Burroughs Corporation
COMPUTER SYSTEMS GROUP
SANTA BARBARA PLANT

PLANT IDENT | 0 1 4

DOCUMENT TRANSMITTAL

PAGE ____ OF ____

CONTROL DOC. None | REV

SYSTEM/PRODUCT B1800/B1700 DISTRIBUTION TYPE

UNIT NAME PROJ. NO.

COORD. CHKR. J Hale RECORDS ISSUE DATE 6-12-77

DOC PFX OR DWG SIZE	DOCUMENT NUMBER	REV OR ISSUE	DOCUMENT DESCRIPTION	ENG. RECORDS ONLY QTY REV PAGE	COPIES EACH PAGE	BOND	VELLUM	REMARKS
P.S.	2212 5207	D	SORT MERGE	15	65			

Burroughs Corporation

COMPUTER SYSTEMS GROUP
SANTA BARBARA PLANT

P.S. 2212 5207
B1800/B1700 SORT MERGE

PRODUCT SPECIFICATION

REV LTR	REVISION ISSUE DATE	APPROVED BY	REVISIONS
D	6-11-79		Changes for Mark 9.0 Release
			3-3 Changed MERGE.MEDIA.NUMBER - BIT(24) to BIT(8) Added FILLER and MERGE.PURGE.INPUT.FILES

BURROUGHS CORPORATION
COMPUTER SYSTEM GROUP
SANTA BARBARA PLANT

TABLE OF CONTENTS

BURROUGHS CORPORATION
COMPUTER SYSTEM GROUP
SANTA BARBARA PLANT

GENERAL

B1800/B1700 SORT/MERGE is a program designed to allow the user to merge designated files that are ordered in the same manner. In addition, the user can alter the virtual collating sequence with which the files are merged.

This specification describes the merge and interface requirements of the B1800/B1700 SORT/MERGE program. It defines the input, output, and memory requirements needed for merging files and presents the various options open to the user during a merge operation. A discussion of MCP interface for B1800/B1700 SORT/MERGE and a description of the Collate Table file are also provided. This specification does not discuss the merge constructs necessary to a particular language. The user must see the language manual of the respective source language for that information.

SORT/MERGE is defined for implementation with SDL, UPL, and COBOL languages for the B1800/B1700 systems. It may also be called by the B1800/B1700 SORT program.

RELATED DOCUMENTATION

NUMBER	NAME
P.S. 2201 2389	SDL S-Language
P.S. 2201 6729	COBOL S-Language
P.S. 2201 6752	B1800/B1700 SORT Language
P.S. 2212 5371	Translate Table Generator
#5000847	B1800/B1700 SDL Manual (BNF Version)

BURROUGHS CORPORATION
COMPUTER SYSTEM GROUP
SANTA BARBARA PLANT

OPERATING SPECIFICATIONS

FUNCTIONAL DESCRIPTION: MERGE

The program combines the records of designated files according to assigned keys in the specified collating sequence. Signed keys are merged algebraically; i.e., negative numbers are less than any positive number. Ascending or descending order for each key may be specified.

The maximum number of keys is 30 unsigned keys, or 15 signed keys. In a combination of signed and unsigned keys, each signed key counts as two keys, and the total must not exceed 30. Each key may have a total length of 4095 bits.

MAIN MEMORY REQUIREMENTS

The program will request enough memory to perform the specified merge with singly buffered files. It assumes a minimum of 8K bytes of memory although it may use less than that amount for small merges. If additional memory is available from calling program roll-out or memory specification, then additional buffers will be used for improved performance.

INPUT

INPUT MEDIA

The input medium to the program can only be one of the following:

A. Cards
B. 7-track magnetic tape(s)
C. 9-track magnetic tape(s)
D. Any B1800/B1700 disk device except diskette

INPUT RESTRICTIONS

Each input file must be wholly contained on a single hardware type. The mixture of hardware types (such as 7-track and 9-track magnetic tapes, or HPT disk and disk pack, for example) is not permitted. A minimum of two input files must be specified; and the allowable maximum is eight.

INPUT ASSUMPTIONS

Magnetic tape input is defaulted to EBCDIC coding if 9-track and BCL coding if 7-track. The user may specify "ANY TAPE" in which case it is his responsibility to make certain that the format is acceptable to the hardware type he has selected.

INPUT LIMITATIONS

The maximum record length for merging is 65,534 bits.

The maximum block-size limit is a function of the available memory space. Input records that are larger than output records (from MERGE) are truncated on the right to match the smaller size. Smaller records are padded on the right with <BLANKS>.

INPUT PARAMETERS

The parameters required by the SORT/MERGE are described in the SORT/MERGE program Interface tables.

The SORT/MERGE program requires information about the disposition of input/output files (e.g., close with purge) and unreadable blocks in the input data file. The keys and the Collate Table file (if used) must also be specified. (see MERGE Key Tables and Collate Table File.)

