604849

CONTROL DATA CORPORATION

SORT/MERGE VERSION 5 INSTANT

CDC® OPERATING SYSTEMS
NOS 2
NOS/BE 1

SORT/MERGE PARAMETERS

PROCEDURE CALLS

6048490

CONTROL DATA CORPORATION

SORT/MERGE VERSION 5 INSTANT

CDC® OPERATING SYSTEMS
NOS 2
NOS/BE 1

REVISION RECORD

Revision	Description
A (11/20/81)	Original release at PSR level 552.
B (02/26/82)	This manual is revised to reflect the support of NOS 2 at PSR level 552. Various technical and editorial changes have been made. This is a complete reprint.

REVISION LETTERS I, O, Q, AND X ARE NOT USED

Address comments concerning this manual to:

CONTROL DATA CORPORATION
Publications and Graphics Division
215 MOFFETT PARK DRIVE
SUNNYVALE, CALIFORNIA 94086

LIST OF EFFECTIVE PAGES

New features, as well as changes, deletions, and additions to information in this manual are indicated by bars in the margins or by a dot near the page number if the entire page is affected. A bar by the page number indicates pagination rather than content has changed.

Page	Revision
Front Cover	-
Inside Front Cover	B
Title Page	-
ii	B
iii/iv	B
v	B
vi	B
vii	B
viii	A
ix	A
1 thru 3	B
4 thru 7	A
8	B
9 thru 11	A
12 thru 15	B
16	A
17	A
18	B
19	A
20	A
21 thru 24	B
25	A
26	B
27	A
28	A
29	B
30 thru 33	A
34	B
35	B
36 thru 40	A
Back Cover	-

PREFACE

This instant provides a convenient summary of the Control Data® Sort/Merge Version 5.0 utility, which operates under the control of the NOS 2 and NOS/BE 1 operating systems for the CDC® CYBER 170 Computer Systems.

This instant assumes that you are an application programmer familiar with Sort/Merge and the operating system that you use. If you use procedure calls to Sort/Merge, the instant assumes that you are familiar with the calling language.

You can find related information in the publications listed below; the publications are listed within groupings that indicate relative importance to you.

The NOS manual abstracts and the NOS/BE manual abstracts are pocket-sized manuals containing brief descriptions of the contents and intended audience of all NOS and NOS/BE manuals and NOS and NOS/BE product set manuals. The manual abstracts can be useful in determining which manuals are of greatest interest to you. The Software Publications Release History serves as a guide in determining which revision level of software documentation corresponds to the Programming System Report (PSR) level of software installed at your site.

The following manuals are of primary interest:

Publication	Publication Number
COBOL Version 5 Reference Manual	60497100
CYBER Record Manager Basic Acess Methods Version 1.5 Reference Manual	60495700
FORTRAN Version 5 Reference Manual	60481300
Sort/Merge Version 5 Reference Manual	60484800

The following manuals are of secondary interest:

Publication	Publication Number
INTERCOM Version 5 Reference Manual	60455010
Network Products Interactive Facility Version 1 Reference Manual	60459330
Network Products Remote Batch Facility Version 1 Reference Manual	60499600
NOS Version 2 Manual Abstracts	60485500
NOS Version 2 Reference Set, Volume 3 System Commands	60459680
NOS/BE Version 1 Manual Abstracts	84000470
NOS/BE Version 1 Reference Manual	60493800
Software Publications Release History	60481000

CDC manuals can be ordered from Control Data Corporation, Literature and Distribution Services, 308 North Dale Street, St. Paul, Minnesota 55103.

This manual describes a subset of the features and parameters documented in the Sort/Merge Version 5 Reference Manual. Control Data cannot be responsible for the proper functioning of any features or parameters not documented in the Sort/Merge Version 5 Reference Manual.

CONTENTS

FIGURES

TABLES

NOTATIONS

Unless otherwise specified, the conventions described in the following paragraphs are used throughout this manual in parameter and procedure call formats.

UPPERCASE	Uppercase letters indicate words, acronyms, or mnemonics required as input by Sort/Merge. Parameter keywords and interactive commands are included in this category.
lowercase	Lowercase letters indicate words or symbols that you must supply.
...	Ellipses indicate that omitted entities repeat the form and function of the last entity given.
[ ]	Brackets indicate an optional portion of a parameter or procedure call format.
..	Two periods in a parameter format indicate a range of letters or digits. The periods are part of the syntax and must be written where shown.
Δ	The delta symbol represents a blank used as a separator.

Unless otherwise indicated, all numbers in text are decimal values.

INTRODUCTION

Sort/Merge Version 5 is a set of powerful and efficient routines that provides high-speed sorting or merging of records for batch processing or interactive processing. This section introduces the available features and the default CYBER Record Manager (CRM) file characteristics for Sort/Merge.

FEATURES

Sort/Merge offers the following features:

- A sort or merge specification with a single control statement. Using the SORT control statement, records are rearranged into the order you specify. Using the MERGE control statement, two or more sorted files are combined into one file.
- An optional directive file for sort or merge specification
- The capability of sorting or merging records from as many as 100 files with one call to Sort/Merge
- Character and noncharacter sort key types
- Five predefined collating sequences
- User-defined collating sequences
- The capability of summing numeric fields
- The capability of sorting any CRM sequential file and any record type (except U)
- A tutorial dialog for interactive sort or merge specification
- A set of procedures for sort or merge processing from within a program written in FORTRAN, COBOL, or some other languages
- Owncode routines to insert, substitute, modify, or delete records during Sort/Merge processing

FILE CHARACTERISTICS

Sort/Merge uses the CRM file processing and input/output capabilities. The default file characteristics specified by CRM apply to all Sort/Merge input and output data files, unless this instant indicates otherwise.

The file organization must be sequential. The default file characteristics are BT=C, RT=Z, FL (or MRL)=150. Other file characteristics can be set by the FILE control statement.

