60497600

CONTROL DATA CORPORATION

SORT/MERGE VERSIONS 4 AND 1 INSTANT

CDC® OPERATING SYSTEMS:
NOS 1
NOS/BE 1
SCOPE 2

60497600

CONTROL DATA CORPORATION

SORT/MERGE VERSIONS 4 AND 1 INSTANT

CDC® OPERATING SYSTEMS:
NOS 1
NOS/BE 1
SCOPE 2

REVISION RECORD	
REVISION	**DESCRIPTION**
A	Original printing.
(02-15-76)	
B	This revision documents the use of key com-
(04-29-77)	parison and key extraction techniques for
	sorting and the use of the Common Memory
	Manager for space allocation, at PSR
	level 446.
C	This revision documents the FORTRAN 5
(01-16-81)	interface at PSR level 528.
Publication No. 60497600	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	Rev
Front Cover	-
Title Page	-
ii	C
iii/iv	C
v	C
vi	C
vii	C
viii	C
1	B
2	A
3 thru 5	B
6	C
7	B
8	C
8.1/8.2	C
9	B
10	B
11	A
12	B
13	C
14	B
15	A
16	A
17	B
18	A
19	B
20	B
21 thru 32	C
Back Cover	-

PREFACE

This instant outlines the Sort/Merge system which is available under the following operating systems:

Sort/Merge Version 4.6 operates under NOS 1 for the CONTROL DATA® CYBER 170 Series, CYBER 70 Models 71, 72, 73, 74, and 6000 Series Computer Systems.

Sort/Merge Version 4.6 operates under NOS/BE 1 for the CDC® CYBER 170 Series, CYBER 70 Models 71, 72, 73, 74, and 6000 Series Computer Systems.

Sort/Merge Version 1.0 operates under SCOPE 2 for the CONTROL DATA CYBER 170 Model 176, CYBER 70 Model 76 and 7600 Computer Systems.

The Sort/Merge record processing facility is available through the use of control statements and directives; user programs can call Sort/Merge with COMPASS assembly language macros, FORTRAN interface routine calls, or through the COBOL language. In this instant, material that applies exclusively to either Sort/Merge Version 1 or Sort/Merge Version 4 is noted accordingly in the text.

This instant is written for a programmer familiar with the Sort/Merge facility and the operating system under which it is running as well as the calling language.

More detailed information can be found in the publications listed below.

Publication	Publication Number
Sort/Merge Versions 4 and 1 Reference Manual	60497500
CYBER Record Manager Basic Access Methods Version 1.5 Reference Manual	60495700
SCOPE 2 Reference Manual	60342600
SCOPE 2 Record Manager Reference Manual	60454690
NOS/BE Version 1 Reference Manual	60493800
NOS Version 1 Reference Manual Volume 1 of 2	60435400

COBOL Version 5 Reference Manual	60497100
FORTRAN Extended Version 4 Reference Manual	60497800
FORTRAN Version 5 Reference Manual	60481300
COMPASS Version 3 Reference Manual	60492600

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

CONTENTS

INTRODUCTION

Sort/Merge is a high-speed record processing program that manipulates and rearranges records into the order specified by the user. Record processing options include:

Sort-only	Sorts records from one or more input files.
Merge-only	Combines a maximum of 100 presorted input files into one output file.
Sort and Merge	Sorts input files and merges the output with presorted files. The combined number of input files and output files cannot exceed 100.

Operations can be specified by a combination of control statements and Sort/Merge directives or by a series of macro calls.

Directives	Macro Calls
SORT	SORT
MERGE	MERGE
BYTESIZE	BYTESIZE
FIELD	
FILE	FILES
KEY	KEY
SEQUENCE	SEQUENCE
EQUATE	EQUATE
OPTIONS	OPTIONS
OWNCODE	OWNCODE
TAPE	TAPE
END	
	POINTER
	SMLIST

Sort/Merge processing initiated by FORTRAN Extended calls is discussed in the section on FORTRAN Extended calls.

SORT KEYS

A sort key is a field of information within each record. The key is used to determine the order in which records are to be written to the output file. At least one sort key must be specified for each run; as many as 100 keys can be specified. Multiple keys can differ as to type, collating sequence, and sort order.

KEY LENGTH

Key field length is specified on a FIELD directive or a KEY macro as the number of bytes and bits in the field. The length of a sort key must not exceed the minimum record length for any file.

KEY POSITION

Starting position of a sort key field, which is specified on a FIELD directive or KEY macro, can be anywhere within the record. The key must maintain the same position relative to the first character of each record in each file. Multiple fields can overlap.

KEY TYPE

Key type is specified on a FIELD directive or a KEY macro. Data in sort key fields can be:

Logical (LOGICAL)	Unsigned binary integers of any length, sorted by magnitude.
Integer (INTEGER)	60-bit integers, starting anywhere within the record, and sorted by numeric value.
Floating Point (FLOAT)	60-bit normalized or unnormalized floating point numbers, starting at any bit position within the record, and sorted by numeric value.
Internal BCD (INTBCD)	Internal BCD character code, any integral number of characters in length, and sorted according to the specified collating sequence.
Display (DISPLAY)	Display code, any integral number of characters in length, and sorted according to the specified collating sequence.

COLLATING SEQUENCE

Collating sequence is specified on the KEY directive or KEY macro; it applies only to character data (key type INTBCD or DISPLAY), and can be any of the following:

ASCII6	6-bit ASCII
COBOL6	6-bit COBOL
DISPLAY	6-bit CDC display code
INTBCD	Internal BCD

The user can specify his own collating sequence with the colseq parameter on the SEQUENCE and EQUATE directives or macros.

SORT ORDER

Sort order for numeric or character keys can be specified on the KEY directive or macro as ascending or descending.

SIGNED NUMERIC DATA

Numeric data with a sign represented by an overpunch character is indicated by the optional keyword SIGN on a FIELD directive or KEY macro or by ≠SIGN≠ on the SMKEY call. The overpunched digit is normally the last digit of the field, but under Sort/Merge 4 it can be specified as the first digit by indicating the keyword LEADING. When the sign character is a digit without an overpunch, the value of the sort key is positive. Consequently, when the SIGN parameter is used, sort key values of 201, 20A, and 20J are equivalent to sort key values of +201, +201, and -201 respectively.

