

# SORT/MERGE Utilities

User's Guide

DPS7000/XTA  
NOVASCAL 7000

File and Volume Management



REFERENCE  
47 A2 08UF 07



# DPS7000/XTA NOVASCALE 7000 SORT/MERGE Utilities User's Guide

File and Volume Management

**February 2005**

**BULL CEDOC  
357 AVENUE PATTON  
B.P.20845  
49008 ANGERS CEDEX 01  
FRANCE**

**REFERENCE  
47 A2 08UF 07**

The following copyright notice protects this book under Copyright laws which prohibit such actions as, but not limited to, copying, distributing, modifying, and making derivative works.

Copyright © Bull SAS 1997, 2005

Printed in France

Suggestions and criticisms concerning the form, content, and presentation of this book are invited. A form is provided at the end of this book for this purpose.

To order additional copies of this book or other Bull Technical Publications, you are invited to use the Ordering Form also provided at the end of this book.

### **Trademarks and Acknowledgements**

We acknowledge the right of proprietors of trademarks mentioned in this book.

Intel® and Itanium® are registered trademarks of Intel Corporation.

Windows® and Microsoft® software are registered trademarks of Microsoft Corporation.

UNIX® is a registered trademark in the United States of America and other countries licensed exclusively through the Open Group.

Linux® is a registered trademark of Linus Torvalds.

# Preface

## OBJECTIVES

This manual describes GCOS 7 Sort and Merge utilities.

It applies to all GCOS 7-V8 systems (from Technical Status TS8560).

## INTENDED READERSHIP

This manual is intended for all users of GCOS 7-V8 who need to sort or merge records in files of all organizations.

## STRUCTURE OF THIS MANUAL

Chapter 1	gives an introduction to the Sort and Merge utilities.
Chapter 2	describes the files and resources needed by Sort/Merge.
Chapter 3	describes the JCL statements SORT and MERGE and their parameters.
Chapter 4	describes the JCL statements GSORT and GMERGE and their parameters.
Chapter 5	presents the DSL (Data Services Language) commands for the Sort utility.
Chapter 6	presents the DSL (Data Services Language) commands for the Merge utility.
Chapter 7	gives additional information and practical advice on how to perform a Sort.

## Sort/Merge User's Guide

- Appendix A contains lists of error messages for Sort and Merge.
- Appendix B contains an example of a Sort report, produced in the SYSOUT file.
- Appendix C gives the collating sequences available with Sort.
- Appendix D gives the syntax of the GCL command SORT\_FILE which can be used to perform a sort (in interactive or batch mode).
- Appendix E gives the syntax of the GCL command MERGE\_FILE which can be used to perform a merge (in interactive or batch mode).
- Appendix F describes the necessary conditions for loading UFAS indexed output files by Sort and Merge.
- Appendix G describes the effect of the parameters DATAFORM (=SARF or =SSF) and TRUNCSSF/NTRNCSSF on files processed by SORT and MERGE.
- Appendix H indicates what restrictions apply to checkpoints with cartridge files for SORT and for MERGE.
- Appendix I discusses Sort/Merge and the UFAS Access Method.

## ASSOCIATED DOCUMENTS

### File Management

<i>UFAS-EXTENDED User's Guide</i> .....	47 A2 04UF
<i>Data Management Utilities User's Guide</i> .....	47 A2 26UF
<i>GAC-EXTENDED User's Guide</i> .....	47 A2 12UF

### Job Management

<i>IOF Terminal User's Reference Manual</i> .....	47 A2 38UJ
.....	47 A2 39UJ
.....	47 A2 40UJ
<i>JCL Reference Manual</i> .....	47 A2 11UJ
<i>JCL User's Guide</i> .....	47 A2 12UJ

### Error Messages

<i>Messages and Return Codes Directory</i> .....	47 A2 10UJ
--	------------

### Languages

<i>COBOL 85 Reference Manual</i> .....	47 A2 05UL
<i>COBOL 85 User's Guide</i> .....	47 A2 06UL

### Library Maintenance

<i>Library Maintenance Reference Manual</i> .....	47 A2 01UP
<i>Library Maintenance User's Guide</i> .....	47 A2 02UP

**SYNTAX NOTATION**

The following conventions are used for presenting JCL command syntax.

- ITEM** An item in upper case is a literal value, to be specified as shown. The upper case is merely a convention; in practice you can specify the item in upper or lower case.
- item** An item in lower case is non-literal, indicating that a user-supplied value is expected.

In most cases it gives the type and maximum length of the value:

```
dec5      a decimal integer value of up to 5 digits
hexa512   a string of up to 512 hexadecimal characters
char105   a string of up to 105 alphanumeric characters
name31    a name of up to 31 characters
file78    a file name of up to 78 characters
```

In some cases, it gives the format of the value:

```
a         a single alphabetic character
nnn       a 3-digit number
hh.mm     a time in hours and minutes
```

In other cases, it is simply descriptive of the value:

```
device-class
condition
any-characters
```

- ITEM An underlined item is a default value.
- bool** A boolean value which is either 1 or 0. A boolean parameter can be specified by its keyword alone, optionally prefixed by "N". Specifying the keyword alone always sets the value to 1. Prefixing the keyword with "N" always sets it to 0.
- { }
- [ ]
- ( )
- ...
- +=\$/
- - -
- Braces indicate a choice of values. Only one may be selected.
- Square brackets indicate that the enclosed item is optional. An item not enclosed in square brackets is mandatory.
- Parentheses indicate that a single value or a list of values can be specified. A list of values must be enclosed by parentheses, with each value separated by a comma or a space.
- Ellipses indicate that the item concerned can be specified more than once.
- These are literal characters to be specified as shown.
- All parameters below a dashed line do not appear in the help menus.



**Example 1:**

```
{
  { IMMED } }
{ WHEN={ [dd.mm.yy.] hh.mm } }
{ { +nnnn{W|D|H|M} } }
{ }
```

This means you can specify:

- Nothing at all (WHEN=IMMED applies).
- WHEN=IMMED (the same as nothing at all).
- WHEN=22.30 to specify a time (and today's date).
- WHEN=10.11.87.22.30 to specify a date and time.
- WHEN=+0002W to specify 2 weeks from now.
- WHEN=+0021D to specify 21 days from now.
- WHEN=+005H to specify 5 hours from now.
- WHEN=+0123M to specify 123 minutes from now.

**Example 2:**

```
[ PAGES=(dec4[-dec4] ...) ]
```

An optional parameter giving either a single value or a list of values enclosed in parentheses, with each value separated by a comma or space. Each value can consist of either a single number or a pair of numbers connected by a hyphen. For example:

```
PAGES=(2,4,10-25,33-36,78 83 90)
```

**Example 3:**

```
[ REPLACE ={ bool | 0 } ]
```

A boolean parameter whose default value is zero. You can specify:

- Nothing at all (REPLACE=0 applies)
- REPLACE=0 or simply NREPLACE
- REPLACE=1 or simply REPLACE

**REMARKS FORM**

Suggestions and criticisms concerning the form, content and purpose of this manual are invited. A Technical Publications Remarks Form is provided at the end of the manual for this purpose.

## Sort/Merge User's Guide

# Table of Contents

<b>1. Introduction.....</b>	<b>1-1</b>
<b>1.1 BASIC CONCEPTS.....</b>	<b>1-1</b>
<b>1.2 SORT/MERGE FUNCTIONALITIES .....</b>	<b>1-2</b>
<b>1.3 STAND ALONE AND SUBROUTINE SORT/MERGE.....</b>	<b>1-3</b>
1.3.1 Using Sort .....	1-3
1.3.2 Using Merge .....	1-3
<b>1.4 JCL STATEMENTS.....</b>	<b>1-4</b>
1.4.1 Activating Sort/Merge .....	1-4
1.4.2 JCL Compatibility .....	1-4
1.4.3 Extended JCL Syntax.....	1-4
<b>1.5 THE DATA SERVICES LANGUAGE (DSL) .....</b>	<b>1-5</b>
<b>1.6 SORT WORK FILE.....</b>	<b>1-5</b>
<b>1.7 SORT/MERGE REPORT .....</b>	<b>1-5</b>
<b>1.8 SORT/MERGE LEVELS.....</b>	<b>1-6</b>
1.8.1 Overview of Basic Sort/Merge.....	1-6
1.8.2 Overview of Extended Sort/Merge.....	1-6
<b>1.9 FUNCTIONS OF NEW SORT/MERGE .....</b>	<b>1-7</b>
1.9.1 Large Memory Sort.....	1-7
1.9.2 Parallel Sort.....	1-7
1.9.3 BPB (Read Ahead/Write Behind) .....	1-8
1.9.4 Fast Read/Write of Relative Files.....	1-9
1.9.5 Support of VOLSET .....	1-9
1.9.6 OUTFILE AS INFILE.....	1-9
<b>1.10 COBOL SORT AND MERGE.....</b>	<b>1-10</b>
<b>1.11 DUMPS IN CASE OF ABORT.....</b>	<b>1-10</b>

<b>2. Files and Resources .....</b>	<b>2-1</b>
<b>2.1 JCL STATEMENTS .....</b>	<b>2-1</b>
2.1.1 Sort.....	2-1
2.1.2 Merge .....	2-1
<b>2.2 SORT FILES AND RESOURCES .....</b>	<b>2-2</b>
<b>2.3 MERGE FILES AND RESOURCES .....</b>	<b>2-3</b>
<b>3. SORT and MERGE Statements .....</b>	<b>3-1</b>
<b>3.1 SORT STATEMENT .....</b>	<b>3-2</b>
3.1.1 Syntax .....	3-2
3.1.2 Explanation of Parameters .....	3-4
3.1.3 Parallel Sort Using SORT Statement.....	3-10
<b>3.2 MERGE STATEMENT .....</b>	<b>3-11</b>
3.2.1 Syntax .....	3-11
3.2.2 Explanation of Parameters .....	3-13
<b>3.3 FILE DEFINITION AND SPECIFICATION .....</b>	<b>3-14</b>
3.3.1 DEFINE Parameters for SORT and MERGE Statements .....	3-14
3.3.2 File Specification for SORT and MERGE Statements.....	3-18
3.3.2.1 Parameters.....	3-18
3.3.2.2 Examples.....	3-21
<b>3.4 EXAMPLES OF SORT AND MERGE STATEMENTS.....</b>	<b>3-22</b>
<b>4. GSORT and GMERGE Statements .....</b>	<b>4-1</b>
<b>4.1 GSORT STATEMENT .....</b>	<b>4-2</b>
4.1.1 Syntax .....	4-2
4.1.2 Explanation of Parameters .....	4-3
4.1.3 Implicit Values .....	4-10
<b>4.2 GMERGE STATEMENT .....</b>	<b>4-11</b>
4.1.1 Syntax.....	4-11
4.1.2 Explanation of Parameters .....	4-12
<b>5. DSL for Sort.....</b>	<b>5-1</b>
<b>5.1 COMMAND FILE ORGANIZATION .....</b>	<b>5-1</b>
<b>5.2 FUNCTION PARAGRAPH .....</b>	<b>5-2</b>
5.1.1 DESCEND Command .....	5-2
5.1.2 COLLATE Command.....	5-3
5.1.3 OUTPUT Command .....	5-5
5.1.4 DUPREC Command.....	5-6
5.1.5 DELETE Command.....	5-6

## Table of Contents

5.1.6	INTINFO Command.....	5-7
5.1.7	SORTSIZE Command (Optimization).....	5-8
5.1.8	MIN Command .....	5-8
5.1.9	AVERAGE Command .....	5-8
5.1.10	WRDISK Command .....	5-9
5.1.11	DEBUG Command .....	5-9
5.1.12	NBSORT Command.....	5-9
5.1.13	LOWYEAR Command.....	5-10
5.1.14	Examples of FUNCTION Paragraph.....	5-10
5.3	RECORD PARAGRAPH .....	5-11
5.1.1	KEY[S] Command.....	5-12
5.1.2	INCLUDE/OMIT Commands .....	5-15
5.1.3	Condition Elements .....	5-15
5.1.4	Example of INCLUDE/OMIT Commands .....	5-19
5.1.5	SUM Command .....	5-20
5.1.6	ARRANGE Command.....	5-22
5.1.7	Examples of RECORD Paragraph.....	5-25
5.4	END PARAGRAPH .....	5-26
6.	DSL for Merge .....	6-1
6.1	FUNCTION PARAGRAPH .....	6-2
6.1.1	DESCEND Command .....	6-2
6.1.2	DELETE Command.....	6-2
6.1.3	COLLATE Command .....	6-2
6.1.4	DEBUG Command .....	6-3
6.1.5	LOWYEAR Command.....	6-3
6.2	RECORD PARAGRAPH .....	6-4
6.3	END PARAGRAPH .....	6-7
7.	Sort Execution .....	7-1
7.1	SORTING VARIABLE LENGTH RECORDS .....	7-1
7.2	SORT RECORD SIZE .....	7-2
7.3	MAXIMUM RECORD SIZE .....	7-3
7.4	SIZE OF DISK WORK FILE .....	7-4
7.5	ESTIMATING MEMORY SIZE FOR A "MEMORY SORT" .....	7-8
7.6	NOTES ON PERFORMANCE.....	7-9
7.7	CHECKPOINT/RESTART .....	7-11
7.8	CONCURRENT ACCESS (GAC).....	7-12

<b>A. Sort and Merge Error Messages .....</b>	<b>A-1</b>
<b>A.1 ERRORS MESSAGES FOR THE SORT STATEMENT .....</b>	<b>A-2</b>
<b>A.2 ERROR MESSAGES FOR THE MERGE STATEMENT .....</b>	<b>A-6</b>
<b>A.3 EXECUTION ERROR MESSAGES FOR SORT/GSORT AND         MERGE/GMERGE.....</b>	<b>A-8</b>
<b>A.3.1 Invalid Records.....</b>	<b>A-8</b>
<b>A.3.2 List of Execution Error Messages .....</b>	<b>A-9</b>
<b>B. Sort/Merge Report.....</b>	<b>B-1</b>
<b>B.1 REPORT = ALL .....</b>	<b>B-1</b>
<b>B.2 REPORT = AUDIT (DEFAULT).....</b>	<b>B-2</b>
<b>B.3 REPORT = PARAM.....</b>	<b>B-3</b>
<b>B.4 REPORT = NONE.....</b>	<b>B-3</b>
<b>C. Collating Sequences .....</b>	<b>C-1</b>
<b>D. SORT_FILE Command (GCL) .....</b>	<b>D-1</b>
<b>D.1 EXECUTING SORT_FILE .....</b>	<b>D-1</b>
<b>D.2 SORT_FILE (SRTF, SORT) COMMAND.....</b>	<b>D-2</b>
<b>E. MERGE_FILE Command (GCL) .....</b>	<b>E-1</b>
<b>E.1 EXECUTING MERGE_FILE .....</b>	<b>E-1</b>
<b>E.2 MERGE_FILE (MRGF) COMMAND.....</b>	<b>E-2</b>
<b>F. UFAS Indexed Output File Loaded by Sort or Merge .....</b>	<b>F-1</b>
<b>G. Use of DATAFORM = {SARF/SSF} and of [TRUNCSSF/NTRNCSSF]     by SORT and MERGE.....</b>	<b>G-1</b>
<b>H. Restrictions to Checkpoint/Restart with Cartridge Files for Sort/Merge.....</b>	<b>H-1</b>
<b>I. Sort/Merge and the UFAS Access Method .....</b>	<b>I-1</b>
<b>I.1 SORT .....</b>	<b>I-1</b>
<b>I.2 MERGE .....</b>	<b>I-2</b>
<b>Index .....</b>	<b>j-1</b>

# Illustrations

## Figures

Figure 3-1. Format of the JCL Statement SORT .....	3-3
Figure 3-2. Format of the JCL Statement MERGE .....	3-12
Figure 3-3. File Definition for SORT and MERGE Statements .....	3-14
Figure 4-1. Format of the JCL Statement GSORT .....	4-2
Figure 4-2. Format of the JCL Statement GMERGE .....	4-11
Figure 5-1. Sample Output Formats .....	5-5
Figure 5-2. Example of KEY[S] Command .....	5-14
Figure 5-3. Using Comparison Operators .....	5-17
Figure 5-4. Using Numeric Condition Operators .....	5-18
Figure 5-5. Example of INCLUDE/OMIT Commands .....	5-19
Figure 5-6. Example of SUM Command .....	5-21
Figure 5-7. Example of the ARRANGE Command .....	5-23
Figure 6-1. Example of RECORD Paragraph .....	6-5

## Tables

Table 3-1. Definition Parameters for Disk Files .....	3-19
Table 3-2. Definition Parameters for Tape Files .....	3-20
Table 5-1. Data Types .....	5-13
Table 5-2. Example of KEY[S] Command .....	5-16
Table 5-3. Possible Data Types in RECORD Paragraph (Sort) .....	5-24
Table 6-1. Possible Data Types in RECORD Paragraph .....	6-6
Table 7-1. Disk Capacities .....	7-7
Table C-1. EBCDIC (DPS 7000) PLW Character Set .....	C-1
Table C-2. Series 100 Character Set .....	C-5
Table C-3. H2000 Collating Sequence for EBCDIC Files .....	C-7
Table C-4. ASCII Collating Sequence .....	C-8

## Sort/Merge User's Guide



# 1. Introduction

This manual describes the Sort and Merge utilities available to a GCOS 7 user.

## 1.1 BASIC CONCEPTS

### Sort

The Sort utility reads one or more input files and outputs the records in increasing or decreasing order of a "key" contained in each record.

### Merge

The Merge utility reads one or more input files which are already in increasing or decreasing order (of a "key" contained in each record) and produces a single output file containing all the input records in "key" order (the same order as in the input files).

### Features

These utilities incorporate facilities to minimize the task of job description and optimize the efficiency and flexibility of the sort or merge operations. The variety of optional parameters permit a user to tailor the sort or merge to the requirements of the application.

### JCL and DSL

The control language used to describe a sort or merge operation consists of Job Control Language (JCL) statements which describe the user's files and operating environment, and a set of commands written in the Data Services Language (DSL) that describe functions based on fields within the user's input records.

### Interactive Use

It is also possible to perform a sort or a merge interactively via the GCL commands SORT\_FILE and MERGE\_FILE which are described in Appendix D and in Appendix E, respectively. A batch file is still necessary for Data Service Language unless you specify COMFILE=::TN or DSLFILE=::TN to input such commands from your terminal.

## 1.2 SORT/MERGE FUNCTIONALITIES

The Sort utility provides the following features:

- Four output record formats
- Specification of up to 64 key fields per sort (up to 8 data types)
- Summation of values in up to 16 fields of identically keyed records
- Pre-sort record selection based on data values in the input record (up to 16 conditions can be specified)
- Field level record restructuring (up to 128 fields can be specified)
- Use of disk for work files
- Allocation of a temporary disk work file if none has been specified by the user
- Use of EBCDIC, Series 100, ASCII or Series 200/2000 collating sequence, or a privately defined collating sequence
- Sorting only part of a file
- Checkpoint/Restart facilities
- File concatenation on input (up to 9 files).

The Merge utility provides the following features:

- Merging of up to eight input files
- Specification of up to 64 fields per merge
- Summation of values in up to 16 fields of identically keyed records
- Pre-merge record selection based on input data values (up to 16 conditions can be specified)
- Field level record restructuring (up to 128 fields can be specified)
- Use of EBCDIC, Series 100, ASCII or Series 200/2000 collating sequences or a privately defined collating sequence.

## **1.3 STAND ALONE AND SUBROUTINE SORT/MERGE**

### **1.3.1 Using Sort**

There are two main ways of using sort:

- As a stand alone load module.  
It reads INFILE and writes the sorted output on OUTFILE. The statements involved are SORT (existing statement) and GSORT (new statement).
- As a "subroutine" called by a user program.  
The statements to define the workfile(s) of the subroutine sort are SORTWORK (existing statement) and GSORTWRK (new statement). For a discussion of these two statements, see the appropriate programming language manuals.

### **1.3.2 Using Merge**

Merge can be used only in stand alone execution mode. The statements involved are MERGE (existing statement) and GMERGE (new statement).

The COBOL verb MERGE executes in-line-code generated by the COBOL compiler. This in-line-code is independent of the GCOS 7 Merge utility.

## 1.4 JCL STATEMENTS

### 1.4.1 Activating Sort/Merge

Both the sort and merge processes are activated as job steps through extended JCL statements. The existing statements are SORT and MERGE. A sequence of parameters is associated with each of these statements. These are described in detail in Chapter 3. A new syntax and new functionalities are provided through 2 new statements: GSORT and GMERGE. These new statements are described in detail in Chapter 4.

The parameters specify the files involved (input, output, work), they describe the location of the Data Services Language commands and select general job control processing options.

To use the new Sort in stand alone, you can use the SORT statement (see Chapter 3) or the GSORT statement (see Chapter 4). To use the new Merge facility, you use the MERGE statement (see Chapter 3) or the GMERGE statement (see Chapter 4). To describe the workfile(s) of a subroutine Sort, you use the SORTWRK or GSORTWRK extended JCL statement. These statements are described in the appropriate programming language manuals (for example, the *COBOL User Guide*, and the *GPL System Primitives*).

### 1.4.2 JCL Compatibility

The statements: SORT, MERGE, SORTWORK remain valid on GCOS 7.

**You can keep the current Sort/Merge JCL statements and use the utilities (which improve performance).** You may also use the new statements and access all the new functions. These functions are listed in paragraph 1.9 and the corresponding interfaces are described in detail later in this manual.

### 1.4.3 Extended JCL Syntax

However, if you want to use all the new Sort/Merge capabilities, you must use the new statements GSORT and GMERGE (described in Chapter 4), and GSORTWRK to describe the workfile(s) to be used by a subroutine sort. GSORTWRK is described in the appropriate programming language manual (for example, the *COBOL User Guide*, and the *GPL System Primitives*).

## 1.5 THE DATA SERVICES LANGUAGE (DSL)

A record level command file, written in DSL, must be provided as an input enclosure or as a previously established member of a source library. Commands in this file are described in Chapter 5 for the Sort utility and Chapter 6 for the Merge utility. All but one of these commands are optional. The mandatory command, KEYS, establishes the key fields for the sort or merge process.

The keys consist of one or more fields in the input record.

For example, a set of records are to be sorted according to an employee code contained in bytes 1 to 4 of each record, and, for each code, they are to be sorted in order of age (contained in bytes 8 and 9).

Set bytes 1 to 4 as the major key and bytes 8 and 9 as the minor key.

Sort sorts the input records in sequence according to employee code, and for each code, in sequence according to age.

Merge uses the major key (employee code) to merge records. It uses also the minor key (age) when some records have the same major key value.

When two records have the same employee code, they are compared on the value of the age field.

## 1.6 SORT WORK FILE

When the input file is small enough to fit in memory, the sort may only require main memory. Otherwise a work file is needed. This file may be preallocated by the user and may be permanent or temporary. If the file is not preallocated by the user, Sort will allocate a temporary file on the implicit VOLSET (if it exists) or on resident disk volume(s). The size of this file can be specified by the user or can be calculated by Sort.

The new Sort may use one workfile (mono-process sort) or several workfiles (multi-process sort). In the case of several workfiles, there will be simultaneous I/Os on the workfiles and possibly a reduced elapsed time.

## 1.7 SORT/MERGE REPORT

Error messages are printed in the Job Occurrence Report as a result of errors detected in the SORT/GSORT and MERGE/GMERGE statements. Error messages are also printed in the Sort/Merge Report and/or in the JOR (Job Occurrence Report) as the result of errors detected at execution time. Both types of error message are discussed in Appendix A.

## 1.8 SORT/MERGE LEVELS

The Sort/Merge products are available at 2 levels:

- **Basic Sort/Merge.**
- **Extended Sort/Merge.**

Each of these is discussed below.

### 1.8.1 Overview of Basic Sort/Merge

The Basic Sort is supplied as part of the GCOS 7 base (that is, no specific MI is needed).

The Basic Sort/Merge offers the following:

- Mono process Sort/Merge,
- Implicit and maximum memory=512K,
- BPB, if available in the access method concerned but limited to BPB-buffers of 64K bytes.

The Basic Sort/Merge offers improved performance. However, it does not offer the extra features which are described below for the Extended Sort.

### 1.8.2 Overview of Extended Sort/Merge

A separate MI is needed (that is, the Extended Sort/Merge has to be bought). This product is not supplied as part of the GCOS 7 base.

The Extended Sort/Merge offers:

- Mono-process or multi-process Sort.  
In mono-process activation, the Extended Sort uses the fast INFILE/OUTFILE read/write.
- Mono-process Merge.  
It uses the fast INFILE/OUTFILE read/write.
- Implicit memory 512K bytes per process.
- Maximum Sort memory = 64 Megabytes for mono-process.
- Maximum Sort memory = 1 Gigabytes for multi-process.
- A special BPB (specific to Sort/Merge) is called for UFAS files without conditions when the UFAS access method is not used (BPB-buffers not limited to 64K bytes).

- BPB of the access method, if an access method is called.
- support of the VOLSET facility.

## 1.9 FUNCTIONS OF NEW SORT/MERGE

The functions of the new Sort/Merge product are discussed in more detail below.

### 1.9.1 Large Memory Sort

For a mono-process sort (implicit case) the Basic Sort is limited to 512K bytes of memory. The Extended Sort is limited to 64 Megabytes in mono-process. But you must note that for a given file size there is a memory size beyond which the performance does not increase. The reason is that even the fastest sort still takes the time necessary to read the input file and then to write the sorted output file. For example, a 1 Gbyte file needs about 4Mbyte of memory to be sorted in a minimum time.

A small memory size (about 100K bytes) does not meet the above condition and also reduces the block size used by the sort to read/write the workfile and the possibility to use read ahead/write behind (also called BPB) on input and output files.

So it is very important to use a sort memory size between 512K and 4Mbytes. Larger memory sizes are useful only to possibly reduce the global number of I/Os of the job flow ("memory sort") or for huge files. Refer to Chapter 7 to see how to estimate the memory size needed for a "memory sort".

### 1.9.2 Parallel Sort

A parallel sort is also known as a **multi-process** sort.

In multi-process activation, the Extended Sort is able to use about 1 Gigabyte of core memory. In addition, it is able to use several CPUs and disks in parallel, and a fast read/write for the INFILE and OUTFILE. Thus the elapsed time will decrease by using more hardware/software resources.

**The parallel capabilities of the new Sort/Merge product are useful even on a mono processor DPS 7000 (for example, as the TDS capabilities are).**

When the input and output files are read/written with BPB (that is, read ahead/write behind), the sort may become "I/O bound" by the workfile. In such a case, (for example, for a large file), the sort can be done as several parallel sub-sorts each one using a private workfile on a private disk. Then workfiles I/Os can be done on each disk at the same time. Then the parallel sort can increase the performance to reach the minimum time. The minimum time is the time to read the input file with BPB and then write the output file with BPB without being slowed down by the workfile(s) I/Os.

A parallelism of workfile I/Os implies several disks, one disk per slave-sort.

The new JCL statement GSORT allows you to assign several workfiles (either static or dynamic allocation). The new JCL introduces the possibility for a parallel sort to define either:

- a list of *n* media for the dynamic allocation of *n* temporary workfiles (WKFILE, WKALC),
- or to allocate *n* workfiles by *n* ASSIGN statements WKASG1, ... WKASG*n*.

Note that you must specify the memory size for a parallel sort according to the number of processes. The minimum memory is 512K bytes per process. The recommended number of processes is between two and four. A larger number of processes is reserved for cases where a very large memory is to be used (up to 32 Mb \* number of processes).

### 1.9.3 BPB (Read Ahead/Write Behind)

GCOS 7-V6 (TS6150) introduced the UFAS Batch Booster or BPB facility. The Batch Booster is a facility to improve I/O performance on UFAS files. The Batch Booster is also known as the BPB (blocks per buffer) option since it is requested via the BPB parameter. This facility is described in the manual *UFAS-EXTENDED User's Guide*.

The BPB parameter exists in the JCL statement DEFINE (in the case of a sort called from a program STEP). BPB is available in the INFILE/OUTFILE parameter groups of the SORT/MERGE statements and in the INDEF/OUTDEF parameter groups of the GSORT/GMERGE statements.

When the Sort/Merge calls an access method to read/write the input/output user files, the BPB is activated as defined by the access method concerned. For performance purposes, Sort/Merge does not call the access method if this is not mandatory.

For UFAS user files, when Sort does not call UFAS:

- The Basic Sort activates the BPB function only if the UFAS BPB product is purchased. In this case, the BPB is limited to 64K BPB buffers as under UFAS (less if the sort memory is under 512K).
- The Extended Sort activates the BPB function even if the UFAS BPB product has not been purchased. In this case, the BPB is limited to (memory size)/8 and not 64K bytes. The BPB factor cannot exceed 32.

In a multi-process sort where the Sort specific BPB function is being used, the number of blocks per buffer is a multiple of the number of sort processes (BPB = *n*\*NBSORT).

Merge activates BPB under the same conditions as Sort does, except that the BPB limit of (memory size)/8 does not apply.

When using the BPB facility, you may specify a value of BPB between 1 and 32, and you must allow for the memory needed for the extra buffers (requested by BPB) in calculating the memory allocated to the sort (SIZE option at step level).



Appendix I defines the conditions under which Sort/Merge calls the UFAS access method to read a UFAS input file or write a UFAS output file.

#### **1.9.4 Fast Read/Write of Relative Files**

When UFAS Relative UFAS files are read/written in sequential access mode, the new Sort/Merge product does not call UFAS as did the previous product. Thus, the CPU time may be significantly lower for the new Sort/Merge product.

#### **1.9.5 Support of VOLSET**

For the new Sort/Merge product, a VOLSET name always represents a list of MEDIA. It will be used exactly as an explicit list of media.

When a VOLSET is used in a workfile assignment of a Parallel Sort, then only 10 media of the VOLSET are used. These 10 media are those which have the most free space (at the moment that the media are selected).

#### **1.9.6 OUTFILE AS INFILE**

If you want a dynamic allocation of the output file with the same characteristics as the input file, you may use OUTALC without any allocation parameters. If you specify OUTALC = (...), the missing characteristics will be copied from the INFILE ones. This applies to both the GSORT and GMERGE statements (see Chapter 4).

## 1.10 COBOL SORT AND MERGE

COBOL programs can perform sorts and merges using the COBOL statements SORT and MERGE. The COBOL statements SORT and MERGE are defined in the *COBOL 85 Language Reference Manual* and the *COBOL 85 User's Guide*.

The *COBOL 85 User's Guide* provides the following information:

- a comparison between the COBOL statements SORT/MERGE and the SORT/MERGE utilities;
- the use of the JCL statement SORTWORK or GSORTWRK when executing programs containing a COBOL statement SORT;
- the use of the keywords AVERAGE, MIN, SIZE and SORTSIZE in the OPTIONS parameter of the JCL statement STEP;
- an example of a COBOL SORT.

The *COBOL 85 User's Guide* also describes the use of the ACTUAL KEY phrase in conjunction with the OUTPUT Data Service Language (DSL) command of the Sort utility.

## 1.11 DUMPS IN CASE OF ABORT

Since the sort memory may be huge, the implicit dump will contain only the information of the processes and segments related to the cause of the abort. The implicit DUMP is the binary dump which is output on the binary dump library (of your configuration).

If you want a total dump, you must specify DUMP=ALL in the step options. However, beware that if a large step memory has been requested (via the SIZE parameter), then DUMP = ALL will result in a very large dump which may overflow your binary dump library and SYSOUT file.

## 2. Files and Resources

### 2.1 JCL STATEMENTS

#### 2.1.1 Sort

There are two JCL statements available for sorting: SORT and GSORT.

- SORT is the existing JCL statement.
- GSORT is a new JCL statement.

Both statements "execute" the new Sort product. However, if you want to use all the features of the new Sort product, you should use the GSORT statement (instead of the SORT statement). The SORT statement is described in Chapter 3. The GSORT statement is described in Chapter 4.

#### 2.1.2 Merge

There are two JCL statements available for merging: MERGE and GMERGE.

- MERGE is the existing JCL statement.
- GMERGE is a new JCL statement.

Both statements "execute" the new Merge product. However, if you want to use all the features of the new Merge product, you should use the GMERGE statement (instead of the MERGE statement). The MERGE statement is described in Chapter 3. The GMERGE statement is described in Chapter 4.

## 2.2 SORT FILES AND RESOURCES

1. Input and output files can be UFAS or Queued, independently of each other.
2. Input and output files can be on disk, tape, or cartridge independently of each other. For example, an input disk file may produce an output disk or tape file, an input tape file may produce an output disk or tape file.
3. Similarly, input and output files can be temporary or permanent independently of each other.
4. Input and output files can have any UFAS organization or be Queued. Input and output files need not have the same organization as each other. See Appendix F for details on the loading of UFAS indexed files as Sort or Merge output files.
5. The output may be written to the input file provided that input file concatenation is not used. See the descriptions of the SORT and GSORT statements.
6. Options are available to reduce the device requirements of Sort:  
  
A single device can be reserved for the input file and the output file, provided, of course, the volumes supporting them are of the same device class (REPLIN).  
  
This option, however, saves device drive(s) at the cost of increasing elapsed time, due to volume mounting and dismounting during the sort.
7. Output records are padded or truncated depending on the size of the output record compared with the input (or altered) record.
8. If a workfile is specified, temporary or not, it will be used by the Sort only if the Sort operation cannot be completed in memory.
9. If no workfile is specified, implicit workfile(s) will be used if needed. At run time, if the sort can be completed in main memory no work file is created. Otherwise a temporary work file is created dynamically.
10. If the DSL command SORTSIZE is used, Sort will check whether there is sufficient space in memory for the sort to be completed. If the memory is not sufficient, Sort checks if the size of the workfile is sufficient when it opens the workfile.

## **2.3 MERGE FILES AND RESOURCES**

1. The file specified as INFILE1 is known as the primary file. Other input files, INFILE2 through INFILE8, must have the same RECSIZE and RECFORM as the primary file.
2. Input and output files may be UFAS or Queued, independently of each other.
3. Any combination of UFAS and Queued files may be merged.
4. Input and output files can have any organization supported by UFAS or Queued. Input and output files need not have the same organization as each other. See Appendix F for precisions on the loading of UFAS indexed files as Sort or Merge output files.

## Sort/Merge User's Guide

## 3. SORT and MERGE Statements

This chapter describes the JCL statements SORT (for sorting) and MERGE (for merging).