OUTPUT

The output medium from the program can only be one of the following:

A. Cards
B. 7-track magnetic tape(s)
C. 9-track magnetic tape(s)
D. Any B1800/B1700 disk device except diskette
E. Line printer

OUTPUT LIMITATIONS

Output records to a line printer that are smaller than a printer line appear left-justified with space fill on the right. The added content of other records defined to be larger than input records is nulls.

Output records that are shorter than input records are truncated from the right.

VARIABLE LENGTH RECORDS

The program can accept as input, and produce for output, magnetic-tape variable-length records. When variable-length records are used, the record size is stored right-justified in the first four bytes of each record. The maximum record and block sizes are specified in the sort information block.

When this type is passed to merge, each input record is standardized at the maximum length of the longest variable record. No sort key location should be specified which is outside the bounds of the shortest variable-length record to be moved.

DUPLICATE CHECKING

The occurrence of duplicate records is checked during the MERGE when the MERGE.DUPCHECK option is set. If any are found, a message will be displayed and a file of relative record pointers will be created named SD.<JOB NUMBER>. It will contain 8-character records which are two 4-character packed decimal pointers to the duplicates and the records duplicated.

SOFTWARE IMPLEMENTATION

SORT/MERGE is invoked by an MCP communicate. For language constructs that apply to a particular source language, the user must see the appropriate source-language manual.

If the optional Collate Table File created by the SORT/COLLATE program is desired, the user must specify the proper file through the appropriate interface control (see Table 3.1 below).

SORT MERGE PROGRAM INTERFACE

Information in the following tables (3.1-3.4) controls MERGE Intrinsic interface:

```
*************************************************************
*                                                           *
* CT.VERB    = 29                                           *
* CT.OBJECT  = BASE RELATIVE ADDRESS OF MERGE               *
*              INFORMATION TABLE                            *
* CT.ADVERB  = 0                                            *
* CT.1       = BASE RELATIVE ADDRESS OF MERGE               *
*              KEY TABLE                                    *
* CT.2       = BASE RELATIVE ADDRESS OF MERGE               *
*              INPUT TABLE                                  *
* CT.3       = OUTPUT FILE NUMBER                           *
* CT.4       = TRANSLATE FILE NUMBER OR NOT ZERO            *
*                                                           *
*************************************************************
```

TABLE 3.1 MERGE COMMUNICATE

```
*****************************************************************
*  MERGE.TYPE              BIT(2)  11 = MERGE                   *
*  MERGE.MEDIA.TYPE        BIT(6)   0 = MERGE                   *
*  FILLER                  BIT(8)                               *
*  MERGE.PURGE.IN-         BIT(8)  1 = CLOSE CORRESPONDING INPUT*
*  PUT.FILES                       FILE WITH PURGE              *
*  MERGE.MEDIA.NUMBER      BIT(8)  NUMBER OF INPUT FILES        *
*  MERGE.RECORD.SIZE       BIT(24) LARGEST KEY DISPLACEMENT     *
*  FILLER                  BIT(67)                              *
*  MERGE.OUTPUT.HDWR       BIT(6)  HARDWARE TYPE                *
*  MERGE.OUTPUT.RECSIZ     BIT(24) MAXIMUM RECORD SIZE IN BITS  *
*  MERGE.OUTPUT.BLKSIZ     BIT(24) MAXIMUM BLOCK SIZE IN BITS   *
*  MERGE.OUTPUT.CLOSE      BIT(12) CLOSE TYPE                   *
*  MERGE.OUTPUT.VARFLAG    BIT(1)  1 = VARIABLE RECORDS         *
*  FILLER                  BIT(2)                               *
*  MERGE.PARITY.ACTION     BIT(1)  0 = DS, 1 = IGNORE RECORD    *
*  FILLER                  BIT(32)                              *
*  MERGE.REPORT.FLAG       BIT(1)  0 = NO REPORT WANTED         *
*                                  1 = PRINT SORT SPECIFICATIONS*
*  MERGE.NUMBER.KEYS       BIT(5)  NUMBER OF MERGE KEYS         *
*  MERGE.KEY.LENGTH        BIT(24) TOTAL LENGTH OF KEY(S) IN BITS*
*  FILLER                  BIT(44)                              *
*  MERGE.DUPCHECK          BIT(1)  1 =CHECK FOR DUPLICATE RECORDS*
*  FILLER                  BIT(19)                              *
*  MERGE.MEMORY.SIZE       BIT(24) MEMORY SIZE IN BITS          *
*                                  TO ALLOCATE                  *
*  FILLER                  BIT(1)                               *
*  MERGE.TRANSLATE         BIT(1)  1 = TRANSLATE                *
*  FILLER                  BIT(55)                              *
*****************************************************************
```

