FAST SORT PROGRAM
FASTSORT
User's Guide

Version 3

August, 1978

Model Code No. 50251

DATAPOINT CORPORATION

The leader in dispersed data processing ™

FAST SORT PROGRAM
FASTSORT

User's Guide

Version 3

August, 1978

Model Code No. 50251

PREFACE

The FASTSORT program produces a sorted copy of a text file on disk. FASTSORT will run on any DATAPOINT processor with the 5500 instruction set and at least 24K bytes of user memory.

FASTSORT is both efficient and easy to use. Many options are provided for key specification, output format, and record selection, but default options allow most sorts to be accomplished with simple commands.

TABLE OF CONTENTS

CHAPTER 1. FASTSORT

1.1 Introduction

The FASTSORT command produces a sorted copy of a text file on disk. FASTSORT will run on any DATAPOINT processor with the 5500 instruction set and at least 24K bytes of user memory.

FASTSORT is both efficient and easy to use. Many options are provided for key specification, output format, and record selection, but default options allow most sorts to be accomplished with simple commands.

1.2 Command Statement Format

The following is the statement format for Datapoint DOS FASTSORT:

```
FASTSORT IN,OUT[,:DRk][,SEQ][;[F][Mn][O][Q][R][GNNNTC][S][K1]...[,On][,Kn]]
```

Information contained within a pair of square brackets [] is optional; information within brackets is order-dependent. Commas may be used to delimit parameters. (NOTE that commas MUST be used to delimit sort-key groups.) The first four fields (those ahead of the semi-colon) are considered to be file specification fields. The fields following the semicolon are considered to be sort key parameters. Default conditions are listed below.

The following list defines the parameters which can be specified:

IN..............This specifies the input file. This file must exist on disk. The default extension is TXT.

OUT.............This specifies the output file. The default extension is the extension of the input file. If no disk drive is specified and the file exists on a drive on-line to the system then the output file will over-write the existing file. If no disk drive is specified and no file of that name exists on a drive on-line to the system then a file of the given name will be created on the same drive as the input file.

:DRk............This specifies the drive for the temporary file. This is only a working scratch file needed during the sort. If no drive is specified, the file will be placed on the same drive as the output file.

SEQ.............NON-ASCII COLLATING SEQUENCE FILE
This specifies the file which contains the collating sequence to be used. If omitted, the ASCII ordering will be used.

F...............OUTPUT FILE FORMAT.
If 'I' is specified, the output file format will be one record per sector. If 'I' is not specified, the output file will be in the normal text file format, with record compression. See the 'S' parameter description for determination of space compression in the output file.

Mn..............INPUT RECORD LENGTH.
This parameter specifies the maximum number of characters in an input record. The default value is the larger of 80 and the length of the first two input records. This parameter must be specified if any input record is longer than 80 characters and longer than the first two input records. If one of the tag format parameters ('K' or 'T') is specified, the default of 80 above is replaced by 1000. Decreasing this parameter will make the sort run faster, so it should be as close to the actual maximum input record size as possible (but not smaller) if speed is important.

O...............ORDER.
This parameter specifies ascending or descending order of a key. 'A' or no parameter indicates ascending, 'D' indicates descending. Note that if some keys are to be sorted in ascending order and other keys

in descending order, the "On" specification described below should preceed each key whose order differs from the order of the key preceeding it. However, if all keys are to be ordered in the same sequence, this parameter need only be specified once.

Q...............STABLE SORT NOT REQUIRED. Without the 'Q' parameter, records with equal keys will be in the same order in the output file as they were in the input file; the sort is said to be stable. The 'Q' parameter specifies that a stable sort is not required; records with equal keys will be in random order in the output file. Specifying this parameter will result in a smaller temporary file and a faster sort; this parameter can never affect the result of a sort if all keys are unique.

R...............RECORD FORMAT.
This parameter specifies a special output record format: Limited output file format or Tag file or Keytag file output. The actual character entered is 'L' or 'T' or 'K'. The default value is NO SPECIAL OUTPUT RECORD FORMAT; that is, neither 'L' nor 'T' nor 'K', so that the records in the output file will be exact copies (FULL IMAGE RECORDS) of the records in the input file.

Normally the sort transfers all the characters of the records of the input file to the output file. It is possible to transfer only part of each record. Including the 'L' parameter in the list of parameters will cause another question to be asked wherein you may specify the limitations. See the section on Limited Output Format Option.