SORT and MERGE each require that the input, output and command files be specified.

All other parameters of the JCL statements are optional.

Default values exist, which, if they are appropriate to the application, allow the specification of certain parameters to be omitted. The optional parameters concern work files (Sort only), reporting, performance optimization, error record processing, part-file processing, checkpointing, and abnormal termination.

## 3.1 SORT STATEMENT

### 3.1.1 Syntax

The syntax of the JCL Statement SORT is shown in Figure 3-1.

The following abbreviations are used in this figure:

sifd:	sequential-input-file-description
sofd:	sequential-output-file-description
smdp:	sort/merge-define-parameters

The symbolic names sifd and sofd refer to standard file descriptions which are to be found in the *JCL Reference Manual*.

The following parameters are not permitted with the SORT/MERGE statements:

- BYPASS
- CATNOW
- VOLWR/NVOLWR
- OPTIONAL
- DEFER
- SITE

Note that the above parameters (with the exception of SITE) are available within the assignment/define parameter groups of the GSORT/GMERGE statements.

The POOL option is not used with Merge.

POOL can be used with Sort. NPOOL can also be used to prevent the POOL option from being applied when dealing with concatenated input files.

The sort/merge define parameters are explained below under "Sort/Merge Define Parameters".

All parameters not explained below are standard parameters of the ASSIGN JCL statement. The parameter groups sequential-input-file-description, sequential-output-file-description, and print-file-description are not expanded in Figure 3-1.

See the *JCL Reference Manual* for all information concerning these standard parameter groups. However, a few of the parameters in these groups, because they are specific to the SORT JCL statement, are summarized below.





### 3.1.2 Explanation of Parameters

**INFILE** This specifies the input file. This parameter, together with INFILE<sub>n</sub>, is used by SORT to generate the ASSIGN and DEFINE JCL statements necessary for SORT execution.

**INFILE<sub>n</sub>** The keywords INFILE1 through INFILE8 may be used to specify files that must be concatenated with INFILE. In this case, all input files, including the file specified in INFILE, must have the same record format and record length and be organized in the same way (in sequence).

If there is disk file concatenation the NPOOL option should be applied to all files in the step. (See the POOL and NPOOL options in the WKDISK paragraph below.)

The INFILE<sub>n</sub> parameters must appear immediately after the INFILE parameter and the values used for n must start at 1 and must be in sequence

For example:

INFILE=(...),INFILE1=(...),INFILE2=(...) is correct.

INFILE=(...),INFILE2=(...),INFILE3=(...) is not correct.

**OUTFILE** This specifies the output file.

REPLIN is used to indicate to SORT that the volume(s) supporting the output file need not be mounted before the sort starts, and may use the device supporting the input file volume(s). This parameter may be used to minimize the device requirements of a sort. In the case of multi-volume input and output files, the MOUNT parameter of the sequential-output-file-description must be set equal to 1 for both input and output file.

INFILE: OUTFILE=INFILE allows the output from SORT to overwrite the input file. It must not be specified when there is file concatenation on input or when SHARE=MONITOR is specified. When OUTFILE=INFILE, the file to be sorted and overwritten will be assigned with ACCESS=SPWRITE unless an explicit access value is given in the INFILE parameter group. This default access value implies exclusive access to the file, but it is the only safe way to use OUTFILE=INFILE.

SYSOUT specifies that the output file is to be written as a standard SYSOUT subfile.

SYS.OUT must be used for the external file name if the SYSOUT parameter is used.

## SORT and MERGE Statements

DATAFORM=SSF can be used for assigning SYS.OUT, as well as INFILE or OUTFILE. In OUTFILE it involves the creation of SSF headers at the beginning of the record. If an input file is declared as SSF, the default processing option is TRUNCSSF for the sort. NTRNCSSF should be used with caution.

The DATAFORM parameter is used to indicate the DATAFORM both on input and on output. Thus the records which are read and written will correspond with the specified DATAFORM. Records which are internal to the sort have no header.

For more details on the use of DATAFORM, refer to Appendix G of this manual.

### COMFILE

This specifies the location of the Data Service Language commands. The language type of the specified DSL command file may be either SSF or SARF. The command file must either be a sequentially readable file or a library member.

### WKDISK [S]

This specifies a disk work file which can be either permanent, in which case it must have been preallocated, or temporary, in which case the space specified in the SIZE parameter will be allocated when the file is opened. The file will be formatted by SORT as needed.

**If the work file is to occupy more than one volume, the devices must all have the same attributes.**

If the WKDISK parameter is not specified, SORT will be executed without a work file being preallocated, in which case a temporary work file on the resident disk will be allocated at run time (unless the sort can be completed in main memory). The SORTSIZE DSL command (see Chapter 5) can be used to declare the number of input records to the sort (if known) so that the file created will be of that size. Otherwise, its size will be 10 cylinders\*.

The preallocation of a permanent file allows the definition of an increment size which will be used by the sort if necessary. For a dynamically created file or if WKDISK has been specified with SIZE option, the increment size is defined by the sort (1/16 of the size or a minimum of 2 cylinders\*).

\* cylinders:

- For VBO organization = physical cylinders. The actual size depends on the disk organization.

- For FBO organization = logical cylinders (1000 KBytes). The actual size does not depend on the disk organization (a disk file that is dynamically assigned is always 10 000 KBytes).

### **IMPORTANT**

When SORT work file preallocation is activated before SORT execution, care must be taken with file definition parameters. Only FILEFORM=UFAS, FILEORG=SEQ) is accepted by SORT on FBO volumes. On VBO volumes, (FILEFORM=BFAS or UFAS, FILEORG=SEQ) is recommended. Other values for FILEFORM and FILEORG will cause a warning message to be issued by SORT. The recommended UFAS CFSIZE or BFAS SEQ BLKSIZE for a preallocated SORT work file is 4096 bytes (with any RECFORM and RECSIZE compatible with that value). However, other BFAS SEQ VBO BLKSIZE or UFAS SEQ CFSIZE values are possible together with suitable SORT memory size values. (See the SIZE parameter for the total amount of memory to be allocated to SORT)

In a multi-programming environment, the use of a resident volume for the work file can cause a considerable increase in the read/write head movement on that volume. Therefore, it is preferable to preallocate a permanent file on a disk other than that used for input and output files, and, if possible, on a disk not being used currently by another job.

External-file-name. The rules for external-file-name are given in the *JCL Reference Manual*.

SIZE (with WKDISK) indicates that the file is temporary, and gives its size in cylinders (dec5). SIZE=0 allows the sort to be ordered in main memory. For more information, see Chapter 7.

POOL and NPOOL. The POOL, FIRST and POOL, NEXT options are explained in the *JCL Reference Manual*. NPOOL specifies that each time a POOL option for assigning a file is met, it is ignored.

During the translation of the SORT statement (Figure 3-1) into basic JCL, POOL, FIRST and POOL, NEXT are automatically generated when there is input file concatenation.

By specifying POOL or NPOOL for each SORT INFILE (and possibly for SORT OUTFILE or SORT WKDISK), the user can correct this automatic generation, if necessary.

**When concatenated input files are used, POOL must be used for tape files and NPOOL for disk files (including workfile disks).**

#### **Mono-process Sort**

You must use either no workfile or only one workfile and then define it via WKDISK[S].

#### **Multi-process (Parallel) Sort**

You can use the WKDISK[S] keyword to introduce a list of MEDIA. Such a list is limited by GCOS 7 to 10 MEDIA. One TEMPARY workfile will be dynamically allocated for each sub-sort, numbered from 1 to n (where NBSORT = n is given in the DSL for a multi-process sort), using the media list in a circular way.

For example, if MEDIA = (A,B,C) and NBSORT = 3, then the 1st workfile is allocated on A, B, C, the 2nd workfile on B, C, A, and the 3rd workfile on C, A, B.

If the MEDIA list is shorter than n MEDIA (NBSORT = n), then n is reduced according to the exact number of MEDIA in the list.

FILESTAT = UNCAT means the file is permanent and uncataloged.

FILESTAT= CAT means the file is permanent and cataloged.

CATALOG is used to override the normal catalog search rules when several private catalogs have been specified in an ATTACH JCL statement. Normally a file is searched for in all the attached catalogs, starting from the first one specified in the ATTACH statement. However, when the CATALOG parameter is used, only the nth catalog specified in the ATTACH statement is searched, where dec1=n.

RESIDENT specifies that the file is stored on one or more disk volumes that are resident.

DEVCLASS specifies the device class for the device associated with an uncataloged file. For example, MS/M500. DEVCLASS should not be specified if FILESTAT=CAT.

MEDIA specifies a list of volume=names. WORK signifies a work volume. MEDIA should not be specified if FILESTAT=CAT. In a list of volume-names specified by MEDIA, it is not possible to mix VBO and FBO disks.

REPORT	<p>This specifies the type of report required from SORT (see Appendix B). Unless PRTFILE is specified, the report is written to the SYSOUT file.</p> <p>AUDIT (default value in batch mode) causes diagnostic and process error information to be printed as well as a statistical summary of record processing.</p> <p>PARAM causes a list of parameters to be printed, in addition to diagnostic and process error information.</p> <p>ALL combines AUDIT and PARAM.</p> <p>NONE limits the report to process information.</p>
PRTFILE	<p>This specifies a permanent SYSOUT file. The file must be a native one with LABEL=NATIVE or LABEL=NONE.</p>
SIZE	<p>This specifies the total memory in units of 1024 bytes to be allocated to SORT for the execution. This includes code, data and buffers.</p> <p>For a mono-process sort, the SIZE is limited to 64 Mbytes.</p> <p>For a parallel sort, the limit is NBSORT*32 Mbytes. The maximum limit is therefore 32*32 Mbytes, though this will depend on the machine configuration. If SIZE is less than NBSORT*32 Mbytes, then each slave sort will use (approximately) SIZE/NBSORT Mbytes.</p> <p><b>The default value is 512 Kbytes per process. To use the Large Memory Sort with a SIZE &gt; 512 via the JCL statement SORT, you must use SIZEOPT instead of SIZE.</b> See the description of the SIZEOPT parameter below.</p> <p>You are recommended to always specify SIZE explicitly as this ensures that the memory used by the Sort is equal to the memory declared to GCOS 7 (the Declared Working Set, DWS).</p>
INVREC	<p>This specifies the action to be taken by SORT if an invalid record is encountered in the input file; that is, a record whose length is smaller than the minimum needed to apply the INCLUDE/OMIT clauses, or the KEY and/or SUM and/or ARRANGE clauses of the DSL commands.</p> <ul style="list-style-type: none"><li>- ABORT (default value) specifies that SORT should terminate when an invalid record is found.</li><li>- CONTINUE specifies that SORT should continue.</li><li>- ERROPT indicates what is to be done with an invalid record.</li></ul>

## SORT and MERGE Statements

PRINTID indicates that the invalid record's number relative to the start of the sort is to be printed in the report. This is the default value. PRINT has the same meaning as PRINTID but the entire invalid record will be printed in hexadecimal and alphanumeric formats. LOG indicates that the invalid record is to be written to the file specified in LOGFILE. IGNORE indicates that no message is to be printed.

LOGFILE	This contains a description of a native output file in which the invalid records are to be logged.
START	This gives the record number with which sorting is to begin. Values from 1 to 2147483647 are acceptable. The default value is 1.
HALT	This gives the record number with which sorting is to end. The same values are acceptable as for START. The default is end-of-file.
REPEAT	This specifies that checkpoints are to be taken during the sort. The default is that no checkpoints are taken.
SIZEOPT	The SIZEOPT parameter can be used to specify parameters used in the JCL statement SIZE (see the <i>JCL Reference Manual</i> ). To use the Large Memory Sort with a SIZE > 512, you must specify the size via SIZEOPT. In this case, SIZEOPT can give a value up to 65536 (that is, 64 Megabytes).
STEPOPT	The STEPOPT parameter can be used to specify the parameters included in the JCL statement STEP (see the <i>JCL Reference Manual</i> ); in particular: DUMP, DEBUG.

Note that the REPEAT parameter inside STEPOPT has the same effect as the REPEAT keyword.

### 3.1.3 Parallel Sort Using SORT Statement

The FUNCTION paragraph of the Sort DSL supports the new parameter NBSORT (where NBSORT = digit2 = number of slave processes of a parallel sort). The maximum value is 32. This is useful only for parallel sorts launched under the current JCL statement SORT whose syntax does not accept NBSORT as a JCL parameter. For a parallel sort launched by the JCL statement SORT, the WKDISK assignment provides the media list as the GSORT WKFILE assignment does.

If you want to run a parallel sort via the JCL statement SORT, you must also specify:

```
... , STEPOPT=(H_PSORT), ...
```

**Example:**

To request a parallel sort via the JCL statement SORT:

```
$JOB PSORT, HOLDOUT;
SORT IF= ..., OF= ..., COMFILE = *DSL, STEPOPT=(H_PSORT),
      SIZEOPT = (2048),
      WKDISK = (AAA, DVC=MS/B10, MD=(MYVOL1, MYVOL2), SIZE=50);
$INPUT DSL, PRINT;
FUNCTION: NBSORT=2
RECORD: KEY= 1 10
END:
$ENDINPUT;
$ENDJOB;
```



## 3.2 MERGE STATEMENT

### 3.2.1 Syntax

The JCL statement for MERGE is shown in Figure 3-2.

The following abbreviations are used in this figure:

sifd:	sequential-input-file-description
sofd:	sequential-output-file-description
smdp:	sort/merge-define-parameters

The following symbolic names refer to standard parameter groups which are described in the *JCL Reference Manual*:

- sequential-input-file-description
- sequential-output-file-description
- print-file-description

The sort/merge-define-parameters are explained below under "Sort/Merge Define Parameters".

The parameter groups sequential-input-file-description, sequential-output-file-description, and print-file-description are not expanded in Figure 3-2. See the *JCL Reference Manual* for all information concerning these standard parameter groups.

## Sort/Merge User's Guide

```
MERGE      INFILE1 = (sifd [smdp])
           INFILE2 = (sifd [smdp])
           [INFILE3 = (sifd [smdp])
           [INFILE4 = (sifd [smdp])
           [INFILE5 = (sifd [smdp])
           [INFILE6 = (sifd [smdp])
           [INFILE7 = (sifd [smdp])
           [INFILE8 = (sifd [smdp])]]]]]]]

           COMFILE = (sifd)

           [           {AUDIT}]
           [           {PARAM}]
           [REPORT = {           }]
           [           {ALL  }]
           [           {NONE  }]

           [PRTFILE = (print-file-description)]
           [BUFFER = dec3]
           [SIZEOPT = (size-parameters)]
           [STEPOPT = (step-parameters)];
```

**Figure 3-2. Format of the JCL Statement MERGE**

### 3.2.2 Explanation of Parameters

INFILEn	This specifies the input files. This parameter is used by MERGE to generate JCL statements ASSIGN and DEFINE during MERGE execution. Note that the NBEFN parameter described in the <i>JCL Reference Manual</i> for the parameter group sequential-file-description is not valid for MERGE.
OUTFILE	<p>This specifies the output files.</p> <p>SYSOUT specifies that the output file is to be written as a standard SYSOUT subfile.</p> <p>SYS.OUT must be used for the external-file-name if the SYSOUT parameter is used.</p> <p>DATAFORM=SSF can be used for assigning SYS.OUT, as well as OUTFILE. In OUTFILE it involves the creation of SSF headers at the beginning of the record. The DATAFORM parameter is used to indicate the DATAFORM on output. Thus the records which are written will correspond with the specified DATAFORM. Records which are internal to the merge have no header.</p>
COMFILE, REPORT, PRTFILE	See the corresponding explanation for SORT.
BUFFER	<p>Normally, 30K bytes of memory are reserved for the input and output buffers of a MERGE (where K = 1024). Depending on the block sizes of the files, and the number of buffers per file, this may be too low or too high. The user can override the default value by using the BUFFER parameter, with dec3 equal to the number of units of 1024 bytes required. The total MERGE memory requirement will be: <math>(12 + \text{dec3}) \times 1024</math> bytes. The buffer requirement is calculated by adding together the products of the block size and the buffering factor of all the input files and the output file. <b>Optimization:</b> It is always better to specify a large BUFFER.</p>
SIZEOPT	The SIZEOPT parameter can be used to specify parameters used in the SIZE JCL statement. However, the size-parameters are not considered in determining the merge memory size (see BUFFER key-word above).
STEOPT	See the corresponding explanation for the SORT statement.

### 3.3 FILE DEFINITION AND SPECIFICATION

#### 3.3.1 DEFINE Parameters for SORT and MERGE Statements

Note that the following paragraphs do not apply to the GSORT and GMERGE statements.

The sort/merge-define-parameters (smdp) which can be included in the INFILE and OUTFILE parameters of the JCL statement SORT and in the INFILEn and OUTFILE parameters of the JCL statement MERGE are shown in Figure 3-3.

```

[          {UFAS} ]
[FILEFORM {    } ]
[          {BAFS} ]

[BLKOFF = dec3]

[          {RETCODE} ]
[          {SKIP  } ]
[ERROPT = {    } ]
[          {ABORT  } ]
[          {IGNORE } ]

[RECSIZE = dec5]
[BLKSIZE = dec5]

[          {FB} ]
[          {F  } ]
[RECFORM = {VB} ] [NBSN]
[          {V  } ]
[          {U  } ]

[WRCHECK]
[DUMMYREC = dec3]

[          {TTRDD} ]
[ADDRFORM = {    } ]
[          {LRRR  } ]

[          {2} ]
[NBBUF = {    } ]
[          {1} ]

[          {1  } ]
[BPB = {    } ]
[          {dec2} ]

For SORT only :
[          {SSF } ]
[DATAFORM = {    } ]
[          {SARF} ]

[ {TRUNCSSF} ]
[ {NTRNCSSF} ]

For V5 MERGE :
[          {SSF } ]
[DATAFORM = {    } ]
[          {SARF} ]

[ {TRUNCSSF} ]
[ {NTRNCSSF} ]

```

**Figure 3-3. File Definition for SORT and MERGE Statements**

## SORT and MERGE Statements

The use of the parameters shown in Figure 3-3 is described below. All these parameters also form part of the JCL statement DEFINE, except for WRCHECK:

FILEFORM	BLKSIZE
BLKOFF	RECFORM
ERROPT	NBSN
RECSIZE	

FILEFORM This specifies the access method, UFAS or BFAS. The following file organizations are supported by these access methods:

- UFAS Sequential, Indexed, Relative,
- BFAS Sequential (disk or tape).

### IMPORTANT

The default value of FILEFORM for Sort/Merge disk output file is UFAS (no longer BFAS). This involves changes in file dimensioning. It is strongly recommended to specify suitable values for BLKSIZE (CISIZE value) and for file SIZE. Remember that only sequential and linked-queued organizations exist for BFAS disk files, and that BFAS sequential disk files explicitly or implicitly allocated on FBO volumes will have the "queued monosubfile" organization and will also require more space than on VBO volumes.

BLKOFF This is used when ANSI Level 1 tape files are to be processed. The data offset from the beginning of the block is specified by dec3. The maximum value is 255.

ERROPT This specifies the action to be taken in case of an I/O error on the file.

RETCODE signifies that the Sort/Merge utility should display the return code via SYSOUT and then terminate. This is the default value.

SKIP signifies that the block at which the I/O error occurred should be skipped and processing should continue with the next block.

ABORT signifies that the Sort/Merge should terminate immediately.

IGNORE signifies that the remainder of the block which has not been read should be padded with blanks and processing should continue with the incorrectly read block. The IGNORE option should be used with care since invalid decimal data may be introduced, resulting in abnormal termination of Sort or Merge.

RECSIZE	<p>This specifies the size of a logical record in a UFAS or BFAS file. For fixed length records, dec5 is the size, in bytes. For variable length records, dec5 is the maximum size, in bytes, not including the record descriptor word (RDW). The RECSIZE parameter is mandatory for non-preallocated output files.</p>
BLKSIZE	<p>This gives the size of a block for UFAS or BFAS files. The BLKSIZE parameter is mandatory for non-preallocated output files.</p> <p>For BFAS disk files and tapes:</p> <p>Fixed - dec5 is the size, in bytes, of the block not including the block serial number.</p> <p>Variable - dec5 is the maximum block size, in bytes, not including the block descriptor word.</p> <p>For UFAS disk files, the block size should be the same as the control interval size. That is, BLKSIZE = CISIZE. When an UFAS disk output file or private printfile or logfile needs a CISIZE value specification, this value must be given through the BLKSIZE keyword.</p> <p>Note that the CISIZE parameter is not available within the Sort/Merge DEFINE parameter. Consequently, if FILEFORM is not indicated as BFAS, and the device-class is not explicitly stated as tape or cartridge, a BLKSIZE parameter given in the Sort/Merge OUTFILE parameters is interpreted as a CISIZE for a UFAS disk file (instead of BLKSIZE for BFAS disk or tape file).</p>
RECFORM	<p>This specifies the logical record format for UFAS and BFAS files:</p> <p>FB-fixed length, blocked (default value).</p> <p>F-fixed length, unblocked.</p> <p>VB-variable length, blocked.</p> <p>V-variable length, unblocked.</p> <p>U-undefined format.</p> <p>NBSN is for UFAS and BFAS tape or cassette files only. It indicates that the file has no block serial numbering. Default is the contrary.</p>

## SORT and MERGE Statements

WRCHECK	<p>This is only used for BFAS disk output files. When this parameter is used, each block written to the output file is re-read to check that it has been correctly written. If the WRCHECK parameter is not used, no read after write will be carried out.</p>
DUMMYREC	<p>This is used when creating a BFAS indexed sequential file. The number of data records between each dummy record in the output file is specified by dec3.</p>
ADDRFORM	<p>This parameter is used only in the JCL statement SORT for the input file. It is used to specify the format of the input record disk addresses to be included in the output record. ADDRFORM is significant only when one of the values ADDATA, ADDROUT or KEYADDR is used in the OUTPUT DSL command.</p> <p>Before using either the TTRDD or the LRRR option the user must ensure that the file organization can support it.</p> <p>TTRDD signifies that the address is to be stored in the form: Bytes 1 and 2 : relative track number in the file (TT). Byte 3 : block number on the track (R). Bytes 4 and 5 : displacement within the block (DD).</p> <p>TTRDD is the default value. UFAS interprets it as SFRA. LRRR signifies that the address is to be stored in the form: Bytes 1, 2 and 3 : Logical record number; that is, the serial number of the record within the file. Bytes 4 and 5 : undefined value. LRRR is applicable to any file organization with fixed length records. However, for the current release, LRRR is not supported for diskettes. LRRR cannot be used in either indexed sequential or random organization.</p>
NBBUF	<p>This specifies the number of buffers to be used by Sort and Merge when processing the file. The default value is 2, that is, double buffering, but single buffering may be required in order to minimize memory usage (the value is then 1)</p>
BPB	<p>This specifies the number of blocks per buffer. That is, the number of blocks to be read at each physical access of the input file, or the number of blocks to be written at each physical access of the output file.</p>
DATAFORM	<p>This parameter specifies whether Sort/Merge input or output file records are to be interpreted (or created) with Standard System Format (SSF) or with Standard Access Record Format (SARF, default).</p>

TRUNCSSF	This parameter specifies that SSF prefixes of input data records must be ignored and that SSF control records of SSF (or SSF-like) input file must be skipped. This is the default value for SORT for Queued and tape input files.
NTRNCSSF	This parameter specifies the contrary of TRUNCSSF. This is the default value for UFAS disk input files.

### 3.3.2 File Specification for SORT and MERGE Statements

#### 3.3.2.1 Parameters

Note that the following paragraphs do not apply to the GSORT and GMERGE statements used with the OUTALC parameter. Neither do they apply to GCL commands SORT\_FILE (see Appendix D) and MERGE\_FILE (see Appendix E) used with the DYNALC parameter.

The parameter values for FILEFORM, RECFORM, RECSIZE and NBSN can be specified when assigning the file; otherwise the pre-existing values from the file label will apply (see the *UFAS EXTENDED User Guide*).

On the other hand, if the file is to be created or if the file is to be input and does not have self-sufficient labels (for example, an unlabelled tape file), then the above parameters must be specified in the file description.

The circumstances under which these parameters must be specified are described below. Parameters are either optional (O) or mandatory (M). Optional means that if the parameter value is not explicitly given in the JCL, the default value for this parameter is used. Mandatory means that a value must be given in JCL for the corresponding parameter since no default value is provided.

The parameters required for disk files are summarized in Table 3-1. The parameters required for tape files are summarized in Table 3-2.



SORT and MERGE Statements

**Table 3-1. Definition Parameters for Disk Files**

File Type	File Organization	Extra Parameters Required	Optional or Mandatory
Permanent or temporary disk input	UFAS	none	
	BFAS		
Permanent or preallocated disk output	UFAS	none	
	BFAS		
Temporary or non-preallocated disk output	UFAS (*)	FILEFORM RECFORM RECSIZE BLKSIZE (*)	O O M O
	BFAS	FILEFORM RECFORM RECSIZE BLKSIZE	M O M M
(*) A default value of 2048 bytes will be taken for CISIZE			

**Table 3-2. Definition Parameters for Tape Files**

<b>File Type</b>	<b>File Organization</b>	<b>Extra Parameters Required</b>	<b>Optional or Mandatory</b>
Permanent or temporary labelled tape inputfile	BFAS UFAS	none	
Unlabelled tape inputfile	BFAS (*) UFAS	FILEFORM RECFORM RECSIZE BLKSIZE	O O M M
Permanent or temporary labelled or unlabelled tapeoutput file	BFAS UFAS	FILEFORM RECFORM NBSN RECSIZE BLKSIZE	O O O (*) M M
(*) Unlabelled tape files are always processed by GCOS 7 assuming they do not have a block serial number.			

For the GSORT/GMERGE statements using OUTALC [= ...] with/without OUTDEF and INDEF[i] parameter groups, the automatic output file allocation and definition applies as for the CREATE utility (or GCL command LOAD\_FILE). See the manual *Data Management Utilities*.

For the SORT/MERGE statements, and for SORT\_FILE/MERGE\_FILE GCL commands without DYNALC, non-existing output files must be specified with consistent definition and allocation parameters.

## SORT and MERGE Statements

### 3.3.2.2 Examples

#### Input Files

UFAS or Queued input files may be specified in the SORT statement as follows:

- On a resident disk:

```
INFILE = (FILE1)                                for UFAS
```

```
INFILE = (LIB1, SUBFILE=MB1)                    for Queued
```

- On a user disk:

```
INFILE = (FILE2, DEVCLASS = MS/FSA, MEDIA = DISK10)
```

- On tape:

```
INFILE = (FILE3, DEVCLASS = MT/T9/D1600, MEDIA = TAPE1)
```

#### Output Files

UFAS or Queued output files may be specified in the SORT statement as follows:

- Preallocated file on disk:

```
OUTFILE = (FL7, DVC = MS/B10, MD = DISK3)      for UFAS
```

- Member of a preallocated cataloged library:

```
OUTFILE = (LIB7, SUBFILE = MB6)                for Queued
```

- Temporary UFAS file on disk:

```
OUTFILE = (FILE9, DEVCLASS = MS/FSA, MEDIA = DISK5,  
           FILESTAT = TEMPRY, FILEFORM = UFAS,  
           SIZE = 4, RECSIZE = 127, RECFORM = V)
```

(Default CISIZE is 2048 bytes (2K))

- Permanent UFAS files on disk are the same as for temporary UFAS files with the following differences:

```
FILESTAT = UNCAT or FILESTAT = CAT (or FILESTAT omitted);  
EXPDATE can be specified.
```

- BFAS tape file :

```
OUTFILE = (FILE0, DVC = MT/T9, MD = TAPE4, RECSIZE = 90,  
           BLKSIZE = 270, EXPDATE = 95/01/01, NBSN)
```

**NOTE:** For disk files that did not pre-exist, the default value is UFAS.

### 3.4 EXAMPLES OF SORT AND MERGE STATEMENTS

An input file named PAYLIST on tape volume TV24 with a record length of 200 bytes and a block size of 4000 bytes is to be sorted. The output file, called SPAYLIST, will also be on tape, on the volume TV81. The block size of the output file will be 2000 bytes. The DSL commands for Sort are held as a source library member, SCPAY, in a RESIDENT source library named SPLL.PY. The memory allocated to the sort is 512K bytes.

If the sort is performed using the default disk work file provided by SORT, then the statement will be:

```
SORT  INFILE= (PAYLIST,DEVCLASS=MT/T9,MEDIA=TV24)
      OUTFILE=(SPAYLIST,EXPDATE=300,DEVCLASS=MT/T9,
              MEDIA=TV81,RECSIZE=200,BLKSIZE=2000)
      SIZE=512
      COMFILE=(SPLL.PY,SUBFILE=SCPAY);
```

If the WKDISK parameter is used then the statement will have the following form:

```
SORT  INFILE= (PAYLIST,DEVCLASS=MT/T9,MEDIA=TV24)
      OUTFILE=(SPAYLIST,EXPDATE=300,DEVCLASS=MT/T9,
              MEDIA=TV81,RECSIZE=200,BLKSIZE=2000)
      WKDISK= (SIZE=15,DEVCLASS=MS/FSA,MEDIA=BD14)
      COMFILE=(SPLL.PY,SUBFILE=SCPAY);
```

In this case the sort file is a temporary file of 15 cylinders on an FSA disk volume named BD14. The temporary file is named H\_SRTWKD.

## SORT and MERGE Statements

If the file SPAYLIST is too large to be accommodated in the disk space available the user may wish to split the sort process into two parts and then subsequently merge the results to form the complete sorted file. If the file contains 10 000 records then a two-part sort and merge would be of the form:

```
$JOB      LSPAY, USER=LX518, PROJECT=INTER;
SORT      INFILE= (PAYLIST, DEVCLASS=MT/T9, MEDIA=TV24)
          OUTFILE=( SPAYLIXA, DEVCLASS=MT/T9, MEDIA=TVVV,
                    RECSIZE=200, BLKSIZE=2000)
          SIZE=50
          COMFILE=(SQLL.PY, SUBFILE=SCPAY);
          HALT=5000;
SORT      INFILE= (PAYLIST, DEVCLASS=MT/T9, MEDIA=TV24)
          OUTFILE=( SPAYLIXB, DEVCLASS=MT/T9, MEDIA=TVVB,
                    RECSIZE=200, BLKSIZE=2000)
          SIZE=50
          COMFILE=(SQLL.PY, SUBFILE=SCPAY);
          START=5001;
MERGE     INFILE1=( SPAYLIXA, DEVCLASS=MT/T9, MEDIA=TVVV)
          INFILE2=( SPAYLIXB, DEVCLASS=MT/T9, MEDIA=TVVB)
          OUTFILE=( SPAYLIST, DEVCLASS=MT/T9, MEDIA=TV81,
                    RECSIZE=200, BLKSIZE=2000)
          COMFILE=MERCAR;
$INPUT   MERCAR;
          .
          .
          .
DSL commands
          .
          .
          .
$ENDINPUT;
$ENDJOB;
```

Note how the DSL commands for Merge are entered as an input enclosure. The two-part operation is performed using the intermediate files SPAYLIXA and SPAYLIXB, on tape volumes TVVV and TVVB.

## Sort/Merge User's Guide

## 4. GSORT and GMERGE Statements

This chapter describes the statements GSORT (for sorting) and GMERGE (for merging). GSORT and GMERGE each require that the input, output and command files be specified.

All other parameters of the statements are optional.

Default values exist, which, if they are appropriate to the application, allow the specification of certain parameters to be omitted. The optional parameters concern work files (Sort only), reporting, performance optimization, error record processing, part-file processing, checkpointing, and abnormal termination.

## 4.1 GSORT STATEMENT

### 4.1.1 Syntax

The GSORT statement may be used in batch or interactive mode (JCL).

The syntax of the JCL statement GSORT is shown in Figure 4-1.

```

GSORT
{
  { INFILE = (asg) | INFILES = ((asg1), (asg2), ... (asgk)) } [,INDEF=(def)]
  ,OUTFILE = (asg) [,OUTDEF = (def)] [,OUTALC [= (alc)] [,SYSOUT = (sysout)]]
  |
  FILE=(asg) [,DEF=(def)]
}
,COMFILE = (asg) [,COMDEF = (def)]

[,PRTFILE = (asg) [,PRTDEF = (def)] [,PRTALC [= (alc)] [,PRTOUT = (sysout)]]

[,LOGFILE = (asg) [,LOGDEF = (def)] [,LOGALC [= (alc)]]]
[,LOGOUT = (sysout)]

[ {
  ,WKFILE = (asg) [,WKALC = (alc)]
  |
  [MODALC = (alc)]
  WKASG1 = (asg) [,WKALC1 = (alc)]
  ,WKASG2 = (asg) [,WKALC2 = (alc)]
  ,WKASG3 = (asg) [,WKALC3 = (alc)]
  ...
  ...
  ,WKASGn = (asg) [,WKALCn = (alc)]
} ]

[,REPORT = { AUDIT | PARAM | ALL | NONE }]

[,SIZE = digits8]

[,REPEAT]

[,INVREC = ({ABORT | CONTINUE}) [,ERROPT = {PRINTID | PRINT | LOG | IGNORE}]]]

[,START = digits10] [,HALT = digits10]

[,MEM]

[,NBSORT = digits2]

[,FILLER = {char1 | hexa2}]

[,SIZEOPT = (sizeopt)]

[,STEOPT = (stepopt)] ;

```

**Figure 4-1. Format of the JCL Statement GSORT**



#### 4.1.2 Explanation of Parameters

##### INFILES

If several files (GSORT accepts up to 9 files, as for the SORT) are to be input to the sort, you must use the syntax:

```
INFILES = ( (asg1), (asg2), .....(asgk)), ...
```

The INDEF option is then valid for any of the concatenated input files.

For example:

```
INFILES=( (MY.SL1,MB=MB1), (MY.SL2,MB=MB2)),  
or  
INFILES=( (MY.FL1), (MY.FL2), (MY.FL3)),
```

Unless specified otherwise by explicit POOL, FIRST or POOL, NEXT, in the INFILES assignments, GSORT implicitly generates POOL, FIRST or POOL, NEXT on concatenated files explicitly assigned on tapes or cartridges.

##### FILE

The existing SORT allows the sorted file to be written onto the INFILE by using the syntax OUTFILE = INFILE. To achieve this function with GSORT, you must replace both INFILE and OUTFILE by the single FILE specification (with DEF instead of INDEF).

