DataGeneral

RDOS
SORT/MERGE
User's Manual

093–000108–00

RDOS
SORT/MERGE
User's Manual

093–000108–00

Ordering No. 093-000108

Printed in the United States of America
Rev. 00, February 1975

Original Release - September 1974
086-000021-00 - February 1975

This manual, 093-000108-00, and Addendum 086-000021-00 is a complete revision of the manual.

TABLE OF CONTENTS

CHAPTER 1 - General Description

Introduction 1-2
Operation 1-2
Functional Description 1-2
Rearrange Records in a File 1-2
Delete Records in a File 1-4
Reformat Records 1-5
Produce Address File 1-6
Merge Files 1-7
Key Fields 1-8
Collating Sequence 1-9

CHAPTER 2 - RDOS Sort/Merge Functional Characteristics

Sort Functional Characteristics 2-1
Merge Functional Characteristics 2-3

CHAPTER 3 - Operating Procedures

General Operating Instructions 3-1
Input/Output Control Field Specifications 3-3
User-Supplied Collating Sequence 3-4
Upper/Lower Control Limit Specifications . . . 3-5
Deletion of Records 3-6
Work File Usage 3-6

APPENDIX A - Error Messages A-1

APPENDIX B - Sample Jobs B-1

Sample Job 1 . . . . . B-1
Purpose B-1
Input Record Format B-1
Output Record Format B-1
Discussion B-1

APPENDIX B - Sample Jobs

Sample Job 2 . . . . . B-3
Purpose B-3
Input Record Format B-3
Output Record Format B-3
Discussion B-3
Sample Job 3 . . . . . B-6
Purpose B-6
Input Record Format B-6
Output Record Format B-6
CLI and BATCH Command Lines B-7
Discussion B-7
Sample Job 4 . . . . . B-9
Purpose B-9
Input Record Format. B-9
Output Record Format B-9
Limit and Collating Sequence Files B-9
Discussion B-10
Sample Job 5 . . . . . B-12
Purpose B-12
Input Record Format B-12
Output Record Format B-12
Discussion B-12
Sample Job 6 . . . . . B-14
Purpose B-14
Input Record Format B-14
Output Record Format B-14
Discussion B-15

APPENDIX C - Timing Considerations C-1

Main Storage Size C-1
Number of Records to be Sorted C-1
Record/Control Field Size C-2
Alternate Collating Sequence C-2
Location of the Files C-2
Installation Requirements C-2

APPENDIX D - System Preparation Guidelines D-1

CHAPTER 1

GENERAL DESCRIPTION

INTRODUCTION

RDOS Sort/Merge is a disk oriented sort/merge utility. It operates as a processing program under control of the Real Time Disk Operating System (RDOS). The program runs in the background of an unmapped DGC computer or in either partition of a mapped DGC NOVA® *840 Computer. Its execution is initiated according to the conventions of the Command Line Interpreter (CLI) or of BATCH; or it can be called for execution through a user program.

RDOS Sort/Merge performs the following functions:

- Rearrange the records in a disk or tape file.
- Delete records from a disk or tape file.
- Reformat the records in a disk or tape file.
- Produce sorted address files for a disk file.
- Merge disk or tape files into a single disk or tape file.

Normally, records in output files produced by RDOS Sort/Merge are exact images of records in the input file. A feature, however, allows the user to select portions of the input records for output. Using this feature, a variety of different output files can be generated from a single input file. This feature saves the user time by relieving the need to generate a unique input file for each desired output file. Another feature allows the user to produce an indexed sequential keyed access list by generating an output file that only contains pointers to the logical records of the input file and the key fields of those records.

RDOS Sort/Merge has additional options that allow the user to specify:

- Collating sequence order and type.
- Major and minor keys.
- Upper and lower bounds.

*NOVA is a registered trademark of Data General Corporation, Southboro, Massachusetts.

OPERATION

RDOS Sort/Merge accepts as input any sequential, random, or contiguous file containing fixed-length records in unblocked format. Maximum record size is 512 bytes and the default record size is 80 bytes.

Only one input file can be sorted during a particular execution; but up to six sorted files can be merged at one time.

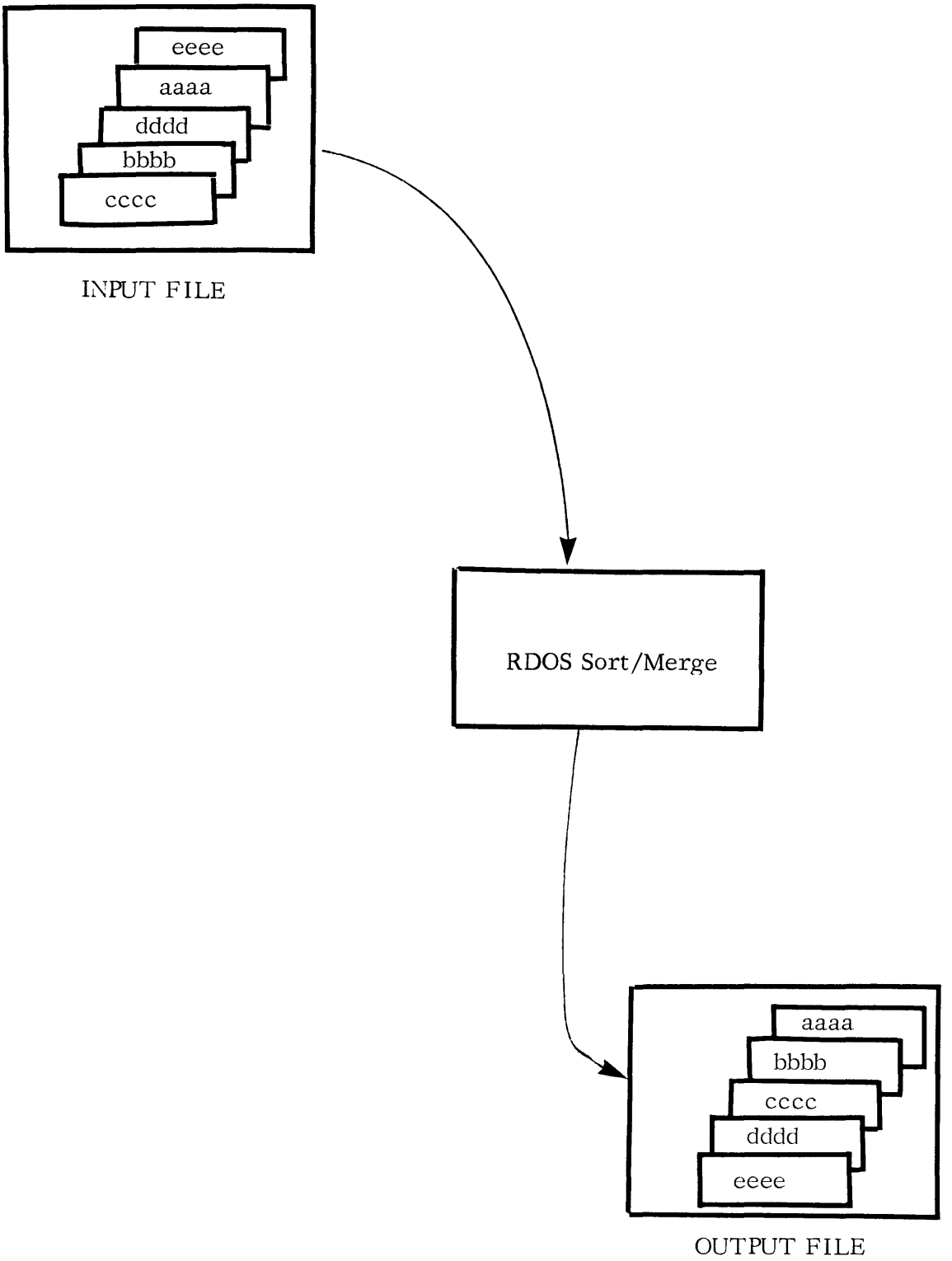
RDOS Sort/Merge requires the use of temporary work space. The amount of work space can be controlled by the user, so that the efficiency of a particular execution can be increased. Work space is on either disk or tape; or it can be on both disk and tape. If the user does not choose to control work space, RDOS Sort/Merge automatically provides it.

FUNCTIONAL DESCRIPTION

The following describes the functions of the RDOS Sort/Merge program.