SORT KEYS

A sort key is a field of data within each record in either an input file to be sorted or in a presorted file to be merged with one or more other files. The key determines the order in which records are output. Key fields must occur in the same position and be the same length in all records. The total number of key characters must be less than 256 (if you also specify sum fields, the total number of characters in the key and sum fields together must be less than 256). The key and sum fields must not overlap.

A file can be sorted on more than one key. The first key you specify is the major sort key and the following keys are minor sort keys.

KEY STARTING LOCATION AND LENGTH

You define the starting location of the sort key by specifying the first byte or bit of the field. Specify the length of the sort key by the number of bytes or bits in the field, or by the last byte or bit of the field. The leftmost byte or bit in a record is counted as number 1. For character data, each character is six bits and occupies one byte. Specify the sort key location and length with the KEY parameter or SM5KEY procedure call parameter.

TYPE OF DATA

You specify the type of data in a key field with a collating sequence for character data or a numeric data format for numeric or noncharacter data. Table 1 summarizes character and noncharacter data types and the associated sort key type.

COLLATING SEQUENCE

To determine character precedence for alphanumeric data, specify one of the predefined collating sequences or define your own collating sequence. Choose the collating sequence with the KEY parameter or SM5KEY procedure call. The data must be display code characters. The predefined collating sequences are ASCII6, COBOL6, DISPLAY, EBCDIC6, or INTBCD. Table 2 shows these predefined collating sequences. The default is ASCII6. See the Sort/Merge reference manual for a description of these collating sequences.

TABLE 1. DATA IN SORT KEY FIELDS

Type	Internal Representation	Data in Field	Type Specified by	Data Ordered According to
Character	Display code	Alphabetic	Name of a collating sequence	Specified collating sequence
		Numeric	Name of a collating sequence	Specified collating sequence
			Name of a numeric data format	Numeric value
Noncharacter	Binary value	Numeric	Name of a numeric data format	Numeric value

TABLE 2. PREDEFINED COLLATING SEQUENCES

Collating Sequence		ASCII6		COBOL6		DISPLAY		EBCDIC6		INTBCD	
Decimal	Octal	Graphics	Display Code	Graphics	Display Code	Graphics	Display Code	Graphics	Display Code	Graphics	CDC INTBCD Code
00	00	blank	55	blank	55	:†	00	blank	55	0	00
01	01	!	66	≤	74	A	01	.	57	1	01
02	02	"	64	%†	63†	B	02	<	72	2	02
03	03	#	60	[	61	C	03	(	51	3	03
04	04	$	53	→	65	D	04	+	45	4	04
05	05	%†	63†	≡	60	E	05	|	66	5	05
06	06	&	67	∧	67	F	06	&	67	6	06
07	07	'	70	↑	70	G	07	$	53	7	07
08	10	(	51	↓	71	H	10	*	47	8	10
09	11	)	52	>	73	I	11	)	52	9	11
10	12	*	47	≥	75	J	12	;	77	:	12
11	13	+	45	⌐	76	K	13	⌐	76	=	13
12	14	,	56	.	57	L	14	-	46	≠	14
13	15	-	46	)	52	M	15	/	50	≤	15
14	16	.	57	;	77	N	16	,	56	%	16
15	17	/	50	+	45	O	17	%†	63†	[	17
16	20	0	33	$	53	P	20	_	65	+	20
17	21	1	34	*	47	Q	21	>	73	A	21
18	22	2	35	-	46	R	22	?	71	B	22
19	23	3	36	/	50	S	23	:	00	C	23

TABLE 2. PREDEFINED COLLATING SEQUENCES (Contd)

Collating Sequence		ASCII6		COBOL6		DISPLAY		EBCDIC6		INTBCD	
Decimal	Octal	Graphics	Display Code	Graphics	Display Code	Graphics	Display Code	Graphics	Display Code	Graphics	CDC INTBCD Code
20	24	4	37	,	56	T	24	#	60	D	24
21	25	5	40	(	51	U	25	@	74	E	25
22	26	6	41	=	54	V	26	'	70	F	26
23	27	7	42	≠	64	W	27	=	54	G	27
24	30	8	43	<	72	X	30	"	64	H	30
25	31	9	44	A	01	Y	31	¢	61	I	31
26	32	:†	00	B	02	Z	32	A	01	<	32
27	33	;	77	C	03	0	33	B	02	.	33
28	34	<	72	D	04	1	34	C	03	)	34
29	35	=	54	E	05	2	35	D	04	≥	35
30	36	>	73	F	06	3	36	E	05	⌐	36
31	37	?	71	G	07	4	37	F	06	;	37
32	40	@	74	H	10	5	40	G	07	-	40
33	41	A	01	I	11	6	41	H	10	J	41
34	42	B	02	∨	66	7	42	I	11	K	42
35	43	C	03	J	12	8	43	!	62	L	43
36	44	D	04	K	13	9	44	J	12	M	44
37	45	E	05	L	14	+	45	K	13	N	45
38	46	F	06	M	15	-	46	L	14	O	46

39	47	G	07	N	16	*	47	M	15	P	47
40	50	H	10	O	17	/	50	N	16	Q	50
41	51	I	11	P	20	(	51	O	17	R	51
42	52	J	12	Q	21	)	52	P	20	∨	52
43	53	K	13	R	22	$	53	Q	21	$	53
44	54	L	14	]	62	=	54	R	22	*	54
45	55	M	15	S	23	blank	55	none	75	↑	55
46	56	N	16	T	24	,	56	S	23	↓	56
47	57	O	17	U	25	.	57	T	24	>	57
48	60	P	20	V	26	≡	60	U	25	blank	60
49	61	Q	21	W	27	[	61	V	26	/	61
50	62	R	22	X	30	]	62	W	27	S	62
51	63	S	23	Y	31	%	63†	X	30	T	63
52	64	T	24	Z	32	≠	64	Y	31	U	64
53	65	U	25	:	00	→	65	Z	32	V	65
54	66	V	26	0	33	∨	66	0	33	W	66
55	67	W	27	1	34	∧	67	1	34	X	67
56	70	X	30	2	35	↑	70	2	35	Y	70
57	71	Y	31	3	36	↓	71	3	36	Z	71
58	72	Z	32	4	37	<	72	4	37	]	72
59	73	[	61	5	40	>	73	5	40	,	73
60	74	\	75	6	41	≤	74	6	41	(	74
61	75	]	62	7	42	≥	75	7	42	→	75
62	76	(	76	8	43	┌	76	8	43	≡	76
63	77	_	65	9	44	;	77	9	44	∧	77

†In installations using a 63-graphic set, the % graphic does not exist. The : graphic is display code 63.