For example:

If you want to read (input) and overwrite (output) the file named INOUT.FILE (an uncataloged file), you must specify:

```
FILE=(INOUT.FILE, MD=MYVOL, DVC=MS/FSA)
```

##### OUTFILE, OUTALC, OUTDEF

These keywords introduce the standard parameters for the output file.

OUTALC has a special meaning, as for the standard GCOS 7 utilities (CREATE, etc.):

If the file referenced by **OUTFILE** does not exist and the **OUTALC** keyword is present, even without any specified allocation values (OUTALC=( ... ), ..), **the Sort/Merge will copy the missing OUTALC and OUTDEF characteristics from the corresponding INFILE ones.**

For Merge, the model file is INFILE1.

This is not applicable to subroutine sort.

**Example:**

```
IF=MY.FILE, OUTALC, OF=(OUT.FILE, MD=MYVOL,  
                        DVC = MS/FSA, CATNOW)
```

For a detailed description of the automatic allocation process, please read the manual *Data Management Utilities*.

PRTDEF, LOGDEF

These keywords introduce the define parameters for PRTFILE and LOGFILE, respectively. They must be correctly supplied if the corresponding file is to be allocated.

PRTALC, LOGALC

These keywords introduce the allocation parameters for PRTFILE and LOGFILE, respectively. The default values SIZE = 1, INCRSIZE = 1, and UNIT = CYL, apply if PRTALC/LOGALC are specified without values.

SYSOUT, PRTOUT, LOGOUT

These keywords introduce the standard sysout parameters for OUTFILE, PRTFILE, and LOGFILE, respectively.

WKFILE, WKALC

These keywords are one of the two ways to allocate the sort workfile(s). The other way is using WKASGi, WKALCi, and MODALC described below.

WKFILE and WKALC define one file assignment, which is used as follows:

**Mono-process Sort**

You must use either no workfile or only one workfile and then define it via WKFILE and WKALC. These keywords allow you to specify the corresponding standard parameter sets (asg) and (alc) for the single workfile.

**Multi-process (Parallel) Sort**

You can use the WKFILE and WKALC keywords to introduce a list of MEDIA. Such a list is limited by GCOS 7 to 10 MEDIA. One TEMPRY workfile will be dynamically allocated for each sub-sort, numbered from 1 to n (where NBSORT=n), using the media list in a circular way.

For example, if MEDIA = (A,B,C) and NBSORT = 3, then the 1st workfile is allocated on A, B, C, the 2nd workfile on B, C, A, and the 3rd workfile on C, A, B.

If the MEDIA list is shorter than n MEDIA (NBSORT = n), then n is reduced according to the exact number of MEDIA in the list.

The SIZE of each TEMPRY file will be the SIZE specified by: WKALC = (... SIZE= ...) divided by n.

If no SIZE is specified under WKALC = (), then a default size is defined by sort as with the JCL statement SORT (see the description of the WKDISK parameter in paragraph 3.1.2). The default size is valid for each of the n workfiles.

A WKFILE given with VOLSET will be considered by Sort as a list of media. This list is retrieved by the H\_RTVLID GCOS 7 primitive. If there are more than 10 media in the VOLSET, the retrieved list is limited to 10 media.

**Examples:**

For a mono-process sort, a TEMPRY workfile of 10 CYLs on RESIDENT or implicit VOLSET is defined as follows:

```
WKFILE=(WORK, FILESTAT=TEMPRY),
      WKALC=(SIZE=10),
```

A multi-process sort with NBSORT=2 may have a workfile defined as:

```
WKFILE=(AAA, DVC=MS/B10, MD=(MYVL1, MYVL2))
      WKALC = (SIZE = 10),
```

The Sort will not use the file named AAA (whether it exists or not). Instead, the Sort will dynamically assign and allocate two TEMPRY workfiles of 5 cylinders each on the media MYVOL1 and MYVOL2. The implicit UNIT is CYL.

MODALC

This keyword implies the presence of the WKASGi assigns, and they specify a multi-workfile sort (parallel sort). MODALC specifies the common parameters for any of the WKASGi workfiles. If used, **MODALC must appear before WKASGi and WKALCi**. Otherwise, an error message will be given and the JCL translation aborted.

MODALC allows you to specify default SIZE, INCRSIZE, and UNIT for any of the workfiles. The rules are:

- if UNIT exists for the ith workfile via its own WKALCi, it remains unchanged.
- if not, this ith workfile inherits the UNIT from MODALC (if current) or of the default unit CYL.

The same rule applies for SIZE and INCRSIZE.

The default values are:

SIZE = 10 CYL | 200 TRACK | 100 x 100KB | 2500 BLOCK,

depending on the value of UNIT.

INCRSIZE = max (SIZE/16,2),

whatever the SIZE and UNIT.

**Examples:**

```
WKASG1=(W1 TEMPRY),  
WKALC1=(SIZE=1, INCRSIZE=1),  
WKASG2=(W2 TEMPRY),  
WKALC2=(SIZE=1, INCRSIZE=1)
```

may also be written:

```
MODALC=(SIZE=1, INCRSIZE=1),  
WKASG1=(W1 TEMPRY),  
WKASG2=(W2 TEMPRY)
```

## GSORT and GMERGE Statements

WKASGi, WKALCi

These keywords introduce the *n* workfiles of a **parallel sort** in the most general form. The workfiles are ranked from 1 to *n* (NBSORT = *n* must be greater than one). They are defined by the standard (asg) and (alc) file parameter sets. The workfiles may be pre-allocated by the user or defined for dynamic allocation by the sort at run time. See also the MODALC parameter above.

With these WKASGi assignments, there is no restriction to VOLSET (no limitation to 10 media).

Note that a **parallel sort may also be activated via the WKFILE and WKALC keywords** as explained above.

A parallel sort is actually parallel if the workfiles, the input file, and the output file are each on a different media (or set of media).

GSORT supports only 16 workfiles assigns. But the limit of 32 slave sorts remains valid for "memory sorts" (NBSORT = 32 and no workfiles) and also for basic JCL STEP level executions.

### **Examples:**

To specify 2 workfiles of 33 VBO-MS/B10 CYLs each, with INCRSIZE=1, on specified media, and another workfile on an MS/FSA media with a specific allocation equivalent to 33 CYLs of VBO-MS/B10, you specify:

```
MODALC= (SIZE=33, INCRSIZE=1) ,  
WKASG1= (W1 FILESTAT=TEMPRY, DVC=MS/B10,  
        MEDIA=V1) ,  
WKASG2= (W2 FILESTAT=TEMPRY, DVC=MS/B10,  
        MEDIA=V2) ,  
WKASG3= (W3 FILESTAT=TEMPRY, DVC=MS/FSA,  
        MEDIA=D1) ,  
WKALC3= (SIZE=17) ,
```

or a more simple way:

```
MODALC= (SIZE=3960, INCRSIZE=100, UNIT=BLOCK) ,  
WKASG1= (W1 FILESTAT=TEMPRY, DVC=MS/B10,  
        MEDIA=V1) ,  
WKASG2= (W2 FILESTAT=TEMPRY, DVC=MS/B10,  
        MEDIA=V2) ,  
WKASG3= (W3 FILESTAT=TEMPRY, DVC=MS/FSA,  
        MEDIA=D1) ,
```

The most simple way to cope with any mix of device class and volume organization, is to use UNIT = BLOCK.

To specify 2 workfiles of 300 blocks each, with INCRSIZE=1, on specified MS/FSA media, you specify:

```
MODALC=(SIZE=300,UNIT=BLOCK,INCRSIZE=1),
WKASG1=(W1 FILESTAT=TEMPRY, DVC=MS/FSA,
        MEDIA=V4),
WKASG2=(W2 FILESTAT=TEMPRY, DVC=MS/FSA,
        MEDIA=V5),
```

REPORT, SIZE, REPEAT, INVREC, ERROPT, START, HALT

These options are the same as in the SORT statement, with exactly the same syntax. However, in interactive mode, REPORT = NONE is the default for both the JCL statement GSORT and the GCL command SORT\_FILE.

The SIZE (memory size in Kbytes) option is incremented to digits8. The implicit memory SIZE in the GSORT statement is 512 K bytes per process (see also the description of the SIZE parameter in paragraph 3.1.2).

#### **Mono-Process Sort**

For a mono-process sort, the memory SIZE is limited to 64 Megabytes.

#### **Multi-Process Sort**

For a multi-process sort, the limit is: (NBSORT x 32Mb), up to 32 x 32 Mb in the limit of the machine configuration. If the user specifies a SIZE value which exceeds NBSORT x 32Mb, then the sort will be done under only NBSORT x 32Mb.

If the specified SIZE does not exceed NBSORT x 32Mb, then each process will run under about SIZE divided by NBSORT.

#### **Examples:**

For a mono-process "in memory" sort:

```
GSORT IF= ..., OF= ..., MEM,
      SIZE=4096, START=12, HALT=15500,
      REPORT=ALL, INVREC=(CONTINUE), REPEAT;
```

For a multi-process sort (NBSORT=2), with two workfiles on specified media:

```
GSORT IF= ..., OF= ..., NBSORT=2,
      SIZE=4096, START=12, REPORT=ALL,
      INVREC=(ERROPT=PRINT), REPEAT,
      WKFILE=(AAA, DVC=MS/FSA, MEDIA=(V1, V2)),
      WKALC = (SIZE = 10);
```

## GSORT and GMERGE Statements

or:

```
GSORT IF= ..., OF= ..., NBSORT=2,  
SIZE=4096, START=12, REPORT=ALL,  
INVREC=(ERROPT=PRINT), REPEAT,  
MODALC=(SIZE=50, INCRSIZE=1),  
WKASG1=(W1, FILESTAT=TEMPRY, DVC=MS/FSA,  
MEDIA=V1),  
WKASG2=(W2, FILESTAT=TEMPRY, DVC=MS/FSA,  
MEDIA=V2);
```

Note that in the these two examples, REPEAT allows an automatic STEP restart, but Sort takes no checkpoints. The checkpoint facility is not allowed for multi-process jobs.

MEM

This keyword forces a memory sort (that is, a sort done "in memory"). The memory allocated to sort will be used only to receive records (no workfile buffers). If the file to be sorted is too large to fit in memory, then the sort aborts (refer to Chapter 7 to see how to estimate memory size). This GSORT keyword is equivalent to WKDISK=(SIZE=0) in a SORT JCL statement.

**Example:**

```
GSORT IF= ..., OF= ..., MEM,  
SIZE=4096, REPORT=NONE, INVREC=(CONTINUE);
```

With the MEM keyword, the sort will not allocate any workfiles. The result is a "memory sort" (which may abort if the memory is insufficient).

NBSORT

This keyword specifies the number of "slaves sorts" of a parallel sort. Its value should be equal to the number of workfiles defined by WKASGi (and WKALCi) or to the number of MEDIA defined by WKDISK. Otherwise, n will be set equal to:

min (NBSORT, number of workfile/media assigned)

If the resulting value is 1, due to the media list, then the sort will abort because of the mismatch between parallel sort and one workfile.

For "non memory sorts", the number of slave sorts depends mainly (for performance reasons) on the number of distinct MEDIA that can be used, and at a lower level on the number of physical channels for these MEDIA. The recommended value, for "non-memory" sorts, is between 2 and 4.

The number of CPUs is irrelevant, except for CPU bound sorts. One CPU allows a useful level of simultaneity between workfile I/Os on several MEDIA.

For "memory sorts", the multi-process sort allows a 1 Gigabyte memory. The maximum memory per process is 32 Megabytes. The NBSORT value will be chosen in order to allow the desired memory value.

For example, if you want to run a sort with SIZE = 128 Megabytes, you choose SIZE = 131072 (that is, 128 x 1024 bytes) and NBSORT = 4 (since 4 x 32 = 128 Megabytes).

At run time, this keyword and its value are placed in the "OPTIONS = string" (by the GSORT statement) to be retrieved by the sort (H\_GETOPT).

**Example:**

```
GSORT IF= ..., OF=..., NBSORT=2, ...
```

FILLER

This keyword specifies the character to be used for padding output records, when necessary. The default option is "blank".

**Example:**

```
GSORT IF = ..., OF = ..., FILLER = *,
```

or

```
GSORT IF = ..., OF = ..., FILLER = 5C,
```

5C being the hexadecimal code for the \* character.

STEOPT=(), SIZEOPT=()

These options are, as for the current SORT JCL commands, the standard JCL step and SIZE options.

### 4.1.3 Implicit Values

The implicit values of the sort keywords remain unchanged (same values as in the current JCL statement SORT), except for the sort memory SIZE which is now 512K per process.



## 4.2 GMERGE STATEMENT

### 4.2.1 Syntax

The GMERGE statement may be used in batch or interactive mode (JCL).

The syntax of the JCL statement GMERGE is shown in Figure 4-2.

```

GMERGE

    INFILE1 = (asg1) [ ,INDEF1 = (def1) ]
    ,INFILE2 = (asg2) [ ,INDEF2 = (def2) ]
    ,INFILE3 = (asg3) [ ,INDEF3 = (def3) ]
    ...
    ...
    ...
    ,INFILEi = (asgi) [ ,INDEFi = (defi) ]

    ,OUTFILE = (asg) [ ,OUTDEF = (def) ] [ ,OUTALC [= (alc) ] ]
    ,COMFILE = (asg) [ ,COMDEF = (def) ]

    [ ,PRTFILE = (asg) [ ,PRTDEF = (def) ] [ ,PRTALC [= (alc) ] ] ]
    [ ,SYSOUT = (sysout) ]
    [ ,PRTOUT = (sysout) ]
    [ ,REPORT = { AUDIT | PARAM | ALL | NONE } ]
    [ ,BUFFER = digits3 ]
    [ ,FILLER = { char1 | hexa2 } ]
    [ ,SIZEOPT = (sizeopt) ]
    [ ,STEOPT = (stepopt) ] ;

```

**Figure 4-2. Format of the JCL Statement GMERGE**

#### 4.2.2 Explanation of Parameters

INFILEi	INFILE1 to INFILEi ( $i \leq 8$ as for MERGE) are the input files to be merged. The implicit values for the DEFINE parameters are the same as for SORT.
REPORT	The REPORT option has the same meaning as in the MERGE statement.
BUFFER	The BUFFER option has the same meaning as in the MERGE statement.

The meanings for the other parameters of GMERGE are the same as for the corresponding keywords in the GSORT statement.

## 5. DSL for Sort

The DSL (Data Services Language) commands for Sort are described in this chapter. These DSL commands apply to Sorts requested via the JCL statements SORT (described in Chapter 3) and GSORT (described in Chapter 4) and the GCL command SORT\_FILE (described in Appendix D).

These DSL commands are placed in a command file. They describe the record-level functions to be performed during the sort. These functions are based on the nature of the input data. Since the KEYS command is always mandatory, a command file must be provided with every sort.

DSL commands are interpreted according to keyword analysis, not position. The command file should be thought of as a continuous stream of characters.

### 5.1 COMMAND FILE ORGANIZATION

The DSL commands are organized into three paragraphs called FUNCTION, RECORD, and END. Of these, the FUNCTION paragraph is optional but, if supplied, must precede the other two. FUNCTION paragraph commands are used to select special processing options. The default options are underlined in the text; if a parameter comprises only a keyword (e.g. DESCEND in the FUNCTION paragraph) the default option is the reverse of that implied when the parameter is specified.

The RECORD paragraph follows FUNCTION (if supplied) and contains a set of commands that provide Sort with information regarding the content and format of individual fields within the user's records. A RECORD paragraph is mandatory since it includes the KEYS command.

The final paragraph in the command file consists simply of the keyword END which terminates the file.

Each paragraph heading can be followed by a colon. Within a paragraph, commands may appear in any order, and must be separated from one another by either blanks or commas.

Elements of a command may be parenthesized at the user's discretion to enhance readability or aid organization. Comments may be inserted to improve readability. A comment has the form */\*comment\*/*.

## 5.2 FUNCTION PARAGRAPH

The format of the FUNCTION paragraph is the following:

```

FUNCTION: [DESCEND]

          [COLLATE={EBCDIC | G100 | H200 | ASCII | 'hexa512' }]
          [OUTPUT {DATA | ADDATA | KEYADDR | ADDROUT}]
          [DUPREC {NONE | FIFO}]
          [DELETE]
          [INTINFO]
          [SORTSIZE=dec11]
          [MIN=dec5]
          [AVERAGE=dec5]
          [WRDISK {WRDATA | WRCKD}]
          [DEBUG]
          [NBSORT=dec2]
          [LOWYEAR=dec2]

```

The commands of the FUNCTION paragraph are described in 5.2.1 to 5.2.11.

### 5.2.1 DESCEND Command

Unless otherwise specified, Sort will output records in ascending order of the key values; that is, a record whose key value is 1 will appear before a record whose key value is 2, and alphabetic characters will appear in the order appropriate to their collating sequence. When DESCEND is specified, this order is reversed and records are sorted into a descending key value order.

## 5.2.2 COLLATE Command

The COLLATE command allows the user to state explicitly which collating sequence is to be employed by Sort. There are five sequences available:

EBCDIC:	Extended Binary Coded Decimal Interchange Code (the standard GC0S 7 code set)
G100:	Series 100 Code Set
H200:	Series 200/2000 Binary Collating Sequence
ASCII	American National Standard Code for Information Interchange
'hexa512':	Privately defined collating sequence

If COLLATE is omitted then the default value is EBCDIC.

The effects of using the above collating sequences are described below.

EBCDIC This is the standard internal code of GC0S 7.

G100 This value should only be specified when the input file has been translated from a Series 100 environment to the GC0S 7 environment. This translation can be performed using the Series 100 File Translator. G100 causes the records to be sorted into the same order as would have occurred if the sort was performed on a Series 100 system (Table C-2 of Appendix C).

The use of G100 for translated files is not mandatory. If the following conditions apply:

- keys do not contain special characters,
- keys do not mix numeric and non-numeric characters in the same positions,

then G100 is not necessary. The EBCDIC collating sequence can be used. The performance of Sort is better using EBCDIC than any other collating sequence.

H200	<p>This value causes the records to be sorted into the same order as would have occurred if the sort was performed on a Series 200/2000 system. It allows the user to maintain the original Series 200/2000 collating system for a file written in the standard native format and code set.</p> <p>The H200 character set contains fewer graphic symbols than the EBCDIC character set (e.g. there are no lower case letters in the H200 character set). Therefore, certain characters, not in the H200 character set, are translated for sorting purposes into specific H200 characters. This translation does not affect the output file (i.e. the original code set is retained). The resulting order is that shown in Table C-3 of Appendix C.</p>
ASCII	<p>This value specifies that the ASCII (American National Standard Code for Information Interchange) collating sequence be applied. The sequence is shown in Table C-4 of Appendix C. This table shows the low-order to high-order sequence as it is applied to the complete range of EBCDIC hexadecimal values available for a character in GCOS. The ASCII collating sequence may only be applied to a GCOS (EBCDIC) standard file.</p>
'hexa512'	<p>To define a collating sequence other than the above, the hexadecimal representation of all the characters in the character set are entered in order, starting with the highest (that which should always win a comparison) and ending with the lowest (that which always loses).</p> <p>The data is assumed to be in EBCDIC code. The collating sequence must include 256 characters, that is 512 hexadecimal digits. For example, if it is desired to sort with numeric characters higher than alphabetic characters (0123456789ABC...), the DSL command would begin:</p> <pre>COLLATE = 'F0F1F2F3... F9C1C2C3...'</pre>

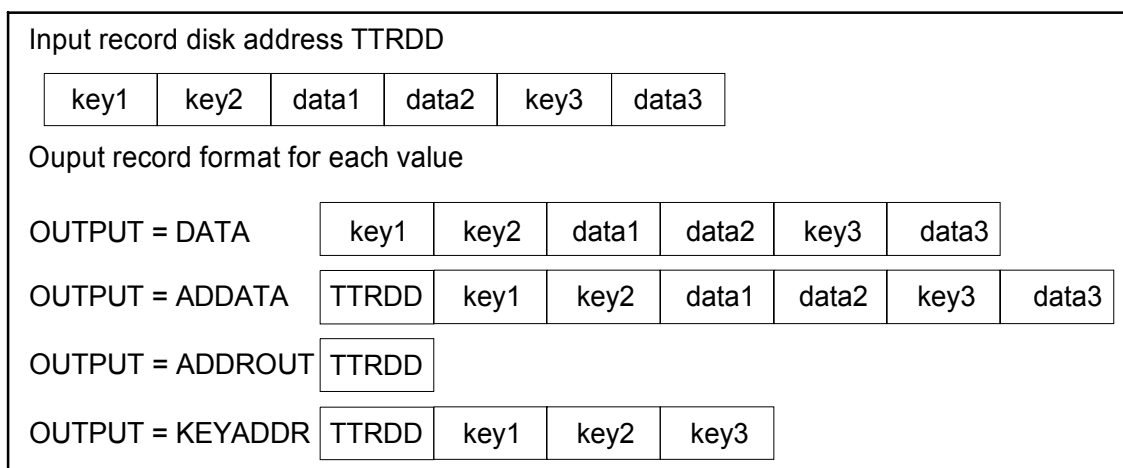
### 5.2.3 OUTPUT Command

The OUTPUT command is used to specify the contents of the records output by Sort. Depending on the value given to the parameter, the output record may contain the entire input record content, or combinations of disk address, data and key fields. The four values of the parameter are given below.

DATA	The output record will consist only of data that appears in the corresponding input record. Either the entire record or only those fields extracted from the record during the sort (via the ARRANGE command) will be included in the output record.
ADDDATA	The output record will consist of original input data as above but will also contain a prefix field giving the original record's disk address.
ADDRROUT	The output record will contain no data at all but will contain the disk address of the original input record. Information on the use of such output is given in the <i>COBOL User's Guide</i> (ACTUAL KEY Phrase).
KEYADDR	The output record will begin with a field giving the disk address of the original record. This will be followed by the contents of the major and minor key fields of the corresponding input record.

The format of the disk address may be TTRDD, where TT is the relative track number, R is the block number relative to the beginning of the track and DD is the record displacement within the block. The format may also be LRRR (logical record number) in which case bytes 1 to 3 of the output record contain the logical record number; the contents of bytes 4 and 5 are undefined.

An example of the output record formats for these values is given in Figure 5-1.



**Figure 5-1. Sample Output Formats**

DATA and ADDATA are used when the user wants the output file to contain all or most of the original data. ADDRROUT and KEYADDR are suitable for applications requiring an index of sorted records, or when there is a limitation on space available for the output file. ADDATA, ADDRROUT and KEYADDR should not be used when concatenated files are specified for input.

The values ADDATA, ADDRROUT and KEYADDR are only valid when the input file is disk resident. Note that for ADDRROUT and KEYADDR output to tape, the record length specified must be at least 18 bytes.

A file containing disk addresses in the form TTRDD can be read into a COBOL program and can be used to access the file that was previously input to the Sort utility (see the *COBOL 85 User's Guide*, ACTUAL KEY Phrase).

#### **5.2.4 DUPREC Command**

The DUPREC command specifies the action to be taken when two or more records have identical values for each of the key fields specified. The meaning of the two values is given below:

- NONE All of the records with identical key fields are output in a random sequence.
- FIFO First-In-First-Out. The duplicate records are output in the order in which they occur in the input file.

#### **5.2.5 DELETE Command**

The DELETE command specifies that all but one record of a set with identical key field values should be deleted. The choice of which of these records to retain for the output file is dependent on the value of the DUPREC command. In a given series of records of equal key values, if DUPREC is NONE, one of the duplicate records will be selected at random; if FIFO has been selected, the first record in will be the one retained.

If, in the RECORD paragraph, summation (SUM) is specified then DELETE must not be specified in the FUNCTION paragraph since deletion is automatic with summation.



## 5.2.6 INTINFO Command

The INTINFO command produces a listing in the Sort/Merge Report of certain internal Sort variables. You are recommended to use this option as it costs nothing in performance and may be useful in checking the sort execution.

The format of the listing is:

```

***** SORT INTERNAL VALUES *****

SORTALGO=dec1, SORTFRM={F|V|C}, ARTBSZ=dec5, BLKART=dec5

NBSORTMEM = dec1, MAXNBSORTMEM = dec11, MAXNBDES = dec6

MRGORDER = dec6

NBMONOPH1 =dec11, NBEXPECTED = dec11, NBPHASE2 = dec6
    
```

where

SORTALGO:	0 = normal sort, 1 = large record sort, 2 = merge sort.
SORTFRM:	intermediate record format, F = Fixed, V = Variable, C = Cell-divided variable. Letters followed by 0 (Fixed group) or by 1 (Variable group). Refer to Chapter 7 for a description of "cell" division.
ARTBSZ:	intermediate record size (bytes).
BLKART:	record block size on work file (bytes).
BLKDES1:	block length of work file descriptors for phases 1, 2, 3 (bytes).
BLKDES2:	block length of work file descriptors for the Sort and Merge phases (bytes).
NBSORTMEM:	number of intermediate record places in memory for initial phase.
MAXNBSORTMEM:	number of intermediate records possible for the initial phase of a sort in memory.
MAXNBDES:	maximum number of channels for the final phase.
MRGORDER:	maximum number of channels for the intermediate phase.
NBMONOPH1:	number of strings expected if there is a random distribution of sort keys.
NBPHASE2:	number of runs in phase 2.

- NOTES:**
1. Phase 1 of Sort (or pre-sort) reads or receives input records and creates strings of sorted records.
  2. Phase 2 of Sort executes the necessary total or partial string merges.
  3. Phase 3 of Sort is the final merge phase that writes or sends sorted output records.

### 5.2.7 SORTSIZE Command (Optimization)

An estimate of the number of input records can be specified in the SORTSIZE command. This number is used by Sort in order to:

- optimize memory utilization, if a "memory" sort is possible,
- check that the size of the work file is adequate, if this has been allocated by the user,
- calculate the size of the work file if none has been allocated by the user.

SORTSIZE is an optional command. However, for efficiency reasons, it is recommended that this command be used whenever possible and that the number of records be estimated as accurately as possible. Over-estimation will produce a more efficient sort than under-estimation. A work file is opened only if the sort cannot be completed in main memory.

### 5.2.8 MIN Command

The MIN command specifies the minimum length of records in the input file. The MIN length covers any element in the record specified by position and length (for example, KEY fields, INCLUDE fields, OMIT fields, etc.). However, if you know that all input records are greater than a given length, then this is the correct MIN value.

MIN is significant only when variable length records are being sorted. See *Sorting Variable Length Records* in Chapter 7.

### 5.2.9 AVERAGE Command

The AVERAGE command specifies the average length of variable length records in the input file. AVERAGE is significant only when variable length records are being sorted.

This value is very important in getting the best out of disk work space. It is advisable never to sort variable length records without using it.

### 5.2.10 WRDISK Command

Options:

WRDATA can be explicitly specified if the work file has been preformatted with physical blocks of 1024 bytes. The physical format is thus not changed by the sort process.

WRCKD is the default option. Sort selects the most convenient physical block size. This is the most usual mode in the absence of segmented disks.

**NOTE:** The WRDISK command is ignored when the Sort work file is allocated on an FBO disk.

### 5.2.11 DEBUG Command

This parameter activates the debugging mechanism, in particular for controlling the sequence of records. CPU time is increased by about 10% as a result. It is mainly used when sort errors occur. It also allows a dump to be obtained when exceptions or internal sort errors occur provided that DUMP=DATA is specified in the STEPOPT parameter of the JCL statement SORT/GSORT.

### 5.2.12 NBSORT Command

NBSORT is valid only for a parallel sort requested via the SORT statement (and not via the GSORT statement).

The NBSORT command specifies the number of slave processes of a parallel sort. The maximum value is 32.

For a parallel sort launched by the JCL statement SORT, the WKDISK assignment provides the media list as the GSORT WKFILE assignment does.

If you want to run a parallel sort via the JCL statement SORT, you must also specify:

```
, STEPOPT=(H_PSORT) , ...
```

See also paragraph 3.1.3.

### 5.2.13 LOWYEAR Command

The keyword "LOWYEAR" (century LOWest YEAR) denotes a year in the range 0 - 99 which is to be the start year of a 100-year window.

For example, if under the SORT or MERGE commands you specify in the FUNCTION paragraph that LOWYEAR = 61, then the decimal fields PDECY, UDECY, DISPY and COMPY are sorted in the ascending sequence (61, 62...98, 99, 00, 01...59, 60).

The value 0 is authorised and gives PDECY, UDECY, DISPY, and COMPY types the same ordering sequence as PDEC, UDEC, DISP and HEXA types.

### 5.2.14 Examples of FUNCTION Paragraph

**Example:**

```
FUNCTION: DUPREC = FIFO SORTSIZE = 10250
```

The Sort output is to be in ascending order. Records with identical key field values will appear in the output file in the same order as in the input file. The output file record will contain the same fields as the input record.

**Example:**

```
FUNCTION: OUTPUT = KEYADDR DELETE SORTSIZE = 5320
```

The FUNCTION paragraph above specifies that the output file, to be ordered in ascending key sequence, should contain only the key and the original disk address of each record. DUPREC = NONE has been chosen by default. Thus the DELETE command ensures that if two or more records are found to have the same key, only one of them, selected at random, will have its address written in the output file.

**Example:**

```
FUNCTION: LOWYEAR = 50
```

This specifies that PDECY, UDECY, DISPY and COMPY fields are to be sorted according to the following ascending sequence : (50, 51...98, 99, 00, 01...48, 49) instead of 00 to 99.

### 5.3 RECORD PARAGRAPH

The commands in the RECORD paragraph specify fields within the user's input records which must be examined by Sort during execution. They are as follows:

```
RECORD: KEY[S]=key-element [key-element]...
        [{INCLUDE=condition-element [AND condition-element]...
        [INCLUDE=condition-element [AND condition-element]...}]...
        [OMIT=condition-element [AND condition-element]...
        [OMIT=condition-element [AND condition-element]...}]...}]
        [SUM=sum-element [AND sum-element]...]
        [ARRANGE=arrange-element [AND arrange-element]...]
```

KEY specifies one or more key fields whose values determine the sort order. The optional spelling KEYS has the same meaning as KEY.

INCLUDE and OMIT are mutually exclusive record selection commands specifying whether or not an input record is to be included in, or omitted from the sort on the basis of field contents. Both commands must not appear in the same paragraph.

SUM specifies a field whose value is to be added together for a series of records with identical key values. The sum of these values will be stored in the record that will be retained for output. However, overflow during summation is not detected.

ARRANGE permits the user to select certain fields from the input record to be omitted, or to be placed elsewhere.

Remember that if an incorrect record is encountered in the input file (particularly for variable length records), the above option will not always suffice. Sometimes, INVREC should be used (see the SORT Statement in Chapter 3 and the GSORT Statement in Chapter 4).

Unless the INVREC parameter group is specified in the JCL statement SORT, any fields referenced by INCLUDE or OMIT commands must be present in Sort input records, and any fields referenced by KEY[S], SUM or ARRANGE commands must be present in included Sort input records.

Introduction of the the 61 Rule in the context of Year 2000 necessitates modification of the syntax of the DSL command. There are new record field types for the commands KEY/KEYS/SUM/INCLUDE/OMIT. The new types are PDECY, UDECY, DISPY and COMPY.

For PDECY, UDECY and DISPY types, the record fields have a length of 2 bytes holding two digits that represent the four-digit years. For the COMPY type, the record fields have a length of 1 byte holding 2 packed decimal digits.

In all cases sorting is in accordance with the 61 Rule and any modification of the LOWYEAR value in the FUNCTION paragraph.

### 5.3.1 KEY[S] Command

Every sort requires the specification of at least one key field. Up to 64 key fields may be specified by means of the KEY[S] command. The maximum length of a key field is 256 bytes. The format of the key-element in this command is as follows:

```
pos length [dtype] [RV]
```

In the key-element, pos is the byte position of the first byte of the key field relative to the first data byte of the record, and length is the length in bytes of the key field. The numbering of data bytes in a record starts at 1. For example, the numbers 1,3 refer to the field consisting of the first three data bytes of a record. They can be separated by a comma or a space. The dtype entry is one of the data type keywords given in Table 5-1. When not specified, the default data type is CHAR. The RV parameter specifies that the contents of this particular key field should be ordered in the opposite sequence from that prevailing for the sort as a whole.

**Table 5-1. Data Types**

<b>Keyword</b>	<b>Meaning</b>
UBIN	Unsigned binary data (2- or 4-byte)
SBIN	Signed fixed binary data (2- or 4-byte)
CHAR	Character string (maximum 256 characters)
UDEC	Unpacked decimal data (maximum 31 decimal digits and sign)
PDEC *	Packed decimal data (maximum 31 decimal digits and sign packed into 16 bytes). The length is given in bytes (maximum 16)
FBIN	Normalized floating point signed hexadecimal data (4-, 8- or 16-byte)
HEXA	Character string (maximum 256 characters), irrespective of whether COLLATE was specified
DISP	Unsigned unpacked decimal data (up to 32 decimal digits).
UDECY	Unpacked decimal (2 decimal digits) to be compared according to the 61 Rule and LOWYEAR
PDECY	Packed decimal (2 bytes) to be compared according to the 61 Rule and LOWYEAR
DISPY	Unsigned unpacked decimal (2 decimal digits) to be compared according to the 61 Rule and LOWYEAR
COMPY	Unsigned packed decimal (1 byte for 2 digits) to be compared according to the 61 Rule and LOWYEAR
* If sign non-existent (COBOL), use CHAR	

Figure 5-2 shows the use of the KEY[S] command to declare two fields in a user record as key fields for sorting. The states of the file before and after sorting are also shown.

RECORD: KEYS = 4 1 5 1												
Input File					Output File							
Record Fields					Record Fields					Key Field Values		
	A	B	C	D	E		A	B	C	D	E	
Rec1	3	1	X	X	9	Rec6	8	2	X	X	4	X4
Rec2	7	2	X	Y	6	Rec3	6	1	X	X	5	X5
Rec3	6	1	X	X	5	Rec7	8	3	X	X	6	X6
Rec4	4	1	X	Z	1	Rec1	3	1	X	X	9	X9
Rec5	7	1	X	Y	3	Rec5	7	1	X	Y	3	Y3
Rec6	8	2	X	X	4	Rec2	7	2	X	Y	6	Y6
Rec7	8	3	X	X	6	Rec4	4	1	X	Z	1	Z1

**Figure 5-2. Example of KEY[S] Command**

The file consists of seven records, labelled Rec1 to Rec7 for the purpose of the example. Each record is five bytes long and contains five 1-byte fields. The KEY[S] command declares fields 4 and 5, in that order as the sorting key fields. The output file is ordered as if Sort has sorted the records according to the content of field 4, and then for a given value of field 4 further sorted the records on the basis of the content of field 5, wherever necessary.