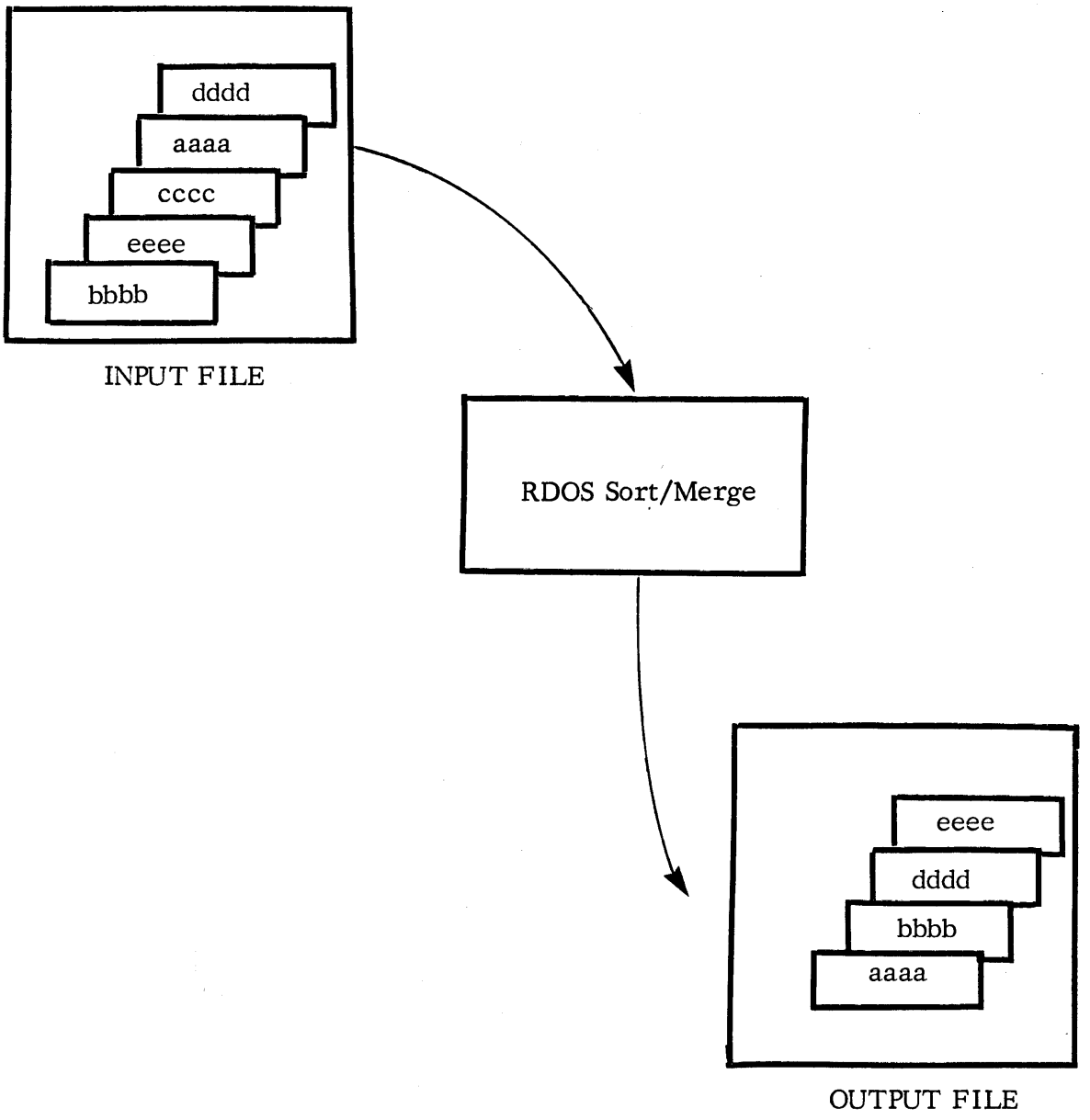
Rearrange Records in a File

When RDOS Sort/Merge rearranges records, the order of the records in the output file is determined by the user specified collating sequence. The user can specify ascending or descending ASCII or can specify a unique collating sequence.


eeee
aaaa
dddd
bbbb
cccc
INPUT FILE
RDOS Sort/Merge
aaaa
bbbb
cccc
dddd
eeee
OUTPUT FILE

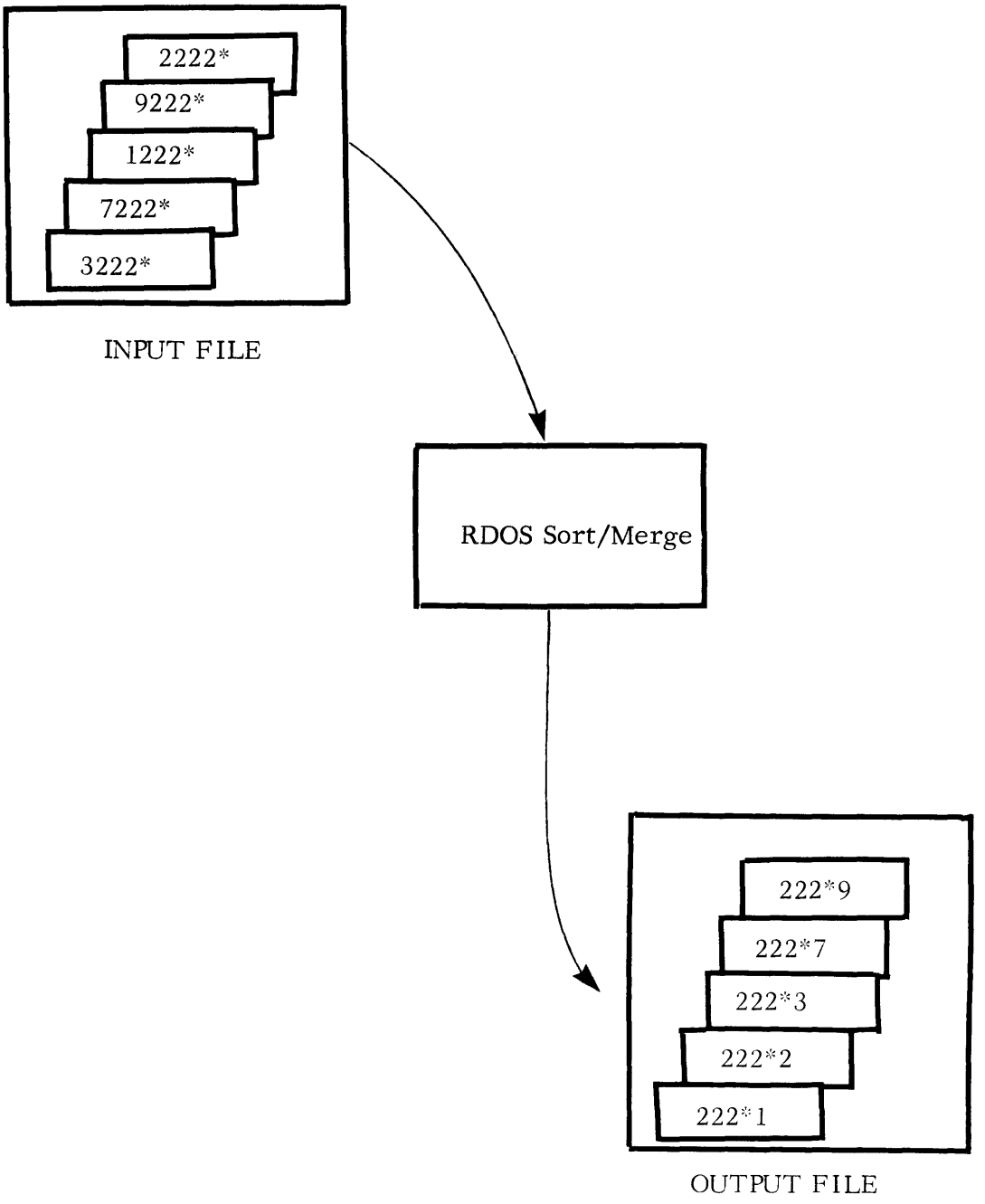
Delete Records from a File

The user can specify that RDOS Sort/Merge not include certain records in the output file.