You can define your own collating sequence with the SEQx parameters in the SORT5 or MERGE control statement or with SM5SEQx procedure calls. See the sections on the SEQx parameters and the SM5SEQx procedure calls for more details.

NUMERIC DATA FORMAT

You can use numeric data in one of the formats shown in table 3. Numeric data can be signed or unsigned. For character numeric data that is signed, the sign can be a floating sign, a leading or trailing separate character, or an overpunch representation over the leading or trailing digit. Table 4 shows the input and output representation of a sign overpunch for numeric data. See the Sort/Merge 5 reference manual for details about these sign types.

You define numeric key fields by specifying the first byte of the field and either the length of the field or the last byte of the field. For types BINARY_BITS and INTEGER_BITS, specify the first bit position of the field and either the length of the field in bits or the last bit of the field.

SORT ORDER

The sort order for keys is ascending or descending. Ascending order for numeric keys places the record with the largest key value last. Ascending order for character keys uses the order of the collating sequence. Descending order for numeric keys places the record with the largest key value first. Descending order for character keys uses the reverse order of the collating sequence. The default sort order is ascending.

TABLE 3. NUMERIC DATA FORMATS

Name	Data Type	Sign	Comments
NUMERIC_LO	Numeric characters	Leading overpunch	All characters are decimal digits except the leading character, which indicates a sign by an overpunch. Data is ordered according to numeric value with all forms of zero ordered equally.
NUMERIC_LS	Numeric characters	Leading separate	All characters are decimal digits except the leading character, which is a negative or positive sign. Specifying a field that is not at least two characters in length causes a fatal error. Data is ordered according to numeric value with all forms of zero ordered equally.
NUMERIC_TO	Numeric characters	Trailing overpunch	All characters are decimal digits except the trailing character, which indicates a sign by an overpunch. Data is ordered according to numeric value with all forms of zero ordered equally.

TABLE 3. NUMERIC DATA FORMATS (Contd)

Name	Data Type	Sign	Comments
NUMERIC_TS	Numeric characters	Trailing separate	All characters are decimal digits except the trailing character, which is a negative or positive sign. Specifying a field that is not at least two characters in length causes a fatal error. Data is ordered according to numeric value with all forms of zero ordered equally.
NUMERIC_NS	Numeric characters	None	All characters are decimal digits. Data is ordered according to numeric value.
NUMERIC_FS	Leading blanks, numeric characters	- sign for negative values; a + character is not allowed	The field contains leading blanks (leading zeros must be converted to blanks); if the value is negative, the right-most leading blank must be converted to a minus sign. If the field contains no leading blanks, the value must be positive. This format is equivalent to the FORTRAN I format. Data is ordered according to numeric value.

BINARY	Binary integer	None	The field starts and ends on character boundaries. Data is ordered according to numeric value.
BINARY_BITS	Binary integer	None	The field does not start or end on character boundaries. Data is ordered according to numeric value.
INTEGER	One's complement binary integer	Positive if leftmost bit is 0; negative if leftmost bit is 1	The field starts and ends on character boundaries. Data is ordered according to numeric value with negative zero ordered before positive zero.
INTEGER_BITS	One's complement binary integer	Positive if leftmost bit is 0; negative if leftmost bit is 1	The field does not start or end on character boundaries. Data is ordered according to numeric value with negative zero ordered before positive zero.
REAL	Normalized binary real or floating-point number	Signed	The field occupies a full computer word and is aligned on word boundaries. Data is ordered according to numeric value with all forms of zero ordered equally. The order of indefinite values is undefined.

TABLE 4. SIGN OVERPUNCH REPRESENTATION

Sign and Digit	Input Punch	Output Representation
+9	12-9	I
+8	12-8	H
+7	12-7	G
+6	12-6	F
+5	12-5	E
+4	12-4	D
+3	12-3	C
+2	12-2	B
+1	12-1	A
+0	12-0	[†
-0	11-0	]†
-1	11-1	J
-2	11-2	K
-3	11-3	L
-4	11-4	M
-5	11-5	N
-6	11-6	O
-7	11-7	P
-8	11-8	Q
-9	11-9	R

†Valid on NOS with 029 keypunches. See a character set table for NOS/BE representations of +0 and -0.

CONTROL STATEMENT SORTS OR MERGES

The control statements SORT5 and MERGE are used to initiate the processing of your files. These statements use parameters and/or directive files to direct the processing. The following paragraphs define the format and usage of the control statements, parameters, and directive file statements.

SORT5

SORT5 initiates the Sort/Merge sort capabilities. In interactive mode, control statements cannot be continued on more than one line. In batch mode, you can continue the control statement line by placing two or more periods at the end of the line and one period at the beginning of the new line. You can only continue lines that are not ranges. You can use a maximum of 240 characters; the maximum processed per line is 80. The control statement begins with the word SORT5, which must be followed by a period. The SORT5 control statement format is shown below:

```
SORT5.p1,p2,...,pn
```

p — Processing parameters; separators must be commas or spaces, which can be used interchangeably

MERGE

MERGE initiates Sort/Merge merge capabilities. In interactive mode, control statements cannot be continued on more than one line. In batch mode, you can continue the control statement line by placing two or more periods at the end of the line and one period at the beginning of the new line. You can use a maximum of 240 characters for the line and continuation; the maximum number of characters processed per line is 80. The control statement begins with the word MERGE, which must be followed by a period. The format for MERGE is shown below:

```
MERGE.p1,p2,...,pn
```

p — Processing parameters; separators must be commas or spaces, which can be used interchangeably

DIRECTIVE FILES

You can use directive files for more complex sorts or merges and for standardizing installation parameters and collating sequences. A directive file contains processing parameters and parameters in the SORT or MERGE control statements.