### 5.3.2 INCLUDE/OMIT Commands

The INCLUDE and OMIT commands establish record selection criteria for the sort by testing fields in the input record. To test a value the user writes up to 16 condition elements, a combination of operators and operands, which yield a value of TRUE or FALSE. The details of constructing condition elements are described below. If they are omitted, variable length records need not contain all key/arrange/sum fields, but if they are included, they must contain all key/arrange/sum fields.

### 5.3.3 Condition Elements

Unless INVREC = CONTINUE is specified as a parameter of the JCL statement SORT, all the input records must be long enough to hold all the fields specified by the INCLUDE and/or OMIT commands.

A condition element consists of an expression that specifies a test to be made. Two types of expression may appear in a condition element:

- a "comparison operation" which uses a comparison operator to compare the field value against another specified value;
- a "numeric condition" which tests the field for a positive, negative or zero value.

#### Comparison Operation

A condition element consisting of a comparison operator has one of the following forms:

field comparison:                pos length operator pos length [dtype]

literal comparison:            pos length operator [dtype] 'literal'.

The field comparison compares one field with another. The literal comparison compares a field of a specific data type with a literal value.

The values of dtype (data type) and their respective keywords can be found in Table 5-1.

Pos is the byte position of the first byte of the field to be compared relative to the first data byte of the record, and length is the length in bytes of that field. The numbering of data bytes in a record starts at 1. For example the numbers 1, 3 refer to the field consisting of the first three data bytes of a record. They can be separated by a comma or a space. The operator is one of the comparison operators shown in Table 5-2.

**Table 5-2. Example of KEY[S] Command**

Operation	True if
a EQ b	a is equal to b
a NE b	a is not equal to b
a LT b	a is less than b
a LE b	a is less than or equal to b
a GT b	a is greater than b
a GE b	a is greater than or equal to b

A literal operand is a character string delimited by apostrophes as shown in the preceding format. If the apostrophe (') is to be included as a character within a string literal, it must be duplicated. For example, the literal 'can't' specifies the string can't.

With the exception of FBIN, each data type listed in Table 5-1 can be used as a literal.

If a literal string of digits is to be compared against a field with a numeric data type, then the literal is converted to the appropriate numeric data type using, if necessary, right justification and leading zeros to equalize the length. If the literal is to be compared with a character field, the user must ensure that the length of the literal is equal to the length of the comparison field, by adding blanks as padding characters in the correct positions, if necessary.

The sample INCLUDE command shown below contains one condition element.

```
INCLUDE = 1,1 EQ 2,1 /*FIRST BYTE = SECOND BYTE*/
```

Here, two fields are being compared to see if they are equal. The fields are one byte long and are bytes 1 and 2 of the record. The operator EQ indicates that the test is for equality. If bytes 1 and 2 are equal, the condition element yields the value TRUE and the record will be included in the sort. If the same condition element were used after an OMIT, yielding the same value, the record would be omitted from the sort.

A single INCLUDE or OMIT command may contain more than one condition element, specifying several tests to be carried out. The parameter AND is used as a condition element separator, as shown below:

```
OMIT = 1,1 EQ 2,1 AND 3,3 EQ 10,3
```

When multiple condition elements, separated by AND, are present in a statement, records will be selected (for omission or inclusion as appropriate) only if all the condition elements yield a TRUE value. Thus in an OMIT command, all condition elements being TRUE would cause the record to be omitted.

Alternatives for record selection can be established by using a sequence of INCLUDE commands or a sequence of OMIT commands. When a sequence of INCLUDE or OMIT commands occurs in a RECORD paragraph, the commands are processed in the order in which they appear.

Processing of a sequence is terminated as soon as a complete command is found to be TRUE; the remaining commands in the sequence are then ignored.

The example below illustrates multiple INCLUDE commands in the same paragraph:

```
INCLUDE 1,5 EQ 6,5 AND 1,1 EQ 10,1
INCLUDE 1,5 EQ 6,5 AND 1,1 NE 10,1
```

The first INCLUDE command performs two tests. First the 5-byte fields beginning at bytes 1 and 6, respectively, are compared. Let us assume they are equal. Byte 1 is then compared to byte 10. If they are equal, the conditions of the command have all been met and the record is included; the second INCLUDE command will be ignored, and will have no effect on the selection of the record. If bytes 1 and 10 are not equal however, the second INCLUDE command will be processed and will cause the record to be included in the sort input record stream.

Figure 5-3 shows more sample condition elements using comparison operators. The numbers indicate the values in the field and not necessarily their internal representation.

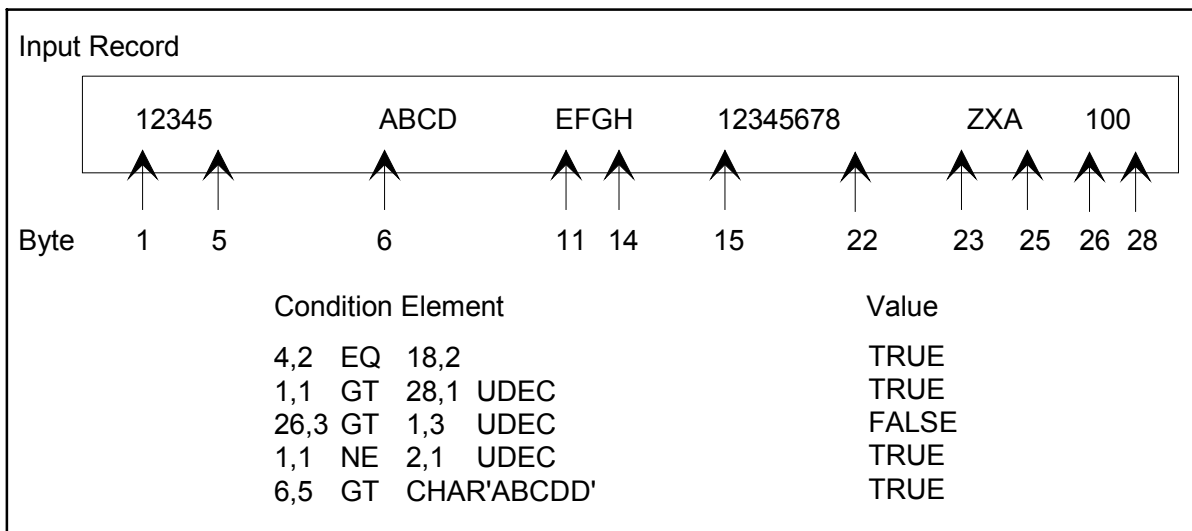


Figure 5-3. Using Comparison Operators

**Numeric Conditions**

A condition element consisting of a numeric expression has the following general form:

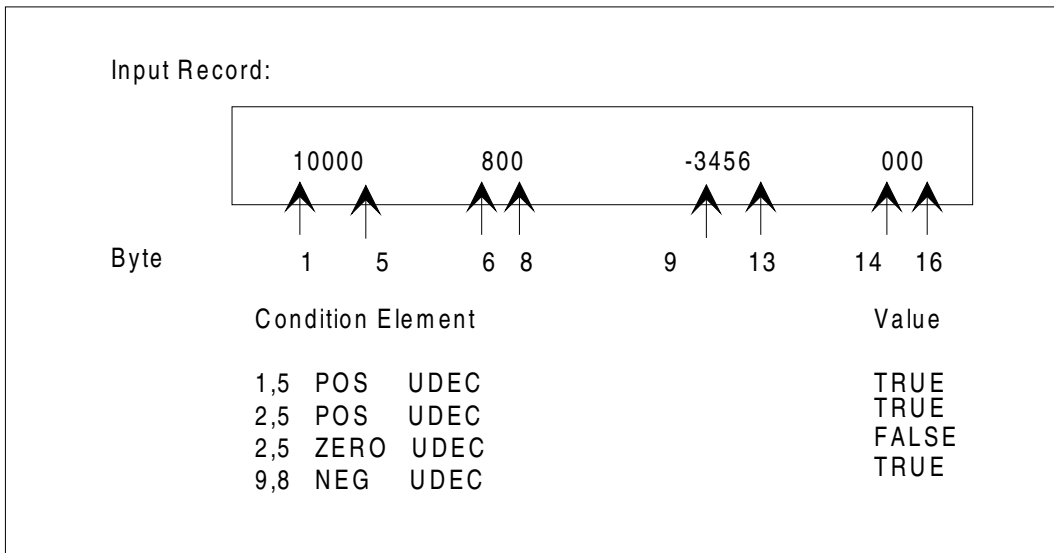
pos length operator dtype

Pos and length are the same as for the comparison operations. The operator here is one of the three numeric condition operators shown below:

POS                      Positive non-zero value  
 NEG                     Negative non-zero value  
 ZERO                    Zero

The permitted values of dtype (data type) are SBIN, UDEC, PDEC and FBIN. They are described in Table 5-1.

Sample condition elements using numeric condition operators are shown in Figure 5-4. Note that the field values are not the true internal numeric representations. Algebraic representations are used for illustrative purposes.



**Figure 5-4. Using Numeric Condition Operators**

### 5.3.4 Example of INCLUDE/OMIT Commands

Figure 5-5 shows the contents of the first 5 records in a file and the results of using certain INCLUDE and OMIT commands with this file.

Rec1	<table border="1" style="border-collapse: collapse; width: 100%;"> <tr> <td style="width: 20%;">G</td> <td style="width: 20%;">H</td> <td style="width: 20%;">I</td> <td style="width: 20%;">J</td> <td style="width: 20%;">K</td> </tr> </table>	G	H	I	J	K	<p>1. Include all records whose first byte is a G</p> <p style="padding-left: 20px;">RECORD : INCLUDE = 1,1 EQ 'G'</p> <p style="padding-left: 20px;">Records selected: All but Rec4.</p>
G	H	I	J	K			
Rec2	<table border="1" style="border-collapse: collapse; width: 100%;"> <tr> <td style="width: 20%;">G</td> <td style="width: 20%;">H</td> <td style="width: 20%;">I</td> <td style="width: 20%;">J</td> <td style="width: 20%;">L</td> </tr> </table>	G	H	I	J	L	<p>2. Omit all records whose first byte is not G.</p> <p style="padding-left: 20px;">RECORD : OMIT = 1,1 NE 'G'</p> <p style="padding-left: 20px;">Records selected: All but Rec4.</p>
G	H	I	J	L			
Rec3	<table border="1" style="border-collapse: collapse; width: 100%;"> <tr> <td style="width: 20%;">G</td> <td style="width: 20%;">H</td> <td style="width: 20%;">I</td> <td style="width: 20%;">J</td> <td style="width: 20%;">M</td> </tr> </table>	G	H	I	J	M	<p>3. Include all records whose first byte is G and whose second byte is I</p> <p style="padding-left: 20px;">RECORD : INCLUDE = 1,1 EQ 'G' AND 2,1 EQ 'I'</p> <p style="padding-left: 20px;">Records selected: Rec5.</p>
G	H	I	J	M			
Rec4	<table border="1" style="border-collapse: collapse; width: 100%;"> <tr> <td style="width: 20%;">F</td> <td style="width: 20%;">H</td> <td style="width: 20%;">I</td> <td style="width: 20%;">J</td> <td style="width: 20%;">N</td> </tr> </table>	F	H	I	J	N	
F	H	I	J	N			
Rec5	<table border="1" style="border-collapse: collapse; width: 100%;"> <tr> <td style="width: 20%;">G</td> <td style="width: 20%;">I</td> <td style="width: 20%;">F</td> <td style="width: 20%;">F</td> <td style="width: 20%;">F</td> </tr> </table>	G	I	F	F	F	
G	I	F	F	F			

**Figure 5-5. Example of INCLUDE/OMIT Commands**

### 5.3.5 SUM Command

The SUM command is used to accumulate the values in one or more specified fields for all the records of the same key. SUM is not valid for HFAS files. The fields to be summed are specified in a sum-element which has the following format

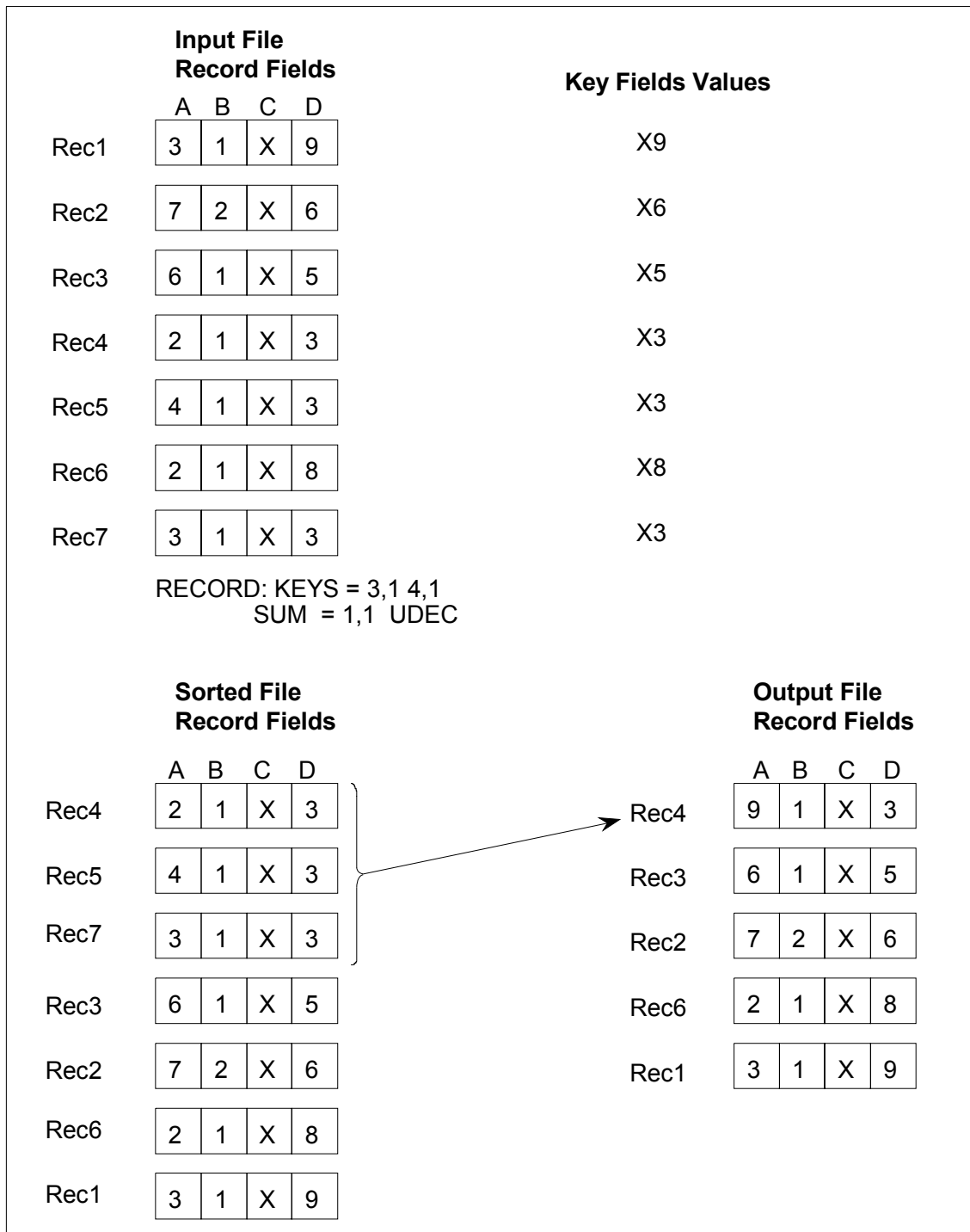
```
pos length dtype
```

POS and length are explained under "Comparison Operations" above. The permitted values of dtype (data type) are SBIN, UDEC, PDEC, FBIN, UDECY, PDECY, COMPY and DISPY. They are described in Table 5-1. Up to 16 fields may be specified for a record; fields must not overlap, and a key field or part of a key field must not be specified for summation.

The SUM command applies only to a group of records with identical key values. Its effect is to accumulate a total of data values for the specified fields throughout the group of duplicate keyed records and to store this total in the record which will be retained for output. Overflow is not detected. The selection of this record is governed by the DUPREC command. The DELETE command must not be specified in the same run as SUM since the deletion of duplicate keyed records is automatic. The selection is done the same way as when the DELETE command is used (see above *DELETE Command*).

Other restrictions on the selection of fields for summation are that each field must be one that will appear on the output record (that is, SUM is incompatible with the ADDRROUT and KEYADDR parameters of the OUTPUT command); and that every sum field must appear as an arrange sub-element in an ARRANGE command if the ARRANGE command is used. Within this context, the sum field must be specified exactly as it was initially; it may appear within the boundaries of a larger field but sum fields must not overlap each other or overlap key fields. A portion of a sum field must not be specified as an ARRANGE element. If a sum field is duplicated in the output record by the ARRANGE command, then summation occurs only for the first occurrence of that field in the output record. The remaining occurrences of that field will have an indeterminate value.

In the example shown in Figure 5-6, three record values are summed into one record for output. The original file and key fields are shown first and then the sorted file, first without summation and deletion and then as it would actually be output.



**Figure 5-6. Example of SUM Command**

In the output file, field A or Rec4 contains the summed values for the three occurrences of records with the identical key field values X and 3. Note that field A in other records is unaffected.

### 5.3.6 ARRANGE Command

The ARRANGE command is used to define the structure of the output record, if it is to be different from that of the input record. The structure is defined by listing all the fields of the record in the sequence in which they are to appear in the output record. Each field is defined in the list by means of an arrange element, which has the following format:

```
(pos, length [,dtype,] [, 'literal'])
```

FBIN-type literals are not permitted.

These types of definition are explained under "Comparison Operations" above. Sort structures each output record to conform to the structure defined in the ARRANGE command, and then writes the record in the output file. Fields not specified in the command are omitted. Note that the arrange-elements in this command refer to the position of the field in the input record; the position of the field in the output record is determined by the order of the arrange-elements in the command. The maximum "length" of an arrange-element is 256.

The ARRANGE command may contain a maximum of 128 arrange-elements. A field may be repeated in the output record by repeating the corresponding arrange-element at the appropriate positions in the command. The limit of 128 on the number of entries includes all the repetitions, and also key fields, irrespective of whether or not the key fields are included in the ARRANGE command. Fields that have been defined as sum fields in the SUM command must appear in the ARRANGE command, either in their entirety or as subfields of other fields. ARRANGE can be used only when OUTPUT is DATA or ADDATA.



## DSL for Sort

A sample RECORD paragraph containing an ARRANGE command is shown in Figure 5-7. Also shown are the input file and the output file.

RECORD: KEYS = 1,1 RV				
INCLUDE = 2,1 EQ 'Y' AND 1,1 LT '7'				
ARRANGE = (3,2), (2,1), (1,1)				
<b>Input File</b>				
<b>Record Fields</b>				
	A	B	C	D
Rec1	4	Y	10	Z
Rec2	3	X	15	W
Rec3	7	Y	24	Z
Rec4	5	Y	14	W
Rec5	1	Y	11	Z
<b>Output File</b>				
<b>Record Fields</b>				
	C	B	A	
Rec4	14	Y	5	
Rec1	10	Y	4	
Rec5	11	Y	1	

**Figure 5-7. Example of the ARRANGE Command**

Field D in Figure 5-7 has been omitted. Field B has retained its original place, and fields A and C have been rearranged. The records are ordered in descending sequence of values in field A, the reverse of the default sequence.

Sort/Merge User's Guide

Table 5-3 gives all the data types which can be used in the RECORD paragraph of the command which executes the sort. An S in the square indicates that this type is possible. (S = Sort).

**Table 5-3. Possible Data Types in RECORD Paragraph (Sort)**

	UBIN	SBIN	CHAR	UDEC UDECY	PDEC PDEPY	FBIN	HEXA	DISP DISPY	COMPY
KEYS	S	S	S	S	S	S	S	S	S
INCLUDE/OMIT Comparison of fields	S	S	S	S	S	S	S	S	S
INCLUDE/OMIT Literal	S	S	S	S	S		S	S	S
INCLUDE/OMIT Numeric Condition in a field		S		S	S	S		S	S
SUM		S		S	S	S		S	
ARRANGE Literal	S	S	S	S	S		S	S	S

### 5.3.7 Examples of RECORD Paragraph

#### **Example 1:**

```
RECORD: INCLUDE = (40,4 EQ CHAR 'MAIN') AND (30,2 LT 32,2 SBIN)
        KEY      = (10,5 CHAR)          (15,3 PDEC RV)
        SUM      = (25,5 PDEC)          (50,4 SBIN)
```

INCLUDE specifies that the records to be included in the sort must meet the following criteria:

- The character string MAIN must appear in bytes 40 through 43.
- The fixed binary value in bytes 30 and 31 must be less than the fixed binary value in bytes 32 and 33.

KEY nominates the following fields:

- Major key is an alphanumeric character string in bytes 10 through 14.
- Minor key is a packed decimal value in bytes 15 through 17 which is to be sorted in reverse order to the sequence specified for the output file.

SUM establishes the following two sum fields:

- Five-byte packed decimal field beginning at byte 25.
- Four-byte binary field beginning at byte 50.

#### **Example 2:**

```
RECORD: KEY   = (8,2 SBIN RV) (15,5 CHAR)
        OMIT  = (1,7 NEG PDEC)
        OMIT  = (1,7 ZERO PDEC)
```

The effects of the RECORD paragraph commands shown above are as follows.

OMIT is specified twice, establishing alternatives for record exclusion; first, if field 1,7 is negative, the record is excluded. Otherwise, the second OMIT is processed and if field 1,7 is zero, the field is excluded. If neither condition is true (that is, field 1,7 is positive), the record is not excluded from the sort.

KEY establishes the following two key fields:

- Major key is a fixed binary value in bytes 8 through 9 to be sorted in reverse order from the order prevailing for the sort as a whole.
- Minor key is the 5-character field beginning at byte 15.

## 5.4 END PARAGRAPH

The END paragraph must be the last item in any DSL command file. It consists solely of the keyword

END:

There are no commands associated with it.

## 6. DSL for Merge

The DSL (Data Services Language) commands for Merge are described in this chapter. These DSL commands apply to Merges requested via the JCL statements MERGE (described in Chapter 3) and GMERGE (described in Chapter 4) and the GCL command MERGE\_FILE (described in Appendix E).

As for Sort, the DSL commands for Merge are provided via the COMFILE parameter. They are organized into FUNCTION and RECORD paragraphs, and are terminated by the paragraph END. The commands within these paragraphs are similar to those used for Sort.

The input to Merge consists of up to eight files, all of which must have identical values for the parameter RECSIZE. The files must all have been ordered on the basis of an identical key field or set of key fields and in the same order (ascending or descending).

One of these input files, specified by the INFILE1 parameter group in the MERGE statement, is treated during the merge as the primary input file; all other input files will be referred to as secondary files in the following description. The difference between primary and secondary files will be explained later in this chapter.

The output file produced by Merge is ordered in the same sequence (i.e. ascending or descending) based on the same set of key fields as the primary file.

## 6.1 FUNCTION PARAGRAPH

The FUNCTION paragraph describes overall characteristics of the merge. All of its commands are optional and the user may leave out the paragraph altogether if the associated default values serve his purpose. If present, the FUNCTION paragraph must precede the RECORD paragraph.

The general form of a Merge FUNCTION paragraph is shown below.

```
FUNCTION: [DESCEND] [DELETE]
          [COLLATE={EBCDIC|G100|H200|ASCII|'hexa512'}]
          [DEBUG]
          [LOWYEAR= digit2)]
```

### 6.1.1 DESCEND Command

When present, the DESCEND command declares that the input files are in descending order and that the output file will also be in descending order.

### 6.1.2 DELETE Command

When present, the DELETE command specifies that multiple occurrences of records with the same key value in the input files will cause all but one of the records to be excluded from the output file.

### 6.1.3 COLLATE Command

The COLLATE command specifies the collating sequence to be used. The collating sequence specified for MERGE/GMERGE must be the same as that used in the preceding SORT/GSORT (used to sort the files). For more details, see the description of COLLATE in Chapter 5.

#### **6.1.4 DEBUG Command**

This is the same as for SORT/GSORT. Merge CPU/ELAPSED times are not increased since input record sequence control is mandatory. If DEBUG is specified together with DUMP=DATA (in the STEPOPT parameter group of the JCL statement MERGE/GMERGE), a dump will be produced if the Merge aborts.

#### **6.1.5 LOWYEAR Command**

The keyword "LOWYEAR" (century LOWest YEAR) denotes a year in the range 0 - 99 which is to be the start year of a 100-year window.

For example, if under the SORT or MERGE commands you specify in the FUNCTION paragraph that LOWYEAR = 61, then the decimal fields PDECY, UDECY, DISPY and COMPY are sorted in the ascending sequence (61, 62...98, 99, 00, 01...59, 60).

The value 0 is authorised and gives PDECY, UDECY, DISPY, and COMPY types the same ordering sequence as PDEC, UDEC, DISP and HEXA types.

## 6.2 RECORD PARAGRAPH

The RECORD paragraph for MERGE/GMERGE is the same as that for SORT/GSORT. The commands are interpreted largely as for SORT/GSORT with the following differences.

KEY[S] is used to specify key fields. Since it is assumed that the primary merge file is in the order desired for output, the key fields are used for reference when merging. That is, a secondary merge file record with the key value A will be placed in the output file contiguous to the primary record, or the set of records whose key field value is A. Minor key fields are likewise used to merge secondary records in the appropriate position within the file.

INCLUDE and OMIT work exactly as for SORT/GSORT but are applied to all the input files.

The SUM command as applied to the merge is based on the primary-file-first order of precedence for retaining one of a group of records with the same key field values. Thus, the sum of the specified field will automatically be stored in the record which is to be retained. If a SUM statement is present then DELETE must not be specified in an associated FUNCTION paragraph.

The ARRANGE command works as for Sort.

If an input record is too short for all the fields referenced in the INCLUDE or OMIT commands, or if an included input record is too short for all the fields referenced by the KEY[S], ARRANGE, and SUM commands, the Merge step will abort with the relevant error messages.

Introduction of the the 61 Rule in the context of Year 2000 necessitates modification of the syntax of the DSL command. There are new record field types for the commands KEY/KEYS/SUM/INCLUDE/OMIT. The new types are PDECY, UDECY, DISPY and COMPY.

For PDECY, UDECY and DISPY types, the record fields have a length of 2 bytes holding two digits that represent the four-digit years. For the COMPY type, the record fields have a length of 1 byte holding 2 packed decimal digits.

In all cases sorting is in accordance with the 61 Rule and any modification of the LOWYEAR value in the FUNCTION paragraph.



DSL for Merge

An example of the RECORD paragraph used in a merge of two files is given in Figure 6-1.

RECORD : KEYS = (1, 1) (2, 1)

<b>INFILE1 (Primary File)</b>	<b>INFILE2</b>	<b>Output FILE</b>	
Record Fields	Record Fields	Record Fields	Key Fields
A 1 2	A 2 1	A 1 2	A 1
A 1 3	A 3 5	A 1 3	A 1
A 2 5	A 7 4	A 2 1	A 2
B 0 0	A 8 0	A 2 5	A 2
B 1 0	B 0 1	A 3 5	A 3
		A 7 4	A 7
		A 8 0	A 8
		B 0 0	B 0
		B 0 1	B 0
		B 1 0	B 1

**Figure 6-1. Example of RECORD Paragraph**

Sort/Merge User's Guide

Table 6-1 gives all the data types which can be used in the RECORD paragraph of the command which executes a Merge of several files. An M in the square indicates that this type is possible.

**Table 6-1. Possible Data Types in RECORD Paragraph**

	UBIN	SBIN	CHAR	UDEC UDECY	PDEC PDECY	FBIN	HEXA	DISP DISPY	COMPY
<b>KEYS</b>	M	M	M	M	M			M	M
<b>INCLUDE/OMIT Comparison of fields</b>	M	M	M	M	M			M	M
<b>INCLUDE/OMIT Literal</b>	M	M	M	M	M			M	M
<b>INCLUDE/OMIT Numeric condition in a field</b>		M		M	M			M	M
<b>SUM</b>		M		M	M			M	
<b>ARRANGE Literal</b>	M	M	M	M	M		M	M	M

### **6.3 END PARAGRAPH**

The END paragraph must be the last item in any DSL command file. It consists solely of the keyword:

END:

There are no commands associated with it.

## Sort/Merge User's Guide

## 7. Sort Execution

### 7.1 SORTING VARIABLE LENGTH RECORDS

The following factors affect the efficiency of a variable length record sort:

- the maximum record length (specified in the RECSIZE parameter or in the file label);
- the minimum record length (may be specified in the MIN\* command);
- the average record length (may be specified in the AVERAGE\* command).

The record lengths specified by the user in the RECSIZE parameter and in the MIN or AVERAGE commands influence the way in which records are stored in memory by Sort. The organization of records in memory has an important effect upon performance. Therefore, it is important that the values chosen for RECSIZE and MIN or AVERAGE be as accurate as possible.

Variable length records are stored in memory in a work area which is divided up into "cells". A cell can store no more than one record, though a record can span more than one cell. Obviously, if the length of a cell is 400 bytes and the average record length is 100 bytes, then 300 bytes will normally be unused in each cell. The way in which the cell length is calculated from RECSIZE and MIN or AVERAGE is described below. If MIN is not specified the minimum record length is equal to the shortest record referred to by the KEY or SUM commands.

\* See Chapter 5, FUNCTION paragraph

Use of the AVERAGE command is strongly recommended. If it is not used the work file can in some cases be much bigger than the file to be sorted.

The cell length is calculated as a function of (AVERAGE, MIN, RECSIZE) which minimizes the space lost on the workfile.

If AVERAGE is not specified, it is calculated as:

$$(MIN + RECSIZE) / 2$$

For variable format the value of AVERAGE must therefore be fixed with care in order to optimize work file space. The workfile space actually used is shown in the Sort Report.

For example, if the sort report indicates:

```
WORKFILE USED IN % OF INFILE : 110
```

it means that the space used on the workfile is 10% more than the INFILE size. The best AVERAGE value will give the lower % of INFILE.

## 7.2 SORT RECORD SIZE

Whether or not it spans more than one cell, a record can be significantly modified by the OUTPUT, ARRANGE, etc. commands.

Let the input record size be  $i$  bytes and the record size after sort be  $s$  bytes. Then  $s$  is calculated as follows:

1. If OUTPUT = DATA, DUPREC = NONE and no ARRANGE elements are specified:  
 $s = i$ .
2. If OUTPUT = DATA, DUPREC = FIFO and no ARRANGE elements are specified:  
 $s = i+4$  bytes.
3. If OUTPUT = DATA, DUPREC = NONE and ARRANGE is used, there are two possibilities:
  - a) If all key fields are contained within the ARRANGE specification:  
 $s = \text{sum-of-the-sizes-of-the-ARRANGE-fields}$ .
  - b) If some key fields are not present in the ARRANGE fields:  
 $s = \text{sum-of-ARRANGE} + \text{sum-of-KEY-fields-not-present-in-ARRANGE}$ .

**For example:**

RECORD: KEYS = (10, 4 30, 6) ARRANGE = (4, 5 20, 3 10, 4)

$$s = \begin{array}{c} (5+3+4) \\ | \\ \text{ARRANGE elements} \end{array} + \begin{array}{c} (6) \\ | \\ \text{excluded key} \end{array}$$

Therefore  $s = 18$

If DUPREC = FIFO, add four bytes to the value of  $s$ .

4. If OUTPUT = ADDATA and DUPREC = NONE, add 5 to the value of  $s$  obtained in step 1, 2 or 3.
5. If OUTPUT = ADDRROUT or KEYADDR,  $s = \text{sum-of-KEY-fields}+5$   
If DUPREC = FIFO,  $s = \text{sum-of-KEY-fields}+5+4$

**For example:**

FUNCTION: OUTPUT = ADDRROUT DUPREC = NONE  
RECORD: KEYS = (10, 7 19, 15 80, 20)

$s = (7+15+20)+5 = 47$

The size of  $s$  determines the size of the work file required. (See paragraph 7.4.)

### 7.3 MAXIMUM RECORD SIZE

A record written to the work file is restricted to the smaller of the following values:

$2048 * NS + 1022$  (NS = number of 1024-byte sectors per track.)

or

$31 * 1024 - 10$

In fact, these maximum values can only occur under certain circumstances which depend on the values given for the Sort parameters.

However, once a record is larger than 10K bytes it is more efficient to execute the sort on the basis of:

keys + record addresses

and then to recreate the sorted file by means of direct access to records. This avoids the need to restrict the record size.

## 7.4 SIZE OF DISK WORK FILE

It is not possible to estimate the size of the disk work file entirely accurately because the user would personally have to do all the calculations executed by the sort program in order to use the allocated disk space. These calculations take a large number of factors into account.

***For example:***

- The choice of block size used in the work file, and therefore the losses on a given track, are determined by the main memory allocated.
- The losses in a given block are determined by the type of disk.
- It is not possible to foresee the number of strings created. At the end of each string a block is written without its necessarily being filled with records. This means that even the best result that can be expected will still be an overestimate.
- For internal reasons the sort program makes justifications at the beginning of each track.
- For variable format, the way records are split into cells depends on MIN, AVERAGE, MAX, etc. Thus, the way records are split depends on the various record sizes in the file.

Many other factors are also involved. It is therefore recommended to use, as a basis:

1. A very simplified formula for estimating the size of the work file.
2. Some additional hints.
3. In all subsequent sorts, the appropriate information given in the Sort Report.

The sort program gives the following assistance for the use of the work file:

1. Extension of the work file as necessary and if an increment was specified either in PREALLOC or in ALLOCATE (limited to a certain number of extents).
2. Sort Report information:
  - Either the percentage of allocated space actually used at the end of a normal sort:

***For example:***

```
% OF WORK FILE SPACE USED: 87
```

- Or the percentage of allocated space which would have been needed to allow the sort to be completed:



## Sort Execution

### **For example:**

MANDATORY % OF ALLOCATED WKDISK: 121

### **Estimation of Work File Size**

The work file size can be estimated by the following formula:

$$\text{WORK} = 1.2 * F * \frac{(1 + 48 * (2C + 6))}{M} \frac{s * \text{CKPT}}{i * T * \text{NCT}}$$

This estimate should be rounded up to the next whole cylinder.