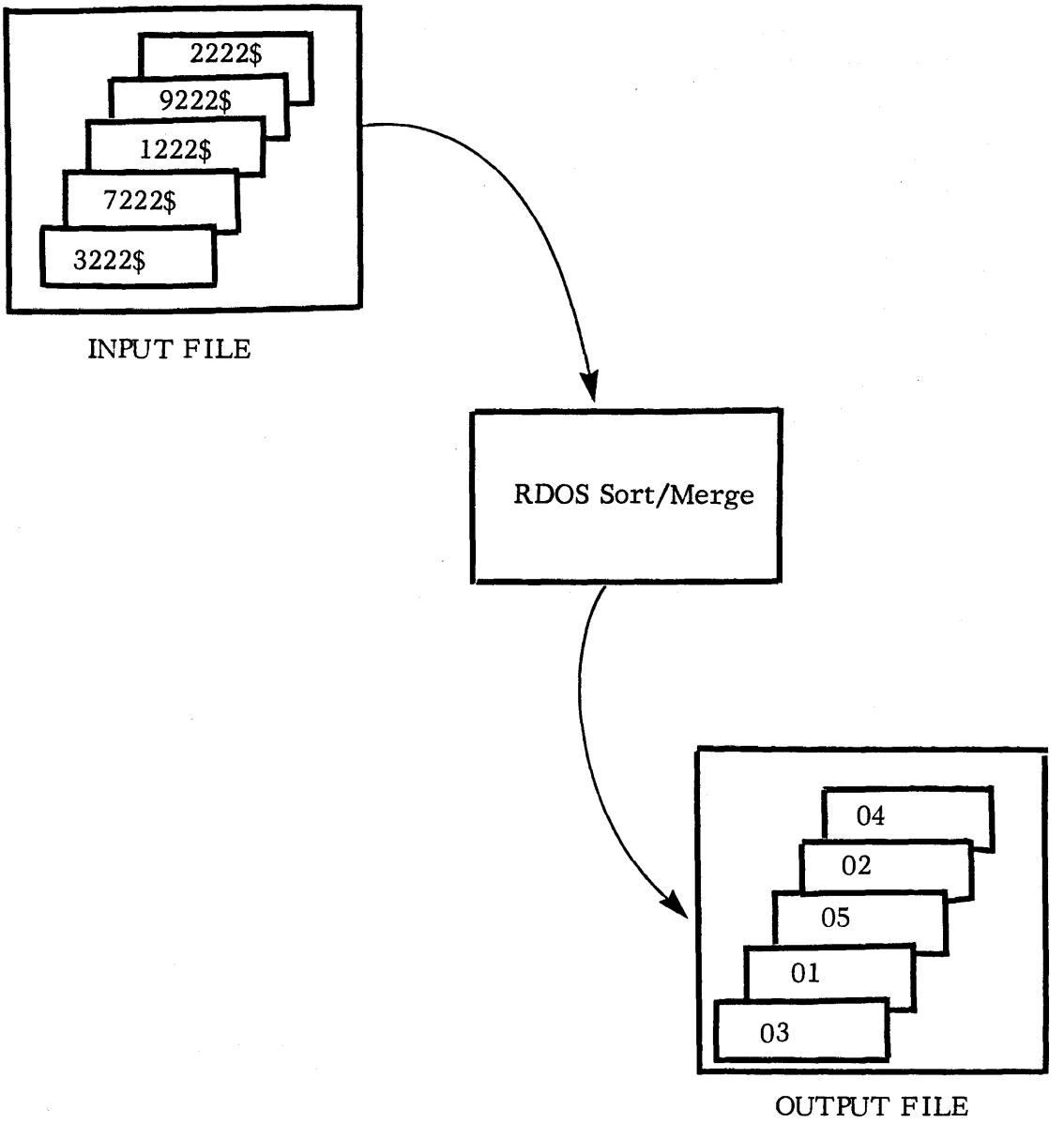
Reformat Records

Fields in an input record can be rearranged for records in the output file.



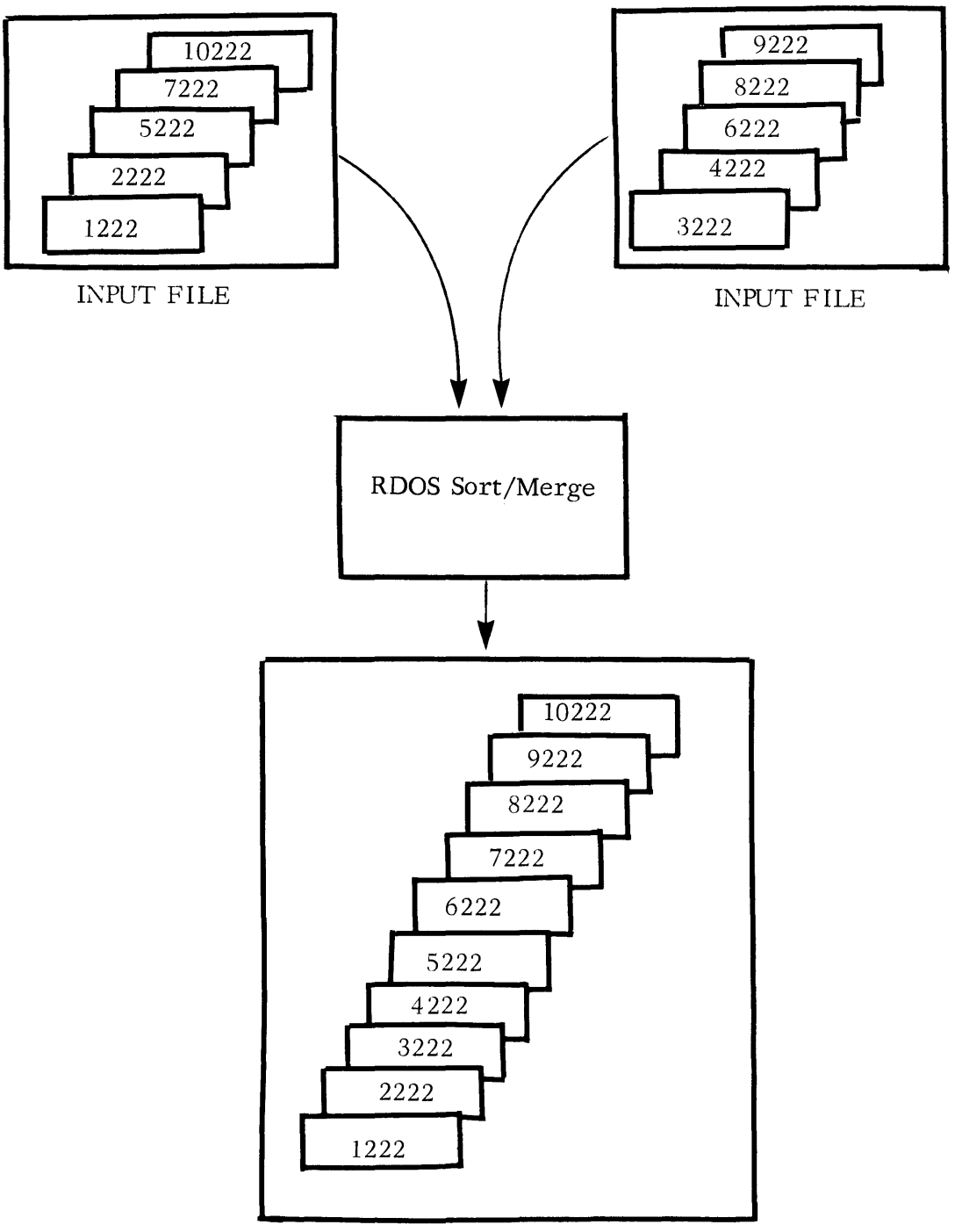
Produce Address File

RDOS Sort/Merge can produce an output file in which the records contain the relative record address of the records in the input file. Such an output file is called a TAG file.



Merge Files

RDOS Sort/Merge can combine up to six previously sorted input files into a single, sorted output file.

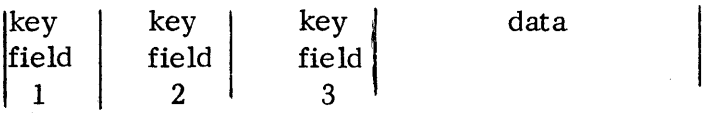


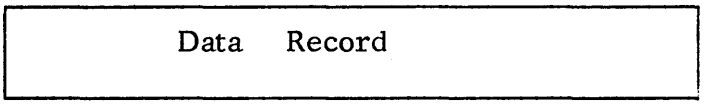
KEY FIELDS

The basis for determining the sequence of records in an output file is a group of characters that make up the Control Word for a record. RDOS Sort/Merge compares the data in the Control Word of each record in the input file to determine the sequence of records in the output file.