Specify directive files with the DIR parameter in the SORT5 or MERGE control statement. The directive file statement formats are:

```
SORT,p1,p2,...,pn

MERGE△ p1△ p2△ ... △pn
```

p — Processing parameter; separators must be commas or spaces, which can be used interchangeably

You can continue the directive file statement by placing two or more periods at the end of the line and one or more spaces at the beginning of the new line. The directive files must be BT=C, RT=Z, and FL=100 or less. Characters in excess of 80 are ignored.

ALPHABETIZED LIST OF PARAMETERS

You can specify the parameters in the SORT5 or MERGE control statement or directive file to direct processing. However, the DIALOG parameter cannot appear in a directive file. When specified with keywords, parameters take one of the following forms, depending on the parameter:

```
keyword=((value-set)[,(value-set)]...)

keyword=value
```

The value or value-set is defined in the individual parameter descriptions. You can delete the keyword by supplying a value or value-set in the position assigned to that particular keyword.

You can mix keyword and positional specifications of parameters. See the Sort/Merge 5 reference manual for more information. Table 5 lists the parameter keywords and the position assigned to each.

CONTROL STATEMENT SORTS OR MERGES

The control statements SORT5 and MERGE are used to initiate the processing of your files. These statements use parameters and/or directive files to direct the processing. The following paragraphs define the format and usage of the control statements, parameters, and directive file statements.

SORT5

SORT5 initiates the Sort/Merge sort capabilities. In interactive mode, control statements cannot be continued on more than one line. In batch mode, you can continue the control statement line by placing two or more periods at the end of the line and one period at the beginning of the new line. You can only continue lines that are not ranges. You can use a maximum of 240 characters; the maximum processed per line is 80. The control statement begins with the word SORT5, which must be followed by a period. The SORT5 control statement format is shown below:

```
SORT5.p1,p2,...,pn
```

p — Processing parameters; separators must be commas or spaces, which can be used interchangeably

MERGE

MERGE initiates Sort/Merge merge capabilities. In interactive mode, control statements cannot be continued on more than one line. In batch mode, you can continue the control statement line by placing two or more periods at the end of the line and one period at the beginning of the new line. You can use a maximum of 240 characters for the line and continuation; the maximum number of characters processed per line is 80. The control statement begins with the word MERGE, which must be followed by a period. The format for MERGE is shown below:

```
MERGE.p1,p2,...,pn
```

p — Processing parameters; separators must be commas or spaces, which can be used interchangeably

DIRECTIVE FILES

You can use directive files for more complex sorts or merges and for standardizing installation parameters and collating sequences. A directive file contains processing parameters and parameters in the SORT or MERGE control statements.

Specify directive files with the DIR parameter in the SORT5 or MERGE control statement. The directive file statement formats are:

```
SORT,p1,p2,...,pn

MERGE△ p1△ p2△ ... △pn
```

p — Processing parameter; separators must be commas or spaces, which can be used interchangeably

You can continue the directive file statement by placing two or more periods at the end of the line and one or more spaces at the beginning of the new line. The directive files must be BT=C, RT=Z, and FL=100 or less. Characters in excess of 80 are ignored.

ALPHABETIZED LIST OF PARAMETERS

You can specify the parameters in the SORT5 or MERGE control statement or directive file to direct processing. However, the DIALOG parameter cannot appear in a directive file. When specified with keywords, parameters take one of the following forms, depending on the parameter:

```
keyword=((value-set)[,(value-set)]...)

keyword=value
```

The value or value-set is defined in the individual parameter descriptions. You can delete the keyword by supplying a value or value-set in the position assigned to that particular keyword.

You can mix keyword and positional specifications of parameters. See the Sort/Merge 5 reference manual for more information. Table 5 lists the parameter keywords and the position assigned to each.

EL

The EL parameter specifies the error level to be reported. The default is EL=W. The EL parameter format is shown below:

EL=el

el — The error level can be:
T for trivial (includes W, F, and C),
W for warning (includes F and C),
F for fatal (includes C), or
C for catastrophic.

ENR

The ENR parameter specifies the estimated number of records to be sorted or merged. The number can be a single value or a range of values. If you omit ENR, a value of 80000 divided by the value specified for the maximum record length (MRL) is used. The ENR parameter format is shown below:

ENR=expr

ENR=expr..expr

expr — Expression that is a decimal integer or CYBER Control Language (CCL) variable R1, R2, R3, R1G, EF, or EFG.

FASTIO

The FASTIO parameter specifies that certain sort or merge input and output operations are to be performed directly by Sort/Merge rather than by CYBER Record Manager (CRM). Specifying the FASTIO parameter results in increased speed, but not all errors are diagnosed. FASTIO=YES indicates that files are read or written by Sort/Merge. The records must be CRM format BT=C and RT=F, or BT=I and RT=W. Other block and record types are processed using CRM. FASTIO=NO indicates that the files are processed using CRM. Omitting the FASTIO parameter is equivalent to FASTIO=NO. The FASTIO parameter format is shown below:

FASTIO=YES (or FASTIO=Y)

FASTIO=NO (or FASTIO=N)

FROM

The FROM parameter specifies the input files from which the records to be sorted or merged are read. As many as 100 files can be sorted or merged in one job step. If you omit the FROM parameter, records are read from file OLD (unless an owncode routine supplies records). The FROM parameter format is shown below:

```
FROM=lfn

FROM=(lfn1,lfn2,...,lfnn)

FROM=$NULL
```

lfn	Logical file name; one to seven letters or digits, beginning with a letter.
$NULL	Records are not read from a file; they must be supplied from an owncode routine.