### **NOTE:** Cylinders and tracks:

in VBO = physical, in FBO = logical.

A logical cylinder has a size of 1000 KBytes and contains 20 logical tracks.

See Table 7-1 for the relationship between physical size and disk model.

Meaning of abbreviations:

WORK = Size of work file in cylinders. (See Note 1 above.)

F = Size of input file in bytes.

M = Size of memory necessary for sort (in bytes):  
total memory allocated:

$2 * \max(\text{input buffer size, output buffer size})$

Note that this means twice the input buffer size, or twice the output buffer size, whichever is the maximum.

i = Size of input record (in bytes).

s = Size of output record (in bytes).

C = Total size of all key items (in bytes).

T = Normal size of disk track used for work file (in bytes). (See Note 1 above.)

NCT = Number of tracks per cylinder on disk used for work file. (See Note 1 above.)

CKPT = 1.2 where checkpoint is requested, otherwise 1.

For variable format, s is the size (in bytes) of the average number of cells spanned by one record.

Each cell contains:

- Key items (size C).
- 7 bytes for:
  - record number (4 bytes)
  - cell number (1 byte)
  - record size (2 bytes).

The rest of the cell contains "non-key" bytes of the record which is to be split.

Each record occupies a whole number of cells, depending on the size chosen for the cell and the size of the record itself.

Once the user knows how the cell size is chosen and used (see above), he can estimate the amount of space needed by the file to be sorted after division into cells.

In this case the size of the work file can be adapted by means of the AVERAGE command which determines the cell length (see Chapter 5, AVERAGE Command).

The estimate for workfile size proposed above is even more reliable in an average situation, i.e. if:

- Allocated memory is more than 512K bytes.
- The record is less than 1 Kbytes.
- The resulting number of cylinders obtained for the work file size is considerably larger than 1.

For variable format, the distribution of record sizes is known and is specified to the sort program.

It is advisable, at least when performing a sort for the first time on a given file, to allocate more disk space than estimated, especially if the estimate is low (e.g. 1 cylinder).

For subsequent (e.g. periodic) runs the appropriate Sort Report information, if any, may be used in order to allocate a smaller work file.

Note that such information is only valid if the main parameters are not changed (same input file, of course; same memory size, record size, etc.).

Sort Execution

**Table 7-1. Disk Capacities**

Disk Type (Device Class)	VBO disks			FBO disks* (on Fixed Sector Architecture: FSA)
	MS/D500 (1)	MS/M500 (1)	MS/B10	MS/FSA
Cylinders per volume	707	838	1730	320, 650, 1330, 2100 logical cylinders
Extra cylinders for back-up tracks	4	5	5	
Tracks per cylinder	2 x 24	2 x 20	15	20 logical tracks per logical cylinder
Total number of tracks (excl. back-up tracks)	33936	33520	25950	6400, 13000, 26600, 42000
Bytes per track	29344	19069	36381	51200 (50 KBytes)
Bytes per cylinder	704254	762760	36381	51200 (50 KBytes)
Total capacity (M bytes)	2 x 498	2 x 639	1000	320, 650, 1330, 2100

(1)Dual-spindle unit

\* The physical data storage unit on FBO/FSA disks is the datablock with a fixed size of 512 bytes - supported on DPS 7000/An.

## 7.5 ESTIMATING MEMORY SIZE FOR A "MEMORY SORT"

To estimate the memory needed to sort a file without a workfile (i.e., in memory), you must first add 10 bytes to the reccsize or cellsize (see below).

If the input records are of variable length, and/or if there are ARRANGE or FIFO commands, or special data type keys (different from CHAR), you must first calculate the sort record size (see paragraph 7.2). You must estimate the number of "cells", because one input record may span several cells. Then, you add 10 bytes per cell.

The memory for an "in memory sort" must be greater than:

$$\text{number of cells} * (\text{size of a cell} + 10)$$

In the simplest case (fixed length input records, no ARRANGE or FIFO commands, no special data type keys), the memory must be greater than:

$$\text{number of input records} * (\text{reccsize} + 10)$$

Note that INCLUDE and/or OMIT commands may change the number of records actually sorted.

To give a margin of security for a memory sort, you should add a few percent (10% say) to the estimated memory calculated as described above.

## 7.6 NOTES ON PERFORMANCE

1. Give sufficient memory to the Sort (SIZE option).

If no BPB is used, the following table gives the recommended memory size as a function of the size of the file to be sorted:

Size of the file to be sorted	18 Mb	100 Mb	460 Mb	1 Gb
Size of the Memory allocated to the sort	512 Kb	1 Mb	2 Mb	4 Mb

In case of a multi-process sort, also apply the following:  
 Size of Memory  $\geq$  NBSORT \* 512Kbytes

2. To the memory determined from the above table, add the memory necessary for the buffers (taking the higher of the INFILE and OUTFILE values):  
 Size of Buffers = 2 \* BPB \* Blocksize
3. File placement: avoid placing the input and output files on the same pack as the disk work files.
4. A multi-process sort is justified in the following 3 cases:
  - performance is I/O bound (by I/Os on the single workfile),
  - performance is CPU bound and more than one CPU is available,
  - you want to do a "memory" sort for a file of more than 64 Megabytes.

The performance limitation by I/O throughput (on the single workfile) occurs mainly if BPB is applied to INFILE and/or OUTFILE. In fact, the workfile is handled with blocks limited to 32Kbytes. Using BPB, you can read/write 32\*Blocksize bytes in a single operation (on INFILE and/or OUTFILE).

5. A multi-process sort may be less efficient than a mono-process sort in the following cases:
  - where there are only a small number of records to be sorted (generally under 10 minutes of elapsed time),
  - where the input file(s) are to be read by the UFAS access method (see Appendix I *Sort/Merge and the UFAS Access Method* for details of the files concerned) unless the number of records to be sorted is substantial (generally over 10 minutes of elapsed time).

See also Appendix I *Sort/Merge and the UFAS Access Method* for the impact of BPB.

6. The NBSORT workfiles must each be on a different disk.
7. Using checkpoint facilities involves slight performance degradation.
8. Use the SORTSIZE command. An exact value will produce a well tailored sort. In case of doubt it is better to supply a value that is too big than no value at all.

## Sort/Merge User's Guide

9. Do not be concerned at the "missing page" condition. When using a large memory the initial access on each page does imply one missing page, but with no loss in elapsed or CPU time.

## **7.7 CHECKPOINT/RESTART**

If the REPEAT parameter is specified in the JCL statement SORT, checkpoints will be taken during the course of the sort (only for a mono-process sort) with a fixed frequency corresponding to 20M bytes written to or read from the work file (depending on the phase of the sort).

For a multi-process sort, the REPEAT parameter will allow a restart only from the beginning of the step.

In the event of error, the operator can decide whether or not to restart Sort from the last checkpoint.

## 7.8 CONCURRENT ACCESS (GAC)

The SORT and MERGE statements can specify concurrent access options (GAC) on its input and output files.

In fact, the step can use files under GAC (ASSIGN statement with SHARE = MONITOR) in only two ways:

- ACCESS=SPREAD or SPWRITE (input or output file),
- READLOCK=STAT (input file only).

Except for the special case of OUTFILE = INFILE, the input file can be explicitly specified with either of the following sets of options (on INFILE):

```
SHARE=MONITOR, READLOCK={NORMAL|EXCL}, ACCESS={SPREAD|SPWRITE}
```

```
SHARE = MONITOR, READLOCK = STAT
```

(Statistical read not taking into account any modifications made to the file during concurrent access but after the actual read).

For an **output file** under GAC (including the case of OUTFILE = INFILE), the only possible combination is:

```
SHARE = MONITOR, ACCESS = SPWRITE
```

It must be included in the OUTFILE description, or in the INFILE description if OUTFILE = INFILE, (in which case the default is ACCESS = SPWRITE).

### IMPORTANT

1. Unless the user wants a statistical read or unless a file is catalogued with SHARE = MONITOR, it is strongly recommended to assign the files involved in the sort with SHARE = NORMAL (default value). The sharing conditions will be the same, but at lower cost.
2. If you want to use SORT with concatenation and with GAC, you should specify ACCESS = SPREAD/SPWRITE with INFILE assign or READLOCK=STAT in defining SORT IF = ... . Without this type of specification, it is mandatory to use REPEAT and to begin the concatenation with the files that are monitored by GAC (SHARE = MONITOR), otherwise the sort processing will abort.

See the *GAC-EXTENDED User's Guide*.



## A. Sort and Merge Error Messages

This Appendix discusses the error messages which are output in the Job Occurrence Report and in the Sort/Merge Report. The Sort/Merge Report is produced in a SYSOUT subfile (see Appendix B).

The diagnostics which can appear in the Job Occurrence Report are caused by syntax or inconsistency errors in the JCL statements: SORT, GSORT, MERGE, and GMERGE. Most of the messages are self-explanatory. Note also that after the expansion of an extended JCL statement, errors may later be found in the resulting basic JCL statement (such as ASSIGN). These errors in basic JCL are described in the JCL Reference Manual.

Execution diagnostics appear in the Sort/Merge Report. They are the result of errors detected in the DSL commands or errors encountered whilst processing the input or output files.

Sort and Merge also set the severity value that can be tested with the JUMP JCL statement. The possible values are:

- |       |  |
|-------|--|
| SEV0: | the function has been correctly executed.  |
| SEV1: | the function has been correctly executed. However a warning is given to the user who should carefully examine the report. For example, an empty output file may have been created. This situation would be normal for Sort if, for example, the input file was empty. But the specification of an empty input file may have been the result of a user error. |
| SEV3: | (Sort only) the function has not been completed, due to an I/O error. If the REPEAT parameter was used in the SORT statement, the function can be restarted (see <i>Checkpoint/Restart</i> Chapter 7).   |
| SEV4: | the function has not been completed, due to a user error or a system error whose origin cannot be precisely diagnosed.   |

**A.1 ERRORS MESSAGES FOR THE SORT STATEMENT**

Below is a list of messages that can be produced in the Job Occurrence Report as a result of errors in the JCL statement SORT. These messages do not apply to the GSORT statement.

Error Number	Text and explanation
106	<p>CONTRADICTIONARY VALUES IN LABEL AND FILEFORM PARAMETERS</p> <p>Verify both parameters. An HFAS fileform implies an H200 type label, a standard label implies a native fileform.</p>
	<p>CONTRADICTIONARY VALUES IN HALT AND START PARAMETERS</p> <p>Verify both values. The value given for HALT must be greater than or equal to the value given for START.</p>
	<p>CONTRADICTIONARY VALUES IN ABORT AND CONTINUE PARAMETERS</p> <p>In INVREC, ABORT and CONTINUE are mutually exclusive.</p>
	<p>ERROR IN OUTFILE PARAMETER; ILLEGAL VALUE OF MEDIA IN CASE OF REPLIN</p> <p>When a disk sort is executed and the input file and the work file are on the same media, the REPLIN parameter is not allowed in OUTFILE because it would cause the work file to be dismounted before the end of the sort.</p>
	<p>ERROR IN COMFILE PARAMETER; ILLEGAL VALUE INPUT-ENCLOSURE-NAME</p> <p>Verify the syntax of the input enclosure name.</p>
	<p>ERROR IN SIZE PARAMETER; ILLEGAL VALUE MUST BE &gt;=22K</p>
	<p>ERROR IN MEMORY PARAMETER; ILLEGAL VALUE MUST LIE BETWEEN 8 AND 120K</p>
	<p>ERROR IN START PARAMETER; ILLEGAL VALUE MUST LIE BETWEEN 1 AND 2 147 483 647</p>
	<p>ERROR IN HALT PARAMETER; ILLEGAL VALUE MUST LIE BETWEEN 1 AND 2 147 483 647</p>
	<p>ERROR IN BUFFER PARAMETER; ILLEGAL VALUE MUST BE &lt; OR = TO 32767</p>

## Sort and Merge Error Messages

### Error Messages for the SORT Statement (Cont'd)

Error Number	Text and explanation
106 (Cont'd)	<p>ERROR IN INFILE PARAMETER; ILLEGAL VALUE REPLIN OR SIZE</p> <p>Neither REPLIN nor SIZE should appear in INFILE description.</p> <p>ERROR IN WKDISK PARAMETER; ILLEGAL VALUE FILESTAT, END, MOUNT, or EXDATE</p> <p>None of these parameters is allowed in WKDISK.</p> <p>ERROR IN WKDISK PARAMETER; ILLEGAL VALUE LABEL OR SORT DEFINITION PAR.</p> <p>Neither LABEL nor sort-definition parameters should appear under WKDISK. The file used as a work file always has native labels.</p> <p>ERROR IN WKDISK PARAMETER; ILLEGAL VALUE SUBFILE</p> <p>SUBFILE is not a valid parameter in WKDISK description.</p> <p>ERROR IN COMFILE PARAMETER; ILLEGAL VALUE MOUNT OR EXPDATE</p> <p>Neither MOUNT nor EXPDATE are valid parameters for the COMFILE description.</p> <p>ERROR IN COMFILE PARAMETER; ILLEGAL VALUE LABEL OR MEDIA NAME TABLE</p> <p>LABEL is not a valid parameter in COMFILE description. The library is assumed to be native. The library cannot be a multivolume file.</p> <p>ERROR IN COMFILE PARAMETER; ILLEGAL VALUE SORT DEFINITION FORBIDDEN</p> <p>Parameters of sort definition are not allowed for COMFILE which must be an already created library.</p> <p>ERROR IN PRTRFILE PARAMETER; ILLEGAL VALUE SORT DEFINITION PARAMETERS</p> <p>The PRTRFILE is either created and there is no need to specify sort-definition parameters, or the PRTRFILE is a tape file to be created by SORT and default values will be used.</p>

**Error Messages for the SORT Statement (Cont'd)**

Error Number	Text and explanation
106 (Cont'd)	<p>ERROR IN WKDISK PARAMETER; ILLEGAL VALUE DATABUF, BPB OR REPLIN</p> <p>None of these parameters is allowed in WKDISK.</p> <p>ERROR IN COMFILE PARAMETER; ILLEGAL VALUE DATABUF, BPB OR REPLIN</p> <p>None of these parameters is allowed in COMFILE.</p> <p>ERROR IN PRTFILE PARAMETER; ILLEGAL VALUE DATABUF, BPB, REPLIN OR SIZE</p> <p>None of these parameters is allowed in PRTFILE.</p> <p>ERROR IN DATBUF PARAMETER; ILLEGAL VALUE MUST BE EQUAL TO 1 OR 2</p> <p>ERROR IN BPB PARAMETER; ILLEGAL VALUE MUST BE GREATER OR EQUAL TO 1</p> <p>ERROR IN LOGFILE PARAMETER; ILLEGAL VALUE DATABUF, BPB OR REPLIN</p>
108	<p>ERROR IN DEVCLASS PARAMETER CONTRADICTIONARY VALUES IN INPUT AND OUTPUT FILES</p> <p>When REPLIN is used in OUTFILE, the input and output file must be of the same device-class</p>
110	<p>CONTRADICTIONARY PARAMETERS RESIDENT, DEVCLASS</p> <p>When a resident volume is described, the device-class should not be specified.</p>
111	<p>MEDIA PARAMETER MAY APPEAR ONLY IF DEVCLASS EXPLICITLY GIVEN</p> <p>No default value is supplied for DEVCLASS. So when the volume is not resident, DEVCLASS and MEDIA must be specified together.</p>

## Sort and Merge Error Messages

### Error Messages for the SORT Statement (Cont'd)

Error Number	Text and explanation
112	<p>ERROR IN NOBSN PARAMETER : NOBSN PARAMETER : NOBSN MAY NOT APPEAR IN THE PRESENT CONTEXT OF DISK FILE</p> <p>The use of NOBSN is restricted to tape files...</p> <p>ERROR IN WKDISK PARAMETER : SIZE MAY NOT APPEAR IN THE PRESENT CONTEXT (EFN ALREADY DEFINED)</p> <p>External-file name and SIZE are mutually exclusive in WKDISK</p> <p>ERROR IN COMFILE PARAMETER : DEVCLASS MAY NOT APPEAR IN THE PRESENT CONTEXT WITHOUT MEDIA</p>
113	<p>MANDATORY PARAMETER RECSIZE MISSING</p> <p>MANDATORY PARAMETER BLKSIZE MISSING</p> <p>MANDATORY PARAMETER DEVCLASS MISSING</p> <p>MANDATORY PARAMETER SUBFILE MISSING</p> <p>MANDATORY PARAMETER INFILE MISSING</p> <p>MANDATORY PARAMETER OUTFILE MISSING</p> <p>MANDATORY PARAMETER COMFILE MISSING</p> <p>MANDATORY PARAMETER EFN MISSING</p> <p>EFN stands for external file name</p> <p>MANDATORY PARAMETER LOGFILE MISSING</p>
126	<p>CONTRADICTIONARY VALUES : MEMORY, BUFFER-SIZE, DATABUF, BPB</p> <p>(MEMORY, BUFFER) form a group of exclusive parameters from the group (SIZE,DATABUF,BPB).</p>

**A.2 ERROR MESSAGES FOR THE MERGE STATEMENT**

These messages do not apply to the GMERGE statement.

Error Number	Text and explanation
106	<p>CONTRADICTIONARY VALUES IN LABEL AND FILEFORM PARAMETERS</p> <p>Verify both parameters. An HFAS fileform implies an H200 type label, a standard label implies a native fileform.</p> <p>CONTRADICTIONARY VALUES IN SIZE AND FILESTAT PARAMETERS</p>
107	<p>ERROR IN COMFILE PARAMETER; ILLEGAL VALUE INPUT-ENCLOSURE-NAME</p> <p>Verify the syntax of the input-enclosure-name.</p> <p>ERROR IN COMFILE PARAMETER; ILLEGAL VALUE OF MOUNT OR EXPDATE</p> <p>Neither MOUNT nor EXPDATE are allowed for COMFILE.</p> <p>ERROR IN COMFILE PARAMETER; ILLEGAL VALUE LABEL OR MEDIA NAME TABLE</p> <p>LABEL is not a valid parameter in COMFILE description. The library is assumed to be native. The library cannot be a multivolume file.</p> <p>ERROR IN COMFILE PARAMETER; ILLEGAL VALUE MERGE DEFINITION FORBIDDEN</p> <p>Parameters of merge definition are not allowed for COMFILE which must be an existing library.</p> <p>ERROR IN PRTFILE PARAMETER; ILLEGAL VALUE MERGE DEFINITION FORBIDDEN</p> <p>Either the PRTFILE exists, and there is no need to specify merge definition parameters, or the PRTFILE is a tape file to be created by Merge and default values will be used.</p> <p>ERROR IN COMFILE PARAMETER; ILLEGAL VALUE DATABUF, BPB OR SIZE</p> <p>None of these parameters is allowed in COMFILE.</p> <p>ERROR IN PRTFILE PARAMETER; ILLEGAL VALUE DATABUF, BPB OR SIZE</p> <p>None of these parameters is allowed in PRTFILE</p>

## Sort and Merge Error Messages

### Error Messages for the MERGE Statement (Cont'd)

Error Number	Text and explanation
107 (Cont'd)	ERROR IN DATBUF PARAMETER; ILLEGAL VALUE MUST BE EQUAL TO 1 OR 2  ERROR IN BPB PARAMETER; ILLEGAL VALUE MUST BE GREATER OR EQUAL TO 1
110	CONTRADICTORY PARAMETERS RESIDENT, DEVCLASS  When a resident volume is described, the device-class should not be specified.
111	MEDIA PARAMETER MAY APPEAR ONLY IF DEVCLASS EXPLICITLY GIVEN
112	ERROR IN NOBSN PARAMETER : NOBSN MAY NOT APPEAR IN THE PRESENT CONTEXT OF DISK FILE  NOBSN is meaningful only for tape files.  ERROR IN COMFILE PARAMETER : DEVCLASS MAY NOT APPEAR IN THE PRESENT CONTEXT WITHOUT MEDIA
113	MANDATORY PARAMETER RECSIZE MISSING  MANDATORY PARAMETER BLOCKSZ MISSING  MANDATORY PARAMETER SUBFILE MISSING  MANDATORY PARAMETER H200FORM MISSING  MANDATORY PARAMETER OUTFILE MISSING  MANDATORY PARAMETER COMFILE MISSING
126	ERROR IN INPUT FILE; ILLEGAL VALUE SIZE  The SIZE parameter cannot appear in an input file description since the file already exists.

### A.3 EXECUTION ERROR MESSAGES FOR SORT/GSORT AND MERGE/GMERGE

These messages apply to the SORT, GSORT, MERGE, and GMERGE statements.

The execution error messages are produced as part of the Sort Report in a SYSOUT subfile (see Appendix B); The general format is as follows:

```
*** TDXX.YY: error-message
           [return-code]
           ["secondary information"]
```

XX = number of procedure giving rise to error.

YY = error code within the procedure.

error-message is a phrase that describes the diagnosis or error.

return-code appears either with the abnormal execution of a system primitive or when there is an internal sort error.

"secondary information" appears when there is an I/O error on the work file. It allows the user to identify the completion code, the device in use, the address of the error, presented as: Cylinder/Track.

#### **Example:**

```
**** SORT EXECUTION LISTING ****
*** TD11.18: I/O ERROR ON WORK FILE: WKDISK-READ
           RC = 5F8B0481 --> TRID11, IOFAIL
           COMPLETION CODE=8104, DEVICE NAME=MS12, ADDRESS=380
**** SORT AUDIT INFORMATION ****
```

In this example: XX = 11 (procedure), YY = 18 followed by a return-code (RC) and ending with secondary information (COMPLETION CODE, DEVICE, ADDRESS).

#### **A.3.1 Invalid Records**

For records with an invalid length the following message is displayed:

```
INVALID RECORD cause: record number
```

The cause may be INCLUDE/OMIT or KEY/ARRANGE/SUM depending on the unsatisfied criteria. Record-number is the record serial number maintained by Sort.

Sort counts the records when reading the input file starting at 1 for the first record.



### A.3.2 List of Execution Error Messages

The execution error messages for SORT/GSORT and MERGE/GMERGE, along with a description of the circumstances which cause them are given below.

TD00.00	DSORT INVALID MESSAGE XX YYYY
Meaning:	XX = module number (hexa digits 2) YYYY = error number (hexa digits 4) An error occurred that caused the module number or the error number to be incorrect. The problem must be reported to Technical Support.
TDXX.YY	INTERNAL SORT ERROR
	XX = Module number (hexa digits 2) where the internal error has been detected. YY = Number (hexa digits 2) of the internal error detected.
Meaning:	The problem must be reported to Technical Support.
TDXX.17	UNABLE TO LAUNCH CHANNEL PROGRAM XX = Module number where the error has been detected.
Meaning:	The problem must be reported to Technical Support.
TDXX.18	I/O ERROR ON WORK FILE : WKDISK-WRITE XX = Module number where the error has been detected.
Meaning:	Detailed information (completion code, involved address) is given below.
TDXX.19	KEY/ARRANGE/SUM FIELD(S) OUT OF INPUT RECORD
	XX = Module number where the error has been detected.
Meaning:	The involved record is too small to hold all the indicated fields. The user program must be corrected.
TDXX.20	OVER 2147483647 INPUT RECORDS
	XX = Module number where the error has been detected.
Meaning:	The number of records exceeds the capacity of 31 bits (more than $2^{31}-1$ records).
TDXX.21	ATTEMPT TO REQUEST A SORTED RECORD AFTER END OF SORT-OUTPUT
Meaning:	Self-explanatory. The user program (Sort output procedure) must be changed ; or problem in system processor must be reported to Technical Support of this system processor.

**List of Execution Error Messages (Cont'd)**

TD01.01	INVALID SCA
Meaning:	SCA version is neither 0 (standard value) nor 1 (BMICK REL/RET). See SH_DCSCA. The user program must be corrected.
TD01.02	INVALID SRTDEF
Meaning:	The SRTDEF is not correct. Details are given by the messages TD01.65 to TD01.81 messages. The user program must be corrected.
TD01.03	INVALID SRTDEF
Meaning:	The option string (see JCL statement STEP) cannot be interpreted. JCL must be corrected.
TD01.04	KEY NUMBER MUST BE 1 TO 64
Meaning:	The number of key fields is greater than 64. The program must be corrected.
TD01.05	INVALID KEY TABLE
Meaning:	The key definition table (pointed to by SRTDEF) is wrong. Details are given by messages TD01.82 to 86. The user program must be corrected.
TD01.06	UNABLE TO CREATE SEGMENT
Meaning:	The main segment (record segment) used for sorting cannot be created. A return code from H_SGCR is given. The error could be internal to Sort but most frequently : either too much memory is required for Sort (SIZE in JCL, SRTDEF) or the number of vacant entries is insufficient (linkage problem).
TD01.07	UNABLE TO CREATE SEGMENT
Meaning:	The common sort segment used for sorting cannot be created. A return code from H_SGCR is given. The error could be internal to Sort but most frequently : either too much memory is required for Sort (SIZE in JCL, SRTDEF) or the number of vacant entries is insufficient (linkage problem).

## Sort and Merge Error Messages

### List of Execution Error Messages (Cont'd)

TD01.08	UNABLE TO CREATE SEGMENT
Meaning:	The generated code segment use for sorting cannot be created. A return code from H_SGCR is given. The error could be internal to Sort but most frequently : either too much memory is required for Sort (SIZE in JCL, SRTDEF) or the number of vacant entries is insufficient (linkage problem).
TD01.09	UNABLE TO CREATE WORK FILE DEFINITION
Meaning:	A dynamic FD (file definition table) is required and cannot be created. A return code from H_CRFD is given. It could be a linkage problem (ENTRYOV).
TD01.10	UNABLE TO DELETE WORK FILE DEFINITION
Meaning:	Sort is deleting useless FD (file definition table). The return code from H_DLFD is given. The problem must be reported to Technical Support.
TD01.11	UNABLE TO IDENTIFY WORK FILE
Meaning:	Sort is unable to know which volume supports the workfile (\$H_RTVLID). The return code from H_RTVLID is given.
TD01.12	INVALID RUN.TIME : SCB
Meaning:	The SCB table in the run time module does not begin with the characters "SCB1". Check for the link. The user program must be corrected.
TD01.13	INVALID RUN.TIME : SCR
Meaning:	The SCR table in the run time module does not begin with the characters "CURR". Check for the link. The user program must be corrected.
TD01.14	UNABLE TO CHANGE SEGMENT SIZE
Meaning:	The main segment used for sorting cannot be adjusted to the desired size. A return code from H_SGSIZE is given. Generally caused by the SIZE parameter (JCL,SRTDEF).
TD01.15	UNABLE TO CHANGE SEGMENT SIZE
Meaning:	The common Sort segment used for sorting cannot be adjusted to the desired size. A return code from H_SGSIZE is given. Generally caused by the SIZE parameter (JCL,SRTDEF).

**List of Execution Error Messages (Cont'd)**

TD01.16	UNABLE TO CHANGE SEGMENT SIZE
Meaning:	The generated code segment used for sorting cannot be adjusted to the desired size. A return code from H_SGFSIZE is given. Generally caused by the SIZE parameter (JCL,SRTDEF).
TD01.17	INVALID SORT CALL NESTING
Meaning:	Several nested Sorts have tried to use the same run time static resources. The user program must be corrected.
TD01.18	UNABLE TO RETRIEVE SORT RUN-TIME RESOURCES
Meaning:	An abnormal return-code from H_SGFSIZE is given when trying to identify a run-time segment. Possible user linkage error or the problem must be reported to Technical Support.
TD01.19	UNABLE TO CHANGE SEGMENT SIZE
Meaning:	The segment for an old COBOL key comparison routine cannot be adjusted to the desired size. A return code from H_SGFSIZE is given. Generally caused by the SIZE parameter (JCL,SRTDEF).
TD01.20	ERROR IN COBOL-COMPILER/SORT INTERFACE
Meaning:	Self-explanatory. The problem must be reported to Technical Support.
TD01.21	NO TAPE-WORK-FILE SUBROUTINE-SORT AVAILABLE
Meaning:	Self-explanatory. JCL must be modified (\$SORTWORK) to supply a disk work file.
TD01.22	PARALLEL SORT IS NOT AVAILABLE IN TDS CONTEXT
Meaning:	Self-explanatory.
TD01.65	SRTDEF : NO VALID KEY SPECIFICATION
Meaning:	The user program must be corrected.
TD01.66	SRTDEF : INVALID COBOL_COMP
Meaning:	The user program must be corrected.
TD01.67	SRTDEF : INVALID LOCAL_PTR
Meaning:	The user program must be corrected.

## Sort and Merge Error Messages

### List of Execution Error Messages (Cont'd)

TD01.68	SRTDEF : INVALID CODE_LENGTH
Meaning:	The user program must be corrected.
TD01.69	SRTDEF : ALL FILLER FIELDS ARE NOT 0
Meaning:	The user program must be corrected.
TD01.70	SRTDEF : SIZE IS NEGATIVE
Meaning:	The user program must be corrected.
TD01.71	SRTDEF : NO_WORK_FILES MUST BE -1 TO +255
Meaning:	The user program must be corrected.
TD01.72	SRTDEF : IN_VOLUME IS NEGATIVE
Meaning:	The user program must be corrected.
TD01.73	SRTDEF : INPUT RECORD LENGTH IS NOT > 0
Meaning:	The user program must be corrected.
TD01.74	SRTDEF : REPORT IS > "01"B
Meaning:	The user program must be corrected.
TD01.75	SRTDEF : COLL_SEQ IS NOT 1
Meaning:	The user program must be corrected.
TD01.76	SRTDEF : DUP_REC_SEQ MUST BE ZERO OR "F"
Meaning:	The user program must be corrected.
TD01.77	SRTDEF : DUP_REC_DEL IS NOT ZERO
Meaning:	The user program must be corrected.
TD01.78	SRTDEF : IN_FD IS NOT NULL
Meaning:	The user program must be corrected.
TD01.79	SRTDEF : OUT_FD IS NOT NULL
Meaning:	The user program must be corrected.
TD01.80	SRTDEF : INVALID KEY_PTR
Meaning:	The user program must be corrected.

**List of Execution Error Messages (Cont'd)**

TD01.81	KEY # XXXXX : INVALID POSITION
Meaning:	The user program must be corrected. XXXXX = rank of the incorrect field.
TD01.82	KEY # XXXXX : INVALID SENSE
Meaning:	The user program must be corrected. XXXXX = rank of the incorrect field.
TD01.83	KEY # XXXXX : LENGTH IS NOT POSITIVE
Meaning:	The user program must be corrected. XXXXX = rank of the incorrect field.
TD01.84	KEY # XXXXX : INVALID TYPE
Meaning:	The user program must be corrected. XXXXX = rank of the incorrect field.
TD01.85	KEY # XXXXX : INVALID LENGTH FOR TYPE
Meaning:	The user program must be corrected. XXXXX = rank of the incorrect field.
TD01.86	SRTDEF : INVALID COBOL KEY COLLATING SEQUENCE
Meaning:	The user program must be corrected.
TD01.87	RECOMPILE AND RELINK COBOL PROGRAM IF YOU WANT UP-TO-DATE DISK-SORT
Meaning:	The merge Sort is used instead of the distribution Sort in this case. This is only a warning.
TD02.01	ABOVE CHARACTER ALREADY MENTIONED IN USER COLLATING SEQUENCE
Meaning:	The collating sequence is incorrect. The sort DSL must be corrected.
TD02.02	SEPARATOR MISSING
Meaning:	The option string or DSL (Data Services Language) must be corrected. Each error is underlined by a "?".
TD02.03	"=" MISSING AFTER PRECEDING KEY_WORD
Meaning:	The option string or DSL (Data Services Language) must be corrected. Each error is underlined by a "?".

## Sort and Merge Error Messages

### List of Execution Error Messages (Cont'd)

TD02.04	FUNCTION PARAGRAPH MISSING
Meaning:	The DSL (Data Services Language) must be corrected. Each error is underlined by a "?".
TD02.05	RECORD PARAGRAPH MISSING
Meaning:	The DSL (Data Services Language) must be corrected. Each error is underlined by a "?".
TD02.06	ABOVE KEY-WORD IS OUT OF CONTEXT IN FUNCTION PARAGRAPH. RECORD PARAGRAPH INSERTED
Meaning:	The DSL (Data Services Language) must be corrected. Each error is underlined by a "?".
TD02.07	ABOVE KEY-WORD IS OUT OF CONTEXT IN RECORD PARAGRAPH. FUNCTION PARAGRAPH INSERTED
Meaning:	The DSL (Data Services Language) must be corrected. Each error is underlined by a "?".
TD02.08	ILLEGAL CHARACTER(S) IGNORED
Meaning:	The option string or DSL (Data Services Language) must be corrected. Each error is underlined by a "?".
TD02.09	NUMERIC VALUE OUT OF BINARY WORD VALUE LIMITS
Meaning:	The option string or DSL (Data Services Language) must be corrected. Each error is underlined by a "?".
TD02.10	LITERAL > 256 INTERNAL CHARACTERS
Meaning:	The option string or DSL (Data Services Language) must be corrected. Each error is underlined by a "?".
TD02.11	ZERO LENGTH LITERAL
Meaning:	The option string or DSL (Data Services Language) must be corrected. Each error is underlined by a "?".
TD02.12	INVALID LENGTH FOR COLLATE SEQUENCE
Meaning:	The option string or DSL (Data Services Language) must be corrected. Each error is underlined by a "?".

**List of Execution Error Messages (Cont'd)**

TD02.13	> 1024 CHARACTERS FOR RECORD PARAGRAPH LITERALS
Meaning:	The option string or DSL (Data Services Language) must be corrected. Each error is underlined by a "?".
TD02.14	ODD LENGTH FOR HEXADECIMAL LITERAL
Meaning:	The option string or DSL (Data Services Language) must be corrected. Each error is underlined by a "?".
TD02.15	INVALID OR UNEXPECTED LITERAL
Meaning:	The option string or DSL (Data Services Language) must be corrected. Each error is underlined by a "?".
TD02.16	MULTIPLE USE OF KEYWORD
Meaning:	The option string or DSL (Data Services Language) must be corrected. Each error is underlined by a "?".
TD02.17	UNEXPECTED END OF OPTION STRING LITERAL
Meaning:	The option string or DSL (Data Services Language) must be corrected. Each error is underlined by a "?".
TD02.18	UNEXPECTED END OF DSL
Meaning:	The option string or DSL (Data Services Language) must be corrected. Each error is underlined by a "?".
TD02.19	INVALID NUMERIC VALUE
Meaning:	The option string or DSL (Data Services Language) must be corrected. Each error is underlined by a "?".
TD02.20	ILLEGAL ZERO VALUE
Meaning:	The option string or DSL (Data Services Language) must be corrected. Each error is underlined by a "?".
TD02.21	PARAMETER VALUE > MAXIMUM ALLOWED
Meaning:	The option string or DSL (Data Services Language) must be corrected. Each error is underlined by a "?".
TD02.22	KEYWORD VALUE IS NOT ALPHANUMERIC
Meaning:	The option string or DSL (Data Services Language) must be corrected. Each error is underlined by a "?".