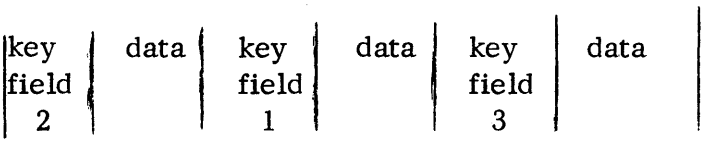
The Control Word can be divided into as many as eight Key Fields. The first Key Field the user specifies is the major Key Field. The remaining Key Fields are minor Key Fields. Key Fields are compared according to their relative order of specification. Minor Key Fields are compared only if the comparison of the next highest Key Field resulted in an equal condition.

Each Key Field can be contiguous to, separated from, or overlapping with other Key Fields. A Key Field can occur anywhere in a data record, but a given Key Field must be located in the same relative position in each record of the input file.



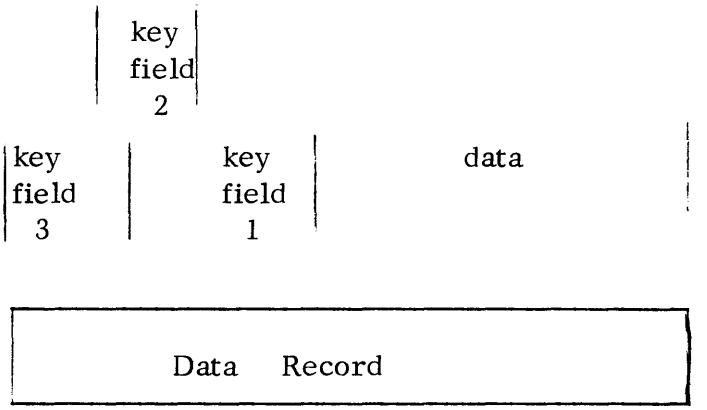


Contiguous Key Fields



Separated Key Fields

KEY FIELDS
(Continued)



Overlapping Key Fields

COLLATING SEQUENCE

RDOS Sort/Merge uses ascending ASCII, descending ASCII, or a user specified collating sequence. Data in Key Fields is compared and the sequence is determined according to the specified collating sequence. When a user specified collating sequence is used, the user supplies a file containing the characters in the order to be used by RDOS Sort/Merge.

*****

CHAPTER 2

RDOS SORT/MERGE FUNCTIONAL CHARACTERISTICS

SORT FUNCTIONAL CHARACTERISTICS

The following section describes how RDOS Sort/Merge works with data files in sort mode.

1. RDOS Sort/Merge reads a data record from the input file. This input file must be disk resident if a sorted output file is desired.

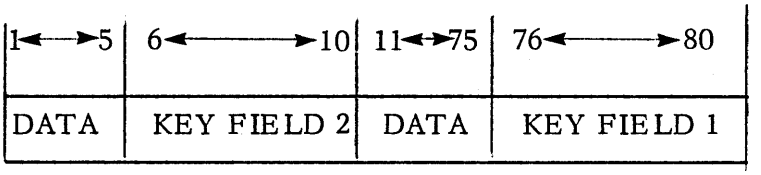
2. RDOS Sort/Merge checks the first word of the data record for a non-zero value. If a zero value is found, RDOS Sort/Merge assumes the record is not active in the data file and bypasses it, going to step 1 again to read another record.

3. RDOS Sort/Merge checks the Key Field first against upper and lower limit bounds (if they have been specified) to see whether the record is to be included in the sort operation.

4. If the record is to be sorted, RDOS Sort/Merge builds a work record with the first word specifying the relative record number within the input data file, and the remainder of the work record formatted according to Key Field specifications. RDOS Sort/Merge extracts the Key Fields from the input file record and translates them using the appropriate collating sequence. The format of the work record is important because it controls the order of sorting operations.

SORT FUNCTIONAL CHARACTERISTICS (Continued)

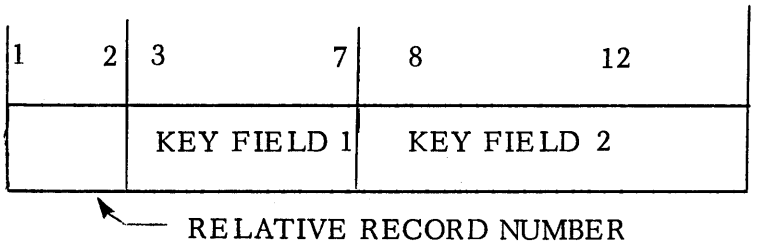
5. Assume the following input record format is specified:



Assume the Key Field specifiers are as follows:

- Use the contents of positions 76-80 in the input record as the primary Key Field.
- Use the contents of positions 6-10 in the input record as the secondary Key Field.

The work record built is:



6. RDOS Sort/Merge builds these work records in main memory until it is filled. It then sorts these records in memory and writes them out to a work file. The sort operation is in ascending or descending order or as specified by a user defined collating sequence. These coreloads of work records are written to one of three work files as illustrated below:

WORK FILE 1	WORK FILE 2	WORK FILE 3
Coreload #1	Coreload #2	Coreload #3
Coreload #4	Coreload #5	Coreload #6
Coreload #7	etc.	

7. RDOS Sort/Merge then merges these work files to produce a single work file of work records in sorted order. It builds this single work file by

SORT FUNCTIONAL CHARACTERISTICS (Continued)

gradually reducing the number of coreloads present in the work files. It merges coreloads #1, #2, and #3 and stores the output in work file 4. Coreloads #4, #5, and #6 are merged and stored in work file 5, and so on, until the contents of work files 1, 2, and 3 are merged and transferred to work files 4, 5 and 6. Work files 1, 2 and 3 are deleted and the above operation is again performed with work files 1, 2 and 3 receiving the merged output. The above series of operations continues until a single work file is produced.

8. RDOS Sort/Merge now produces a sorted output file (step 9) and /or a sorted tag file (step 10).

9. To produce a sorted output file, RDOS Sort/Merge reads records from the input file in the order specified by the sorted work file. The record is either completely written out to the output file or selected parts of the input record (output control field specifiers) are written out sequentially to the output file.

10. RDOS Sort/Merge retains the sorted work file for the user. This "tag" file is in the same format as the work file specified in step 5 above.

MERGE FUNCTIONAL CHARACTERISTICS

The following section describes how RDOS Sort/Merge works with data files in merge mode.

1. RDOS Sort/Merge reads a data record from each of the merge input files. These input files may be either disk or tape files.

2. RDOS Sort/Merge extracts the Key Fields from each record, translates them using the appropriate collating sequence, and compares these record Key Fields to find the lowest order Key Field.

MERGE FUNCTIONAL CHARACTERISTICS (Continued)

3. Once the lowest order Key Field is found, this record is output to the merge output file in total, or the specified output Key Fields are extracted from the input record and are output to the merge output file.

4. RDOS Sort/Merge reads a data record from the input file to replace the record just written to the merge output file and returns to step 2. This loop continues until all input records have been read and written to the merge output file.

*****

CHAPTER 3

OPERATING PROCEDURES

GENERAL OPERATING INSTRUCTIONS

RDOS Sort/Merge is called into execution by a command line to CLI entered through the operator console, by a BATCH job file, or by a user program calling for its execution. The following page shows the CLI command format. The BATCH Command line is similar in format; the major difference being the "!" character before RDOSSORT and the "#" character appearing as the first character in any continuation lines (instead of the " ↑ ").

If global switch /N is not used in the command line, a listing file of RDOS Sort/Merge statistics is produced. The listing is output to the console output device ($TTO or $TTO1) by default, if a listing file is not specified. The listing can be redirected by specifying the desired listing file name followed by a local switch /L. Local switch /L overrides the specification of no list file indicated by the global switch /N.

RDOS Sort/Merge assumes the input record size is 80 characters unless otherwise specified. To indicate the size of the record if the size is not 80, the user places a decimal number followed by a local switch /R in the command line. For example, for a record of 128 bytes the user enters 128/R.

RDOS Sort/Merge assumes the program is to operate in sort mode unless a global switch /M is used to indicate it is to operate in merge mode.

RDOS Sort/Merge assumes that the Key Fields are to be arranged in ascending ASCII Collating sequence unless a global switch /D is used to indicate descending ASCII collating sequence. The user may specify his own non-ASCII collating sequence if desired, by using the local switch /S.