KEY

The KEY parameter specifies the key fields that determine the sorted or merged order of output records. The total number of key characters must be less than 256 (if you also specify sum fields, the total number of characters in the key and sum fields together must be less than 256). The three KEY parameter formats are shown below:

```
KEY=((value-set)[,(value-set)]...)

KEY=first..last

KEY=first
```

value-set	(first[,length[,type[,ad]]]) or (first..last[,type[,ad]])
first	First byte or bit of the key field
length	Number of bytes or bits in the key field; default is 1
last	Last byte or bit of the key field

type	Name of a numeric data format or collating sequence; default is ASCII6
ad	Order; either A for ascending or D for descending; default is A

Two sets of parentheses are still needed if you specify only one key field with the first format. If you omit the KEY parameter, KEY=1...mnr (minimum record length) is used.

L

The L parameter specifies the file to which listing information is written. The default value is L=OUTPUT, which lists information to the file OUTPUT. The L parameter format is shown below:

L=lfn

L=$NULL

lfn	Logical file name; one to seven letters or digits, beginning with a letter.
$NULL	Listing information is not written.

LO

The LO parameter selects the listing options. Omitting the LO parameter is equivalent to LO=S. The format for the LO parameter and the possible options are shown below:

LO=option

option	You can select one of the following options:
S	A copy of the directive file is written.
A	A copy of the resource map is written.
(S,A) or (A,S)	A copy of the directive file and the resource map is written.
OFF	Nothing is written.

OWNF

The OWNF parameter specifies the file that is the source of owncode routines. If the owncode routine is not on the file specified by OWNF, the global library set is searched. The OWNF parameter format is shown below:

OWNF=lfn

lfn — Logical file name; one to seven letters or digits, beginning with a letter.

OWNFL

The OWNFL parameter specifies the exact number of characters in all records entering the sort from an owncode routine. You cannot specify both the OWNFL and OWNMRL parameters for the same sort. The OWNFL parameter format is shown below:

OWNFL=integer (or OFL=integer)

integer — Integer from 1 to 5000 (larger records can be sorted if the field length is increased). The default is the record length specification of the input and output files.

OWNMRL

The OWNMRL parameter specifies the maximum length in characters of any record entering the sort from an owncode routine. You cannot specify both the OWNFL and OWNMRL parameters for the same sort. The OWNMRL parameter format is shown below:

OWNMRL=integer (or OMRL=integer)

integer — Integer from 1 to 5000 (larger records can be sorted if the field length is increased). The default is the length specification of the input and output files.

OWNn

The OWNn parameter specifies the name of an owncode procedure that is executed each time point n is reached during the sort or merge. The owncode routine named must be in the OWNF file or be in the global library set. The OWNn parameter format is shown below:

OWNn=proc

n	Point in Sort/Merge processing at which the owncode routine is executed; must be a 1, 2, 3, 4, or 5. See the section on Owncode Routines for more detailed information.
proc	Procedure name; 1 to 31 letters, digits, or the special characters ? # @ ___. Only seven characters are significant and the first must be a letter.

RETAIN

The RETAIN parameter (RETAIN=YES) directs Sort/Merge to output records with equal sort keys in the same order as the records are input. RETAIN=NO or not specifying the parameter might cause records with equal sort keys to be output in a different order than the records are input. Omitting the RETAIN parameter is equivalent to RET=N. The RETAIN parameter format is shown below:

RETAIN=YES (or RET=Y)

RETAIN=NO (or RET=N)

SEQx

The SEQx parameters define your own collating sequence. The SEQN signals the start of your sequence definition. The definition of one collating sequence continues with the SEQS, SEQR, or SEQA parameters; it is terminated by any parameter other than these three. You can define as many as 100 collating sequences with a separate series of SEQx parameters. The SEQx parameter formats for each collating sequence are shown below:

```
SEQN=name
SEQS=('char',...,'char')
.
.
.
[SEQR=YES or SEQR=Y]
[SEQA=YES or SEQA=Y]
```

SEQN — This parameter specifies that the name of the collating sequence will be 'name'. 'Name' cannot be the same name as a standard collating sequence or a previous user-defined name. 'Name' must be 1 to 31 letters, digits, or the special characters ? # @ _ ; only ten characters are significant and the first must be a letter.

SEQS — This parameter specifies the collating positions of the characters in your collating sequence. Each SEQS parameter specifies either a single step or a range of steps, indicated by 'char'. A single character may be specified by the character enclosed in apostrophes, or by $CHAR(n), where n is the number of the character in your character set. You can specify the SEQS parameter more than once in a directive file.

SEQR — This parameter defines a special value step that consists of all characters not explicitly or implicitly specified with SEQS; either YES or Y is allowed. This parameter can appear once in any place between the start and end of the sequence definition.

SEQA — This parameter alters all equated characters in output records so they become the first characters in the appropriate SEQS parameter. Either YES or Y is allowed. This parameter can appear once in any place between the start and end of the sequence definition.

STATUS

The STATUS parameter specifies that a CYBER Control Language (CCL) variable be set to a value representing the highest level of error that occurred during the sort or merge. The values that can be returned in the specified variable are shown in table 6. This parameter prevents an abort if a fatal or catastrophic error occurs before any data records are input. Some types of execution errors do not allow recovery and an abort can occur regardless of the STATUS parameter. The STATUS parameter format is shown below:

STATUS=variable (or ST=variable)

variable — CCL variable R1, R2, R3, R1G, EF, or EFG

TABLE 6. ERROR LEVEL CODES, STATUS PARAMETER

Error Level	Code
No errors	0
Trivial	20
Warning	30
Fatal	40
Catastrophic	50

SUM

The SUM parameter specifies the fields that you want to combine from records with equal key values. As many as 255 bytes can be summed. The records with all key fields equal are combined into one new record, and the other records with equal keys are deleted. Sum fields

cannot be key fields. See the Sort/Merge 5 reference manual for further details about the summing algorithm. The SUM parameter format is shown below:

SUM=((value-set)[,(value-set)]...)