By entering the 'T' character an output file is generated which consists only of binary record number and buffer byte pointers to the input file records. See the section on Tag File Output Format Option.

By entering the 'K' character a standard text format output file is generated which consists of records containing a 5 byte user logical record number, a 3 byte buffer address, and the key. These records are space-compressed and have trailing spaces truncated. See the section on Keytag File Output Format Option.

G..............GROUP INDICATOR
This parameter specifies that the input file consists of PRIMARY and SECONDARY records and specifies which GROUP is to be sorted. The actual character entered is 'P' for PRIMARY or 'S' for SECONDARY. There is no default value.

IF the 'G' option is entered THEN the NNNTC options MUST ALSO be entered.

In a file with PRIMARY and SECONDARY records a string of records with a PRIMARY record as the first record and SECONDARY records following it is considered one block, or group, of records.

When the file is sorted on PRIMARY records the output file has the blocks of records re-ordered so that the PRIMARY records are in the sorted sequence; no change is made in the sequence of the secondary records following each PRIMARY record. When the file is sorted on SECONDARY records the output file has the blocks of records in the same order as in the input file, but the SECONDARY records within each block are in the sorted sequences.

FASTSORT has no provision for the sorting of PRIMARY and SECONDARY records in the same run.

NNN............NUMERIC position of PRIMARY/SECONDARY flag.
This parameter specifies the character position for the character (the 'C' parameter) indicating whether the record is a PRIMARY or SECONDARY record . The number MUST be specified if the option is taken.

T...............TYPE of evaluation. This parameter specifies equivalence or inequivalence of the group indicator character; that is, whether the character in the record will be EQUAL to or NOT EQUAL to the character specified. The actual character entered is '=' for equal or '#' for not equal. There is no default character, '=' or '#' must be given if the option is taken.

If '=' is given then if the character in the NNNth position of an input file record is EQUAL to the group indicator character -- indicated by 'C' below -- then the record is a member of the specified sort group -- indicated by 'G' above. Otherwise, it is not a member of the specified group.

C...............CHARACTER, group indicator This parameter specifies the actual test character for determination of a record's membership in the sort group. The actual character entered is any member of the available character set -- this means any combination of eight bits -- except 015. There is no default character: the character immediately following the 'T' parameter is taken to be the 'C' parameter -- except a 015.

S...............OUTPUT FILE SPACE COMPRESSION. This parameter affects whether the output file is space compressed. If 'N' is specified, spaces in the output file will not be compressed. If 'C' is specified, spaces in the output file will be compressed. The 'N' parameter is implied by the 'I' parameter unless the 'C' parameter is also specified. The 'C' parameter is implied by the 'K' or 'L' parameter unless the 'I' or 'N' parameter is also specified. If none of the parameters 'C', 'I', 'K', 'L', or 'N' is specified, spaces in the output file will be compressed if and only if the input file contained compressed spaces.

K1..............SSS-EEE

This is the first sort key specification. If no key is specified, FASTSORT will assume the entire record.
SSS is the starting key position.
EEE is the ending key position. EEE must be greater than or equal to SSS.

On..............This specifies the order for the nth key (ascending and descending are indicated by 'A' or 'D'). If omitted the order used on the previous key is assumed.

Kn..............SSS-EEE
The nth sort key specification.

1.3 Collating Sequence File

By specifying a sequence file, the user may substitute any collating sequence for the standard ASCII character set. The file name contains eleven characters, eight of which are the file name and three of which are the extension (example, EBCDIC/SEQ:DRn). The last three characters (the extension) must be "SEQ". If the disk drive number on which the file resides is omitted, FASTSORT defaults to the same drive from which FASTSORT itself was loaded. This table may be supplied by the user but must meet certain requirements to be loaded:

1. It must be an absolute object file.
2. It must begin loading at location 027400.
3. The first eleven bytes must contain the file name and the extension must be SEQ.
4. The table itself must begin loading at location 027400 and occupy 256 bytes (overstoring the file name described in 3). For instance, the source for the EBCDIC sequence file begins:

```
SET     027400
DC      'EBCDIC  SEQ'
SET     027400
DC      0,1,2,3,4,5,6,7,
.
.
.
```

1.4 Ascending and Descending sequences

Changing the collating sequence from ascending to descending is the same as 'reversing' the file, or placing the last first, etc. Sorting a telephone directory in ascending sequence on name produces the familiar order. Should it be sorted in descending sequence, then Mr. Zyk would be first and Mr. Aardvark would be last. The order of collation, when alphabetic, numeric, and punctuation characters all can occur in a column together, follows the character set order. The sequence may be specified for each sort key. However, it need not be specified if it is the same as the key which preceeds it. Therefore, it is possible to sort portions of the key in ascending order and portions in descending order.