RDOS SORT/MERGE PROGRAM COMMAND LINE FORMAT

CLI COMMAND LINE:

```
RDOSSORT</D></N></M> ^
         INPUT FILENAME</D> ^
         <INPUT FILE 2> <INPUT FILE 3> ^
         <INPUT FILE 4> <INPUT FILE 5> ^
         <INPUT FILE 6> ^
         <OUTPUT FILENAME/O> ^
         KEY SPECIFIER 1 ^
         <KEY SPECIFIER 2> ^
         <KEY SPECIFIER 3> <KEY SPECIFIER 4> ^
         <KEY SPECIFIER 5> <KEY SPECIFIER 6> ^
         <KEY SPECIFIER 7> <KEY SPECIFIER 8> ^
         <OUTPUT FIELD 1> <OUTPUT FIELD 2> ^
         <OUTPUT FIELD 3> <OUTPUT FIELD 4> ^
         <OUTPUT FIELD 5> <OUTPUT FIELD 6> ^
         <OUTPUT FIELD 7> <OUTPUT FIELD 8> ^
         <WORK FILE 1/W> <WORK FILE 2/W> ^
         <WORK FILE 3/W> <WORK FILE 4/W> ^
         <WORK FILE 5/W> <WORK FILE 6/W> ^
         <SORTED KEY FILENAME/K> ^
         <LISTING FILENAME/L> ^
         <SEQUENCE FILENAME/S> ^
         <RECORD SIZE/R> ^
         <LOWER LIMIT FILENAME/B> ^
         <UPPER LIMIT FILENAME/U>
```

NOTE: INFORMATION CONTAINED IN <> IS OPTIONAL.

GLOBAL SWITCHES:

```
/D-SORT DATA IN DESCENDING ORDER.
   (ASCENDING IS OTHERWISE ASSUMED.)
/M-MERGE OPERATION IS REQUESTED
/N-NO LISTING OF STATISTICS
```

LOCAL SWITCHES:

```
/B-NAME OF FILE CONTAINING LOWER FIELD LIMIT
/D-DELETE INPUT FILE AFTER SORT COMPLETED.
   (OVERRIDEN IF NO OUTPUT FILE SPECIFIED)
/K-FILENAME OF SORTED KEYS
/L-LISTING FILENAME FOR SORTING STATISTICS
   (DEFAULT IS CONSOLE OUTPUT DEVICE)

/O-FILENAME OF SORTED DATA FILE (OUTPUT FILE)
/S-USER SPECIFIED COLLATING SEQUENCE FILENAME.
/R-RECORD SIZE SPECIFIER (IN DECIMAL)
/U-NAME OF FILE CONTAINING UPPER FIELD LIMIT
/W-USER SPECIFIED WORK OR TEMPOARARY FILES
   (UP TO SIX MAY BE SPECIFIED)
```

GENERAL OPERATING INSTRUCTIONS (Continued)

Descriptions of the other local switches are given in the following sections.

INPUT/OUTPUT CONTROL FIELD SPECIFICATIONS

The operation of the RDOS Sort/Merge Program is governed by Key Specifiers 1 through 8. As indicated, at least one Key Specifier must be present for any sort of merge operation. The other seven are optional. The format of a Key Specifier is as follows:

<starting byte number> . <field length>

For example, the following Key Specifiers can be used.

6.10 18.4 103.1

which means that there is a 10 byte key in position 6-15, a 4 byte key in position 18-21, and a single byte key at position 103.

The format of the records in the output file (if one is specified) is control by the Output Field 1 through 8 specifiers. All eight of the Output Field specifiers are optional.

If Output Field specifiers are not used, RDOS Sort/Merge defaults, and outputs the complete input record to the output file. The format of an Output Field specifier is as follows:

<starting byte number> : <field length>

For example, the following output field specifiers can be used.

1:30 50:64

which means that the output record would be 94 bytes in length made up of a 30 byte field from positions 1-30 of the input records and a 64 byte field from positions 50-113 of the input record.

INPUT/OUTPUT CONTROL FIELD SPECIFICATIONS (Continued)

A number of conditions must be remembered when specifying either Key Specifiers or Output Field Specifiers. They are:

Starting byte number must be less than or equal to the size of a record.

The field length must be in the range of 1 to 64.

The starting byte number plus the field length must not be more than the record size plus 1.

There must be at least one Key Specifier.

There may be a maximum of eight Key Specifiers.

There may be up to a maximum of eight Output Field Specifiers.

If Output Field Specifiers are not present, the complete input record is output.

The Key Specifiers or Output Fields may be separated, contiguous, or overlapping.

USER - SUPPLIED COLLATING SEQUENCE

The standard (or default) collating sequence is ascending ASCII. In ascending ASCII, the lower the byte numerically the more significant the byte; but this can be reversed using the global switch /D. The user may not want to use the straight numerical comparison because characters like +, -, !, etc. are more significant than numbers which are more significant than the alphabetic characters.

RDOS Sort/Merge provides the user the option of specifying the name of a file followed by a local switch/S to indicate that a different collating sequence should be used. The content of this file should be the characters ordered in increasing priority and terminated by a null character. This file need only contain the characters that the user is concerned with, since any characters not specified are used in the ASCII collating sequence

USER - SUPPLIED COLLATING SEQUENCE (Continued)

at a lesser priority than those specified by the user. For example, the user may desire that alphabetic characters are more significant than numeric characters which are more significant than a space or dollar sign and the rest of the characters can be sorted in the standard order. A file can be built as follows:

```
XFER/A  $TTI  COLLAT
ABC ...XYZ  0123456789  $
```

and this file name is then specified as the collating sequence as follows:

```
COLLAT/S
```

This means that records get ordered as follows:

```
AA
A1
A$
BA
B$
CA
CC
```

UPPER/LOWER CONTROL LIMIT SPECIFICATIONS

Frequently it is desirable to select only certain records from an input file. This is done by specifying upper and/or lower limits for the major Key Field. The lower limit is specified by placing the name of the file containing the lower limit followed by a local switch /B in the command line. The upper limit is specified by placing the name of the file containing the upper limit followed by a local switch /U in the command line.

The limit files should be as long as the major Key Field specified. The upper and lower limit file names specified in the command line may be the same if it is desired to extract only those records that exactly match the limit file contents. Limit files can be produced under program control or from the operator console.

DELETION OF RECORDS

As an additional feature, RDOS Sort/Merge allows the user to easily mark records for deletion from the file by setting the first word of the record to a zero. This means that a maintenance program that desires to delete records need only mark them for deletion, allowing the next operation of RDOS Sort/Merge on the file to remove the inactive record.

WORK FILE USAGE

During a sort operation, RDOS Sort/Merge uses up to six work files. By default, work files are named SORTW1.TP, SORTW2.TP, SORTW3.TP, etc. They are created in the current disk partition in which RDOS Sort/Merge is operating. The current disk partition is DP0.

To take advantage of other bulk storage peripherals in the system, the user can specify alternative work files. Because the level of activity on work files is not uniform, the user should consider work file activity when specifying alternative work files. The level of activity is:

Work File 1
Work File 4
Work File 2
Work File 5
Work File 3
Work File 6

Alternative work files must be specified in the order of their level of activity. For example, with the use of two magnetic tape drives and the disk, work files would be specified as:

```
MT0: O/W  WORK2/W  WORK3/W  MT1:O/W
```

If the system only contains a dual disk unit, alternative work files should be specified such that:

1. The first three work files are not on the same disk unit as the input file.

2. The last three work files are on the same disk unit as the input file.

WORK FILE USAGE (Continued)

When only a dual disk unit is in the system, alternative work files should be specified as follows (assuming that DP0 is the current partition):

```
INPUT DP1:W1/W DP1:W2/W DP1:W3/W WORK4/W +
WORK5/W WORK6/W
```

*****

APPENDIX A

ERROR MESSAGES

Two types of error messages are produced by RDOS Sort/Merge. They result from either a non-system error or a system error. The non-system error message is output to the listing file (if one has been specified) or to the console output device. It takes the following format:

GENERATOR - <error message>

where <error message> is one of the messages described in the following text.

The system error message is produced any time RDOS Sort/Merge is unable to complete a system action like creating, deleting, opening, closing, reading, or writing a file, loading an overlay, or extending the size of the user program space. This message has the following format:

<filename> - <error message>

where <filename> is the name of the last file or program segment reference by the program. It could be the name of one of the phases of RDOS Sort/Merge like

GENERATOR
PRESORT
LASTPASS
SORT/MERGE
MERGE

or the name of a file referenced by one of the phases of RDOS Sort/Merge like

RDOSSORT.OL
<INPUTFILE>
<OUTPUTFILE>
<COLLATE FILE>
<KEY FILE>

ERROR MESSAGES
(Continued)

UNIDENTIFIED LOCAL SWITCH

A local switch has been used that RDOS Sort/Merge cannot recognize during the generation phase.