(value-set)	(first,length,type[,rep]) or (first..last,type[,rep])
first	First byte or bit of the sum field
length	Number of bytes or bits in the sum field
last	Last byte (or bit for formats BINARY_BITS and INTEGER_BITS) in the sum field
type	Name of a numeric data format (except REAL)
rep	Number of adjacent fields to be summed; default is 1

TO

The TO parameter specifies the file to which sorted or merged records are written. If you omit the TO parameter, records are written to the file NEW (if there is not an owncode routine to process records). The TO parameter format is shown below:

TO=lfn	Logical file name; one to seven letters or digits, beginning with a letter.
TO=$NULL	Records are not written to a file; you must supply an owncode routine.

VERIFY

The VERIFY parameter directs Sort/Merge to check merge input records for correct order. If a record is out of order when you have specified VERIFY=YES, the merge is aborted, even if the STATUS parameter has been specified. A dayfile message is then issued identifying the file that is out of order. If you specify VERIFY=NO or omit the VERIFY parameter, merge input records are not checked for correct order. The

VERIFY parameter cannot be used for a sort. Omitting the VERIFY parameter is equivalent to VERIFY=NO. The VERIFY parameter format is shown below:

VERIFY=YES (or VER=Y)

VERIFY=NO (or VER=N)

CYBER RECORD MANAGER FILE CONTROL STATEMENT

The CYBER Record Manager (CRM) FILE control statement specifies the characteristics of all input and output data files named in a SORT5 or MERGE control statement, or in a directive file. You must specify the FILE control statement before the SORT5 or MERGE control statements for files with other than default characteristics.

The defaults are automatic for files INPUT, OUTPUT, and PUNCH. The record length must be supplied if the default does not apply to the file. The default file characteristics that apply to some different file types are shown in table 7.

TABLE 7. FILE CHARACTERISTICS

File	Fit Field		
	BT	RT	FL
FROM files	C	Z	FL≤150
TO file	C	Z	FL≤150
DIR files	C	Z	FL≤100
L file	C	Z	FL=132
E file	C	Z	FL=72†
INPUT	C	Z	FL=80
OUTPUT	C	Z	FL=140
PUNCH	C	Z	FL=80

†If E=OUTPUT, FL=140 is used.

INTERACTIVE USE

You can access Sort/Merge from your terminal by specifying parameters interactively or by using an interactive dialog. After you connect your terminal to the computer you can use either interactive method.

INTERACTIVE DIALOG

The DIALOG=YES invokes a dialog between you and Sort/Merge for sort or merge specification. You can specify owncode routines and a nonstandard collating sequence. Before beginning the dialog you must assign files to be used and enter CRM FILE control statements for any files not written in standard unit record format. Standard unit record format files include files created by punched cards in a batch job or lines entered at a terminal, and the special files INPUT, OUTPUT, and PUNCH.

During the dialog you are asked questions. Your response to a question determines the next question. For answers requiring YES or NO, you can use Y or N. When you have completed the dialog, the following message is displayed:

```
THANK YOU. SORT/MERGE NOW BEGINS.
```

If sorted or merged records are written to a file (not processed by an owncode routine), Sort/Merge creates a local file and gives this file the output name that you specified during the dialog. When the dialog is complete, you can display or print this local file at the terminal, you can route this file to a printer, or you can save the file.

INTERACTIVE PARAMETERS AND COMMANDS

You can use all Sort/Merge parameters at your terminal with the same SORT5 or MERGE control statement requirements as used in batch processing. You must first login, REQUEST and ATTACH all necessary files, and type in any needed FILE control statements.

You can use the commands HELP, RESTART, and QUIT to control the use of the interactive dialog. These commands can be used at any time during the interactive dialog. Table 8 lists the interactive commands and their uses.

TABLE 8. INTERACTIVE COMMANDS

Command	Usage
HELP	Displays explanation of Sort/ Merge usage.
RESTART	Restarts interactive dialog at beginning.
QUIT	Terminates interactive dialog. Control returns to the operating system.

PROCEDURE CALLS

You can invoke a sort or merge from within a program by a sequence of procedure calls. You can call the procedures from a FORTRAN program or from a program in any language that uses the standard FORTRAN calling sequence, such as COBOL.

To use procedure calls you must observe all coding conventions for the language, and assign any needed permanent files. The default file characteristics are used unless you use the FILE control statement.

You can call the procedures in any order, except you must call SM5SORT or SM5MERG first, and you must call SM5END last. Since there are no default values, all of the parameters in the following call statements must be included.

SM5E

The SM5E procedure specifies the file to which diagnostic messages are written. The file is written in CRM format BT=C, RT=Z, and FL=72. If you do not call SM5E, diagnostic messages are written to file OUTPUT. If you specify file $NULL, no diagnostic messages are written. The SM5E procedure format is shown below:

```
CALL SM5E('file')
```

'file'	File name; one to seven letters or digits, beginning with a letter.

SM5EL

The SM5EL procedure specifies the error level to be reported. If you do not call SM5EL, errors of levels T, W, F, and C are reported. Errors are written to the file specified by SM5E, or to OUTPUT if you do not call SM5E. The SM5EL procedure format is shown below:

```
CALL SM5EL (error-level)
```

error-level	Alphabetic character enclosed in apostrophes or an integer as shown in table 9.