Key Punch Code	Sign Character	Sign of Key Value	Value of Digit
12-9	I	+	9
12-8	H	+	8
12-7	G	+	7
12-6	F	+	6
12-5	E	+	5
12-4	D	+	4
12-3	C	+	3
12-2	B	+	2
12-1	A	+	1
12-0	<	+	0
11-0	v	-	0
11-1	J	-	1
11-2	K	-	2
11-3	L	-	3
11-4	M	-	4
11-5	N	-	5
11-6	O	-	6
11-7	P	-	7
11-8	Q	-	8
11-9	R	-	9

Under Sort/Merge 4, a separate + or - sign can be specified for numeric data by including the keywords SIGN and SEPARATE on a FIELD directive or KEY macro or by specifying ≠SEPARATE≠ on the SMKEY call. The separate sign of the value must always be the first or last character in the field, as indicated by the keyword LEADING or TRAILING.

SORT/MERGE DIRECTIVES

Sort/Merge directives are structured according to the following rules:

- Directives can begin in any column, but cannot extend beyond column 72.
- Any number of continuation statements can be used. Each continuation must have a comma in column 1.
- Any number of comment statements can be included. Comments require an asterisk in column 1.
- User-defined parameters can consist of any number of alpha and numeric characters. The first character of the parameter must be alpha, and the first seven characters for each specification within a single Sort/Merge run input deck must be unique.
- Although alternative separators are accepted for some directives by Sort/Merge, the comma has been used throughout this manual.
- Embedded blanks are illegal in user-defined parameters.

SORT

SORT,VAR=type,LCMSB=ba

Specifies sort-only or sort and merge processing.

type Processing indicators:

DISK	Mass storage Sort/Merge (default)
TAPE POLYPHASE POLY	Polyphase tape (Sort/Merge 4 only)
BALANCED BAL	Balanced tape (Sort/Merge 4 only)

ba Optional parameter designating total large core memory (LCM) buffer area for all intermediate scratch files (Sort/Merge 1 only).

MERGE

MERGE

Specifies merge-only processing.

BYTESIZE

BYTESIZE, n

Specifies the number of bits per byte that are to be used by the FIELD directive. If this directive is omitted, six bits per byte is assumed.

n Decimal number of bits per byte.

FIELD

FIELD,keyname (start, length, type, SIGN, location, SEPARATE)$_1$, . . . , keyname (. . .)$_n$

Specifies the starting position, length, and data type of a key field. At least one FIELD directive must be included in a Sort/Merge run; no more than 100 fields can be specified. Any sort key can be defined or referenced more than once during a single Sort/Merge run providing a new keyname is specified each time the sort key is defined.

keyname$_i$ User-assigned sort key name.

start$_i$ Starting position of the sort key:

byte
.bit
byte.bit } Byte number and number of the first bit within that byte in which the sort key first appears. Bytes and bits are numbered from 1; if either is omitted, 1 is assumed.

length$_i$ Length of the sort key:

nbytes
.nbits
nbytes.nbits } Number of bits and bytes in the sort key. A default value of six bits per byte is assumed unless specified differently by the BYTESIZE directive. A value of 0 is assumed if either of these parameters is omitted.

type$_i$ Sort key type identifier LOGICAL, INTEGER, FLOAT, INTBCD, or DISPLAY.

SIGN Optional parameter for sort keys containing numeric data in display code.

location Optional position indicator, valid only if SIGN is used; LEADING or TRAILING (Sort/Merge 4 only).

SEPARATE Optional parameter signifying sign is separate; valid only if SIGN is used (Sort/Merge 4 only).

KEY

KEY, keyname(order,colseq)$_1$, . . . ,keyname(. . .)$_n$

Specifies the order and collating sequence of keys.

keyname$_i$ — Sort key name; it must also be specified in the FIELD directive.

order$_i$ — Order in which keys are to be sorted and merged. Ascending order is assumed if the parameter is omitted.

A = ascending order
B = descending order

colseq$_i$ — User-specified collating sequence name specified in the SEQUENCE directive, or one of the standard collating sequence identifiers (ASCII6, COBOL6, DISPLAY, or INTBCD).

FILE

FILE,type=name(action),name(action) , . . . ,type= . . .

Specifies the files that are to be used. Only MERGE and OUTPUT file types can be specified for a merge run. SORT, MERGE, INPUT, and OUTPUT file types can be specified for a sort run. All files of a particular type must be specified in one group.

type — File type identifier:

INPUT or SORT	Sort input file
MERGE	Merge input file
OUTPUT	Sort or merge output file

name — Logical file name of a file to be processed by Sort/Merge.

action — System action to be performed after file processing is complete:

C	Close file
R	Rewind file
U	Unload file
N	No action
CR	Close and rewind file (default)
RC	Close and rewind file (default)
CU	Close and unload file

SEQUENCE

SEQUENCE, colseq (c_1, c_2, . . . ,c_n)

Specifies the user's own collating sequence; or redefines the default to be a user collating sequence or one of the standard collating sequences.

colseq — Name of the user's collating sequence or one of the standard sequence identifiers (ASCII6, COBOL6, DISPLAY, or INTBCD). If the parameter is omitted, the collating sequence in parentheses (c_1, c_2, . . .) is used by Sort/Merge.

c_i — Character or octal value specified in the sequence in which it is to appear in the user collating sequence. Characters or octal values not specified are considered equal and collated after the specified characters or values. This parameter should be omitted when colseq specifies a standard collating sequence.

The octal value must not be followed by a B.

If the following symbols are included in the user's collating sequence, they must be specified in the c parameter according to their octal equivalent:

Character	Display Code Octal Equivalent	Internal BCD Octal Equivalent
(	51	74
)	52	34
,	56	73
→	65	75

EQUATE

EQUATE,colseq(c_1,c_2, . . . ,c_n)(c_1,c_2, . . . ,c_n)

Specifies two or more characters in the collating sequence as equal for comparison purposes.

colseq — ASCII6, COBOL6, DISPLAY, INTBCD, or the name of the user-defined collating sequence.

c — Character or octal values to be equated. The collating positions of leading characters within the parentheses are equated to the collating position of the last character within the parentheses. (See the discussion of the SEQUENCE directive for additional information on the specification of characters or octal values.)