TOO MANY KEY SPECIFIERS

More than eight input or output control fields have been specified in the command line.

INVALID KEY SPECIFIERS

The key (control field) for either input or output has been setup incorrectly, i.e. a non-numeric character, start position and length not separated by . or : .

KEY RANGE ERROR

The input or output Key Field specified did not exist completely within the record as indicated by the user or the default record (size 80 bytes).

NO KEY SPECIFIERS

At least one Key Field was not specified in the command line. One or more are required for operation in either sort or merge modes.

DUPLICATE SEQUENCE FILES

Two or more files containing alternate collating sequences have been specified in the command line.

DUPLICATE KEY FILES

Two or more files have been specified to contain the record number and control field data.

ERROR MESSAGES
(Continued)

DUPLICATE LISTING FILES

Two or more listing files have been specified in the command line.

INSUFFICIENT MEMORY FOR PRESORT

Enough memory must be present to sort at least ten records at one time. To proceed, the user could use a smaller control field.

INPUT RECORD TOO SMALL

Record length of zero or a length less than the record length needed to contain the specified control fields was specified in the command line.

ILLEGAL RECORD SIZE

Record length specified in the command line specified was larger than 512.

TOO MANY WORK FILES SPECIFIED

More than six work files have been specified in the command line.

NO OUTPUT FILE SPECIFIED FOR MERGE MODE

As the message indicates

NO INPUT FILE SPECIFIED FOR RDOSSORT

At least one input file is necessary for RDOS Sort/Merge to operate.

NO KEY OR OUTPUT FILE FOR RDOSORT

At least one of these files must be specified for RDOS Sort/Merge to perform useful work.

ERROR MESSAGES
(Continued)

DUPLICATE OUTPUT FILES

Attempt to specify more than one output file in the command line.

TOO MANY INPUT FILES SPECIFIED

Attempt to specify more than one input file for a SORT operation or more than six input files for the MERGE operation.

DUPLICATE LOWER LIMIT KEY FILE NAMES

Attempt to specify more than one lower limit file-name in the command line.

DUPLICATE UPPER LIMIT KEY FILE NAMES

Attempt to specify more than one upper limit file name in the command line.

*****

APPENDIX B

SAMPLE JOBS

This appendix illustrates the usage of the RDOS Sort/Merge Program by examples.

SAMPLE JOB 1

Produce an Ordered Item File.

Purpose

This example issustrates how RDOS Sort/Merge can be used to perform an ordering of a data file based on a given control key.

All records are selected from the transaction file and are used in creating the output records. The output records could then be used to produce a history of activity involving various stock item. Output records are sorted in ascending order by item number (ITEM field).

Input Record Format

Record Positions	Field Names	Contents
1	ID	Record Type
2-7	ITEM	Number of the stock item
8-12	QTY	Number of units ordered
•		
•		
•		
65-72	DATE	Date of Order
73-80	COST	Total Cost of Items Ordered

Output Record Format

The output record format is identical to the input format. The only difference is that the output file (TRANSACTION.NW) is a sorted (ascending order by ITEM) version of the input file TRANSACTION.DA.

Discussion

The resultant output file is a copy of the input file sorted by the six digit item number contained in byte positions 2 through 7 of the record. The file is sorted into ascending order (the default case) and the record size of 80 characters (also the default size) is used.

SAMPLE JOB NUMBER ONE
****

PURPOSE
****

PRODUCE AN ORDERED ITEM FILE

CLI COMMAND LINE
****

```
RDOSSORT SLPT/L TRANSACTION.DA TRANSACTION.NW/O 2,6
```

BATCH JCL STATEMENTS
****

```
!JOB  SAMPLE JOB # 1
!RDOSSORT SYSOUT/L TRANSACTION.DA TRANSACTION.NW/O 2,6
!EOF
```

SUMMARY SORT/MERGE PRINTOUT
****

```
RDOS SORT/MERGE PROGRAM -- SORT MODE     06:42:05  07/02/74

INPUT FILENAME(S)        :- TRANSACTION.DA

OUTPUT FILENAME          :- TRANSACTION.NW

RECORD SIZE (BYTES)      :-    80

COLLATING SEQUENCE       :-  ASCENDING ORDER
SORTED KEY FILENAME      :-  NONE SPECIFIED

INPUT  FIELD SPECIFIERS  :-  START BYTE/LENGTH(BYTES)
                                 2  -    6

OUTPUT FIELD SPECIFIERS  :-  START BYTE/LENGTH(BYTES)
                                 1  -   80

INPUT FILE RECORDS       :-   195
SORT IN RECORD COUNT     :-   195
SORT OUT RECORD COUNT    :-   195
```

SAMPLE JOB 2 — Produce a Record Address File for an Entire Inventory File.

Purpose

This example shows how the RDOS Sort/Merge Program can be used to create a record address file containing relative record numbers of all records in the inventory file. It sorts them into ascending order by class field. Within each class, addresses are also sorted into ascending order by item field .

Input Record Format

Record Positions	Field Names	Contents
1-6	ITEM	Number of Stock Item
7-10	CLASS	Class of Item
.		
.		
.		
106-111	ISSUES	Number of Units Issued during this Period

Output Record Format

Word Positions	Field Names	Contents
1	RECNO	Integer (16 bit) of relative record number
2-4	ITEM	Number of Stock Item
5-6	CLASS	Class of Item

Discussion

The output from this operation of RDOS Sort/Merge is a key file containing minimum key information in addition to the relative record number. The input record was 111 bytes in length. Rather than make a complete copy of the file sorted in ascending order based on a primary key (ITEM number) and the secondary key (CLASS of item), a key file was built for the inventory data base contained in INVENTORY.DA. The resultant key file (containing only key information and a relative record pointer to the logical record it represents) provides an indexed-sequential

Discussion
(Continued)

keyed access list. This list can be quickly and easily accessed by ITEM or CLASS to point to the desired record.

As the printout illustrates, five records were dropped during the sort operation and this accounts for the difference in the number of records found in the input file and the number finally included in the sort operation.

SAMPLE JOB NUMBER TWO
****

PURPOSE
****

PRODUCE A RECORD ADDRESS FILE FOR AN INVENTORY FILE

CLI COMMAND LINES
****

```
RDOSSORT INVENTORY.DA 111/R KEYINVENT.KF/K 1.6 7.4 SLPT/L
```

BATCH JCL STATEMENTS
****

```
!JOB  SAMPLE JOB # 2
!RDOSSORT INVENTORY.DA 111/R KEYINVENT.KF/K 1.6 7.4 SYSOUT/L
!EOF
```

SUMMARY SORT/MERGE PRINTOUT
****

```
RDOS SORT/MERGE PROGRAM -- SORT MODE     06:42:33  07/02/74

INPUT FILENAME(S)        :- INVENTORY.DA

OUTPUT FILENAME          :-  NONE SPECIFIED

RECORD SIZE (BYTES)      :-   111

COLLATING SEQUENCE       :-  ASCENDING ORDER
SORTED KEY FILENAME      :- KEYINVENT.KF

INPUT  FIELD SPECIFIERS  :-  START BYTE/LENGTH(BYTES)
                                1   -    6
                                7   -    4

INPUT FILE RECORDS       :-   15140
SORT IN RECORD COUNT     :-   15135
```

SAMPLE JOB 3

Produce a File of Inventory Information about Selected Items.

Purpose

This example illustrates how the RDOS Sort/Merge Program can select inventory records for stock items in classes 019-0356.

Output records are to contain only inventory information that indicates the activity (transactions) involving the stock items. The information is in the fields ITEM, CLASS, TRANS, and ISSUES.

Output records are to be sorted in ascending order by class (CLASS field), and in ascending order by number of transactions (TRANS field) within each class.

Input Record Format

Record Positions	Field Names	Contents
1-6	ITEM	Number of Stock Item
7-10	CLASS	Class of the Item
11-24	DESC	Description of the Item
.		
.		
.		
97-99	TRANS	Number of Transactions this Period
100-105	TDATE	Data of Last Transaction
106-111	ISSUES	Number of Units issued during this Period

Output Record Format

Record Positions	Field Names	Contents
1-4	CLASS	Contents
5-10	ITEM	Number of Stock Item
11-13	TRANS	Number of Transactions this Period

Output Record Format (Continued)

Record Positions	Field Names	Contents
14-19	ISSUES	Number of Units Issued during this Period

CLI And BATCH Command Lines

Before the RDOS Sort/Merge operation is initiated, the user must insure that the two "class" limit files exist on the disk. The lower limit file LCLASS should contain the ASCII characters 0195 and the upper limit file UCLASS should contain 0356.