TABLE 3.2 MERGE Information Table

```
*************************************************************************
*                                                                       *
*   UNSIGNED KEY(S)                                                     *
*     01  KEY.FLAGS             BIT(4),                                 *
*         02 SIGN FLAG          BIT(1),    %0 = UNSIGNED                *
*         02 DIRECTION          BIT(1),    %0 = ASCENDING               *
*                                          %1 = DESCENDING              *
*         02 FILLER             BIT(1),    %  = 0                       *
*         02 TRAN KEY           BIT(1)     %TRANSLATE OPTION            *
*     01  KEY.LENGTH            BIT(12),   %LENGTH OF KEY IN BITS       *
*     01  KEY.DISPLACEMENT      BIT(20),   %LOCATION FROM               *
*                                           BEGINNING OF RECORD         *
*   SIGNED KEY(S)                                                       *
*     01  KEY FLAGS            BIT(4),                                 *
*         02 SIGN.FLAG          BIT(1),    %1 = SIGN                    *
*         02 SIGN.DIRECTION     BIT(1),    %0 = ASCENDING               *
*                                          %1 = DESCENDING              *
*         02 NEW.FORMAT         BIT(1),    %1 = THIS FORMAT             *
*         02 FILLER             BIT(1),    %  = 0                       *
*     01  LENGTH.AND.SIGN                  %=A40DA                      *
*         02 SIGN.LENGTH        BIT(4),    %SIGN LENGTH                 *
*         02 SIGN               BIT(8),    %SIGN                        *
*     01  SIGN DISPLACEMENT     BIT(20),   %LOCATION OF SIGN FROM       *
*                                           BEGINNING OF RECORD         *
*                                                                       *
*                                                                       *
*************************************************************************
```

TABLE 3.3 MERGE Key(s)

MERGE INPUT FILES

The file numbers of the input files to be merged are put into the MERGE Input Table (Table 3.4 below) in a top-down manner. The first eight bits of the table are set to zero for table format.

```
**************************************************************************
*                                                                        *
*  01  MERGE.INPUT.TABLE          BIT(80)                                *
*      02  MERGE.VERSION          BIT(8)     = 0                         *
*      02  MERGE.DISK.INPUT       BIT(8)     NUMBER OF DISK INPUT        *
*                                             FILES                      *
*      02  MERGE.INPUT.1          BIT(8)     INPUT FILE NUMBER           *
*      02  MERGE.INPUT.2          BIT(8)     INPUT FILE NUMBER           *
*      02  MERGE.INPUT.3          BIT(8)     INPUT FILE NUMBER           *
*      02  MERGE.INPUT.4          BIT(8)     INPUT FILE NUMBER           *
*      02  MERGE.INPUT.5          BIT(8)     INPUT FILE NUMBER           *
*      02  MERGE.INPUT.6          BIT(8)     INPUT FILE NUMBER           *
*      02  MERGE.INPUT.7          BIT(8)     INPUT FILE NUMBER           *
*      02  MERGE.INPUT.8          BIT(8)     INPUT FILE NUMBER           *
*                                                                        *
**************************************************************************
```

TABLE 3.4 MERGE Input Table (Format 1)

MERGE KEY DESCRIPTORS

The Key Table may contain a maximum of thirty 36-bit key descriptions. Each description will specify ascending or descending order, key length, and key displacement. An additional description must be supplied for signed keys which specify the type, length, and displacement of the sign.

COLLATE TABLE FILE

GENERAL: COLLATE

Use of the optional Collation Table file created by the Sort/Collate program invokes the virtual collating sequence capability of the MERGE INTRINSIC.

This option permits the alteration of the sequence in which MERGE combines records during the merging process. Normally, all characters encountered in the merge keys are compared in the hardware collating sequence; i.e., 2002 through 2FF2. Only those elements of the merge key described as unsigned alphanumeric are affected by the translation capability. Computational merge keys are always processed according to the hardware collating sequence.

The Collate Table file can be created to:

a. Specify a new collating sequence for the particular programs invoking merge.

b. Retain the normal collating sequence of 2002 through 2FF2 except for certain characters whose rank in the sequence it is desired to interchange.

c. Make a number of characters have the same rank for the ordering of records.

The Collate Table is frequently required for foreign alphabets or conversion from other processing systems.

FUNCTIONAL DESCRIPTION: COLLATE

The MERGE Intrinsic with the Collate Table file option is invoked by specifying the name of the Collate Table file in the merge key parameters (see MERGE INTRINSIC INTERFACE).

The MCP interface verifies that the Collate Table file is on disk before proceeding with the merge. If the file is not present at the time merge is invoked, the user is directed to load the file.

The names of the Collate Table files should be unique to an installation so the files are not inadvertently merged into the wrong sequence. The Collate Table file consists of two 256-byte records in a single area file on disk.

The MCP interface verifies the header information prior to opening the Collate Table file to ensure the file is 256-byte records in a single-area file. If not, a syntax error is printed. For the format of the Collate Table file, see SORT/COLLATE (P. S. 2212 5371).

The Intrinsic brings the Collate Table file into main memory and, as the merge key is extracted from each record, those elements of the merge key which are declared as unsigned eight-bit (byte) format are processed through a translation operation before being passed to the Intrinsic comparison logic. Computational (i.e., 4-bit) MERGE key fields are not affected by the translation.

INDEX