## Sort and Merge Error Messages

### List of Execution Error Messages (Cont'd)

TD02.23	UNKNOWN KEYWORD OR OUT OF CONTEXT
Meaning:	The option string or DSL (Data Services Language) must be corrected. Each error is underlined by a "?".
TD02.24	INVALID FIELD POSITION
Meaning:	The option string or DSL (Data Services Language) must be corrected. Each error is underlined by a "?".
TD02.25	INVALID FIELD LENGTH
Meaning:	The option string or DSL (Data Services Language) must be corrected. Each error is underlined by a "?".
TD02.26	INVALID FIELD LENGTH FOR FIELD TYPE
Meaning:	The option string or DSL (Data Services Language) must be corrected. Each error is underlined by a "?".
TD02.27	2ND CHAR FIELD LENGTH NOT = 1ST LENGTH
Meaning:	The option string or DSL (Data Services Language) must be corrected. Each error is underlined by a "?".
TD02.28	UNEXPECTED LITERAL
Meaning:	The option string or DSL (Data Services Language) must be corrected. Each error is underlined by a "?".
TD02.29	UNEXPECTED NUMERIC VALUE
Meaning:	The option string or DSL (Data Services Language) must be corrected. Each error is underlined by a "?".
TD02.30	UNEXPECTED ALPHANUMERIC VALUE
Meaning:	The option string or DSL (Data Services Language) must be corrected. Each error is underlined by a "?".
TD02.31	WKTAPE CANNOT BE SUPPORTED BY DSORT
Meaning:	The option string or DSL (Data Services Language) must be corrected. Each error is underlined by a "?".
TD02.32	OUTPUT DATA MUST BE (DEFAULT) FOR H200 FILES
Meaning:	self-explanatory

**List of Execution Error Messages (Cont'd)**

TD02.33	INVALID COLLATE SPECIFICATION
Meaning:	The option string or DSL (Data Services Language) must be corrected. Each error is underlined by a "?".
TD02.34	COLLATE SEQUENCE FOR H200 FILE MUST GIVE 64 CHARACTERS
Meaning:	self-explanatory
TD02.35	FUNCTION PARAGRAPH MUST APPEAR 1ST AND ONCE
Meaning:	The option string or DSL (Data Services Language) must be corrected. Each error is underlined by a "?".
TD02.36	ARRANGE/SUM IMPOSSIBLE WITH OUTPUT = KEYADDR/ADDRROUT
Meaning:	The option string or DSL (Data Services Language) must be corrected. Each error is underlined by a "?".
TD02.37	SUM AND DELETE CANNOT BE USED TOGETHER
Meaning:	The option string or DSL (Data Services Language) must be corrected. Each error is underlined by a "?".
TD02.38	INCLUDE AND OMIT CANNOT BE USED TOGETHER
Meaning:	The option string or DSL (Data Services Language) must be corrected. Each error is underlined by a "?".
TD02.39	MULTIPLE RECORD PARAGRAPH
Meaning:	The option string or DSL (Data Services Language) must be corrected. Each error is underlined by a "?".
TD02.40	OVER 64 KEYS SPECIFIED
Meaning:	The option string or DSL (Data Services Language) must be corrected. Each error is underlined by a "?".
TD02.41	OVER 16 SUM FIELDS SPECIFIED
Meaning:	self-explanatory
TD02.42	OVER 128 ARRANGE FIELDS SPECIFIED
Meaning:	The option string or DSL (Data Services Language) must be corrected. Each error is underlined by a "?".

## Sort and Merge Error Messages

### List of Execution Error Messages (Cont'd)

TD02.43	OVER 16 INCLUDE/OMIT CONDITIONS SPECIFIED
Meaning:	The option string or DSL (Data Services Language) must be corrected. Each error is underlined by a "?".
TD02.44	INVALID FIELD SPECIFICATION
Meaning:	The option string or DSL (Data Services Language) must be corrected. Each error is underlined by a "?".
TD02.45	FIELD LENGTH MISSING
Meaning:	The option string or DSL (Data Services Language) must be corrected. Each error is underlined by a "?".
TD02.46	ILLEGAL FIELD TYPE
Meaning:	The option string or DSL (Data Services Language) must be corrected. Each error is underlined by a "?".
TD02.47	FIELD TYPE MISSING
Meaning:	The option string or DSL (Data Services Language) must be corrected. Each error is underlined by a "?".
TD02.48	MULTIPLE LITERAL TYPE
Meaning:	The option string or DSL (Data Services Language) must be corrected. Each error is underlined by a "?".
TD02.49	LITERAL MISSING
Meaning:	The option string or DSL (Data Services Language) must be corrected. Each error is underlined by a "?".
TD02.50	"AND" MISSING
Meaning:	The option string or DSL (Data Services Language) must be corrected. Each error is underlined by a "?".
TD02.51	END PARAGRAPH MUST BE EMPTY OR COMMENT
Meaning:	The option string or DSL (Data Services Language) must be corrected. Each error is underlined by a "?".
TD02.52	LENGTH OF ARRANGED RECORD > 32767
Meaning:	The option string or DSL (Data Services Language) must be corrected. Each error is underlined by a "?".

**List of Execution Error Messages (Cont'd)**

TD02.53	NO ZONE SPECIFIED IN ARRANGE LIST
Meaning:	The option string or DSL (Data Services Language) must be corrected. Each error is underlined by a "?".
TD02.54	OVER 128 ERRORS ON ABOVE LINE
Meaning:	The option string or DSL (Data Services Language) must be corrected. Each error is underlined by a "?".
TD02.55	NO VALID KEY SPECIFICATION OCCURRED
Meaning:	The option string or DSL (Data Services Language) must be corrected. Each error is underlined by a "?".
TD02.56	UNABLE TO CREATE SEGMENT
Meaning:	The main segment (record segment) used for sorting cannot be created. A return code from H_SGCR is given. The error could be internal to Sort but most frequently : either too much memory is required for Sort (SIZE in JCL, SRTDEF) or the number of vacant entries is insufficient (linkage problem).
TD02.57	UNABLE TO OPEN COMFILE
Meaning:	A return code from H_OPEN is given. JCL must be corrected.
TD02.58	UNABLE TO GET DSL RECORD
Meaning:	A return code from H_OPEN is given. The problem must be reported to Technical Support.
TD02.59	UNABLE TO CLOSE COMFILE
Meaning:	A return code from H_OPEN is given. The problem must be reported to Technical Support.
TD02.60	ERRORS IN STATE IOF PARAMETERS
Meaning:	A return code is given from H_STIOF. The problem must be reported to Technical Support.
TD02.61	UNABLE TO IDENTIFY PRTFILE
Meaning:	A return code is given from H_RTVLID. A user JCL error has occurred, or the problem must be reported to Technical Support.

## Sort and Merge Error Messages

### List of Execution Error Messages (Cont'd)

TD02.62	INFILES NUMBER IS < 2 OR > 8
Meaning:	Use INFILE if only one input file; 8 files is the maximum.
TD02.64	SUM FIELD VV {OVERLAPS { SUM FIELD } WW} {                    { KEY FIELD } } { {OUT OF ALL ARRANGE FIELDS } }
Meaning:	DSL must be corrected. VV, WW are the ranks of the fields involved among the sum or key fields.
TD03.01	USER SORT KEYS ARE TOO LONG
Meaning:	The total length of the sum or key fields exceeds 7000 bytes. The user program must be corrected.
TD03.02	USER RECORD IS TOO LONG
Meaning:	The sort record size exceeds 15000 bytes. Either the user record is too long or too many keys overlap. The user program must be corrected.
TD03.03	INSUFFICIENT MEMORY SIZE
Meaning:	The size parameter is too small according to the other parameters. The user program must be corrected.
TD03.04	UNABLE TO CHANGE SEGMENT SIZE
Meaning:	The main segment used for sorting cannot be adjusted to the desired size. A return code from H_SGFSIZE is given. Generally caused by the SIZE parameter (JCL,SRTDEF).
TD03.05	UNABLE TO CREATE SEGMENT
Meaning:	The node segment (record segment) used for sorting cannot be created. A return code H_SGCR is given. The error could be internal to Sort but most frequently: either too much memory is required for Sort (SIZE in JCL,SRTDEF) or the number of vacant entries is insufficient (linkage problem).
TD03.06	UNABLE TO CHANGE SEGMENT SIZE
Meaning:	The node segment used for sorting cannot be adjusted to the desired size. A return code from H_SGFSIZE is given. Generally caused by the SIZE parameter (JCL,SRTDEF).

**List of Execution Error Messages (Cont'd)**

TD03.07	UNABLE TO CHANGE SEGMENT SIZE
Meaning:	The generated code segment used for sorting cannot be adjusted to the desired size. A return code from H_SGFSIZE is given. Generally caused by the SIZE parameter (JCL,SRTDEF).
TD03.08	UNABLE TO MODIFY SEGMENT ATTRIBUTES
Meaning:	The problem must be reported to Technical Support.
TD03.09	UNABLE TO CREATE SEGMENT
Meaning:	The segment used for large COBOL key tables cannot be created. A return code from H_SGCR is given. The error could be internal to Sort but most frequently: either too much memory is required for Sort (SIZE in JCL,SRTDEF) or the number of vacant entries is insufficient (linkage problem).
TD03.10	UNABLE TO CHANGE SEGMENT SIZE
Meaning:	The segment used for large COBOL key tables cannot be adjusted to the desired size. A return code from H_SGFSIZE is given. Generally caused by the SIZE parameter (JCL,SRTDEF).
TD03.11	UNABLE TO CHANGE SEGMENT SIZE
Meaning:	The common sort segment used for sorting cannot be adjusted to the desired size. A return code from H_SGFSIZE is given. Generally caused by the SIZE parameter (JCL,SRTDEF).
TD03.12	UNABLE TO CHANGE SEGMENT SIZE
Meaning:	A record storage area segment cannot be adjusted to the desired size. A return code from H_SGFSIZE is given. Generally caused by the SIZE parameter (JCL,SRTDEF).
TD03.13	UNABLE TO MODIFY SEGMENT ATTRIBUTES
Meaning:	The problem must be reported to Technical Support.
TD03.14	UNABLE TO CREATE SEGMENT
Meaning:	A return code H_SGCR is given. The error could be internal to Sort but most frequently: either too much memory is required for Sort (SIZE in JCL,SRTDEF) or the number of vacant entries is insufficient (linkage problem).

## Sort and Merge Error Messages

### List of Execution Error Messages (Cont'd)

TD03.15	UNABLE TO CREATE SEGMENT
Meaning:	A return code H_SGCR is given. The error could be internal to Sort but most frequently: either too much memory is required for Sort (SIZE in JCL,SRTDEF) or the number of vacant entries is insufficient (linkage problem).
TD03.16	PARALLEL SORT ASKED BUT NOT PURCHASED
Meaning:	Parallel sorts are available if you have the MI for the Extended Sort. If you have purchased the Extended Sort and you get this message, please report the problem to your Service Center.
TD03.17	UNABLE TO CREATE SEGMENT
Meaning:	A return code H_SGCR is given. The error could be internal to Sort but most frequently: either too much memory is required for Sort (SIZE in JCL,SRTDEF) or the number of vacant entries is insufficient (linkage problem).
TD03.18	MEMORY FORCED TO 512K (LIMIT FOR NOT PURCHASED SORT)
Meaning:	For the Basic Sort, the memory limit is 512K bytes. If you want to use more memory during sorts, you must install the Extended Sort (separate MI).
TD04.01	UNABLE TO DEFINE CHANNEL PROGRAM
Meaning:	The problem must be reported to Technical Support.
TD04.02	NO DISK WORK FILE ASSIGNED
Meaning:	NO WORK FILES (SRTDEF) implies an already assigned work file which is not supplied. The user program must be corrected.
TD04.03	UNABLE TO CREATE WORK FILE DEFINITION
Meaning:	A dynamic FD (file definition table) is required and cannot be created. A return code from H_CRFD is given. It could be a linkage problem (ENTRYOV).
TD04.04	UNABLE TO OPEN WORK FILE
Meaning:	Check the work file state and assignment. If these are correct, then the problem must be reported to Technical Support.

**List of Execution Error Messages (Cont'd)**

TD04.05	UNABLE TO RETRIEVE WORK FILE EXTENTS
Meaning:	If RC = BUSY then SHARE = NORMAL. Otherwise the problem must be reported to Technical Support.
TD04.06	MUCH TOO MANY EXTENTS IN WORK FILE
Meaning:	The problem must be reported to Technical Support. Installation's resident disc space should be reorganized.
TD04.07	WORK FILE IS NOT ALLOCATED IN CYLINDERS
Meaning:	The user program must be corrected.
TD04.08	ALL THE WORK FILE VOLUMES ARE NOT PREMOUNTED
Meaning:	JCL must be corrected.
TD04.09	WORK FILE IS NOT ON DISK
Meaning:	JCL must be corrected.
TD04.10	ALL THE WORK FILE EXTENTS ARE NOT IN CYLINDERS
Meaning:	Abnormal file. The problem must be reported to Technical Support.
TD04.11	SEVERAL DISK TYPES FOR WORK FILE
Meaning:	JCL must be corrected. See your Service Center.
TD04.12	INSUFFICIENT MEMORY SIZE
Meaning:	The SIZE parameter is too small in relation to the other parameters. The user program must be corrected.
TD04.13	WORK FILE OVERFLOW
Meaning:	The work file has been extended as much as possible. It is still insufficient to hold the whole input file. Either the work file is actually too short or some incorrect parameter (record size, memory size) has caused an unexpected work file occupancy (e.g. one record per block).
TD04.14	WORK FILE OVERFLOW: CHECK FOR AVERAGE RECORD SIZES
Meaning:	The work file is too small according to the Sort parameters (record size, number of records). The work file must be re-defined.



## Sort and Merge Error Messages

### List of Execution Error Messages (Cont'd)

TD04.15	UNABLE TO CREATE SEGMENT
Meaning:	The buffer segment used for sorting cannot be created. A return code from H_SGCR is given. The error could be internal to Sort but most frequently : either too much memory is required for Sort (SIZE in JCL, SRTDEF) or the number of vacant entries is insufficient (linkage problem).
TD04.16	UNABLE TO CHANGE SEGMENT SIZE
Meaning:	The buffer segment used for sorting cannot be adjusted to the desired size. A return code from H_SGFSIZE is given. Generally caused by the SIZE parameter (JCL, SRTDEF).
TD04.17	UNABLE TO LOCK B SEGMENT
Meaning:	A return code from H_SGLOCK is given. Either SIZE is smaller than the locked memory needed or the problem must be reported to Technical Support.
TD04.18	UNABLE TO CHANGE SEGMENT SIZE
Meaning:	The main segment used for sorting cannot be adjusted to the desired size. A return code from H_SGFSIZE is given. Generally caused by the SIZE parameter (JCL, SRTDEF).
TD04.19	UNABLE TO GET SEMAPHORES
Meaning:	Linkage problem (insufficient semaphore pool). Could be caused by nested sorts.
TD04.20	UNABLE TO CHANGE SEGMENT SIZE
Meaning:	The node segment used for sorting cannot be adjusted to the desired size. A return code from H_SGFSIZE is given. Generally caused by the SIZE parameter (JCL, SRTDEF).
TD04.21	USER SORT KEYS ARE TOO LONG
Meaning:	The total length of the sum or key fields exceeds 7000 bytes. The user program must be corrected.
TD04.22	USER RECORD IS TOO LONG
Meaning:	The sort record size exceeds 15000 bytes. Either the user record is too long or too many keys overlap. The user program must be corrected.
TD04.23	UNABLE TO ALLOCATE DYNAMIC DISK WORK FILE
Meaning:	A return code from H_DYNAL is given

**List of Execution Error Messages (Cont'd)**

TD04.24	UNABLE TO ASSIGN DYNAMIC DISK WORK FILE
Meaning:	A return code from H_DYNASG is given.
TD04.25	UNABLE TO DEASSIGN DYNAMIC DISK WORK FILE
Meaning:	A return code from H_DDASG is given. The problem must be reported to Technical Support.
TD04.26	UNABLE TO CLOSE DYNAMIC DISK WORK FILE
Meaning:	A return code from H_CLOSE is given. The problem must be reported to Technical Support.
TD04.27	MEMORY_SORT OVERFLOW
Meaning:	A memory-sort is required and the memory is too small. The user program must be corrected.
TD04.28	UNABLE TO DELETE WORK FILE DEFINITION
Meaning:	Sort is deleting useless FD (file definition table). A return code from H_DFLD is given. The problem must be reported to Technical Support.
TD04.29	UNABLE TO IDENTIFY WORK FILE
Meaning:	Sort is unable to determine which volume supports the work file (\$H_RTVLID). A return code from H_RTVLID is given.
TD04.30	SORT NESTING OVERFLOW
Meaning:	Too many dynamic resources needed by nested sorts.
TD04.31	SORT MEMORY TOO SHORT FOR WORK FILE FILE-BLOCK SIZE RC = 5F841807 --> TRID 4, LNERR
Meaning:	If you receive this message, refer to the description of WKDISK[S] in Chapter 3 (JCL statement SORT).
TD04.32	INTERNAL SORT ERROR
Meaning:	The problem must be reported to Technical Support.
TD04.33	INTERNAL SORT ERROR
Meaning:	The problem must be reported to Technical Support.

## Sort and Merge Error Messages

### List of Execution Error Messages (Cont'd)

TD04.34	UNABLE TO EXTEND WORK FILE
Meaning:	Self explanatory.
TD04.35	UNABLE TO RETRIEVE WORK FILE DEFINITION
Meaning:	The problem must be reported to Technical Support.
TD04.36	NON SEQUENTIAL FILEORG OF WORKFILE IS RISKY AND SHOULD BE CHANGED
Meaning:	Self-explanatory. JCL should be changed.
TD04.37	UNABLE TO CREATE SEGMENT
Meaning:	The segment used for work file extents description cannot be created. Return code from H_SGCR is given. The error could be internal to Sort but most frequently : either too much memory is required for Sort (SIZE in JCL, SRTDEF) or the vacant entries are insufficient (linkage problem).
TD04.38	UNABLE TO CHANGE SEGMENT SIZE
Meaning:	The segment used for work file extents description cannot be adjusted to the desired size. Return code from H_SGFSIZE is given. Generally caused by the SIZE parameter (JCL, SRTDEF).
TD04.39	WORK FILE ASSIGNED ON FBO OR FSA VOLUMES IS NOT UFAS SEQUENTIAL
Meaning:	Self-explanatory. JCL must be corrected. Return code from H_SGFSIZE is given. Generally caused by the SIZE parameter (JCL,SRTDEF).
TD06.01	UNABLE TO DEFINE CHANNEL PROGRAM
Meaning:	The problem must be reported to Technical Support.
TD06.02	UNABLE TO DELETE CHANNEL PROGRAM
Meaning:	The problem must be reported to Technical Support.
TD06.03	UNABLE TO CHANGE SEGMENT SIZE
Meaning:	The main segment used for sorting cannot be adjusted to the desired size. A return code from H_SGFSIZE is given. Generally caused by the SIZE parameter (JCL,SRTDEF).

**List of Execution Error Messages (Cont'd)**

TD06.04	UNABLE TO CHANGE SEGMENT SIZE
Meaning:	The buffer segment used for sorting cannot be adjusted to the desired size. A return code from H_SGFSIZE is given. Generally caused by the SIZE parameter (JCL,SRTDEF).
TD06.05	INTERNAL SORT ERROR
Meaning:	The problem must be reported to Technical Support.
TD06.06	INTERNAL SORT ERROR
Meaning:	The problem must be reported to Technical Support.
TD06.07	INTERNAL SORT ERROR
Meaning:	The problem must be reported to Technical Support.
TD06.08	UNABLE TO LOCK B_SEGMENT
Meaning:	Return code from H_SGLOCK is given. Either \$SIZE is smaller than the locked memory needed or the problem must be reported to Technical Support.
TD08.01	UNABLE TO DEFINE CHANNEL PROGRAM
Meaning:	The problem must be reported to Technical Support.
TD08.02	UNABLE TO DELETE CHANNEL PROGRAM
Meaning:	The problem must be reported to Technical Support.
TD08.03	UNABLE TO CHANGE SEGMENT SIZE
Meaning:	The main segment used for sorting cannot be adjusted to the desired size. A return code from H_SGFSIZE is given. Generally caused by the SIZE parameter (JCL,SRTDEF).
TD08.04	UNABLE TO CHANGE SEGMENT SIZE
Meaning:	The main segment used for sorting cannot be adjusted to the desired size. A return code from H_SGFSIZE is given. Generally caused by the SIZE parameter (JCL,SRTDEF).
TD08.05	INTERNAL SORT ERROR
Meaning:	The problem must be reported to Technical Support.
TD08.06	INTERNAL SORT ERROR
Meaning:	The problem must be reported to Technical Support.

## Sort and Merge Error Messages

### List of Execution Error Messages (Cont'd)

TD08.07	INTERNAL SORT ERROR
Meaning:	The problem must be reported to Technical Support.
TD08.08	INTERNAL SORT ERROR
Meaning:	The problem must be reported to Technical Support.
TD08.09	UNABLE TO LOCK B_SEGMENT
Meaning:	Return code from H_SGLOCK is given. Either \$SIZE is smaller than the locked memory needed or the problem must be reported to Technical Support.
TD10.01	UNABLE TO DEFINE CHANNEL PROGRAM
Meaning:	The problem must be reported to Technical Support.
TD10.02	UNABLE TO DELETE CHANNEL PROGRAM
Meaning:	The problem must be reported to Technical Support.
TD10.03	UNABLE TO CHANGE SEGMENT SIZE
Meaning:	The main segment used for sorting cannot be adjusted to the desired size. A return code from H_SGFSIZE is given. Generally caused by the SIZE parameter (JCL,SRTDEF).
TD10.04	UNABLE TO CHANGE SEGMENT SIZE
Meaning:	The buffer segment used for sorting cannot be adjusted to the desired size. A return code from H_SGFSIZE is given. Generally caused by the SIZE parameter (JCL,SRTDEF).
TD10.05	INTERNAL SORT ERROR
Meaning:	The problem must be reported to Technical Support.
TD10.06	INTERNAL SORT ERROR
Meaning:	The problem must be reported to Technical Support.
TD10.07	UNABLE TO MODIFY SEGMENT ATTRIBUTES
Meaning:	The problem must be reported to Technical Support.
TD10.08	UNABLE TO MODIFY SEGMENT ATTRIBUTES
Meaning:	The problem must be reported to Technical Support.

**List of Execution Error Messages (Cont'd)**

TD10.09	UNABLE TO CHANGE SEGMENT SIZE
Meaning:	The generated code segment used for sorting cannot be adjusted to the desired size. A return code from H_SGFSIZE is given. Generally caused by the SIZE parameter (JCL,SRTDEF).
TD10.10	UNABLE TO CHANGE SEGMENT SIZE
Meaning:	The node segment used for sorting cannot be adjusted to the desired size. A return code from H_SGFSIZE is given. Generally caused by the SIZE parameter (JCL,SRTDEF).
TD10.11	UNABLE TO CHANGE SEGMENT SIZE
Meaning:	The node segment used for sorting cannot be adjusted to the desired size. A return code from H_SGFSIZE is given. Generally caused by the SIZE parameter (JCL,SRTDEF).
TD10.12	UNABLE TO CHANGE SEGMENT SIZE
Meaning:	A buffer segment used for sorting cannot be adjusted to the desired size. A return code from H_SGFSIZE is given. Generally caused by the SIZE parameter (JCL,SRTDEF).
TD10.13	UNABLE TO LOCK B_SEGMENT SIZE
Meaning:	A return code from H_SGLOCK is given. Either SIZE is smaller than the lock memory needed or the problem must be reported to the Service Center.
TD10.14	UNABLE TO CHANGE SEGMENT SIZE
Meaning:	A work segment used for sorting cannot be adjusted to the desired size. A return code from H_SGFSIZE is given. Generally caused by the SIZE parameter (JCL,SRTDEF).
TD10.15	UNABLE TO CHANGE SEGMENT SIZE
Meaning:	A work segment used for sorting cannot be adjusted to the desired size. A return code from H_SGFSIZE is given. Generally caused by the SIZE parameter (JCL,SRTDEF).
TD10.16	UNABLE TO CHANGE SEGMENT SIZE
Meaning:	A pseudo-buffer segment used for sorting cannot be adjusted to the desired size. A return code from H_SGFSIZE is given. Generally caused by the SIZE parameter (JCL,SRTDEF).

## Sort and Merge Error Messages

### List of Execution Error Messages (Cont'd)

TD10.17	INTERNAL SORT ERROR
Meaning:	The problem must be reported to Technical Support.
TD12.01	UNABLE TO DEFINE CHANNEL PROGRAM
Meaning:	The problem must be reported to Technical Support.
TD12.02	UNABLE TO DELETE CHANNEL PROGRAM
Meaning:	The problem must be reported to Technical Support.
TD12.03	UNABLE TO CHANGE SEGMENT SIZE
Meaning:	The main segment used for sorting cannot be adjusted to the desired size. A return code from H_SGFSIZE is given. Generally caused by the SIZE parameter (JCL,SRTDEF).
TD12.04	UNABLE TO CHANGE SEGMENT SIZE
Meaning:	The node segment used for sorting cannot be adjusted to the desired size. A return code from H_SGFSIZE is given. Generally caused by the SIZE parameter (JCL,SRTDEF).
TD12.05	UNABLE TO CHANGE SEGMENT SIZE
Meaning:	The generated code segment used for sorting cannot be adjusted to the desired size. A return code from H_SGFSIZE is given. Generally caused by the SIZE parameter (JCL,SRTDEF).
TD12.06	UNABLE TO CHANGE SEGMENT SIZE
Meaning:	The buffer segment used for sorting cannot be adjusted to the desired size. A return code from H_SGFSIZE is given. Generally caused by the SIZE parameter (JCL,SRTDEF).
TD12.07	UNABLE TO MODIFY SEGMENT ATTRIBUTES
Meaning:	The problem must be reported to Technical Support.
TD12.08	UNABLE TO MODIFY SEGMENT ATTRIBUTES
Meaning:	The problem must be reported to Technical Support.
TD12.09	UNABLE TO CHANGE SEGMENT SIZE
Meaning:	The segment used for large COBOL key tables cannot be adjusted to the desired size. A return code from H_SGFSIZE is given. Generally caused by the SIZE parameter (JCL,SRTDEF).

**List of Execution Error Messages (Cont'd)**

TD12.10	UNABLE TO CHANGE SEGMENT SIZE
Meaning:	A pseudo-buffer segment used for sorting cannot be adjusted to the desired size. A return code from H_SGFSIZE is given. Generally caused by the SIZE parameter (JCL,SRTDEF).
TD12.11	UNABLE TO LOCK B_SEGMENT
Meaning:	A return code from H_SGLOCK is given. Either SIZE is smaller than the locked memory needed or the problem must be reported to Technical Support.
TD14.01	UNABLE TO OPEN PRTFILE
Meaning:	JCL error or the problem must be reported to Technical Support.
TD14.02	UNABLE TO CREATE PRTFILE BANNER
Meaning:	Incorrect PRTFILE or the problem must be reported to Technical Support.
TD14.03	UNABLE TO CREATE RECORD 118 ON PRTFILE
Meaning:	Incorrect PRTFILE or the problem must be reported to Technical Support.
TD14.04	UNABLE TO PUT RECORD ON PRTFILE
Meaning:	Incorrect PRTFILE or the problem must be reported to Technical Support.
TD14.05	INCONSISTENT DATA : DATA LOSS
Meaning:	When the ENDSRT procedure was called, the number of returned entities was lower than the number of released entities.
TD14.06	INCONSISTENT DATA : DATA GAIN
Meaning:	When the ENDSRT procedure was called, the number of returned entities was greater than the number of released entities.
TD14.07	UNABLE TO OPEN LOGFILE
Meaning:	The LOGFILE JCL parameter must be corrected or the problem must be reported to Technical Support.



## Sort and Merge Error Messages

### List of Execution Error Messages (Cont'd)

TD14.08	LOGFILE OVERFLOW
Meaning:	The LOGFILE assignment is not correct or the DSL implies that many invalid records have been put in the LOGFILE.
TD14.09	UNABLE TO PUT RECORD ON LOGFILE
Meaning:	Check file labels, characteristics, assignment. If they are correct, the problem must be reported to Technical Support.
TD14.10	UNABLE TO CLOSE LOGFILE
Meaning:	The problem must be reported to Technical Support.
TD14.11	UNABLE TO CORRECTLY RESET SORT RESOURCES ** ** ** **
Meaning:	The problem must be reported to Technical Support.
TD14.12	END OF FILE REACHED BEFORE LAST RECORD TO BE PROCESSED
Meaning:	The HALT option indicated more records than there are in the file. This is only a warning issued by the module XX.
TD14.13	OUTFILE IS EMPTY
Meaning:	The user is advised that there is no output record. This is only a warning issued by the module XX.
TD14.14	CHECK FOR RECORD TRUNCATION
Meaning:	The user has defined the output record shorter than the input record. This is only a warning issued by the module XX.
TD14.15	CHECK FOR RECORD PADDING
Meaning:	The user has defined the output record longer than the input record. This is only a warning issued by the module XX.
TD14.16	UNABLE TO CHANGE SEGMENT SIZE
Meaning:	The main segment used for sorting cannot be adjusted to the desired size. A return code from H_SGSIZE is given.
TD18.01	UNABLE TO GET RECORD/BLOCK FROM INPUT FILE
Meaning:	Check file labels, characteristics, assignment. If they are correct, the problem must be reported to Technical Support.

**List of Execution Error Messages (Cont'd)**

TD18.02	UNABLE TO CLOSE INPUT FILE
Meaning:	The problem must be reported to Technical Support.
TD18.03	UNABLE TO CLOSE LOGFILE
Meaning:	The problem must be reported to Technical Support.
TD19.01	UNABLE TO RETRIEVE INPUT FILE IDENTIFICATION
Meaning:	The problem must be reported to Technical Support.
TD19.02	UNABLE TO OPEN INPUT FILE
Meaning:	Check file labels, characteristics, assignment. If they are correct, the problem must be reported to Technical Support.
TD19.03	UNABLE TO GET RECORD/BLOCK FROM INPUT FILE
Meaning:	Check file labels, characteristics, assignment. If they are correct, the problem must be reported to Technical Support.
TD19.04	UNABLE TO RETRIEVE GETCI, CHECKCI ADDRESS FOR INPUT FILE
Meaning:	Check file labels, characteristics, assignment (UFAS involved). If they are correct, the problem must be reported to Technical Support.
TD19.05	INVALID INPUT FILE PARAMETER (FILEFORM OR RECFORM)
Meaning:	Check file labels, characteristics, assignment. If they are correct, the problem must be reported to Technical Support.
TD22.01	UNABLE TO PUT RECORD/BLOCK ON OUTPUT FILE
Meaning:	Check file labels, characteristics, assignment. If they are correct, the problem must be reported to Technical Support.
TD22.02	UNABLE TO CLOSE OUTPUT FILE
Meaning:	Check file labels, characteristics, assignment. If they are correct, the problem must be reported to Technical Support.
TD22.03	OUTPUT FILE OVERFLOW
Meaning:	Either the output file is actually too small or some incorrect parameter (e.g. output record or block size) implies an abnormal output file occupancy (e.g. one record per block).

## Sort and Merge Error Messages

### List of Execution Error Messages (Cont'd)

TD22.04	UNABLE TO GET RECORD/BLOCK FROM INPUT FILE
Meaning:	Check file labels, characteristics, assignment. If they are correct, the problem must be reported to Technical Support.
TD22.05	UNABLE TO CLOSE INPUT FILE
Meaning:	The problem must be reported to Technical Support.
TD23.01	UNABLE TO OPEN OUTPUT FILE
Meaning:	Check file labels, characteristics, assignment. If they are correct, the problem must be reported to Technical Support.
TD23.02	UNABLE TO PUT RECORD/BLOCK ON OUTPUT FILE
Meaning:	Check file labels, characteristics, assignment. If they are correct, the problem must be reported to Technical Support.
TD23.03	UNABLE TO RETRIEVE GETCI, CHECKCI ADDRESS FOR OUTPUT FILE
Meaning:	Check file labels, characteristics, assignment. If they are correct, the problem must be reported to Technical Support.
TD23.04	OUTPUT FILE OVERFLOW
Meaning:	Either the output file is actually too small or some incorrect parameter (e.g. output record or block size) implies an abnormal output file occupancy (e.g. one record per block).
TD23.05	INVALID INPUT FILE PARAMETER (FILEFORM OR RECFORM)
TD23.06	UNABLE TO PUT RECORD/BLOCK ON OUTPUT FILE
Meaning:	Check file labels, characteristics, assignment. If they are correct, the problem must be reported to Technical Support.
Meaning:	Check file labels, characteristics, assignment. If they are correct, the problem must be reported to Technical Support.
TD25.01	CHECKPOINT ERROR
Meaning:	The checkpoint conditions are probably not satisfied. If so, then the program or the JCL must be corrected.
TD25.02	BREAK POINT : QUIT
Meaning:	An abort has been requested by the response QUIT.