1.5 Limited output format option

In many cases, especially when making reports, directories etc. from the data base, it isn't necessary to have the entire record transferred from the input file to the output file during a sort. For instance, an entire personnel data base can be sorted by name to produce an internal company telephone directory. However, it is obvious that all that is needed is the name and telephone number, NOT all the other payroll information. Therefore, FASTSORT permits transferring only that part of the data base desired.

The following is the generalized statement format for the limited output specification which is entered as a second line of parameters:

```
SSS[-EEE][,<DUPLICATE OF PRECEEDING>]...
```

Items within square brackets [] are optional.

The following list defines the parameters which can be specified:

```
SSS............STARTING position within input record.
EEE............ENDING position within input record.
               These parameters specify the character
               positions within the input record to be
               copied to the output record.  The EEE
               specification is optional; if it is not
               specified then only one character, the
               character at SSS, will be copied from the
               input record to the output record.
```

In the same manner that the key of the records is specified by fixed column number, i.e. 1-10 for the first ten characters, the limited output feature specifies that part of the records to be transferred. Should the response 1-10 be given to the limited output format request, only the first ten characters of each record will be transferred to the output file. The limited output format specifier operates in the same manner as the specification of multiple discontiguous sort key fields. For instance, 1-10,50-70 would transfer thirty-one characters from each record of the input file to the output file. The eleventh character in the output record would be the fiftieth character of the input record, etc.

To invoke the limited output format option, the operator includes the 'L' parameter in the specifier list. If and only if the L is specified during the FASTSORT call, will there be a second question asked of the operator on the next line:

LIMITED OUTPUT FILE FORMAT:

This question requires at least one non-trivial field specification. The number of field specifications is only limited by that which can fit on the keyed in line.

Note that the output file requires proportionally less room than the input file when limited. Often this fact can be put to use when the disk file space is nearly exhausted and a sort is required.

1.6 TAG file output format option

For some applications it is useful to have a data file sorted into several different sequences. However, to have several copies of a file on disk merely to have it in different sequences consumes a lot of disk space, and indeed if the file is a very large file many copies of it may not fit onto one or even four disk packs.

This problem could be avoided if there were a way to index into the one main file in any of several different sequences. The index pointers could exist as a file, and the index entry for each record in the main file would only have to be three bytes long -- two bytes for the LRN (Logical Record Number) and one byte for the BUFPTR (Buffer Pointer -- a pointer to the beginning of the actual desired record within the disk physical buffer).

FASTSORT provides for the generation of such an indexing file, a TAG file, by the 'T' variation of the 'R' option. The format of a TAG file is simple:

1. For each record in the input file, the TAG file will have a three byte binary pointer to the first byte of the record.

2. The format of the pointer is:
   Byte 1: MSPLRN (Most Significant Portion of LRN),
   Byte 2: LSPLRN (Least Significant Portion of LRN),
   Byte 3: BUFPTR (Buffer Pointer).

3. The three-byte binary pointers are blocked 83 to a physical disk record.

4. The Physical-End-Of-Record mark is an 003 and the rest 000's.

5. The End-Of-File mark is: beginning at the first byte in the physical record, six 000's, one 003, and the rest 000's.

TAG files may be used by assembly language programs or by RPG II (as Record Address files).

For users writing their own Assembly language code to use a TAG file, it is important to know that the MSPLRN and LSPLRN are together a 16-bit binary pointer to the DOS LOGICAL RECORD NUMBER of the input file, as opposed to the USER LOGICAL RECORD NUMBER. The difference is this: The DOS LOGICAL RECORD NUMBER of a file points to the actual Nth record (starting with zero, the primary RIB) in the file, whereas the USER LOGICAL RECORD NUMBER of a file points to the Nth DATA RECORD (starting with the zeroth data record) in the file. Thus a DOS LRN of zero points to the very first record of the file, which is the master copy of the RIB, a DOS LRN of one points to the second record of the file which is the RIB copy, a DOS LRN of two points to the third record of the file (which is the FIRST DATA RECORD of the file and the USER LOGICAL RECORD NUMBER zero), and so on. The LRN given in the TAG file can NOT be used with the POSIT$ routine unless it is biased by -2. It is much easier to simply place the LRN from the TAG file directly into the LOGICAL FILE TABLE ENTRY for the file that is indexed.

