Traditional Chinese 139
RAF 115
read in arrival sequence 11
reading sequentially 41
Record Address (RA) file 41, 83, 150
as output 41
description 41
record address file, definition 150
record format
description 12
filed-level view 13
input and output 12
record length
actual 11
input records 11
output file 11
print file 12
work area 13
record number
copy-only requests 13
relative 41
record selection, alternative collating sequence 121
record specification 5
record specifications 87
AND 28, 89
begin/continue record definition 89
blank 89
coding of negative constants 97
column entries 87
column summary 87
comments 98
comparing Factor 1 and Factor 2 95
comparison operator 95
constant value for Factor 2 96
data type 91
Factor 1 end position 95
Factor 1 start position 95
Factor 2 end position 96
Factor 2 is a keyword 97
Factor 2 start position 96
OR 90
page 89, 101
sequence checking 89
set of 87
specification identifier 89
statement number 89, 101
type of Factor 2 data 96

record specifications, input record type 7
record specifications, purpose of 6
record type 6, 7, 87
  beginning a new 7
  defining 6
  input 7
  order of fields within 99
  output 7
  specifying different 87
records 1
  format considerations 12
  input, include/exclude 87
  input, including 89
  input, location of input field 107
  input, omitting 89
  limitations on number of
    copy request 13
    sort request 13
  number in work area 13
  output, exceeding maximum length for 109
  output, padding with blanks 102
  Sort treatment of 87
redirecting output 12
reformatting 1, 7
related printed information 147
relative record number 41
restricting access 10
restricting file access 12
restrictions on Factor 1 field length 95
restrictions, Factor/Factor2 field length 92
return code after processing 14
right justification 4
RTVJOBA command 14
running the sort job 75

S

saving space in memory 102
screen displays
  first FMTDTA prompt display 74
  messages 14
  second FMTDTA prompt display 75
second-level messages 78
security 11
Sei-On Katakana sequence, Japanese 130
select and rearrange fields 1
selection criteria 9
sequence
  arrival 41
  key 41
sequence checking 78, 101
  record specifications 89
sequence number
  description 101
  field specification 101
  header specification 83
  record specification 89, 101
sequence, alternative by field 109
sequence, collating, alternative, definition 149
sequence, collating, definition 149
sequence, key, ascending, definition 149
sequencing specifications 6
sequential processing of input files 11
sequential reading 41
service information
set of record specifications 87
set, definition 150
SEU 4
shared-read lock 10
shared-update lock 12
shift-in character, definition 150
shift-out character, definition 150
signed constants, Factor 2 field 97
signed numeric data 92
signs
  binary representation of 94
  placement in packed and zoned numbers 94
Simplified Chinese double-byte character set, definition 150
Simplified Chinese double-byte character sets
  character-type 135
  DBCS code sequence 136
  radical 135
  single-pronunciation 135
  strokes 135
  strokes beyond radical 135
  tie-breaker 136
Simplified Chinese, definition 150
single-pronunciation sequence, Japanese 129
single-pronunciation, Simplified Chinese 135
sort order, overriding 84
sort process
  how to use 2—3
  maximization 9
sort requests, storage 13
sort sequence
  header specification 84
  making characters equal 123
sort table