**List of Execution Error Messages (Cont'd)**

TD25.03	INTERNAL SORT ERROR
Meaning:	The problem must be reported to Technical Support.
TD26.01	ONLY OUTPUT=DATA IS CORRECT FOR TAPE INPUT FILES
Meaning:	JCL or DSL must be corrected.
TD26.02	MIN IS GREATER THAN MAXIMAL INPUT RECORD LENGTH
Meaning:	JCL or DSL or SRTDEF must be corrected. MIN is defined as the length of the shortest record which can hold all the KEY, ARRANGE, SUM and INCLUDE/OMIT fields.
TD26.03	INCLUDE/OMIT FIELD END OUT OF MINIMAL INPUT RECORD
Meaning:	JCL or DSL must be corrected. MIN is defined as the length of the shortest record which can hold all the KEY, ARRANGE, SUM and INCLUDE/OMIT fields.
TD26.04	KEY/SUM/ARRANGE FIELD END OUT OF MINIMAL INPUT RECORD
Meaning:	JCL or DSL must be corrected. MIN is defined as the length of the shortest record which can hold all the KEY, ARRANGE, SUM and INCLUDE/OMIT fields.
TD26.05	AVERAGE IS GREATER THAN MAXIMAL INPUT RECORD LENGTH
Meaning:	JCL or DSL must be corrected.
TD26.06	AVERAGE IS LOWER THAN MINIMAL INPUT RECORD LENGTH
Meaning:	JCL or DSL must be corrected. MIN is defined as the length of the shortest record which can hold all the KEY, ARRANGE, SUM and INCLUDE/OMIT fields.
TD26.07	DUMMY INPUT FILE
Meaning:	This function is not available. The user program must be corrected.

## Sort and Merge Error Messages

### List of Execution Error Messages (Cont'd)

TD26.08	UNABLE TO RETRIEVE INPUT FILE DEFINITION
Meaning:	The sort cannot determine the characteristics of the input file.  A return code from H_RFLDEF or a return code set by Sort is given. JCL error or non-standard file.
TD26.09	ERRORS IN SORT PARAMETERS
Meaning:	This message is only a general title for specific messages.
TD26.10	USER SORT KEYS ARE TOO LONG
Meaning:	The SUM or KEY field length exceeds 7000 bytes. The user program must be corrected.
TD26.11	USER RECORD IS TOO LONG
Meaning:	The sort record size exceeds 15000 bytes. Either the user record is too long or too many keys overlap.  The user program must be corrected.
TD26.12	UNABLE TO OPEN INPUT FILE
Meaning:	Check file labels, characteristics, assignment. If they are correct, the problem must be reported to Technical Support.
TD26.13	UNABLE TO GET RECORD/BLOCK FROM INPUT FILE
Meaning:	Check file labels, characteristics, assignment. If they are correct, the problem must be reported to Technical Support.
TD26.14	UNABLE TO CLOSE INPUT FILE
Meaning:	The problem must be reported to Technical Support.
TD26.15	UNABLE TO CHANGE SEGMENT SIZE
Meaning:	The main segment used for sorting cannot be adjusted to the desired size. A return code from H_SGFSIZE is given. Generally caused by the SIZE parameter (JCL,SRTDEF).
TD26.16	INVALID MONITORED SORT_INPUT_FILE ASSIGNMENT
Meaning:	Conditions for use of "SHARE = MONITOR" (GAC) for input file assignment are not satisfied. JCL must be corrected.

**List of Execution Error Messages (Cont'd)**

TD26.17	UNABLE TO GET INPUT FILE SIZE
Meaning:	Check the input file assignment in your JCL. If it is correct, please contact your Service Center, or submit a STAR.
TD26.64	<pre> END OF { INCLUDE } ZZZZZ IS BEYOND INPUT RECORD SIZE       { OMIT      }       { KEY       }       { SUM       }       { ARRANGE  } </pre> <p>ZZZZZ = rank of the field which is beyond the record size.</p>
Meaning:	<p>The involved field ends beyond the end of the input record.</p> <p>The DSL or the user program must be corrected.</p>
TD27.01	UNABLE TO RETRIEVE OUTPUT FILE DEFINITION
Meaning:	The sort cannot discover the characteristics of the output file. A return code from H_RFLDEF or a return code set by Sort is given. JCL error or non-standard file.
TD27.02	DUMMY OUTPUT FILE
Meaning:	This function is not available.
TD27.03	UNABLE TO OPEN OUTPUT FILE
Meaning:	Check file labels, characteristics, assignment. If they are correct, the problem must be reported to Technical Support.
TD27.04	UNABLE TO CLOSE OUTPUT FILE
Meaning:	Check file labels, characteristics, assignment. If they are correct, the problem must be reported to Technical Support.
TD27.05	UNABLE TO ALLOCATE OUTPUT FILE
Meaning:	Check the other messages which accompany this one. These will be of the form DUxx.yy (PREALLOC). See the description of PREALLOC in the <i>Data Management Utilities User Guide</i> . Generally, the problem is due either to a shortage of disk space or to a bad file definition.

## Sort and Merge Error Messages

### List of Execution Error Messages (Cont'd)

TD30.21	ATTEMPT TO REQUEST A SORTED RECORD AFTER END OF SORT_OUTPUT
Meaning:	The program or processor calling Sort is erroneously starting an infinite loop after the H_RETURNS primitive has returned the return code DATALIM, or after the COBOL verb RETURN has activated the paragraph or Chapter "AT END".
TD31.22	KEY SEQUENCE ERROR
Meaning:	User error : at least one merge input file is out of sequence according to merge key(s) specifications. The unsorted input file and the concerned input record rank appear in the subsequent merge report lines.
TD33.01	UNABLE TO CREATE SEGMENT
Meaning:	A return code from H_SGCR is given. The error could be internal to Sort but most frequently : either too much memory is required for Sort (SIZE in JCL, SRTDEF) or the
TD33.02	UNABLE TO GET SEMAPHORES
Meaning:	Linkage problem (insufficient semaphore pool). Could be caused by nested sorts.
TD33.03	UNABLE TO DEFINE CHANNEL PROGRAM
Meaning:	The problem must be reported to Technical Support.
TD33.04	UNABLE TO DEFINE CHANNEL PROGRAM
Meaning:	The problem must be reported to Technical Support.
TD33.05	UNABLE TO GET RECORD/BLOCK FROM INPUT FILE
Meaning:	Check file labels, characteristics, assignment. If they are correct, the problem must be reported to Technical Support.
TD33.06	UNABLE TO GET RECORD/BLOCK FROM INPUT FILE
Meaning:	Check file labels, characteristics, assignment. If they are correct, the problem must be reported to Technical Support.
TD33.07	UNABLE TO GET RECORD/BLOCK FROM INPUT FILE
Meaning:	Check file labels, characteristics, assignment. If they are correct, the problem must be reported to Technical Support.
TD33.08	UNABLE TO START A SERVANT PROCESS
Meaning:	Check if you have tried to run a multi-process LM. If not, the problem must be reported to Technical Support.

**List of Execution Error Messages (Cont'd)**

TD33.09	INTERNAL SORT ERROR
Meaning:	The problem must be reported to Technical Support.
TD33.10	INTERNAL SORT ERROR
Meaning:	The problem must be reported to Technical Support.
TD33.11	INTERNAL SORT ERROR
Meaning:	The problem must be reported to Technical Support.
TD34.01	UNABLE TO DELETE CHANNEL PROGRAM
Meaning:	The problem must be reported to Technical Support.
TD34.02	UNABLE TO DELETE CHANNEL PROGRAM
Meaning:	The problem must be reported to Technical Support.
TD34.03	UNABLE TO CREATE SEGMENT
Meaning:	A return code from H_SGCR is given. The error could be internal to Sort but most frequently : either too much memory is required for Sort (SIZE in JCL, SRTDEF) or the
TD34.04	UNABLE TO CREATE SEGMENT
Meaning:	A return code from H_SGCR is given. The error could be internal to Sort but most frequently : either too much memory is required for Sort (SIZE in JCL, SRTDEF) or the
TD35.01	UNABLE TO CREATE SEGMENT
Meaning:	A return code from H_SGCR is given. The error could be internal to Sort but most frequently : either too much memory is required for Sort (SIZE in JCL, SRTDEF) or the
TD35.02	UNABLE TO CREATE SEGMENT
Meaning:	A return code from H_SGCR is given. The error could be internal to Sort but most frequently : either too much memory is required for Sort (SIZE in JCL, SRTDEF) or the
TD35.03	UNABLE TO CREATE SEGMENT
Meaning:	A return code from H_SGCR is given. The error could be internal to Sort but most frequently : either too much memory is required for Sort (SIZE in JCL, SRTDEF) or the



## Sort and Merge Error Messages

### List of Execution Error Messages (Cont'd)

TD35.04	UNABLE TO CREATE SEGMENT
Meaning:	A return code from H_SGCR is given. The error could be internal to Sort but most frequently : either too much memory is required for Sort (SIZE in JCL, SRTDEF) or the
TD35.05	UNABLE TO CREATE SEGMENT
Meaning:	A return code from H_SGCR is given. The error could be internal to Sort but most frequently : either too much memory is required for Sort (SIZE in JCL, SRTDEF) or the
TD35.06	UNABLE TO CREATE SEGMENT
Meaning:	A return code from H_SGCR is given. The error could be internal to Sort but most frequently : either too much memory is required for Sort (SIZE in JCL, SRTDEF) or the
TD35.07	UNABLE TO GET SEMAPHORES
Meaning:	Linkage problem (insufficient semaphore pool). Could be caused by nested sorts.
TD35.08	UNABLE TO DEFINE CHANNEL PROGRAM
Meaning:	The problem must be reported to Technical Support.
TD35.09	UNABLE TO DEFINE CHANNEL PROGRAM
Meaning:	The problem must be reported to Technical Support.
TD35.10	UNABLE TO GET RECORD/BLOCK FROM INPUT FILE
Meaning:	Check file labels, characteristics, assignment. If they are correct, the problem must be reported to Technical Support.
TD35.11	UNABLE TO GET RECORD/BLOCK FROM INPUT FILE
Meaning:	Check file labels, characteristics, assignment. If they are correct, the problem must be reported to Technical Support.
TD35.12	UNABLE TO GET RECORD/BLOCK FROM INPUT FILE
Meaning:	Check file labels, characteristics, assignment. If they are correct, the problem must be reported to Technical Support.
TD35.13	UNABLE TO GET RECORD/BLOCK FROM INPUT FILE
Meaning:	Check file labels, characteristics, assignment. If they are correct, the problem must be reported to Technical Support.

**List of Execution Error Messages (Cont'd)**

TD35.14	UNABLE TO GET RECORD/BLOCK FROM INPUT FILE
Meaning:	Check file labels, characteristics, assignment. If they are correct, the problem must be reported to Technical Support.
TD35.15	UNABLE TO DELETE CHANNEL PROGRAM
Meaning:	The problem must be reported to Technical Support.
TD35.16	UNABLE TO DELETE CHANNEL PROGRAM
Meaning:	The problem must be reported to Technical Support.
TD35.17	UNABLE TO START A SERVANT PROCESS
Meaning:	Check if you have tried to run a multi-process LM. If not, the problem must be reported to Technical Support.
TD35.18	INTERNAL SORT ERROR
Meaning:	The problem must be reported to Technical Support.
TD36.01	UNABLE TO CREATE SEGMENT
Meaning:	A return code from H_SGCR is given. The error could be internal to Sort but most frequently : either too much memory is required for Sort (SIZE in JCL, SRTDEF) or the
TD36.02	UNABLE TO DEFINE CHANNEL PROGRAM
Meaning:	The problem must be reported to Technical Support.
TD36.03	UNABLE TO DEFINE CHANNEL PROGRAM
Meaning:	The problem must be reported to Technical Support.
TD36.04	UNABLE TO DELETE CHANNEL PROGRAM
Meaning:	The problem must be reported to Technical Support.
TD36.05	UNABLE TO DELETE CHANNEL PROGRAM
Meaning:	The problem must be reported to Technical Support.
TD37.01	UNABLE TO CREATE WORK FILE DEFINITION
Meaning:	A dynamic FD (file definition table) is required and cannot be created. A return code from H_CRFD is given. It could be a linkage problem (ENTRYOV).

## Sort and Merge Error Messages

### List of Execution Error Messages (Cont'd)

TD37.02	UNABLE TO ASSIGN DYNAMIC DISK WORK FILE
Meaning:	A return code from H_DYNASG is given.
TD37.03	UNABLE TO RETRIEVE WORK FILE DEFINITION
Meaning:	The problem must be reported to Technical Support.
TD37.04	UNABLE TO DELETE WORK FILE DEFINITION
Meaning:	Sort is deleting useless FD (file definition table). The return code from H_DLFD is given. The problem must be reported to Technical Support.
TD37.05	UNABLE TO ALLOCATE DYNAMIC DISK WORK FILE
Meaning:	A return code from H_DYNAL is given.
TD37.06	TOO FEW WORK DISKS (<2) FOR A PARALLEL SORT
Meaning:	Self-explanatory.
TD37.07	UNABLE TO RETRIEVE WORK FILE DEFINITION
Meaning:	The problem must be reported to Technical Support.
TD37.08	UNABLE TO IDENTIFY WORK FILE
Meaning:	Sort is unable to know which volume supports the workfile (\$H_RTVLID). The return code from H_RTVLID is given.
TD37.09	STATIC H_WSTAxX ASSIGNMENTS MUST BE CONSECUTIVE FROM H_WSTA01
Meaning:	Self-explanatory.
TD37.10	H_SRTWKD AND H_WSTAxX CAN'T BE TOGETHER ASSIGNED
Meaning:	Self-explanatory.
TD99.99	USER SORT CALL VIOLATION
Meaning:	The primitives H_BEGSRT, H_RELSRT, H_SRT, H_RETSRT and H_ENDSRT have been called in the wrong order. In COBOL, this incorrect order could be caused by wrong end of paragraph, section ... etc.

## Sort/Merge User's Guide

## B. Sort/Merge Report

The contents of the Sort/Merge Report, which is produced in a SYSOUT subfile are determined by the value assigned by the user to the REPORT parameter of the SORT/GSORT or MERGE/GMERGE JCL statement. Specifiable values are AUDIT, PARAM, ALL, or NONE. The appearance of Sort Execution Diagnostics or Sort Parameter Diagnostics terminate the report. If Sort itself aborts (independently from DSL) the information corresponding to INTINFO will still be available.

### Report for the New Sort Product

There is no change in the sort report for a mono-process sort.

For a parallel sort, there is a global sort report as for a mono-process sort. If you specify INTINFO (internal information, in the DSL), you will get the global sort report and a report of the same form for each slave sort.

There is no change in the merge report.

### B.1 REPORT = ALL

The following is an example of a Sort Report with REPORT = ALL (full report):

```
***** SORT REPORT *****
**** SORT PARAMETER LISTING ****
FUNCTION: DUPREC = FIFO INTINFO
RECORD : KEY = 1 10
END :
**** END OF SORT REPORT ****
**** SORT AUDIT INFORMATION ****
INPUT FILE : TRID.ALEA150
OUTPUT FILE : ;102478.OUTALEA
TOTAL NUMBER OF RECORDS READ      :      10000
TOTAL NUMBER OF RECORDS WRITTEN   :      10000
TOTAL NUMBER OF RECORDS OMITTED   :           0
TOTAL NUMBER OF RECORDS DELETED   :           0
TOTAL NUMBER OF INVALID RECORDS   :           0
% OF WORK FILE SPACE USED         :          62
INFILE: FILEFORM=BFAS, FILEORG=SEQUENTIAL, RECFORM=FB, BLKSZ=5000, RECSZ=50
OUTFILE: FILEFORM=BFAS, FILEORG=SEQUENTIAL, RECFORM=FB, BLKSZ=6000, RECSZ=50
**** SORT INTERNAL VALUES ****
SORTALGO: SORTFRM=FO, ARTBSZ=54, BLKART=3890, BLKDES1=3890, BLKDES2=3890
NBSORTMEM=500, MAXNBSORTMEM=731, MAXNBDES=14, MRGORDER= 6
NBMONOPH1=11, NB EXPECTED=10, NBPHASE2= 0
**** END OF SORT REPORT ****
```

**B.2 REPORT = AUDIT (DEFAULT)****Without INTINFO**

```

***** SORT REPORT *****
**** SORT AUDIT INFORMATION ****
INPUT FILE : TRID.ALEA150
OUTPUT FILE : ;102478.OUTALEA
TOTAL NUMBER OF RECORDS READ      :      10000
TOTAL NUMBER OF RECORDS WRITTEN   :      10000
TOTAL NUMBER OF RECORDS OMITTED   :           0
TOTAL NUMBER OF RECORDS DELETED   :           0
TOTAL NUMBER OF INVALID RECORDS   :           0
% OF WORK FILE SPACE USED         :          62
INFILE: FILEFORM=BFAS, FILEORG=SEQUENTIAL, RECFORM=FB, BLKSZ=5000, RECSZ=50
OUTFILE: FILEFORM=BFAS, FILEORG=SEQUENTIAL, RECFORM=FB, BLKSZ=6000, RECSZ=50
**** END OF SORT REPORT ****

```

**With INTINFO**

```

***** SORT REPORT *****
**** SORT AUDIT INFORMATION ****
INPUT FILE : TRID.ALEA150
OUTPUT FILE : ;102478.OUTALEA
TOTAL NUMBER OF RECORDS READ      :      10000
TOTAL NUMBER OF RECORDS WRITTEN   :      10000
TOTAL NUMBER OF RECORDS OMITTED   :           0
TOTAL NUMBER OF RECORDS DELETED   :           0
TOTAL NUMBER OF INVALID RECORDS   :           0
% OF WORK FILE SPACE USED         :          62
INFILE : FILEFORM=BFAS, FILEORG=SEQUENTIAL, RECFORM=FB, BLKSZ=5000, RECSZ=50
OUTFILE : FILEFORM=BFAS, FILEORG=SEQUENTIAL, RECFORM=FB, BLKSZ=6000, RECSZ=50
**** SORT INTERNAL VALUES ****
SORTALGO=SORTRFM=FO, ARTBSZ= 54, BLKART=3890, BLKDES1=3890, BLKDES2=3890
NBSORTMEM=500, MAXNBSORTMEM=731, MAXNBDES=14, MRGORDER=6
NBMONOPH1=11, NB EXPECTED=10, NBPHASE2=0
**** END OF SORT REPORT ****

```

**With Variable Length Record Format**

```

***** SORT REPORT *****
**** SORT AUDIT INFORMATION ****
INPUT FILE : UFRELV
OUTPUT FILE : ;208711.A
TOTAL NUMBER OF RECORDS READ      :      3740
TOTAL NUMBER OF RECORDS WRITTEN   :      3740
TOTAL NUMBER OF RECORDS OMITTED   :           0
TOTAL NUMBER OF RECORDS DELETED   :           0
TOTAL NUMBER OF INVALID RECORDS   :           0
AVERAGE INPUT RECORD SIZE        :          50
SORT INTERNAL RECORD SIZE         :          90
% OF WORK FILE SPACE USED         :           3
WORK USED IN % OF INFILE          :         174
INFILE : FILEFORM=UFAS, FILEORG=DIR./REL., RECFORM=V, BLKSZ=4096, RECSZ=129
OUTFILE : FILEFORM=UFAS, FILEORG=DIR./REL., RECFORM=V, BLKSZ=4096, RECSZ=129
**** END OF SORT REPORT ****

```

## Sort/Merge Report

### **B.3 REPORT = PARAM**

In this case, only SORT PARAMETER LISTING is available (see above).

```
***** SORT REPORT *****  
FUNCTION: DUPREC=FIFO INTINFO  
RECORD: KEY = 1 10  
END:
```

### **B.4 REPORT = NONE**

A Sort Report is not given unless there are errors, in which case the same listing as for REPORT = AUDIT with INTINFO will be given.

## Sort/Merge User's Guide



## C. Collating Sequences

The table below shows the collating sequences available with Sort. The table shows the characters in ascending order, from top to bottom, and left to right. They begin with the "low-value" and finish with the "high-value".

**Table C-1. EBCDIC (DPS 7000) PLW Character Set (1/4)**

<b>EBCDIC hexa</b>	<b>Graphic</b>	<b>H36 punch</b>	<b>EBCDIC decimal</b>	<b>Graphic</b>	<b>H36 punch</b>
00	low-val	12-0-1-8-9	20		11-0-1-8-9
01		12-1-9	21		0-1-9
02		12-2-9	22		0-2-9
03		12-3-9	23		0-3-9
04		12-4-9	24		0-4-9
05		12-5-9	25		0-5-9
06		12-6-9	26		0-6-9
07		12-7-9	27		0-7-9
08		12-8-9	28		0-8-9
09		12-1-8-9	29		0-1-8-9
0A		12-2-8-9	2A		0-2-8-9
0B		13-3-8-9	2B		0-3-8-9
0C		12-4-8-9	2C		0-4-8-9
0D		12-5-8-9	2D		0-5-8-9
0E		12-6-8-9	2E		0-6-8-9
0F		12-7-8-9	2F		0-7-8-9
10		12-11-8-9	30		12-11-0-8-9
11		11-1-9	31		1-9
12		11-2-9	32		2-9
13		11-3-9	33		3-9
14		11-4-9	34		4-9
15		11-5-9	35		5-9
16		11-6-9	36		6-9
17		11-7-9	37		7-9
18			38		8-9
19		11-8-9	39		1-8-9
1A		11-1-8-9	3A		2-8-9
1B		11-2-8-9	3B		3-8-9
1C		11-3-8-9	3C		4-8-9
1D		11-4-8-9	3D		5-8-9
1E		11-5-8-9	3E		6-8-9
1F		11-6-8-9	3F		7-8-9
		11-7-8-9			

Table C-1. EBCDIC (DPS 7000) PLW Character Set (2/4)

EBCDIC hexa	Graphic	H36 punch	EBCDIC decimal	Graphic	H36 punch
40	space	no punches	60	-	11
41		12-0-19	61	/	0-1
42	â	12-0-2	62	Â	11-0-2-9
43	ä	12-0-3-9	63	Ä	11-0-3-9
44	à	12-0-4-9	64	À	11-0-4-9
45	á	12-0-5-9	65	'A	11-0-5-9
46	ã	12-0-6-9	66	Ã	11-0-6-9
47	â	12-0-7-9	67	Á	11-0-7-9
48	ç	12-0-8-9	68	Ç	11-0-8-9
49	ñ	12-1-8	69	Ñ	0-1-8
4A	[	12-2-8	6A	⋮	12-11
4B	.	12-3-8	6B		0-3-8
4C	<	12-4-8	6C	%	0-4-8
4D	(	12-5-8	6D	-	0-5-8
4E	+	12-6-8	6E	>	0-6-8
4F	!	12-7-8	6F	?	0-7-8
50	&		70	∅	12-11-0
51	é	12-11-1-9	71	È	12-11-0-1-9
52	ê	12-11-2-9	72	Ê	12-11-0-2-9
53	ë	12-11-3-9	73	Ë	12-11-0-3-9
54	è	12-11-4-9	74	E	12-11-0-4-9
55	í	12-11-5-9	75	Ì	12-11-0-5-9
56	î	12-11-6-9	76	Î	12-11-0-6-9
57	ï	12-11-7-9	77	Ï	12-11-0-7-9
58	ì	12-11-8-9	78	ì	12-11-0-8-9
59	ß	11-1-8	79	'	1-8
5A	]	11-2-8	7A	:	2-8
5B	\$	11-3-8	7B	#	3-8
5C	*	11-4-8	7C	@	4-8
5D	)	11-5-8	7D	'	5-8
5E	:	11-6-8	7E	=	6-8
5F	^	11-7-8	7F	"	7-8

Collating Sequences

**Table C-1. EBCDIC (DPS 7000) PLW Character Set (3/4)**

<b>EBCDIC hexa</b>	<b>Graphic</b>	<b>H36 punch</b>	<b>EBCDIC decimal</b>	<b>Graphic</b>	<b>H36 punch</b>
80	∅	12-0-1-8	A0	μ	11-0-1-8
81	a	12-0-1	A1	~	11-0-1
82	b	12-0-2	A2	s	11-0-2
83	c	12-0-3	A3	t	11-0-3
84	d	12-0-4	A4	u	11-0-4
85	e	12-0-5	A5	v	11-0-5
86	f	12-0-6	A6	w	11-0-6
87	g	12-0-7	A7	x	11-0-7
88	h	12-0-8	A8	y	11-0-8
89	i	12-0-9	A9	z	11-0-9
8A		12-0-2-8	AA	i	11-0-2-8
8B		12-0-3-8	AB	?	11-0-3-8
8C		12-0-4-8	AC		11-0-4-8
8D		12-0-5-8	AD		11-0-5-8
8E		12-0-6-8	AE		11-0-6-8
8F	±	12-0-7-8	AF		11-0-7-8
90	°	12-11-1-8	B0		12-11-0-1-8
91	j	12-11-1	B1	£	12-11-0-1
92	k	12-11-2	B2		12-11-0-2
93	l	12-11-3	B3		12-11-0-3
94	m	12-11-4	B4		12-11-0-4
95	n	12-11-5	B5	§	12-11-0-5
96	o	12-11-6	B6	¶	12-11-0-6
97	p	12-11-7	B7		12-11-0-7
98	q	12-11-8	B8		12-11-0-8
99	r	12-11-9	B9		12-11-0-9
9A	a	12-11-2-8	BA		12-11-0-2-8
9B	o	12-11-3-8	BB		12-11-0-3-8
9C	æ	12-11-4-8	BC	..	12-11-0-4-8
9D		12-11-5-8	BD	,	12-11-0-5-8
9E	Æ	12-11-6-8	BE		12-11-0-6-8
9F	α	12-11-7-8	BF		12-11-0-7-8

Table C-1. EBCDIC (DPS 7000) PLW Character Set (4/4)

EBCDIC hexa	Graphic	H36 punch	EBCDIC decimal	Graphic	H36 punch
C0	{	12-0	E0	\	0-2-8
C1	A	12-1	E1		11-0-1-9
C2	B	12-2	E2	S	0-2
C3	C	12-3	E3	T	0-3
C4	D	12-4	E4	U	0-4
C5	E	12-5	E5	V	0-5
C6	F	12-6	E6	W	0-6
C7	G	12-7	E7	X	0-7
C8	H	12-8	E8	Y	0-8
C9	I	12-9	E9	Z	0-9
CA		12-0-2-8-9	EA		11-0-2-8-9
CB	ô	12-0-3-8-9	EB	Ô	11-0-3-8-9
CC	ö	12-0-4-8-9	EC	Ö	11-0-4-8-9
CD	ò	12-0-5-8-9	ED	Ò	11-0-5-8-9
CE	ó	12-0-6-8-9	EE	Ó	11-0-6-8-9
CF	õ	12-0-7-8-9	EF	Õ	11-0-7-8-9
D0	}	11-0	F0	0	0
D1	J	11-1	F1	1	1
D2	K	11-2	F2	2	2
D3	L	11-3	F3	3	3
D4	M	11-4	F4	4	4
D5	N	11-5	F5	5	5
D6	O	11-6	F6	6	6
D7	P	11-7	F7	7	7
D8	Q	11-8	F8	8	8
D9	R	11-9	F9	9	9
DA		12-11-2-8-9	FA		12-11-0-2-8-9
DB	û	12-11-3-8-9	FB	Û	12-11-0-3-8-9
DC	ü	12-11-4-8-9	FC	Ü	12-11-0-4-8-9
DD	ù	12-11-5-8-9	FD	Û	12-11-0-5-8-9
DE	ú	12-11-7-8-9	FE	Ù	12-11-0-6-8-9
DF		12-11-2-8-9	FF		12-11-0-7-8-9

## Collating Sequences

Table C-2 below shows the collating sequence for the Series 100 character set. The left-most column gives the original internal value of the character in a Series 100 system. The right-most column shows the hexadecimal value which the character has after translation into the DPS 7000 environment.

**Table C-2. Series 100 Character Set (1/2)**

Hexadecimal pre-translation	Graphic	H14 punch	H36 punch	EBCDIC
00	low-val			00
40	0	0	0	F0
41	1	1	1	F1
42	2	2	2	F2
43	3	3	3	F3
44	4	4	4	F4
45	5	5	5	F5
46	6	6	6	F6
47	7	7	7	F7
48	8	8	8	F8
49	9	9	9	F9
4A	[	8-2	12-8-2	4A
4B	#	8-3	8-3	7B
4C	@	8-4	8-4	7C
4D	:	8-5	8-5	7A
4E	>	8-6	8-6	6E
4F	?	8-7	8-7	6F
50	space	no punches	no punches	40
51	A	12-1	12-1	C1
52	B	12-2	12-2	C2
53	C	12-3	12-3	C3
54	D	12-4	12-4	C4
55	E	12-5	12-5	C5
56	F	12-6	12-6	C6
57	G	12-7	12-7	C7
58	H	12-8	12-8	C8
59	I	12-9	12-9	C9
5A	&	12	12	50
5B	.	12-8-3	12-8-3	4B
5C	]	12-8-4	12-8-2	4F
5D	(	12-8-5	12-8-5	4D
5E	<	12-8-6	12-8-4	4C
5F	\	12-8-7	0-8-2	E0

**Table C-2. Series 100 Character Set (2/2)**

Hexadecimal pre-translation	Graphic	H14 punch	H36 punch	EBCDIC
A0		11-0	11-8-7	5F
A1	J	11-1	11-1	D1
A2	K	11-2	11-2	D2
A3	L	11-3	11-3	D3
A4	M	11-4	11-4	D4
A5	N	11-5	11-5	D5
A6	O	11-6	11-6	D6
A7	P	11-7	11-7	D7
A8	Q	11-8	11-8	D8
A9	R	11-9	11-9	D9
AA	-	11	11	60
AB	\$	11-8-3	11-8-3	5B
AC	*	11-8-4	11-8-4	5C
AD	)	11-8-5	11-8-5	5D
AE	;	11-8-6	11-8-6	5E
AF	,	11-8-7	8-5	7D
B0	+	12-0	12-8-6	4E
B1	/	0-1	0-1	61
B2	S	0-2	0-2	E2
B3	T	0-3	0-3	E3
B4	U	0-4	0-4	E4
B5	V	0-5	0-5	E5
B6	W	0-6	0-6	E6
B7	X	0-7	0-7	E7
B8	Y	0-8	0-8	E8
B9	Z	0-9	0-9	E9
BA	← -	0-8-2	0-8-5	6D
BB	,	0-8-3	0-8-3	6B
BC	%	0-8-4	0-8-4	7C
BD	=	0-8-5	8-6	7E
BE	"	0-8-6	8-7	7F
BF	!	0-8-7	11-8-2	5A
FF	hi-val			FF

**NOTE:** When two entries appear in the graphic column in Table C-5, the first graphic is the original Series 100 character and the second is the graphic value after translation into EBCDIC.

## Collating Sequences

Table C-3 shows the effect of COLLATE = H200 on EBCDIC files.

When two graphics are present (special characters), the first is the Series 200 value and the second is the equivalent in DPS 7000.

**Table C-3. H2000 Collating Sequence for EBCDIC Files**

H200 graphic	EBCDIC hexa	H200 graphic	EBCDIC hexa	H200 graphic	EBCDIC hexa	H200 graphic	EBCDIC hexa
0	F0	A	C1	J	D1	/	61
1	F1	B	C2	K	D2	S	E2
2	F2	C	C3	L	D3	T	E3
3	F3	D	C4	M	D4	U	E4
4	F4	E	C5	N	D5	V	E5
5	F5	F	C6	O	D6	W	E6
6	F6	G	C6	P	D6	X	E7
7	F7	H	C8	Q	D8	Y	E8
8	F8	I	C9	R	D9	Z	E9
9	F9	:	E5	#	7B	@	7C
\	7D	.	4B	\$	5B	,	6B
=	7E	)	5D	*	5C	(	4D
:	7A	%	6C	"	7E		4F
space	40	␣	5F	\	E0	.	6D
>	6E	?	6F	!	5A	-	4A
&	50	-	60	<	4C	¢	
+ {	4E C0	}	D0				

**NOTE:** All other hexadecimal values have the same collating sequence value as EBCDIC hexa 5F. Double entries in the table such as "A a C1 81" indicate that the values are equivalent for collating purposes.

Table C-4 shows the effect of COLLATE = ASCII for an EBCDIC file.

**Table C-4. ASCII Collating Sequence (1/2)**

Graphic	EBCDIC hexa	Graphic	EBCDIC hexa	Graphic	EBCDIC hexa	Graphic	EBCDIC hexa
NUL	00	space	40		7C		79
SOH	01	!	4F	A	C1	a	81
STX	02	"	7F	B	C2	b	82
ETX	03	#	7B	C	C3	c	83
EOT	37	\$	5B	D	C4	d	84
ENQ	2D	%	6C	E	C5	e	85
ACK	2E	&	50	F	C6	f	86
BEL	2F	\	7D	G	C7	g	87
BS	16	(	4D	H	C8	h	88
HT	05	)	5D	I	C9	i	89
LF	25	*	5C	J	D1	j	91
VT	0B	+	4E	K	D2	k	92
FF	0C	,	6E	L	D3	l	93
CR	0D	-	60	M	D4	m	94
SO	0E	.	4B	N	D5	n	95
SI	0F	/	61	O	D6	o	96
DLE	10	0	F0	P	D7	p	97
DC1	11	1	F1	Q	D8	q	98
DC2	12	2	F2	R	D9	r	99
TM	13	3	F3	S	E2	s	A2
DC4	3C	4	F4	T	E3	t	A3
NAK	3D	5	F5	U	E4	u	A4
SYN	32	6	F6	V	E5	v	A5
ETB	26	7	F7	W	E6	w	A6
CAN	18	8	F8	X	E7	x	A7
EM	19	9	F9	Y	E8	y	A8
SUB	3F	:	7A	Z	E9	z	A9
ESC	27	;	5E	[	4A	{	C0
IFS	1C	<	4C	\	E0	:	6A
IGS	1D	=	7E	!	5A	}	D0
IRS	1E	>	6E	~	5F	-	A1
IUS	1F	?	6F	-	6D	DEL	07



Collating Sequences

**Table C-4. ASCII Collating Sequence (2/2)**

Graphic	EBCDIC hexa	Graphic	EBCDIC hexa	Graphic	EBCDIC hexa	Graphic	EBCDIC hexa
DS	20		41		76	L	B8
SOS	21		42		77	V	B9
FS	22		43		78	M	BA
	23		44		80		BB
BYP	24		45		8A		BC
NL	15		46		8B		BD
LC	06		47		8C		BE
IL	17		48		8D		BF
	28		49		8E		CA
	29		51		8F		CB
SM	2A		52		90		CC
CU2	2B		53		9A		CD
	2C		54		9B		CE
RLF	09		55		9C		CF
SMM	0A		56		9D		DA
CU1	1B		57		9E		DB
	30		58		9F		DC
	31		59		A0		DD
CC	1A		62		AA		DE
	33		63		AB		DF
PN	34		64		AC		EA
RS	35		65		AD		EB
	35		65		AD		EB
UC	36		66		AE		EC
	08		67		AF		ED
	38		68		B0		EE
	39		69		B1		EF
	3A		70		B2		FA
CU3	3B		71		B3		FB
RF	04		72		B4		FC
RES	14		73		B5		FD
	3E		74		B6		FE
	E1		75		B7		FF

## Sort/Merge User's Guide

## D. SORT\_FILE Command (GCL)

### D.1 EXECUTING SORT\_FILE

You can execute a sort in interactive or in batch mode using the GCL command SORT\_FILE (alias SRTF or SORT).