OPTIONS

OPTIONS, option, option, ...

Specifies special record handling options.

option	VERIFY	Output file is checked for correct sequencing. If the order of records on the output file is incorrect, the job terminates and the output file is lost. When this option is omitted during a merge run, out-of-sequence records produce a nonfatal diagnostic and are included in the output file.
	RETAIN	Records with identical sort keys are sequenced according to the order in which they are read. When this option is omitted, records with equal sort keys are sequenced arbitrarily.
	VOLDUMP	Checkpoint dump is taken when an end-of-volume condition on the input file or new-volume condition on the output file exists. A checkpoint file must have been previously requested (Sort/Merge 4 only).
	DUMP(n)	Checkpoint dump is taken when the specified number of records is read from the input file or the same number of records is written on the output file. If n is omitted, default is 50,000. A checkpoint file must have previously been requested (Sort/Merge 4 only).
	NODUMP	No checkpoint dumps are taken (Sort/Merge 4 only).
	ORDER(n)	Specifies the number of strings to be merged in core at one time. If core is not sufficient to merge at the order specified, a fatal error occurs. Default merge order is an installation option (Sort/Merge 4 only).
	COMPARE	Key comparison technique is used internally for comparing sort keys (Sort/Merge 4 only).
	EXTRACT	Key extraction technique is used internally for comparing sort keys (Sort/Merge 4 only).

The COMPARE and EXTRACT options are mutually exclusive within a single sort. If neither option is specified, Sort/Merge attempts to choose the best technique.

If more than one OPTIONS directive is specified, only the last OPTIONS directive specified applies.

OWNCODE

OWNCODE, MRL=mrl, exitno$_1$=entry$_1$, . . . ,exitno$_n$=entry$_n$

Specifies entry point names for the user's relocatable owncode exit routines.

mrl	Maximum decimal record length in 6-bit characters. Required if an input file has not been specified.
exitno$_i$	Number of the owncode exit. Owncode exit 6 is allowed only under Sort/Merge 4.
entry$_i$	Corresponding entry point name of the exit.

TAPE

TAPE, filename, filename, . . .

Specifies that intermediate merge files are to be assigned to magnetic tapes. If files have not been defined in a previous job step, Sort/Merge issues requests for scratch tapes to be assigned to intermediate files as needed. (Sort/Merge 4 only.)

filename	Logical file name assigned to intermediate merge file.

END

END

Signifies the end of a Sort/Merge run. This directive is required and must be the last directive in each run.

CYBER RECORD MANAGER FILE CONTROL STATEMENT

A FILE control statement must be provided for every input and output file except INPUT, OUTPUT, and PUNCH under Sort/Merge 4. Under Sort/Merge 1, CF=N and OF=N must be specified on FILE control statements for the special system files INPUT, OUTPUT, and PUNCH.

SORTMRG CONTROL STATEMENT

Processing of Sort/Merge directives is initiated by the SORTMRG control statement. Parameters can be specified in any order or can be omitted. Sort/Merge supplies a default value for all omitted parameters.

PARAMETERS

SORTMRG (parameter list)

Directive format — Specifies the format of the Sort/Merge directives.

- 6C — Directives are in Sort/Merge 3 format.
- 7C or omitted — Directives are in Sort/Merge 4 format.

Source input — Specifies the file on which the Sort/Merge directives are located.

- l=lfn or l=lfn/NR — File (lfn) is not rewound before opening.
- l=lfn/R — File (lfn) is rewound before opening.
- l — Directives are on file COMPILE.
- omitted — Directives are on file INPUT.

List file — Identifies the file to which listings output by Sort/Merge are written. These listings include directives, dayfile messages, and any diagnostics.

- O=lfn or O=lfn/NR — File (lfn) is not rewound before opening.
- O=lfn/R — File (lfn) is rewound before opening.
- O or omitted — Listings are written to file OUTPUT.

Merge order — Specifies the intermediate merge order. Overrides specification on the OPTIONS directive. (Sort/Merge 4 only.)

- MO=n — Merge order is $2 \leq n \leq 64$.
- omitted — Merge order is based on available core.

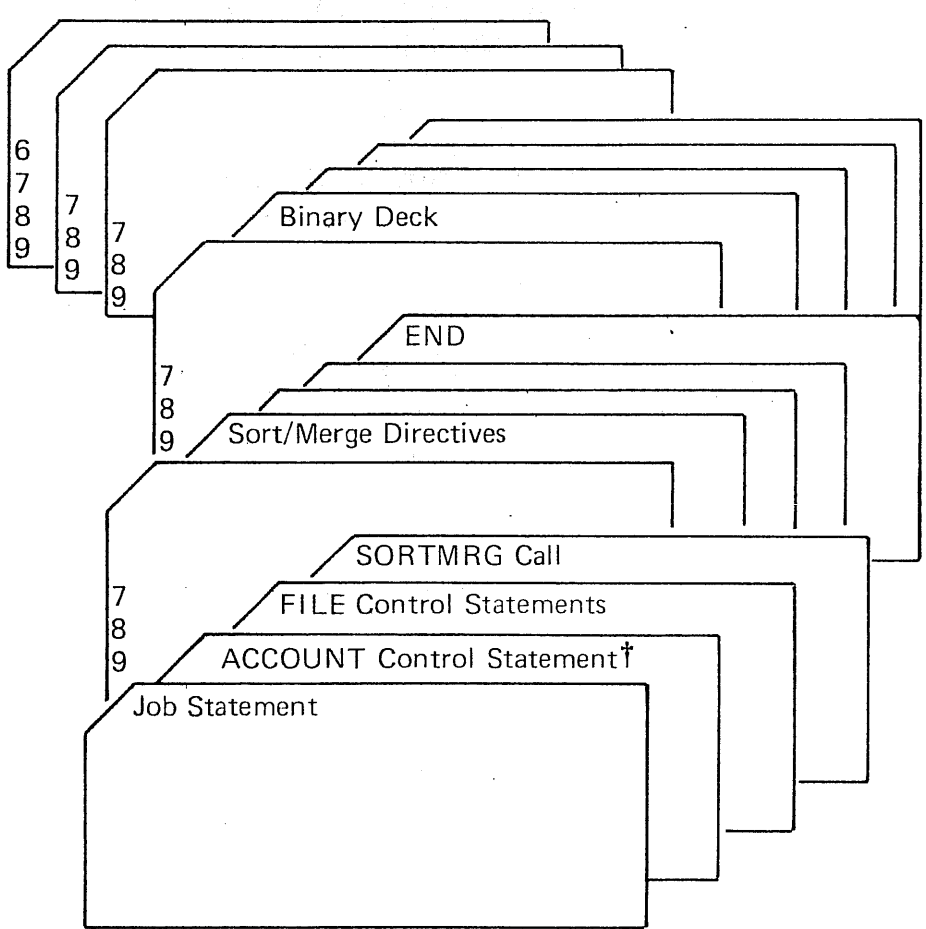
Owncode file	Specifies the file on which the owncode binaries are located. Sort/Merge reads this file after the directive file is read. Therefore, owncode binaries should be placed after directives in the job deck.
OWN=lfn OWN=lfn/NR	File (lfn) is not rewound before opening.
OWN=lfn/R	File (lfn) is rewound before opening.
OWN	Owncode binaries are on file LGO.
omitted	Owncode binaries are on file INPUT.