sort table, active, definition 149
sort, master table, definition 150
SORTA entry, header specification 83
SORTA in place of RAF 115
sorting 8
  an RA file 43
  five input files 36
  in a different order 120
  include/exclude input records 87
  Japanese DBCS 129
  Korean DBCS 142
  opposite 29
  produce an RA file 41
  several input files 35
  Simplified Chinese DBCS 135
  single input file 17
  single input record type 17
  special characters 6
  Traditional Chinese DBCS 139
  two record types 22
  with reformatting 30, 33
  with totals 47, 83
sorting and summing
  detecting overflow 64
  one record type 49
  two input record types 54
  using an overflow field 63
SORTR entry, header specification 83
SORTR in place of FILE 115
SORTRS entry, header specification 83
SORTRS option 48
source file 5—8
  data starts in byte 13 11
  default 9
  description 12
  order of field specification within 99
  overriding default 12
  sort specifications, location of 12
  specifying on FMTDTA command 12
source file name parameter 77
specification identifier
  comment specification 111
  field specification 101
  header specification 83
  record specification 89
specification set
  name 85
specification sheets, definition 151
specifications 5, 6, 101
  comment 111
specifications *(continued)*
  entering no record or field specifications 10
  field 5, 100
  field, entered without record specifications 89
  force-all 103
  forms 2, 4
  forms, completing 3
  header 5, 81
    alternative collating sequence 84
    column entries 82
    comments 85
    file-name 85
    identifier 83
    include/exclude control field 84
    maximum control field length 83
    output file type 83
    page 83
    sort sequence 84
  include-all 68
  omit record 68
  optional 6
  order 78
  processing of 8
  record 5, 87
    AND 89
    begin/continue record definition 89
    blank 89
    column entries 87
    comments 98
    comparison operator 95
    constant value for Factor 2 96
    data type 91
    Factor 1 end position 95
    Factor 1 start position 95
    Factor 2 end position 96
    Factor 2 is a keyword 97
    Factor 2 start position 96
    OR 90
    page 89, 101
    sequence checking 89
    set of 87
    specification identifier 89
    statement number 89, 101
    type of Factor 2 data 96
  sequence checking 78
  sequence number 6
  sequencing of 6
  storing 5, 6
  type 83, 89, 101, 111
  types of 5

speed of processing 11
standard collating sequence 117
start/end positions in input records 8
status message, System/36 Sort Utility 113
status of job 14
storage
  auxiliary 13
  sort requests 13
storing specifications 5
strings of bytes 8
strokes beyond radical, Simplified Chinese 135
strokes beyond radical, Traditional Chinese 139
strokes sequence, Japanese 129
strokes, Simplified Chinese 135
strokes, Traditional Chinese 139
STRSEU, CL commands 4
substitution character, field specification 108
substitution of characters in output record 36
substitution of characters, forced, definition 149
summary data field 48, 102, 104
  description 47
  increasing size of 48
  overflow 48
summary data field, definition 151
summary overflow indicator, field
 specification 108
summary sort 47, 48, 83
summing 47
summing and sorting
  detecting overflow 64
  two input record types 54
  using an overflow field 63
syntax checking, System/36 Sort Utility 113
syntax of FMTDTA command 74
system data description 11
system printer 12, 77
system security 11

T

table, active sort, definition 149
tables
  generation of 13
  internal 78
temporary work library 12
testing for overflow indicator 64
tie-breaker, Japanese 130
tie-breaker, Simplified Chinese 136
total, overflow 108
totals 104
tracking problems 14
Traditional Chinese double-byte character sets
  character-type 139
  DBCS code sequence 139
  radical 139
  strokes 139
  strokes beyond radical 139
Traditional Chinese, definition 151
truncation of output record 11
type
  data, field specification 104
  field, field specification 101
  record 87
type of Factor 2 data, record specification 96

U

UDATE keyword
  Factor 1 field 95
  Factor 2 field 97
UDAY keyword, Factor 2 field 97
UMONTH keyword
  Factor 1 field 95
  Factor 2 field 97
unconditional control field 102
unconditional force
  description 104
unconditional force, definition 151
unconditional forced control field 104
unconditionally forced character
unique control field value, summary sort 48
unpacked format 19
updating, automatic 42
useful things to know 10—14
using all input records, default 89
UYEAR keyword
  Factor 1 field 95
  Factor 2 field 97

W

warning messages 78
work area 9, 10, 12, 151
  automatic deletion of 10
  copy-only requests 10
  description 12
  number of records 13
work area, definition 151

work file, definition 151
work library 12
work record length 13
work record, definition 151
writing to a file 11

Z

zone portion
  EBCDIC 117
  of character byte 91
zoned decimal format, definition 151
zoned field, definition 151
zoned format, numeric fields 91

Program Number: 5763-SS1

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

SC09-1826-00