These files can be created simply, through the CLI as follows:

```
XFER/A $TTI LCLASS )
0195 ↑Z
R

XFER/A $TTI UCLASS )
0356 ↑Z
R
```

Discussion

The output from this operation of RDOS Sort/Merge is an output file containing 3120 output records (each record is 19 characters in length). The program retained only those records with a class number in the range 0195 to 0356. The only thing to note about the summary printout on the next page is the difference between the number of INPUT FILE RECORDS and the SORT IN RECORD COUNT. This difference indicates that five records in the input file INVENTORY.DA has word one set to a zero to indicate the record was to be deleted.

SAMPLE JOB NUMBER THREE

PURPOSE

PRODUCE A FILE OF INVENTORY INFORMATION ABOUT SELECTED ITEMS

CLI COMMAND LINES

```
RDOSSORT INVENTORY.DA 111/R ACTIVITY/O 7.4 97.3 SLPT/L ^
1:6 7:4 97:3 106:6 LCLASS/B UCLASS/U
```

BATCH JCL STATEMENTS

```
!JOB  SAMPLE JOB # 3
!RDOSSORT INVENTORY.DA 111/R ACTIVITY/O 7.4 97.3 SYSOUT/L
# 1:6  7:4 97:3 106:6 LCLASS/B UCLASS/U
!EOF
```

SUMMARY SORT/MERGE PRINTOUT

```
RDOS SORT/MERGE PROGRAM -- SORT MODE      06:42:49  07/02/74

INPUT FILENAME(S)          := INVENTORY.DA

OUTPUT FILENAME            := ACTIVITY

RECORD SIZE (BYTES)        :=   111

COLLATING SEQUENCE         :=  ASCENDING ORDER
SORTED KEY FILENAME        :=  NONE SPECIFIED

LOWER LIMIT FILENAME       := LCLASS

UPPER LIMIT FILENAME       := UCLASS

INPUT  FIELD SPECIFIERS    :=  START BYTE/LENGTH(BYTES)
                                 7  =    4
                                97  =    3

OUTPUT FIELD SPECIFIERS    :=  START BYTE/LENGTH(BYTES)
                                 1  =    6
                                 7  =    4
                                97  =    3
                               106  =    6

INPUT FILE RECORDS         :=   15140
SORT IN RECORD COUNT       :=   15135
SORT OUT RECORD COUNT      :=    3120
```

SAMPLE JOB 4

Produce a Sorted File of Selected Members of the User's Group.

Purpose

This example illustrates the use of the RDOS Sort/Merge Program to select records from a file based on upper and lower limits and then to produce an output file ordered by zip-code/country.

Input Record Format

Record Positions	Field Names	Contents
1-30	NAME	Member's Name
31-60	TITLE	Member's Job Title
61-90	CMPNY	Company Affiliation
91-120	ADR	Company Address
121-140	CITY	City Name
141-150	ZIP	State/Zipcode or Country
151-180	CODES	Special Interest Group Codes

Output Record Format

Record Positions	Field Names	Contents
1-30	NAME	Data General Employee's Name
31-60	CMPNY	"Data General"
61-90	ADR	Local Branch Office Address
91-110	CITY	City
111-120	ZIP	State/Zipcode or Country

Limit And Collating Sequence Files

A non-standard ASCII collating sequence was desired. COLLAT was produced so that alphabetic characters would have higher priority over numeric characters. The file COLLAT was produced as follows on the next page.

Limit and Collating Sequence Files (Continued)

```
XFER/A  $TTI COLLATE )
ABCDEF...XYZ0123456789↑Z
```

Only members that worked for Data General were to be included and so a limit file was produced as follows:

```
XFER/A  $TTI DGLIMIT
DATA GENERAL↑Z
```

Discussion

This operation produced an output file DGSORT that included only Data General employees that was arranged in ascending order by state and zipcode or country.

SAMPLE JOB NUMBER FOUR
****

PURPOSE
****

PRODUCE A SORTED FILE OF SELECTED MEMBERS OF USER'S GROUP

CLI COMMAND LINES
****

```
RDOSSORT MEMBERSHIP.DA DGSORT/O COLLAT/S 61,12 141,10 ^
SLPT/L 180/R 1:30 61:60 121:30 DGLIMIT/B DGLIMIT/U
```

BATCH JCL STATEMENTS
****

```
!JOB  SAMPLE JOB # 4
!RDOSSORT MEMBERSHIP.DA DGSORT/O COLLAT/S 61,12 141,10
# SYSOUT/L
# 180/R 1:30 61:60 121:30 DGLIMIT/B DGLIMIT/U
!EOF
```

SUMMARY SORT/MERGE PRINTOUT
****

```
RDOS SORT/MERGE PROGRAM -- SORT MODE     06:43:07  07/02/74

INPUT FILENAME(S)        :- MEMBERSHIP.DA

OUTPUT FILENAME          :- DGSORT

RECORD SIZE (BYTES)      :-   180

COLLATING SEQUENCE       :-  USER SPECIFIED
SEQUENCE FILENAME        :- COLLAT
SORTED KEY FILENAME      :-  NONE SPECIFIED

LOWER LIMIT FILENAME     :- DGLIMIT

UPPER LIMIT FILENAME     :- DGLIMIT

INPUT  FIELD SPECIFIERS  :-  START BYTE/LENGTH(BYTES)
                               61  -   12
                              141  -   10

OUTPUT FIELD SPECIFIERS  :-  START BYTE/LENGTH(BYTES)
                                1  -   30
                               61  -   60
                              121  -   30

INPUT FILE RECORDS       :-    3458
SORT IN RECORD COUNT     :-    3458
SORT OUT RECORD COUNT    :-     366
```

SAMPLE JOB 5

Produce a New Membership File with New Member File Merged with Current Membership File.

Purpose

A file of new members (MEMBERSHIP.UD) is merged with the existing membership file (MEMBERSHIP.DA) to produce the updated membership file (MEMBERSHIP.NW).

Input Record Format

Record Positions	Field Names	Contents
1-30	NAME	Member's Name
31-60	TITLE	Member's Job Title
61-90	CMPNY	Company Affiliation
91-120	ADR	Company Address
121-140	CITY	City Name
141-150	ZIP	State/Zipcode or Country
151-180	CODES	Special Interest Group Codes

Output Record Format

Same as input record format.

Discussion

This operation or RDOS Sort/Merge showed how easily two sorted input files could be merged together to produce another sorted output file. The records were arranged in a sorted order based on an input control field that included characters 151 through 180.

```
        SAMPLE JOB NUMBER FIVE
        ======================

PURPOSE
=======

 PRODUCE A NEW MEMBERSHIP FILE WITH THE NEW MEMBERSHIP FILE
 MERGED WITH THE CURRENT MEMBERSHIP FILE


CLI COMMAND LINES
=================

 RDOSSORT/M MEMBERSHIP.DA MEMBERSHIP.UD MEMBERSHIP.NW/O ^
 $LPT/L 180/R 151,30


BATCH JCL STATEMENTS
====================

 !JOB  SAMPLE JOB # 5
 !RDOSSORT/M MEMBERSHIP.DA MEMBERSHIP.UD MEMBERSHIP.NW/O
 # SYSOUT/L 180/R 151,30
 !EOF


SUMMARY SORT/MERGE PRINTOUT
===========================


RDOS SORT/MERGE PROGRAM -- MERGE MODE      06:43:21  07/02/74

INPUT FILENAME(S)        :- MEMBERSHIP.DA, MEMBERSHIP.UD
OUTPUT FILENAME          :- MEMBERSHIP.NW

RECORD SIZE (BYTES)      :-   180

COLLATING SEQUENCE       :-  ASCENDING ORDER


INPUT  FIELD SPECIFIERS :-  START BYTE/LENGTH(BYTES)
                                151  -   30

OUTPUT FIELD SPECIFIERS :-  START BYTE/LENGTH(BYTES)
                                  1  -  180

INPUT RECORD COUNT       :- MEMBERSHIP.DA        3343
                            MEMBERSHIP.UD        86

MERGE IN RECORD COUNT    :-   3429
TOTAL RECORDS IN MERGE   :-   3429
```

SAMPLE JOB 6 File Reorganization.

Purpose

Periodically the contents and the organization of the records of a data file must be modified to delete unused data fields or to make provision for new fields.