SORTMRG

This call is equivalent to:

```
SORTMRG(7C,I=INPUT/NR,O=OUTPUT/NR,
OWN=INPUT/NR)
```

SAMPLE INPUT STRUCTURES



†If required by the operating system

MACRO CALLS

When specifying macro instructions, the user must ensure that:

- A SORT or MERGE macro precedes all other macros.
- Macros are specified in a continuous sequence.
- Under Sort/Merge 1, the system control statement LIBRARY(SRTLIB) precedes the relocatable load of the macro binaries.
- Under Sort/Merge 4, the control statement LIBRARY(COBOL) or LDSET(LIB=COBOL) precedes the relocatable load of the macro binaries.
- The macros are not assembled with the COMPASS LIST M or F options because use of these options causes a large amount of source statements to be generated.
- Under Sort/Merge 4, S=SMTEXT is included on the call for COMPASS assembly to ensure proper expansion of macros.
- Under Sort/Merge 1, the control statement ATTACH (SRTMACS, SRTMACS, ID=PRDLIB) precedes the COMPASS control statement and the COMPASS program includes the statement

  ```
  SRTMACS        XTEXT
  ```

  with SRTMACS in the location field and XTEXT in the operation field.

Sort/Merge macros are structured according to rules for formatting COMPASS macros. Since the parameter functions of macros and directives are usually identical, a detailed explanation of previously described parameters is not included in this section. (See Sort/Merge Directives and Control Statement Requirements for detailed parameter functions.)

SORT ba,MAXCM=n,CM=BELOWHHA

Initiates Sort/Merge functions as a subprogram within a job requesting Sort/Merge processing.

SORTB MAXCM=n,CM=BELOWHHA

Specifies balanced processing under Sort/Merge 4.

SORTP MAXCM=n,CM=BELOWHHA

Specifies polyphase processing under Sort/Merge 4.

MERGE MAXCM=n,CM=BELOWHHA

Specifies merge-only processing.

The following parameters can be specified on the SORT macro call and its two alternate formats as well as on the MERGE macro call.

MAXCM=n Required parameter specifying the maximum number of central memory words to be used by Sort/Merge for working storage. At least 22000B words should be provided. If zero is specified, Sort/Merge uses a default size. n can specify a register. (Sort/Merge 4 only.)

CM=BELOWHHA Optional parameter specifying that the Common Memory Manager preferentially allocates the working storage area for Sort/Merge between the last word address (LWA) of the last loaded overlay and the highest LWA of all overlays. If the parameter is omitted, the working storage area begins higher than the highest LWA of all overlays. (Meaningful only for overlay programs under Sort/Merge 4.)

BYTESIZE n

Specifies the number of bits per byte. If the macro is omitted, six bits per byte is assumed.

FILES (type,name,name),(type,name,. . .)

Specifies input and output files.

KEY bytepos,bitpos,nbytes,nbits,type,colseq,order, SIGN,location,SEPARATE

Specifies sort key characteristics. A maximum of 100 sort keys can be specified in a run.

bytepos Position of the first byte of the sort key in relation to the first byte of the record, counting from 1. (The default is 1.)

bitpos Position of the first bit of the sort key in the byte indicated by bytepos, counting from 1. (The default is 1.)

nbytes Number of complete bytes in the sort key. (The default is 0.)

nbits Number of bits in the sort key, in addition to number of bytes specified in the previous parameter. (The default is 0.)

SEQUENCE colseq,(c,c, . . . ,),END

Specifies the user's own collating sequence, or redefines as default a user collating sequence or a standard collating sequence.

EQUATE colseq,(c,c, . . . ,),(c,c, . . .),END

Specifies two or more characters as equal in a collating sequence for comparison purposes.

OPTIONS option,option,. . .

Specifies special record handling options. The options are: VERIFY, RETAIN, VOLDUMP, DUMP, NODUMP, ORDER, COMPARE, and EXTRACT.

OWNCODE (MRL,mrl),(exitno,entry),(exitno, . . .)

Specifies entry point names to relocatable owncode exit routines. Owncode exit 6 is allowed under Sort/Merge 4 only.

TAPE filename,filename, . . .

Specifies that intermediate merge files are to be assigned to magnetic tapes (Sort/Merge 4 only).

POINTER addr

Required with SMLIST macro to give SMLIST macro address (Sort/Merge 4 only).

addr Address of an SMLIST macro.

SMLIST macaddr,macaddr, . . .

Specifies the addresses of Sort/Merge macros that need not exist in the sequence following the SORT macro; processed only when a POINTER macro is encountered within a sequence following a SORT macro (Sort/Merge 4 only).

macaddr Address of Sort/Merge macro: BYTESIZE, FILES, KEY, SEQUENCE, EQUATE, OPTIONS, OWNCODE, or TAPE.