TABLE 9. ERROR LEVEL SPECIFICATION

Error Level	Errors Reported
'T' or 1	T, W, F, C
'W' or 2	W, F, C
'F' or 3	F, C
'C' or 4	C

SM5END

The SM5END procedure terminates your sort or merge specification and initiates Sort/Merge processing. SM5END must be the last procedure you call. The SM5END procedure format is shown below:

```
CALL SM5END
```

SM5ENR

The procedure SM5ENR specifies the estimated number of records to be sorted or merged. If you do not call SM5ENR, a value of 80000 divided by the value specified for the maximum record length (MRL) is used. The SM5ENR procedure format is shown below:

```
CALL SM5ENR(integer)
```

integer — Number of records to be sorted or merged. The range is 0 to 1000000000.

SM5FAST

The SM5FAST procedure specifies that certain sort or merge input and output operations are to be performed directly by Sort/Merge rather than by CRM. If the option is YES, the records must be BT=C and RT=F, or BT=I and RT=W. If the option is NO, CRM is used. If

you omit SM5FAST, the default is SM5FAST='YES'. The SM5FAST procedure format is shown below:

```
CALL SM5FAST(option)
```

option	The option can be: 'YES' or 'Y', 'NO' or 'N'

SM5FROM

The SM5FROM procedure specifies the input files from which the records to be sorted or merged are read. You can specify up to 100 files with each SM5FROM or you can call SM5FROM up to 100 times. The files are read in the order you specify them. If you do not call SM5FROM, the default file is OLD (unless an owncode routine supplies records). The SM5FROM procedure format is shown below:

```
CALL SM5FROM('file1'[,'file2']...)
```

'file'	File name; one to seven letters or digits, beginning with a letter.

SM5KEY

The SM5KEY procedure specifies a single key field for the sort or merge. You can call SM5KEY as many as 100 times during a sort or merge. The total number of key characters you can use must be less than 256 characters (if you also specifiy sum fields, the total number of characters in the key and sum fields together must be less than 256). If you do not specify SM5KEY, the entire record is used as a character key field and the default key type and sort order are used. The SM5KEY procedure format is shown below:

```
CALL SM5KEY(first,length,'type','ad')
```

first	First byte or bit of the key field; the leftmost byte or bit is counted as number 1.
length	Number of bytes or bits in the key field.
'type'	Name of a numeric data format or collating sequence.
'ad'	Order; either 'A' for ascending or 'D' for descending.

SM5MERG

The SM5MERG procedure signals the beginning of a merge specification. SM5MERG must be the first procedure you call for a merge. The result array is a 16-element integer array; each element returns a result as shown in table 10. You set the first element before calling the procedure. The SM5MERG procedure format is shown below:

```
CALL SM5MERG(option)
```

option — Result array name; one to seven letters or digits, beginning with a letter, or the integer zero. The integer zero returns no statistics or results.

SM5OFL

The SM5OFL procedure specifies the exact number of characters in each fixed length record entering the sort from an owncode routine. If you do not specify SM5OFL, the records can be no longer than the longest allowed input or output record. You cannot call SM5OFL and SM5OMRL in the same sort. The SM5OFL procedure format is shown below:

```
CALL SM5OFL(integer)
```

integer — Exact number of characters in each fixed length record entering the sort from an owncode routine.

SM5OMRL

The SM5OMRL procedure specifies the maximum length of any record entering the sort from an owncode routine. If you do not call SM5OMRL, records entering the sort from an owncode routine can be no longer than the longest allowed input or output record. You cannot call SM5OMRL and SM5OFL in the same sort. The SM5OMRL procedure format is shown below:

```
CALL SM5OMRL(integer)
```

integer — Maximum length in characters of any record entering the sort from an owncode routine.

TABLE 10. RESULT ARRAY FORMAT

Array Element Number	Contents
1	Number of elements of results you want returned (0 through 15)
2	Number of records read from sort or merge input files
3	Number of records deleted by an owncode 1 routine
4	Number of records inserted by an owncode 1 routine
5	Number of records inserted by an owncode 2 routine
6	Number of records sorted or merged
7	Number of records deleted by an owncode 3 routine
8	Number of records inserted by an owncode 3 routine
9	Number of records inserted by an owncode 4 routine
10	Reserved
11	Number of records deleted by an owncode 5 routine
12	Number of records combined by summing
13	Number of records written to output file
14	Minimum record length in characters
15	Average record length in characters
16	Maximum record length in characters

SM5OWNn

The SM5OWNn procedure specifies an owncode routine that executes each time point n is reached during the sort or merge. If you do not specify SM5OWNn, no owncode routine executes. The SM5OWNn procedure format is shown below:

```
CALL SM5OWNn(proc)
```

n	Point at which the owncode routine is executed; must be a 1, 2, 3, 4, or 5. See the section on Owncode Routines for more detailed information.
proc	Owncode routine name; one to seven letters or digits, beginning with a letter. In a FORTRAN program, the name must be declared in an EXTERNAL statement.

SMRETA

The SMRETA procedure specifies that records with equal sort keys are output in the same order as they are input. If you do not call SM5RETA, records with equal keys might not be retained in the original order. The YES option assures that records with equal keys retain their original order. The SMRETA procedure format is shown below:

```
CALL SMRETA(option)
```

option	The option can be: 'YES' or 'Y', 'NO' or 'N'.

SM5SEQx

The SM5SEQx procedure specifies a user-defined collating sequence that is referenced by SM5KEY. SM5SEQx specifies something different about the collating sequence depending upon x. The x can be N, S, R, or A.

The SM5SEQN procedure names the collating sequence. It must be the first called in the series of SM5SEQx procedures. The procedure format is shown below:

```
CALL SM5SEQN('name')
```

'name' Collating sequence name enclosed in apostrophes; 1 to 31 letters, digits, or the special characters ? # @ _ (only the first 10 characters are significant and the first character must be a letter). It also must be a different name from any predefined collating sequence.