The SORT\_FILE command has a format similar to the format of the JCL statement SORT. The visibility is similar to Data Management Utility commands.

The SORT\_FILE command is described in the *IOF Terminal User's Reference Manual*.

Note that the JCL statement GSORT (described in Chapter 4) can also be executed in interactive mode.

## D.2 SORT\_FILE (SRTF, SORT) COMMAND

### Purpose:

Sorts one or more files into a new file or into an existing one.

### Syntax:

```
{ SORT_FILE }
{ SRTF      }
{ SORT      }

      { FILE      }      { INFILE          }
      {           } = {                   }
      { OUTFILE  }      { ( output-file-description ) }

      { INFILE   }
      {           } = ( input-file-description )
      { INFILE1  }

      { UNCAT          }
      { TEMPRY        }
[ DYNALC = {           } ]
      { CAT            }
      { CAT{1|2|3|4|5} }

      { ALLOCATE }
[ {           } = ( file-allocation-parameters ) ]
      { OUTALC   }

      { DEF      }
[ {           } = ( file-define-parameters ) ]
      { OUTDEF   }

[ INDEF = ( file-define-parameters ) ]

{ DSLFILE }
{           } = { file78 | ::TN }
{ COMFILE }

[ WRKSIZE = ( dec5 [ dec5 ] ) ]

[ WRKVOL = { volume40 | RESIDENT | $RES } ]

[ WRKFILE = lib78 ]

[ WRKALC = ( file-allocation-parameters ) ]
```

## SORT\_FILE Command (GCL)

```
-----  
[ START = { 1 | dec10 } ]  
[ HALT = dec10 ]  
[ REPORT = { AUDIT | PARAM | ALL | NONE } ]  
[ INVREC = { ABORT | CONTINUE } ]  
[ ERROPT = { PRINTID | PRINT | LOG | IGNORE } ]  
[ LOGFILE = file78 ]  
[ INFILE2 = ( input-file-description ) ]  
  .  
  .  
[ INFILE9 = ( input-file-description ) ]  
[ PRTFILE = file78 ]  
[ REPEAT = { bool | 0 } ]  
[ SIZE = dec7 ]  
[ NBSORT = dec2 ]  
[ PADCHAR = { char1 | hexa2 } ]  
[ WMODALC = (file-allocation-parameters) ]  
[ WRKFILE1 = lib78 ]  
[ WRKALC1 = ( file-allocation-parameters ) ]  
  .  
  .  
[ WRKFILE16 = lib78 ]  
[ WRKALC16 = ( file-allocation-parameters ) ]
```

**Parameters:**

FILE	the file resulting from the sort, expressed as an output-file description. If FILE=INFILE is specified, the resulting file overwrites the input file.								
INFILE	the file to be sorted, expressed as an input-file-description.								
DYNALC	requests the dynamic allocation of the outfile and specifies its file status:  <table> <tr> <td>CAT</td> <td>cataloged</td> </tr> <tr> <td>CATi</td> <td>cataloged in one of the attached catalogs (i=1 to 5)</td> </tr> <tr> <td>UNCAT</td> <td>uncataloged</td> </tr> <tr> <td>TEMPRY</td> <td>temporary</td> </tr> </table> <p>When DYNALC is not specified for a non-preallocated output file, you must supply consistent DEF/OUTDEF and ALC/OUTALC parameter values. DYNALC is not allowed if FILE = INFILE is specified.</p>	CAT	cataloged	CATi	cataloged in one of the attached catalogs (i=1 to 5)	UNCAT	uncataloged	TEMPRY	temporary
CAT	cataloged								
CATi	cataloged in one of the attached catalogs (i=1 to 5)								
UNCAT	uncataloged								
TEMPRY	temporary								
PADCHAR	the character to be used for padding output records (the character itself or its hexadecimal equivalent are accepted). The default filler is the "blank" character.								
NBSORT	the number of "slave sorts" for a parallel sort ( $\geq 2$ , if it is specified).  <p>For a non-memory sort with (WRKSIZE, WRKVOL), or with (WRKFILE, WRKALC): <math>2 \leq \text{NBSORT} \leq 32</math>. In this case, the media list of the workfile assigned may limit NBSORT at execution time.</p> <p>For a non-memory sort with (WMODALC, WRKFILEi, WRKALCi), NBSORT must be equal to the number of assigned WRKFILEi (<math>\geq 2</math>).</p> <p>If no workfile is assigned at all, the implicit VOLSET or the RESIDENT disks will be used to dynamically assign NBSORT workfiles, except if the input file can be sorted in memory.</p> <p>If NBSORT is unspecified, the sort is mono-process and only one workfile may be assigned (WRKSIZE, WRKVOL), or (WRKFILE, WRKALC), or no workfile at all (implicit dynamic allocation if needed, or memory sort).</p>								
ALLOCATE	the resulting file-allocation parameters expressed as a file-allocation parameter group.								
DEF/INDEF	additional processing parameters for the output and input files, expressed as file-define parameter groups.								

## SORT\_FILE Command (GCL)

DSLFILE	the file containing the Data Services Language (DSL) commands that define the Sort options. The DSL for Sort is described in Chapter 6. The DSL description of the file may also be entered directly from the terminal by setting the parameter DSLFILE=:TN. The terminal file is closed by typing the character "/".
WRKSIZE	(to be used in conjunction with WRKVOL) the number of cylinders to be used as auxiliary space for sorting, optionally followed by the increment by which it is extended when exhausted. When WRKSIZE=0 is specified, the auxiliary space is in main memory (memory sort).
WRKVOL	the volume on which auxiliary space for sorting is to be allocated. Default allocation is on a resident disk volume if the VOLSET facility is not active, and on the default volset if the VOLSET facility is active. If the full VOLSET facility is active, a non-default volset can be specified using the syntax \$VOLSET:volset-name.
WRKFILE	the identification of the sort workfile.
WRKALC	introduces the file-allocation parameter group for the sort workfile.
-----	
START	the rank of the record at which the sort is to begin. Values from 1 to 2147483647 are acceptable. The default value is 1.
HALT	the rank of the record at which the sort is to end. Acceptable values are the same as for START. The default value is end of file.
REPORT	the type of report required, as follows:  AUDIT causes diagnostic and process error information to be printed as well as a statistical summary of record processing (default in batch mode). PARAM causes a list of parameters to be printed, in addition to diagnostic and process error information. ALL combines AUDIT and PARAM. NONE limits the report to process error information (default in interactive mode).
INVREC	the action to be taken if an invalid record is encountered in the input file, as follows:  ABORT Sort terminates (default value), CONTINUE Sort continues.

## Sort/Merge User's Guide

ERROPT	<p>the action to be taken with invalid record, as follows:</p> <table><tr><td>PRINTID</td><td>print the rank of the record (default value),</td></tr><tr><td>PRINT</td><td>print the rank and the contents of the record,</td></tr><tr><td>LOG</td><td>write the record in LOGFILE,</td></tr><tr><td>IGNORE</td><td>print no message.</td></tr></table>	PRINTID	print the rank of the record (default value),	PRINT	print the rank and the contents of the record,	LOG	write the record in LOGFILE,	IGNORE	print no message.
PRINTID	print the rank of the record (default value),								
PRINT	print the rank and the contents of the record,								
LOG	write the record in LOGFILE,								
IGNORE	print no message.								
LOGFILE	the file into which errors are logged.								
INFILEi	up to eight files to be concatenated to INFILE in order to define the input file to be sorted. Each file is expressed as an input-file-description parameter group. Restrictions applicable to these files are the same as those for concatenated files in the ASGi parameter group.								
PRTFILE	the report file.								
REPEAT	if 1, checkpoints are to be taken during the sort; if 0 (default), no checkpoint is to be taken. A parallel sort never takes checkpoints except at the beginning of the step.								
SIZE	<p>the total memory in units of 1024 bytes to be allocated to Sort for the execution. This includes code, data, and buffers.</p> <p>For a mono-process sort, the SIZE is limited to 64 Mbytes.</p> <p>For a parallel sort, the limit is <math>NBSORT*32</math> Mbytes. The maximum limit is therefore <math>32*32</math> Mbytes, though this will depend on the machine configuration. If SIZE is less than <math>NBSORT*32</math> Mbytes, then each slave sort will use (approximately) <math>SIZE/NBSORT</math> Mbytes.</p>								
WMODALC	(for parallel sort only) provides default SIZE, INCRSIZE, and UNIT values for the sort workfiles allocated by the WRKALCi parameters. If WRKALCi does not specify its own SIZE, INCRSIZE or UNIT, then the value specified for WMODALC applies.								
WRKFILEi	(for parallel sort only) the identification of workfile i.								
WRKALCi	(for parallel sort only) introduces a file-allocation parameter group for workfile i.								



## SORT\_FILE Command (GCL)

### **Constraints:**

- START may not exceed HALT.
- If ERROPT is LOG, LOGFILE must be specified.
- WRKFILEi and WRKALCi are exclusive of WRKSIZE, WRKVOL, WRKFILE, WRKALC.
- WMODALC, WRKALCi, and WRKFILEi are meaningful only if NBSORT is greater than 1 (or unspecified).

### **Examples:**

```
SORT MYDIR.SORTEDFILE MYDIR.SOURCE DSLFILE=MYDIR.SRTCMSDS
      {sort a file into a new one}
```

```
SORT INFILE MYDIR.SOURCE DSLFILE=MYDIR.SRTCMSDS
      {sort a file overwriting it with the result}
```

```
SORT MYDIR.RES ALLOCATE=(SIZE=10)
      INFILE1=MYDIR.S1 INFILE2=MYDIR.S2 INFILE3=MYDIR.S3
      DSLFILE=MYDIR.SRTCMSDS
      {sort three concatenated files into a new one}
```

## Sort/Merge User's Guide

## E. MERGE\_FILE Command (GCL)

### E.1 EXECUTING MERGE\_FILE

You can execute a merge in interactive or batch mode using the GCL command MERGE\_FILE (alias MRGF).

The MERGE\_FILE command has a format similar to the format of the JCL statement MERGE. The visibility is similar to Data Management Utility commands.

The MERGE\_FILE command is described in the *IOF Terminal User's Reference Manual*.

You can also execute the JCL statement GMERGE (described in Chapter 4) in interactive mode.

## E.2 MERGE\_FILE (MRGF) COMMAND

### Purpose:

Activates the Merge utility which may be used to merge two to eight sorted files into a new one or into an existing one.

### Syntax:

```
{ MERGE_FILE }
{
{ MRGF
}

{ INFILE1 }
{
} = ( input-file-description )
{ IF1
}

{ INFILE2 }
{
} = ( input-file-description )
{ IF2
}

{ FILE
}
{ OUTFILE } = ( output-file-description )
{ OF
}

{ CAT
}
{ CAT{1|2|3|4|5} }
[ DYNALC = {
} ]
{ UNCAT
}
{ TEMPRY
}

[ INDEF1 = ( file-define-parameters ) ]
[ INDEF2 = ( file-define-parameters ) ]

{ DEF
}
[ {
} = ( file-define-parameters ) ]
{ OUTDEF
}

{ ALLOCATE }
[ {
} = ( file-allocation-parameters ) ]
{ OUTALC
}

{ DSLFILE }
{
} = { file78 | ::TN }
{ COMFILE }
```

## MERGE\_FILE Command (GCL)

```
-----  
  
    { INFILE3 }  
[ {           } = ( input-file-description ) ]  
  { IF3       }  
  
    { INFILE4 }  
[ {           } = ( input-file-description ) ]  
  { IF4       }  
  
    { INFILE5 }  
[ {           } = ( input-file-description ) ]  
  { IF5       }  
  
    { INFILE6 }  
[ {           } = ( input-file-description ) ]  
  { IF6       }  
  
    { INFILE7 }  
[ {           } = ( input-file-description ) ]  
  { IF7       }  
  
    { INFILE8 }  
[ {           } = ( input-file-description ) ]  
  { IF8       }  
  
[ INDEF3 = ( file-define-parameters ) ]  
[ INDEF4 = ( file-define-parameters ) ]  
[ INDEF5 = ( file-define-parameters ) ]  
[ INDEF6 = ( file-define-parameters ) ]  
[ INDEF7 = ( file-define-parameters ) ]  
[ INDEF8 = ( file-define-parameters ) ]  
[ PRTFILE = file78 ]  
[ REPORT = { AUDIT | PARAM | ALL | NONE } ]  
[ BUFFER = dec3 ]  
[ PADCHAR = { char1 | hexa2 } ]
```

**Parameters:**

INFILEi	from two to eight file descriptions defining the files to be merged. These are expressed as input-file-description parameter groups.								
FILE/OUTFILE	the file resulting from the merge, expressed as an output-file description. It must be different from all the input files to be merged.								
DYNALC	requests dynamic allocation of the output file by specifying its file status: <table> <tr> <td>CAT</td> <td>cataloged</td> </tr> <tr> <td>CATi</td> <td>cataloged in one of the attached catalogs (i=1 to 5)</td> </tr> <tr> <td>UNCAT</td> <td>uncataloged</td> </tr> <tr> <td>TEMPRY</td> <td>temporary</td> </tr> </table> <p>When DYNALC is not specified for a non-preallocated output file, you must supply consistent DEF/OUTDEF and ALC/OUTALC parameter values.</p>	CAT	cataloged	CATi	cataloged in one of the attached catalogs (i=1 to 5)	UNCAT	uncataloged	TEMPRY	temporary
CAT	cataloged								
CATi	cataloged in one of the attached catalogs (i=1 to 5)								
UNCAT	uncataloged								
TEMPRY	temporary								
INDEFi/DEF	additional processing parameters for the input files and the output file, expressed as file-description parameter groups.								
ALLOCATE	the resulting file-allocation parameters expressed as a file-allocation parameter group.								
DSLFILE	the file containing the Data Services Language (DSL) commands that define the Merge options. The DSL for Merge is described in Chapter 6. The DSL description of the file may also be entered directly from the terminal by setting the parameter DSLFILE=::TN. The terminal file is closed by typing the character "/".								
PRTFILE	the report file. Interactive merge works exactly like batch merge, and the default merge PRTFILE is the public SYSOUT, never the user's terminal.								
REPORT	the type of report required, as follows: <table> <tr> <td>AUDIT</td> <td>causes diagnostic and process error information to be printed as well as a statistical summary of record processing (default in batch mode).</td> </tr> <tr> <td>PARAM</td> <td>causes a list of parameters to be printed, in addition to diagnostic and process error information.</td> </tr> <tr> <td>ALL</td> <td>combines AUDIT and PARAM.</td> </tr> <tr> <td>NONE</td> <td>limits the report to process information (default in interactive mode).</td> </tr> </table>	AUDIT	causes diagnostic and process error information to be printed as well as a statistical summary of record processing (default in batch mode).	PARAM	causes a list of parameters to be printed, in addition to diagnostic and process error information.	ALL	combines AUDIT and PARAM.	NONE	limits the report to process information (default in interactive mode).
AUDIT	causes diagnostic and process error information to be printed as well as a statistical summary of record processing (default in batch mode).								
PARAM	causes a list of parameters to be printed, in addition to diagnostic and process error information.								
ALL	combines AUDIT and PARAM.								
NONE	limits the report to process information (default in interactive mode).								

## MERGE\_FILE Command (GCL)

BUFFER	the total input/output file buffer memory space to be used, expressed in units of 1024 bytes. The default value is 30, but any sufficient lower or higher value may be specified, provided that it is lower than 512.
PADCHAR	the character to be used for padding output records (the character itself or its hexadecimal equivalent are accepted). The default filler is the "blank" character.

### **Constraints:**

None.

### **Examples:**

```
MRGF MYDIR.MRG1 MYDIR.MRG2 MYDIR.MRGOUT DSLFILE=MYDIR.CMDS  
      {merge two files into a third one}
```

```
MRGF OUTFILE=MYDIR.RES ALLOCATE=(SIZE=10)  
      INFILE1=MYDIR.M1 INFILE2=MYDIR.M2  
      INFILE3=MYDIR.M3 DSLFILE=MYDIR.MRGCMS  
      {merge three files into a new one being allocated}
```

## Sort/Merge User's Guide



## F. UFAS Indexed Output File Loaded by Sort or Merge

This Appendix applies to the SORT, GSORT, MERGE, and GMERGE statements.

A UFAS Indexed file is defined with a primary key field (see KEYLOC and KEYSIZE in the JCL statement DEFINE) and possibly with secondary key(s).

This type of file must be pre-allocated or must pre-exist before being loaded by Sort or Merge.

When such a file is to be loaded sequentially, as done by Sort and Merge, the primary key field values must appear in strictly ascending order with no duplicated key values (this is a general UFAS Indexed file loading specification).

This requirement can be met if, in the KEY (or KEYS) entry of the RECORD paragraph (COMFILE), the Sort or Merge key is defined exactly as in the file definition, and the possible duplicate key values are eliminated (see DELETE and SUM commands, respectively in FUNCTION and RECORD paragraph of COMFILE).

For example, if a file to be loaded by Sort or Merge is defined with:

KEYLOC = 5 and KEYSIZE = 8 in a \$DEFINE basic JCL statement, or in equivalent JCL/GCL objects

KEYLOC = 6 and KEYSIZE = 8 in a PREALLOC or equivalent JCL/GCL statement

the RECORD paragraph for Sort or Merge would be RECORD: KEY = (6,8)

If the Sort or Merge input records are also submitted to ARRANGE commands, remember that the key specifications describe the key positions and key lengths before ARRANGE processing.

Any other Sort or Merge key definition having the same effect (known by the user) as the simple example above is possible.

If UFAS secondary keys also exist, a SORTIDX step (or a SORT\_INDEX GCL command) is necessary after loading the Indexed file using Sort or Merge, before the file can be used through secondary keys.

## Sort/Merge User's Guide

## **G. Use of DATAFORM = {SARF/SSF} and of [TRUNCSSF/NTRNCSSF] by SORT and MERGE**

Unless TRUNCSSF is explicitly specified in the INFILE parameter group, the **read of the SORT input file is implicitly submitted to TRUNCSSF processing** when it resides **on tape or cartridge** or when its FILEFORM is **BFAS disk or LINKQD**. This processing is common to any concatenated sort input files.

**Sort or Merge output files are implicitly created in SARF format** (no SSF control records nor SSF prefixes are added) unless DATAFORM = SSF is specified inside the OUTFILE parameter group concerning a tape or cartridge file, or a LINKQD file.

When DATAFORM = SSF is so validated for a SORT or MERGE output file, its first record will be an "SSF control record 101" and the data part of each data record will be SSF-prefixed by the value "008400404040E240" X.

When the SORT or MERGE output file is the public SYSOUT, it may be created either in SARF format (default) or in SSF format by SORT.

## Sort/Merge User's Guide

## **H. Restrictions to Checkpoint/Restart with Cartridge Files for Sort/Merge**

This Appendix applies only to a mono-process sort.

The low level cartridge volumes (DVC = CT/M6) do not allow checkpoint/restart processing when they are opened in output or append mode, and not positioned at file beginning, file end or volume end.

When this type of media supports Sort/Merge PRTFILE, Sort/Merge OUTFILE or Sort LOGFILE, a REPEAT specification in Sort/Merge JCL statement will cause a Sort/Merge abnormal termination with appropriate message and return-code.

No improvement of Sort/Merge checkpoints is possible with such a device because all that could be done is to avoid taking checkpoints when the device context is not suitable, and this would not be a real improvement compared to what exists already.

## Sort/Merge User's Guide

# I. Sort/Merge and the UFAS Access Method

## I.1 SORT

The conditions under which Sort calls the UFAS Access Method are given below.

This Appendix gives detailed technical information. It is not essential for reading for users.

### **Mono-Process Sort**

For files of UFAS Indexed Organization, Sort always calls the UFAS access method (but the UFAS BPB does not apply in this case).

For **Input** UFAS Sequential or UFAS Relative files, Sort calls the UFAS access method in the following cases:

- SHARE = FREE, DIR, ONEWRITE, or (SHARE=MONITOR and READLOCK=STAT)
- or "all volumes are not mounted for the file",
- or TRUNCSSF,
- or concatenation,
- or REPEAT and CKPTLIM,
- or the DSL contains: KEYADDR or ADDATA or ADDRROUT.

For **Output** UFAS Sequential or UFAS Relative files, Sort calls the UFAS access method in the following cases:

- SHARE not = NORMAL,
- or "all volumes are not mounted for the file",
- or REPEAT and CKPTLIM.

**NOTE:** For SHARE = MONITOR, (READLOCK = STAT) or (ACCESS = SPREAD or SPWRITE) are mandatory for INFILE. For OUTFILE, ACCESS = SPWRITE is mandatory when SHARE = MONITOR.

### **Multi-Process Sort**

For files of UFAS Indexed Organization, Sort always calls the UFAS access method (but the UFAS BPB does not apply in this case).

For **Input** UFAS Sequential or UFAS Relative files, Sort calls the UFAS access method in the following cases:

- SHARE = FREE, DIR, ONEWRITE, or (SHARE=MONITOR and READLOCK=STAT)
- or "all volumes are not mounted for the file",
- or TRUNCSSF,
- or concatenation,
- or the DSL contains:  
(KEYADDR or ADDATA or ADDRROUT)

For **Output** UFAS Sequential or UFAS Relative files, Sort calls the UFAS access method in the following cases:

- SHARE not = NORMAL,
- or "all volumes are not mounted for the file".

**NOTE:** For SHARE = MONITOR, (READLOCK = STAT) or (ACCESS = SPREAD or SPWRITE) are mandatory for INFILE. For OUTFILE, ACCESS = SPWRITE is mandatory when SHARE = MONITOR.

## **I.2 MERGE**

Merge calls the UFAS access method under the same conditions as Sort (except that the DSL conditions do not apply).



# Index

## A

ABORT  
     sort/merge define-parameter 3-15  
 ADDATA DSL parameter 5-5, 5-6  
 ADDRFORM  
     sort/merge define-parameter 3-17  
 ADDROUT DSL parameter 5-5, 5-6, 5-20  
 ALLOCATE JCL statement 7-4  
 ALLOCATE parameter  
     MERGE\_FILE command E-4  
     SORT\_FILE command D-4  
 ARRANGE DSL command 5-20, 5-22, 5-23, 6-6, 7-2  
 ASCII collating sequence 5-3, C-8, C-9  
 ASSIGN JCL statement 3-2, 3-4, 3-13  
 ATTACH JCL statement 3-6  
 AVERAGE DSL command 5-8, 7-1

## B

Basic Sort/Merge 1-6  
 Batch Booster 1-8  
 batch merge E-1  
 batch sort D-1  
 BFAS files 3-15, 3-16, 3-20  
 BLKOFF  
     sort/merge define-parameter 3-15  
 BLKSIZE  
     sort/merge define-parameter 3-16  
 BPB 3-17  
     sort/merge define-parameter 3-17  
     usage with sort 1-8  
 BUFFER parameter  
     GMERGE statement 4-12  
     MERGE statement 3-13  
     MERGE\_FILE command E-5

## C

cartridge files H-1  
 checkpoint/restart 7-11, H-1  
 COBOL merge 1-10  
 COBOL sort 1-10  
 COLLATE DSL command 5-3, 5-4, 6-2  
 collating sequences  
     ASCII 5-3  
     EBCDIC 5-3  
     H200 5-3, 5-4  
     'hexa512' 5-3, 5-4  
     Series 100 5-3  
 Collating sequences  
     ASCII C-8, C-9  
     EBCDIC C-1, C-4  
     H200 C-7  
     Series 100 C-5, C-6  
 COMFILE parameter  
     MERGE statement 3-13  
     SORT statement 3-5  
 comparison operations 5-15  
 Comparison operations 5-17  
 condition elements 5-15, 5-16, 5-17  
 Condition elements 5-16

## D

DATA DSL parameter 5-5  
 Data Services Language (DSL) 1-5  
 DATAFORM G-1  
     sort/merge define-parameter 3-17  
 DATAFORM parameter  
     SORT statement 3-5  
 DEBUG DSL command 5-9, 6-3  
 DEF parameter  
     MERGE\_FILE command E-4  
     SORT\_FILE command D-4  
 DEFINE JCL statement 3-4, 3-13, 3-15



## Index

MERGE\_FILE command E-4  
 INFILE parameter  
   SORT statement 3-4  
   SORT\_FILE command D-4  
 INFILEi parameter  
   GMERGE statement 4-12  
   MERGE\_FILE command E-4  
   SORT\_FILE command D-6  
 INFILEn parameter  
   MERGE statement 3-13  
   SORT statement 3-4  
 INFILES parameter  
   GSORT statement 4-3  
 interactive merge E-1  
 interactive sort D-1  
 INTINFO DSL command 5-7, B-2  
 INVREC parameter  
   GSORT statement 4-8  
   SORT statement 3-8  
   SORT\_FILE command D-5

## J

JCL statements  
   ALLOCATE 7-4  
   ASSIGN 3-4, 3-13  
   ATTACH 3-6  
   DEFINE 3-4, 3-13, 3-15  
   PREALLOC 7-4  
   SIZE 3-13  
   STEP 3-9, 3-13

## K

KEY(S) DSL command 5-12, 5-14, 6-4, 6-6, 7-1  
 KEYADDR DSL parameter 5-5, 5-6, 5-20

## L

large memory sort  
   dumps 1-10  
   overview 1-7  
 LOGALC parameter  
   GSORT statement 4-4  
 LOGDEF parameter  
   GSORT statement 4-4  
 LOGFILE parameter  
   SORT statement 3-9  
   SORT\_FILE command D-6  
 LOGOUT parameter  
   GSORT statement 4-4

## M

maximum record size 7-3  
 MEM parameter  
   GSORT statement 4-9  
 Merge  
   definition 1-1  
   error messages 1-5  
   examples 3-22, 3-23  
   features 1-2  
   rules 2-3  
   using 1-3  
 MERGE  
   syntax 3-11  
 MERGE\_FILE command E-2  
 MIN DSL command 5-8, 7-1  
 MODALC parameter  
   GSORT statement 4-6  
 mono-process sort 1-6  
 MRGF command E-2  
 multi-process sort 1-7

## N

NBBUF  
   sort/merge define-parameter 3-17  
 NBSORT DSL command 5-9  
 NBSORT parameter  
   GSORT statement 4-9  
   SORT\_FILE command D-4  
 NPOOL parameter  
   SORT statement 3-6  
 NTRNCSSF  
   sort/merge define-parameter 3-18  
 numeric condition 5-18

## O

OMIT DSL command 5-15, 5-19  
 OUTALC parameter  
   GSORT statement 4-3  
 OUTDEF parameter  
   GSORT statement 4-3  
 OUTFILE parameter  
   GSORT statement 4-3  
   MERGE statement 3-13  
   MERGE\_FILE command E-4  
   SORT statement 3-4  
 OUTPUT DSL command 5-5, 5-6, 5-20, 6-4, 6-6, 7-2

## Sort/Merge User's Guide

### P

PADCHAR parameter	
MERGE_FILE command	E-5
SORT_FILE command	D-4
parallel sort	1-7
POOL parameter	
SORT statement	3-6
PREALLOC JCL statement	7-4
PRTALC parameter	
GSORT statement	4-4
PRTDEF parameter	
GSORT statement	4-4
PRTFILE parameter	
MERGE statement	3-13
MERGE_FILE command	E-4
SORT statement	3-8
SORT_FILE command	D-6
PRTOUT parameter	
GSORT statement	4-4

### Q

Queued files	2-2
--------------	-----

### R

read ahead	1-8
RECFORM	
sort/merge define-parameter	3-16
RECORD (DSL paragraph)	5-1, 5-11, 6-1, 6-4, 6-6
RECSIZE	
sort/merge define-parameter	3-16
REPEAT parameter	
GSORT statement	4-8
SORT statement	3-9
SORT_FILE command	D-6
REPORT parameter	
GMERGE statement	4-12
GSORT statement	4-8
MERGE statement	3-13
MERGE_FILE command	E-4
SORT statement	3-8
SORT_FILE command	D-5
RETCODE	
sort/merge define-parameter	3-15

### S

SARF format	G-1
-------------	-----

Series 100 collating sequence	5-3, C-5, C-6
SIZE JCL statement	3-9, 3-13
size of disk work file	7-4
size of sorted records	7-2
SIZE parameter	
GSORT statement	4-8
SORT statement	3-6, 3-8
SORT_FILE command	D-6
SIZEOPT parameter	
GSORT statement	4-10
MERGE statement	3-13
SORT statement	3-9
SKIP	
sort/merge define-parameter	3-15
sort	
dumps	1-10
large memory	1-7
mono-process	1-6
multi-process	1-7
parallel	1-7
using	1-3
with BPB	1-8
with volset	1-9
Sort	
definition	1-1
error messages	1-5
examples	3-22
features	1-2
rules	2-2
work file space	1-5
SORT	
examples	3-23
performance	7-9
sort record size	7-2
sort/merge define-parameters	3-14
ABORT	3-15
ADDRFORM	3-17
BLKOFF	3-15
BLKSIZE	3-16
BPB	3-17
DATAFORM	3-17
DUMMYREC	3-17
ERROPT	3-15
FILEFORM	3-15
IGNORE	3-15
NBBUF	3-17
NTRNCSSF	3-18
RECFORM	3-16
RECSIZE	3-16
RETCODE	3-15
SKIP	3-15
TRUNCSSF	3-18
WRCHECK	3-17
Sort/Merge report	7-4, B-1, B-3
Sort/Merge Report	7-6
SORT_FILE GCL command	D-1, D-2

## Index

sorting variable length records 7-1  
 SORTSIZE DSL command 2-2, 3-5, 5-8  
 SRTF GCL command D-2  
 SSF format G-1  
 START parameter  
     GSORT statement 4-8  
     SORT statement 3-9  
     SORT\_FILE command D-5  
 STEP JCL statement 3-9, 3-13  
 STEPOPT parameter  
     GSORT statement 4-10  
     MERGE statement 3-13  
     SORT statement 3-9  
 SUM DSL command 5-20, 6-4, 6-6, 7-1  
 SYSOUT parameter  
     GSORT statement 4-4

### T

TRUNCSSF G-1  
     sort/merge define-parameter 3-18

### U

UFAS files 2-2, 2-3, 3-15, 3-20, F-1  
 using merge 1-3  
 using sort 1-3

### V

variable length records 7-1  
 VBO (Variable Block Organization) 3-15  
 volset 1-9

### W

WKALC parameter  
     GSORT statement 4-4  
 WKALCi parameter  
     GSORT statement 4-7  
 WKASGi parameter  
     GSORT statement 4-7  
 WKDISK parameter  
     SORT statement 3-5  
 WKFILE parameter  
     GSORT statement 4-4  
 WMODALC parameter  
     SORT\_FILE command D-6  
 WRCHECK  
     sort/merge define-parameter 3-17  
 WRDISK DSL command 5-9  
 write behind 1-8  
 WRKALC parameter  
     SORT\_FILE command D-5  
 WRKALCi parameter  
     SORT\_FILE command D-6  
 WRKFILE parameter  
     SORT\_FILE command D-5  
 WRKFILEi parameter  
     SORT\_FILE command D-6  
 WRKSIZE parameter  
     SORT\_FILE command D-5  
 WRKVOL parameter  
     SORT\_FILE command D-5

## Sort/Merge User's Guide

## Technical publication remarks form

<b>Title :</b>	DPS7000/XTA NOVASCALE 7000 SORT/MERGE Utilities User's Guide File and Volume Management
----------------	--

<b>Reference N° :</b>	47 A2 08UF 07
-----------------------	---------------

<b>Date:</b>	<b>February 2005</b>
--------------	----------------------

### ERRORS IN PUBLICATION

--

### SUGGESTIONS FOR IMPROVEMENT TO PUBLICATION

--

Your comments will be promptly investigated by qualified technical personnel and action will be taken as required.  
If you require a written reply, please include your complete mailing address below.

NAME : \_\_\_\_\_ Date : \_\_\_\_\_

COMPANY : \_\_\_\_\_

ADDRESS : \_\_\_\_\_

Please give this technical publication remarks form to your BULL representative or mail to:

Bull - Documentation Dept.  
1 Rue de Provence  
BP 208  
38432 ECHIROLLES CEDEX  
FRANCE  
info@frec.bull.fr

# Technical publications ordering form

To order additional publications, please fill in a copy of this form and send it via mail to:

**BULL CEDOC**  
**357 AVENUE PATTON**  
**B.P.20845**  
**49008 ANGERS CEDEX 01**  
**FRANCE**

**Phone:** +33 (0) 2 41 73 72 66  
**FAX:** +33 (0) 2 41 73 70 66  
**E-Mail:** [srv.Duplicopy@bull.net](mailto:srv.Duplicopy@bull.net)

CEDOC Reference #	Designation	Qty
-- -- [ ]		
-- -- [ ]		
-- -- [ ]		
-- -- [ ]		
-- -- [ ]		
-- -- [ ]		
-- -- [ ]		
-- -- [ ]		
-- -- [ ]		
-- -- [ ]		
-- -- [ ]		
-- -- [ ]		
[ ] : The latest revision will be provided if no revision number is given.		

NAME: \_\_\_\_\_ Date: \_\_\_\_\_

COMPANY: \_\_\_\_\_

ADDRESS: \_\_\_\_\_

PHONE: \_\_\_\_\_ FAX: \_\_\_\_\_

E-MAIL: \_\_\_\_\_

**For Bull Subsidiaries:**

Identification: \_\_\_\_\_

**For Bull Affiliated Customers:**

Customer Code: \_\_\_\_\_

**For Bull Internal Customers:**

Budgetary Section: \_\_\_\_\_

**For Others: Please ask your Bull representative.**





**BULL CEDOC**  
**357 AVENUE PATTON**  
**B.P.20845**  
**49008 ANGERS CEDEX 01**  
**FRANCE**

REFERENCE  
**47 A2 08UF 07**