OWNCODE ROUTINES

When entry to an owncode routine is initiated by the OWNCODE directive or the OWNCODE macro, the user should ensure that:

- When entry is initiated by the directive, the owncode routine has been assembled in relocatable binary form and, when necessary, is identified on the SORTMRG control statement.

- When entry is initiated by the macro, the owncode routine is assembled in the program calling Sort/Merge, or is assembled and referenced in a program occupying core at the same time as the program calling Sort/Merge.

Transfer to a user's owncode routine is accomplished with a return jump (RJ) which fills the entry point of the owncode routine with a return to Sort/Merge. To reestablish Sort/Merge control, the user normally jumps to the entry point of the owncode routine. The user can request specific processing action by jumping to an altered return address.

Registers A2 and X0 are used by Sort/Merge as a pointer to the current data record. Upon entry to all owncode exits, register A2 contains the address of the record and X0 contains the record length. The user must ensure that the contents of these registers are accurate when control is returned to Sort/Merge during record insertion or record modification.

FORTRAN Extended users should consult the section on FORTRAN Extended calls for information on owncode routines.

EXIT 1: PROCESSING OF INPUT RECORDS

Exit 1 is taken after each record is read from the input file or during a search for a new record when no input file is specified. This exit is not allowed in a merge-only run.

Normal return address	Sort/Merge accepts the current data record.
Normal return address+1	Sort/Merge deletes the current input record.
Normal return address+2	Sort/Merge inserts the user-specified current data record. Control returns to exit 1 and the current input record remains unchanged until a normal return is executed or the record is deleted.

Normal return address+3	Sort/Merge terminates record input from the current input file and proceeds to the next input file. If the current file is the last, Sort/Merge proceeds to exit 2 (if specified), or to the merge phase.

EXIT 2: PROCESSING OF INPUT FILES

Exit 2 is taken after the last record is read from each file. This exit is not allowed in a merge-only run.

Normal return address	Sort/Merge proceeds to the next input file or to the merge phase.
Normal return address+1	Sort/Merge inserts the user-specified current data record after the last record read. Control returns to exit 2 until a normal return is executed.

EXIT 3: PROCESSING OF OUTPUT RECORDS

Exit 3 is taken before each record is moved into the final output area or when no output file has been specified.

Normal return address	Sort/Merge writes the current output record.
Normal return address+1	Sort/Merge deletes the output record.
Normal return address+2	Sort/Merge inserts the user-specified current data record. Control returns to exit 3, and the current output record remains unchanged until a normal return is executed or the record is deleted.
Normal return address+3	Sort/Merge terminates record output to the current output file and proceeds with exit 4 (if specified), or normal end-of-file processing.

EXIT 4: PROCESSING OF OUTPUT FILES

Exit 4 is taken after the last record is moved into the final output area.

Normal return address	Sort/Merge continues normal processing by advancing to the end-of-file procedures for the output file.
Normal return address+1	Sort/Merge inserts a user-specified current data record after the last record in the output area. Control returns to exit 4 until a normal return is executed.

EXIT 5: PROCESSING OF DUPLICATE KEYS

Exit 5 is taken when two records with equal sort keys are encountered.

Normal return address	Sort/Merge accepts two records: the current data record, and the record whose address and record length are contained in registers A3 and X4.
Normal return address+1	Sort/Merge deletes one of two records with equal sort keys. The user must ensure that the record to be retained has been designated as the current data record.

EXIT 6: PROCESSING OF NONSTANDARD LABELS

(Sort/Merge 4 only)

Exit 6 is taken for all nonstandard input and output labels on files whose FILE control statements specify LT=NS, ULP≠NO.

Normal return address	Sort/Merge accepts a nonstandard label.
Normal return address+1	Sort/Merge accepts a nonstandard label. Control returns to exit 6 and Sort/Merge accepts additional nonstandard labels until a normal return is executed.

Processing Actions	Exit 1	Exit 2	Exit 3	Exit 4	Exit 5	Exit 6
Substitute Record	NR		NR		NR†	
Insert Record	NR+2	NR+1	NR+2	NR+1		
Delete Record	NR+1		NR+1		NR+1††	
Verify A Label Record During A Read						NR+1
Verify Last Label Record During A Read						NR
Supply A Label Record During A Write						NR+1
Supply Last Label Record During A Write						NR
Terminate File†††	NR+3		NR+3			
Normal Record Processing	NR	NR	NR	NR	NR	

†Both records, designated by the A2 and X0 and the A3 and X4 registers respectively, can be substituted with new address and length specifications.

††The record designated by the A3 and X4 registers is the record deleted.

†††The current record designated by the A2 and X0 register is not included in sort and/or merge processing.

FORTRAN EXTENDED CALLS

Sort/Merge processing can be initiated by a FORTRAN Extended program through special FORTRAN Extended call statements.

CALL SMSORT(mrl,ba)

CALL SMSORTB(mrl,ba)

CALL SMSORTP(mrl,ba)

CALL SMMERGE(mrl,ba)

One of these statements must be the first call of any sort.

SMSORT	Specifies sort-only or sort and merge processing.
SMSORTB	Specifies a balanced tape sort under Sort/Merge 4.
SMSORTP	Specifies a polyphase tape sort under Sort/Merge 4.
SMMERGE	Specifies merge-only processing.
mrl	Maximum record length in characters of the record to be sorted.
ba	Under Sort/Merge 1, optional parameter designating the number of words of large core memory (LCM) buffer area for all intermediate scratch files. Under Sort/Merge 4, optional parameter designating the number of words of central memory to be used by Sort/Merge for working storage. If the parameter is omitted, the amount is computed by Sort/Merge.