The SM5SEQS procedure assigns collating positions to the characters in your collating sequence. Each SM5SEQS procedure specifies a value step in the sequence. A value step consists of one or more characters. Characters specified in the same step collate equally. This procedure can be called more than once. The procedure format is shown below:

```
CALL SM5SEQS ('char',...,'char')
```

'char' Character in the collating sequence.

The SM5SEQR procedure assigns equal collating position to characters not specified by SM5SEQS. The procedure format is shown below:

```
CALL SM5SEQR('YES') or CALL SM5SEQR('Y')
```

The SM5SEQA procedure alters characters in the same value step to output the same as the first character in the step. The procedure format is shown below:

```
CALL SM5SEQA('YES') or CALL SM5SEQA('Y')
```

SM5SORT

The SM5SORT procedure signals the beginning of a sort specification. SM5SORT must be the first procedure you call for a sort. A 16-element integer array returns results of the sort back to your program; each element returns a result as shown in table 10. The first element in the array is set by you, before calling the procedure. The SM5SORT procedure format is shown below:

```
CALL SM5SORT(option)
```

option Result array name; one to seven letters or digits, beginning with a letter, or the integer zero. The integer zero returns no statistics or results.

SM5ST

The SM5ST procedure specifies an integer variable that is set to a value representing the highest level of error that occurred during the sort or merge. Table 11 shows the error levels and the representative values. If you call the SM5ST procedure, Sort/Merge does not abort if a catastrophic error occurs before any data records are input. Some types of errors do not allow recovery and an abort can occur regardless of the SM5ST procedure. The SM5ST procedure format is shown below:

```
CALL SM5ST(variable)
```

variable — Symbolic name of an integer; one to seven letters or digits, beginning with a letter.

TABLE 11. ERROR LEVEL CODES

Error Level	Code
No Errors	0
Trivial	10
Warning	20
Fatal	30
Catastrophic	40

SM5SUM

The SM5SUM procedure specifies fields that are to be summed in records with equal key values. As many as 255 bytes can be summed. The Sort/Merge summing procedure combines records with all key fields equal into one new record and deletes other records with equal keys. The SM5SUM procedure format is shown below:

```
CALL SM5SUM (first,length,'type',rep)
```

first — First byte or bit of the sum field

length — Number of bytes or bits in the sum field

'type' Name of a numeric data format

rep Number of fields to be summed

You can call SM5SUM as many as 100 times during one sort or merge. You cannot call SM5SUM and SM5OWN5 in the same sort or merge.

SM5TO

The SM5TO procedure specifies the file to which sorted or merged records are written. If your program also reads or writes the file, you must include an OPEN or REWIND statement in your program after calling the procedures. If you do not call SM5TO, records are written to file NEW (unless an owncode routine processes all records). The SM5TO procedure format is shown below:

```
CALL SM5TO('file')
```

'file' File name; one to seven letters or digits, beginning with a letter.

COBOL INTERFACE

You can use Sort/Merge 5 from COBOL 5 through a control statement option or through direct calls. If you specify SORT5 in your COBOL 5 control statement, Sort/Merge 5 is used when the SORT or MERGE statement in your COBOL program executes. See the COBOL 5 reference manual for information on coding the SORT or MERGE statement in a program.

Direct calls from your COBOL program to the Sort/Merge procedures, can be made with the ENTER FTN5 SM5xxx statements. See the COBOL 5 reference manual for information on coding ENTER statements.

OWNCODE ROUTINES

You can write a routine to insert, substitute, modify, or delete input and output records each time the sort or merge processing reaches a certain point. The owncode routines are written in FORTRAN, COBOL, or some other language that has been compiled and saved in relocatable binary form.

Specify the OWNF parameter when you are using control statement calls to Sort/Merge. The OWNF parameter supplies Sort/Merge with the name of the file containing your owncode routines. The OWNn parameter specifies the name of your owncode routines.

Specify SM5OWNn when you are using procedure calls to Sort/Merge. SM5OWNn specifies the name of your owncode routine and indicates at which point in processing the routine is executed.

Owncode routines are called with parameters, as shown in the SUBROUTINE format in figure 1. A summary of the owncode routines and the parameters passed by each routine is shown in table 12.

```
SUBROUTINE proc(return_code,reca,rla[,recb,rlb])

proc          Procedure name

return_code   Parameter passed by Sort/Merge that
              can be changed by the owncode routine.
              The value returned to Sort/Merge indi-
              cates a specific action to be taken by
              Sort/Merge.

reca          Parameter that is an otherwise empty
              array that is used to pass the current
              record.

rla           Parameter that passes the number of
              characters in the current record being
              sorted or merged.

recb          Parameter used when processing two
              records with equal keys; passes the
              second record.

rlb           Parameter used when processing two
              records with equal keys; passes the
              number of characters in the second
              record.
```

Figure 1. Subroutine Format

TABLE 12. OWNCODE ROUTINE SUMMARY

Routine Type	Processing	Parameters Passed					Return_code Value	
		Return_code	reca	rla	recb	rlb		
Owncode 1	Input records	x	x	x			0	Sort current record.
							1	Delete current record.
							2	Insert new record.
							3	Terminate input from current file.
Owncode 2	Input files	x	x	x			0	Begin processing next input file.
							1	Insert new record.

TABLE 12. OWNCODE ROUTINE SUMMARY (Contd)

Routine Type	Processing	Parameters Passed					Return_code Value	
		Return_code	reca	rla	recb	rlb		
Owncode 3	Output records	x	x	x			0	Modify current record.
							1	Delete current record.
							2	Create new record.
							3	Terminate output.
Owncode 4	Output files	x	x	x			0	Sort or merge is complete.
							1	Insert new record.
Owncode 5	Equal keys	x	x	x	x	x	0	Retain both records.
							1	Replace both records with new record.

CORPORATE HEADQUARTERS, 8100 34th AVE. SO.
MINNEAPOLIS, MINN. 55440

SALES OFFICES AND SERVICE CENTERS
IN MAJOR CITIES THROUGHOUT THE WORLD