Input Record Format

Record Positions	Field Names	Contents
1-6	ITEM	Number of the Stock Item
7-10	CLASS	Class of the Item
11-24	DESC	Description of the Item
.		
.		
.		
87-91	VEND1	First Vendor
92-96	VEND2	Second Vendor
97-102	TDATE	Date of Last Transaction

Output Record Format

Record Positions	Field Names	Contents
1-6	ITEM	Number of the Stock Item
7-10	CLASS	Class of the Item
11-24	DESC	Description of the Item
.		
.		
.		
87-91	VEND1	First Vendor
92-96	VEND2	Second Vendor
97-101	VEND3	Third Vendor
102-106	VEND4	Fourth Vendor
107-128	SPARE	Spare Field

SAMPLE JOB 6
(Continued)

Discussion

This operation of RDOS Sort/Merge shows how the MERGE mode can be utilized to rearrange the format of records within a file. In this example it was desired to eliminate the TDATE field and extend the file to allow two additional vendors to be specified and also to leave some space at the end of the record for later expansion. The output field specifiers were chosen to keep the input record (character position 1-96) intact in the output records. The two additional vendor fields were provided by copying the first two vendor fields again into the output record. The space field was produced by copying the item description into the output files. It might be noted that the record size was chosen as 128 bytes since this means that four complete records can be stored per sector reducing disk access requirements.

SAMPLE JOB NUMBER SIX

PURPOSE

REORGANIZE A FILE BY EXTENDING IT AND DELETING OTHER INFORMATION

CLI COMMAND LINES

```
INIT MT0
RDOSSORT/M OLDFILE MT0:0/O 102/R $LPT/L 1.6 1:64 65:32 ^
87:10 11:14 11:8
RELEASE MT0
```

BATCH JCL STATEMENTS

```
!JOB  SAMPLE JOB # 6
!MTA TAPE   SCRATCH TAPE
!RDOSSORT/M OLDFILE TAPE:0/O 102/R 1.6 1:64 65:32
# 87:10 11:14 11:8 SYSOUT/L
!RELEASE TAPE
!EOF
```

SUMMARY SORT/MERGE PRINTOUT

```
RDOS SORT/MERGE PROGRAM -- MERGE MODE     06:43:41  07/02/74

INPUT FILENAME(S)        := OLDFILE
OUTPUT FILENAME          := MT0:0

RECORD SIZE (BYTES)      :=   102

COLLATING SEQUENCE       :=  ASCENDING ORDER


INPUT  FIELD SPECIFIERS :=  START BYTE/LENGTH(BYTES)
                              1  -    6

OUTPUT FIELD SPECIFIERS :=  START BYTE/LENGTH(BYTES)
                              1  -   64
                             65  -   32
                             87  -   10
                             11  -   14
                             11  -    8

INPUT RECORD COUNT       := OLDFILE      5363

MERGE IN RECORD COUNT    :=   5363
TOTAL RECORDS IN MERGE   :=   5363
```

APPENDIX C

TIMING CONSIDERATIONS

The time it takes to run a sort job can vary greatly and is dependent on the following factors.

- The amount of main storage available for the sort program.
- The number of records to be sorted.
- Record size.
- Size of the control fields.
- Whether an alternate collating sequence is used.
- Location and number of input, work, and output files.

Here is how each factor affects the length of RDOS Sort/Merge job.

MAIN STORAGE SIZE

The more main memory storage the program has to use means the following:

- The more control fields that can be specified
- The longer input records can be
- The faster the sort job will be executed

RDOS Sort/Merge utilizes all of main storage that is available to the program.

NUMBER OF RECORDS TO BE SORTED

The more records you want to sort, the longer the sort job will take. For the most efficient sort job, therefore, use the record limit feature whenever possible so only the records you are interested in are included.

RECORD/CONTROL FIELD SIZE

The larger the keys, the longer it will take to sort a file because fewer keys can be sorted in main memory. So, if program runtime is important to you, do not include control fields that contain unnecessary information. Every field you do not include decreases the work record size by the length of that control field.

ALTERNATE COLLATING SEQUENCE

Use of an alternate collating sequence slightly increases the time it takes to run a sort job because a translation of the control fields is necessary.

LOCATION OF THE FILES

Because the RDOS Sort/Merge Program must move each record from the input file to the output file, file location is an important factor in how long it takes to run a sort job.

It is important, then, to place the high activity files such as the work files on a fast access head/track disk like the NOVADISC® * whenever possible. If multiple disk packs or disk cartridge systems are available, the work files should be distributed over these disk units. Magnetic tape units can be used for work files or the output file.

INSTALLATION REQUIREMENTS

The computer system must be operating under the Data General Real Time Disk Operating System (RDOS) and have at least 10K word user program partition available to the RDOS Sort/Merge Program.

Intermediate storage (also called work areas) is required for every sorting job. It may be assigned on the following devices:

Cassette Tape Units
Magnetic Tape Units
Fixed Head Disks
Moving Head Disks

*NOVADISC is a registered trademark of Data General Corporation, Southboro, Massachusetts.

APPENDIX D

SYSTEM PREPARATION GUIDELINES

This appendix details the steps to be followed to get an operational RDOS Sort/Merge program for the user RDOS environment (mapped and unmapped). RDOS Sort/Merge is supplied to licensed users in the form of an RDOS dump file containing relocatable binaries of the RDOS Sort/Merge mainline program and the overlays.

The RDOS dump file contains:

RDOSSORT.RB
SORTG.RB
SORTL.RB
SORTM.RB
SORTP.RB
SORTT.RB
MERGE.RB
SORTE.RB

After loading the dump tape on the disk, a relocatable load must be performed to link all these modules together into a save file and an overlay file.

The load command is:

```
RLDR RDOSSORT [SORTG, SORTP, SORTM, SORTL,↑
SORTE, SORTT, MERGE]
```

The current system library (SYS.LB) must be used since RDOS Sort/Merge is a multi-tasking program and the multi-tasking modules are different between the mapped and unmapped RDOS systems.

The user could build both a mapped and an unmapped version on the disk by executing the load commands on the following page.

```
RENAME RDOSSORT.SV URDOSSORT.SV
RENAME RDOSSORT.OL URDOSSORT.OL

RLDR/N RDOSSORT MRDOSSORT/S
[SORTG, SORTP, SORTM, SORTL, SORTE, SORTT,↑
MERGE] MSYS.LB
```

To operate the RDOS Sort/Merge program on a non-mapped machine, the user should link the files as follows:

```
LINK RDOSSORT.SV URDOSSORT.SV
LINK RDOSSORT.OL URDOSSORT.OL
```

To operate the RDOS Sort/Merge program on a mapped machine like the NOVA® 840, the user should link the files as follows:

```
LINK RDOSSORT.SV MRDOSSORT.SV
LINK RDOSSORT.OL MRDOSSORT.OL
```

DataGeneral

Software Documentation Remarks Form

How Do You Like This Manual?

Title _____ No. _____

We wrote the book for you, and naturally we had to make certain assumptions about who you are and how you would use it. Your comments will help us correct our assumptions and improve our manuals. Please take a few minutes to respond.

If you have any comments on the software itself, please contact your Data General representative. If you wish to order manuals, consult the Publications Catalog (012-330).

Who Are You?

- ☐ EDP Manager
- ☐ Senior System Analyst
- ☐ Analyst/Programmer
- ☐ Operator
- ☐ Other _____

What programming language(s) do you use? _____

_____

How Do You Use This Manual?

(List in order: 1 = Primary use)

- ____ Introduction to the product
- ____ Reference
- ____ Tutorial Text
- ____ Operating Guide
- ____ _____

Do You Like The Manual?

Yes	Somewhat	No	
☐	☐	☐	Is the manual easy to read?
☐	☐	☐	Is it easy to understand?
☐	☐	☐	Is the topic order easy to follow?
☐	☐	☐	Is the technical information accurate?
☐	☐	☐	Can you easily find what you want?
☐	☐	☐	Do the illustrations help you?
☐	☐	☐	Does the manual tell you everything you need to know?

Comments?

(Please note page number and paragraph where applicable.)

From:

Name _____ Title _____ Company _____

Address _____ Date _____

FOLD DOWN FIRST FOLD DOWN

FIRST
CLASS
PERMIT
No. 26
Southboro
Mass. 01772

BUSINESS REPLY MAIL

No Postage Necessary if Mailed in the United States

Postage will be paid by:

Data General Corporation

Southboro, Massachusetts 01772

ATTENTION: Software Documentation

FOLD UP SECOND FOLD UP

STAPLE