CALL SMFILE(smo,type,lfn,action)

One call to SMFILE must be issued for each file used during Sort/Merge processing with one exception: if the output file is handled by SMOWN, no call to SMFILE is required for that file.

smo	Specifies file processing: ≠SORT≠ ≠MERGE≠ ≠OUTPUT≠

type — Indicates type of input/output used for file access:

≠FORMATTED≠ or ≠CODED≠	Formatted
≠BINARY≠	Unformatted
0 (zero)	Record Manager interface routines

lfn — File name can be a tape number or the name left-justified with zero fill (nLlfn).

action — Indicates action to be taken with the file upon Sort/Merge completion:

≠REWIND≠

≠UNLOAD≠

≠NONE≠ (Default)

CALL SMKEY(bytepos,bitpos,nbyte,nbits,type,colseq,order)

One SMKEY call is required for each sort key.

bytepos, bitpos — Starting position of the sort key in relation to the first 6-bit byte of the record, counting from 1. bytepos gives the byte; bitpos gives the bit within the byte.

nbyte, nbits — Length of the sort key in 6-bit bytes, or characters (nbyte) plus bits (nbits).

The remaining three parameters are optional:

type — Specifies type of code used to interpret keys (under Sort/Merge 4, type is a single option, except ≠LEADING≠ or ≠TRAILING≠ can be used in any order with either ≠SIGN≠ or ≠SEPARATE≠ separated by commas):

≠DISPLAY≠	Internal display code
≠FLOAT≠	Floating point data
≠INTEGER≠	Signed integer data
≠LOGICAL≠	Unsigned integer data (default)

The following identifiers must be preceded by type ≠DISPLAY≠. The identifiers must be separated by commas as indicated.

≠DISPLAY≠, ≠SIGN≠, ≠LEADING≠	Numeric data in display code; + or - sign present as overpunch at beginning of field (Sort/Merge 4 only).
≠DISPLAY≠, ≠SIGN≠, ≠TRAILING≠	Numeric data in display code; + or - sign present as an over punch at the end of the field.
≠DISPLAY≠, ≠SEPARATE≠, ≠LEADING≠	Numeric data in display code, sign is a separate character at beginning of the field (Sort/Merge 4 only).
≠DISPLAY≠, ≠SEPARATE≠, ≠TRAILING≠	Numeric data in display code; sign is a separate character at the end of the field (Sort/Merge 4 only).

colseq — Name of the user-supplied collating sequence defined by an SMSEQ call, or one of the following collating sequences:

≠ASCII6≠	6-bit ASCII collating sequence (default for installations using ASCII character set)
≠COBOL6≠	6-bit COBOL collating sequence (default for installations using CDC character set)
≠DISPLAY≠	Internal display code collating sequence
≠INTBCD≠	Internal BCD collating sequence
name	Name of a collating sequence specified by a call to SMSEQ

A colseq parameter cannot be used unless the type parameter specifies ≠DISPLAY≠.

order — Sequencing order for sort processing. It can be either of the following:

≠A≠	Ascending (default)
≠D≠	Descending

CALL SMSEQ(colseq,array)

Specifies a user's collating sequence, or redefines the default to be a user collating sequence.

colseq — Names the collating sequence being defined.

array — Name of the array containing characters in the order they are to be collated. Each character should be in nRx format (right-justified with zero fill or ijB format (octal). Unspecified characters collate high and equal. The collating sequence is terminated by a negative number.

CALL SMEQU(colseq,array)

Specifies that two or more characters in the collating sequence are equal for comparison purposes.

colseq — Specifies the collating sequence containing the characters to be equated.

array — Name of the array specifying the characters to be equated. Each character should be in nRx format (right-justified with zero fill) or ijB format (octal). The end of the list of characters to be equated is indicated by a negative number.

CALL SMOPT(option$_1$,...,option$_n$)

Specifies special record handling options. The options are:

≠VERIFY≠
≠RETAIN≠
≠VOLDUMP≠
≠DUMP≠,n
≠NODUMP≠
≠NODAY≠
≠ORDER≠,mo
≠COMPARE≠
≠EXTRACT≠

All of the above options except ≠NODAY≠ are explained in detail under directives.

≠NODAY≠ Specifies suppression of dayfile messages.

CALL SMOWN(exitnum$_1$,subname$_1$,exitnum$_2$,subname$_2$)

Specifies entry point names for the user's relocatable owncode routines.

exitnum$_i$ Number of the owncode exit.

subname$_i$ Corresponding entry point name of the exit.

CALL SMRTN(retaddr,a_1,rl_1,b_1,rl_1)

This call is used to return to Sort/Merge processing from the user's owncode routine when the owncode routine is a FORTRAN Extended subroutine.

retaddr Specifies normal return address or altered return address to continue Sort/Merge processing.

a_i Specifies the integer array where Sort/Merge stores a record when subname is called. Storing into a causes indeterminate results.

b_i Specifies the integer array where the user stores a record when subname is called. b should not be the same as a.

rl_i Record length in characters.

CALL SMTAPE(lfn_1, . . . ,lfn_n)

This call is required for the tape variant of Sort/Merge 4 to specify all magnetic tape intermediate merge files.

lfn_i Name assigned to an intermediate merge file. Each file name must be in nLln format (left-justified with zero fill) and must not be defined by the FORTRAN Extended program.

CALL SMEND

This call initiates Sort/Merge processing and must be the last call for any one sort or merge.

CALL SMABT

This call terminates a sequence of Sort/Merge interface calls without initiating execution of Sort/Merge.

FORTRAN 5 CALLS

Sort/Merge processing can be initiated by a FORTRAN 5 program through special FORTRAN 5 call statements.

CALL SMSORT (mrl,ba)

CALL SMSORTB (mrl,ba)

CALL SMSORTP (mrl,ba)

CALL SMMERGE (mrl,ba)

One of these statements must be the first call of any sort.

SMSORT Specifies sort-only or sort and merge processing.

SMSORTB Specifies a balanced tape sort under Sort/Merge 4.

SMSORTP Specifies a polyphase tape sort under Sort/Merge 4.

SMMERGE Specifies merge-only processing.

mrl Maximum record length in characters of record to be sorted.

ba Under Sort/Merge 1, optional parameter designating the number of words of large core memory (LCM) used as a buffer area for all intermediate scratch files.

Under Sort/Merge 4, optional parameter designating the number of words of Central Memory to be used by Sort/Merge for working storage. If the parameter is omitted, the amount is computed by Sort/Merge.

CALL SMFILE(dis,i/o,lfn,action)

One call to SMFILE must be issued for each file used during Sort/Merge processing with one exception: if the output file is handled by SMOWN, no call to SMFILE is required for that file.

dis Specifies file processing:

'SORT'

'MERGE'

'OUTPUT'

i/o — Indicates type of input/output used for file access:

'FORMATTED' or 'CODED'	Formatted
'BINARY'	Unformatted
0 (zero)	CYBER Record Manager interface routines

lfn — File name can either be a tape number or the name left justified with zero fill. For Sort/Merge 4 when i/o is specified as zero, lfn can be an array containing the file information table.

action — Indicates action to be taken with the file upon Sort/Merge completion:

'REWIND'

'UNLOAD'

'NONE' (default)

CALL SMKEY(charpos,bitpos,nchar,nbits,type,colseq,order)

charpos
bitpos — Starting position of the sort key in relation to the first 6-bit byte of the record, counting from 1. Charpos gives the position of the first character; bitpos specifies the bit within the character (or 6-bit byte) specified by charpos.

nchar
nbits — Length of the sort key in 6-bit bytes, or characters (nchar) plus bits (nbits).

The remaining three parameters are optional:

type — Specifies type of code used to interpret keys:

'DISPLAY'	Internal display code
'FLOAT'	Floating point data
'INTEGER'	Signed integer data
'LOGICAL'	Unsigned integer data (default)

The following identifers must be preceded by type 'DISPLAY'. The identifiers must be separated by commas as indicated.

'DISPLAY', 'SIGN', 'LEADING'	Numeric data in display code; + or - sign present as overpunch at beginning of field (Sort/Merge 4 only).
'DISPLAY', 'SIGN', 'TRAILING'	Numeric data in display code; + or - sign present as overpunch at end of field.
'DISPLAY' 'SEPARATE', 'LEADING'	Numeric data in display code; sign is a separate character at beginning of field (Sort/Merge 4 only).
'DISPLAY', 'SEPARATE', 'TRAILING'	Numeric data in display code; sign is a separate character at end of field (Sort/Merge 4 only).
colseq	Name of the user-supplied collating sequence defined by a SMSEQ call, or one of the following collating sequences.
'ASCII6'	6-bit ASCII collating sequence (default for installations using ASCII character set).
'COBOL6'	6-bit COBOL collating sequence (default for installations using CDC character set).
'DISPLAY'	Internal display code collating sequence.
'INTBCD'	Internal BCD collating sequence.
name	Name of a collating sequence specified in a call to SMSEQ. A colseq parameter cannot be used unless the type parameter specifies 'DISPLAY'.
order	Sequencing order for sort processing. It can be either of the following: 'A' Ascending (default) 'D' Descending

CALL SMSEQ(segname,segspec)

Specifies a user's collating sequence or redefines the default to be a user collating sequence.

segname Names the collating sequence being defined.

segspec — Names the integer array containing the characters in the order they are to be collated. Each character should be in nR"s" format (right-justified with zero fill) or O"o" format (octal). Unspecified characters collate high and equal. The collating sequence is terminated by a negative number.

CALL SMEQU(colseg,equspec)

Specifies that two or more characters in the collating sequence are equal for comparison purposes.

colseq — Specifies the collating sequence containing the characters to be equated.

equspec — Name of the array specifying the characters to be equated. Each character should be in nR"s" format (right-justified with zero fill) or O"o" format (octal). The end of the list of characters to be equated is indicated by a negative number.

CALL SMOPT(opt ,opt ,...)

Specifies special record handling options. The options are:

'VERIFY'
'RETAIN'
'VOLDUMP'
'DUMP',n
'NODUMP'
'NODAY'
'ORDER',mo
'COMPARE'
'EXTRACT'

All of the above options except 'NODAY' are explained in detail under directives.

'NODAY' — Specifies suppression of dayfile messages.

CALL SMOWN(exitnum$_1$,subname$_1$,exitnum$_2$,subname$_2$)

Specifies entry point names for the user's relocatable owncode routines.

exitnum$_i$ — Number of the owncode exit.

subname$_i$ — Corresponding entry point name of the exit.

CALL SMRTN(retaddr,a_1,rl_1,b_1,rl_1)

This call is used to return to Sort/Merge processing from the user's owncode routine when the owncode routine is a FORTRAN 5 subroutine.

retaddr — Specifies normal return address or altered return address to continue Sort/Merge processing.

a_i — Specifies the integer array where Sort/Merge stores a record when subname is called. Storing into a causes indeterminate results.

b_i — Specifies the integer array where the user stores a record when subname is called. b should not be the same as a.

rl — Record length in characters.

CALL SMTAPE(taplist)

This call is required for the tape variant of Sort/Merge 4 to specify all magnetic tape intermediate merge files.

taplist — List of logical file names, each in the form L"filename". The file names in taplist must not be declared in the PROGRAM statement.

CALL SMEND

This call initiates Sort/Merge processing and must be the last call for any one sort or merge.

CALL SMABT

This call terminates a sequence of Sort/Merge interface calls without initiating execution of Sort/Merge.

STANDARD CHARACTER SETS

CDC Graphic	ASCII Graphic Subset	Display Code	Hollerith Punch (026)	External BCD Code	ASCII Punch (029)	ASCII Code
:†	:	00†	8-2	00	8-2	3A
A	A	01	12-1	61	12-1	41
B	B	02	12-2	62	12-2	42
C	C	03	12-3	63	12-3	43
D	D	04	12-4	64	12-4	44
E	E	05	12-5	65	12-5	45
F	F	06	12-6	66	12-6	46
G	G	07	12-7	67	12-7	47
H	H	10	12-8	70	12-8	48
I	I	11	12-9	71	12-9	49
J	J	12	11-1	41	11-1	4A
K	K	13	11-2	42	11-2	4B
L	L	14	11-3	43	11-3	4C
M	M	15	11-4	44	11-4	4D
N	N	16	11-5	45	11-5	4E
O	O	17	11-6	46	11-6	4F
P	P	20	11-7	47	11-7	50
Q	Q	21	11-8	50	11-8	51
R	R	22	11-9	51	11-9	52
S	S	23	0-2	22	0-2	53
T	T	24	0-3	23	0-3	54
U	U	25	0-4	24	0-4	55
V	V	26	0-5	25	0-5	56
W	W	27	0-6	26	0-6	57
X	X	30	0-7	27	0-7	58
Y	Y	31	0-8	30	0-8	59
Z	Z	32	0-9	31	0-9	5A
0	0	33	0	12	0	30
1	1	34	1	01	1	31
2	2	35	2	02	2	32
3	3	36	3	03	3	33
4	4	37	4	04	4	34
5	5	40	5	05	5	35
6	6	41	6	06	6	36
7	7	42	7	07	7	37

CDC Graphic	ASCII Graphic Subset	Display Code	Hollerith Punch (026)	External BCD Code	ASCII Punch (029)	ASCII Code
8	8	43	8	10	8	38
9	9	44	9	11	9	39
+	+	45	12	60	12-8-6	2B
-	-	46	11	40	11	2D
*	*	47	11-8-4	54	11-8-4	2A
/	/	50	0-1	21	0-1	2F
(	(	51	0-8-4	34	12-8-5	28
)	)	52	12-8-4	74	11-8-5	29
$	$	53	11-8-3	53	11-8-3	24
=	=	54	8-3	13	8-6	3D
blank	blank	55	no punch	20	no punch	20
(comma)	(comma)	56	0-8-3	33	0-8-3	2C
(period)	(period)	57	12-8-3	73	12-8-3	2E
≡	#	60	0-8-6	36	8-3	23
[	[	61	8-7	17	12-8-2	5B
]	]	62	0-8-2	32	11-8-2	5D
%††	%	63	8-6	16	0-8-4	25
≠	" (quote)	64	8-4	14	8-7	22
→	– (underline)	65	0-8-5	35	0-8-5	5F
∨	!	66	11-0	52	12-8-7	21
∧	&	67	0-8-7	37	12	26
↑	' (apostrophe)	70	11-8-5	55	8-5	27
↓	?	71	11-8-6	56	0-8-7	3F
<	<	72	12-0	72	12-8-4	3C
>	>	73	11-8-7	57	0-8-6	3E
≤	@	74	8-5	15	8-4	40
≥	\	75	12-8-5	75	0-8-2	5C
¬	∧ (circumflex)	76	12-8-6	76	11-8-7	5E
; (semicolon)	; (semicolon)	77	12-8-7	77	11-8-6	3B

†Twelve or more zero bits at the end of a 60-bit word are interpreted as end-of-line mark rather than two colons. End-of-line mark is converted to external BCD 1632.

††In installations using the CDC 63-graphic set, display code 00 has no associated graphic or Hollerith code; display code 63 is the colon (8-2 punch).

STANDARD COLLATING SEQUENCES

COBOL6†		Display†		INTBCD		ASCII6††	
Graphics	Display Code	Graphics	Display Code	Graphics	CDC INTBCD	Graphics	Display Code
blank	55	:†	00†	0	00	blank	55
≤	74†	A	01	1	01	!	66
%†	63	B	02	2	02	"	64
[	61	C	03	3	03	#	60
→	65	D	04	4	04	$	53
≡	60	E	05	5	05	%††	63
∧	67	F	06	6	06	&	67
↑	70	G	07	7	07	'	70
↓	71	H	10	8	10	(	51
>	73	I	11	9	11	)	52
≥	75	J	12	:	12	*	47
¬	76	K	13	=	13	+	45
.	57	L	14	≠	14	,	56
)	52	M	15	≤	15	-	46
;	77	N	16	%	16	.	57
+	45	O	17	[	17	/	50
$	53	P	20	+	20	0	33
*	47	Q	21	A	21	1	34
-	46	R	22	B	22	2	35
/	50	S	23	C	23	3	36
,	56	T	24	D	24	4	37
(	51	U	25	E	25	5	38
=	54	V	26	F	26	6	39
≠	64	W	27	G	27	7	40
<	72	X	30	H	30	8	41
A	01	Y	31	I	31	9	42
B	02	Z	32	<	32	:	00
C	03	0	33	.	33	;	77
D	04	1	34	)	34	<	72
E	05	2	35	≥	35	=	54
F	06	3	36	¬	36	>	73
G	07	4	37	;	37	?	71
H	10						
I	11	5	40	-	40	@	74

COBOL6†		Display†		INTBCD		ASCII6††	
Graphics	Display Code	Graphics	Display Code	Graphics	CDC INTBCD	Graphics	Display Code
V	66	6	41	J	41	A	01
J	12	7	42	K	42	B	02
K	13	8	43	L	43	C	03
L	14	9	44	M	44	D	04
M	15	+	45	N	45	E	05
N	16	-	46	O	46	F	06
O	17	*	47	P	47	G	07
P	20	/	50	Q	50	H	10
Q	21	(	51	R	51	I	11
R	22	)	52	V	52	J	12
]	62	$	53	$	53	K	13
S	23	=	54	*	54	L	14
T	24	blank	55	↑	55	M	15
U	25	,	56	↓	56	N	16
V	26	.	57	>	57	O	17
W	27	≡	60	blank	60	P	20
X	30	[	61	/	61	Q	21
Y	31	]	62	S	62	R	22
Z	32	%†	63†	T	63	S	23
:	00†	≠	64	U	64	T	24
0	33	→	65	V	65	U	25
1	34	v	66	W	66	V	26
2	35	^	67	X	67	W	27
3	36	↑	70	Y	70	X	30
4	37	↓	71	Z	71	Y	31
5	40	<	72	]	72	Z	32
6	41	>	73	,	73	[	61
7	42	≤	74	(	74	\	75
8	43	≥	75	→	75	]	62
9	44	¬	76	≡	76	^	76
		;	77	^	77	-	65

†Under the CDC 63-character set, there is no percent graphic; the colon is display code 63. Display code 00 is not used.

††Under the ASCII 63-character set, there is no percent graphic; the colon collates in position 05, not position 32.

CORPORATE HEADQUARTERS
P.O. BOX 0
MINNEAPOLIS, MINNESOTA 55440

SALES OFFICES AND SERVICE CENTERS
IN MAJOR CITIES
THROUGHOUT THE WORLD

PRINTED IN U.S.A.

CONTROL DATA CORPORATION