The case with the BUFFER POINTER byte is similar to the LRN pointer bytes. The BUFFER POINTER byte from the tag file is the DOS BUFFER POINTER as opposed to the USER BUFFER POINTER. The difference is this: the DOS BUFFER POINTER points to the actual

Nth byte of a disk buffer (starting with zero), whereas the USER BUFFER POINTER points to the Nth DATA BYTE in the disk buffer; the beginning (zeroth) DATA BYTE in the buffer is the fourth byte in the buffer; the first three bytes are reserved for the DOS. Thus, a DOS BUFPTR of zero points to the very first byte in the buffer, which is the PFN (Physical File Number) of the file, a DOS BUFPTR of one points to the second byte in the buffer, which is the DOS LSPLRN, a DOS BUFPTR of two points to the third byte in the buffer, which is the DOS MSPLRN, a DOS BUFPTR of three points to the fourth byte of the buffer (which is the very first DATA BYTE in the buffer), and so on. The BUFPTR given in the TAG file can NOT be used with the GETR$ or PUTR$ routines unless it is biased by -3. It is much easier to simply place the BUFPTR from the TAG file directly into the LOGICAL FILE TABLE ENTRY for the file that is indexed.

If the TAG file option is specified then the LIMITED OUTPUT FILE FORMAT can NOT be specified.

If a TAG file is generated when the 'P' (PRIMARY SORT) option is specified then TAG file pointers will be generated only to the PRIMARY records in the input file.

If a TAG file is generated when the 'S' (SECONDARY SORT) option is specified then TAG file pointers will be generated only to the SECONDARY records in the input file.

1.7 KEYTAG file output format option

Requesting a Keytag file output will cause a file to be created. This GEDIT- compatible text file contains the record pointers and the key. The record pointers (first 8 bytes of the record) consist of a 5 byte logical record number (range 0 to 65,535) and a 3 byte buffer address. The record number is the user logical record, that is, zero points to the first data sector. Therefore, the user logical record number, converted to binary, may be used with the POSIT$ routine. The buffer address is the modified buffer pointer, that is, one points to the first data byte in a sector. It may be used by the GETR$ routine if biased by -1.

1.8 Disk space requirements

FASTSORT uses a temporary file (SORTMRG/SYS) during a sort. This file is deleted at the end of the sort. The size of this file will usually be approximately 10 to 50 percent larger than the output file, depending on the record size and the sort options.

1.9 Using CHAIN to cause a merge

Consider a situation wherein a system has a master file called 'MASTER' and a file of records to be added, in sequence, to the master file called 'ADDFILE'. To merge these two files in sorted sequence at the end of each day would normally require a sequence of keyed in operations which are somewhat complicated and error prone. CHAIN can cause an effective MERGE and assign it a single name as follows:

```
SAPP MASTER,ADDFILE,MASTER
FASTSORT MASTER,SCRATCH;1-20
KILL MASTER/TXT
NAME SCRATCH/TXT,MASTER/TXT
```

Note that the procedure:
1) appends the ADDFILE to the MASTER file.
2) Sorts the extended MASTER file into a SCRATCH file.
3&4) Renames the SCRATCH file as the new MASTER file. Thus, it is apparent that a merge can be effectively achieved using FASTSORT and by using chain to pre-define the procedure.

1.10 Incompatibilities with the old sort (SORT)

1. FASTSORT will only run on a processor with the 5500 instruction set and at least 24K bytes of user memory.
2. If the maximum input record length is longer than 80 characters (1000 characters if the 'K' or 'T' parameter is specified) and longer than the first record, FASTSORT requires the user to specify the length with the Mn parameter.
3. The temporary file can grow much larger under FASTSORT. This file will usually be a little larger than the output file. If no drive is specified for this file, it will be placed on the same drive as the output file. SORT tried to place its temporary files on the optimum drive, but often picked a write protected drive, or one with

insufficient space.

4. The limited output specification is restricted to columns of the input record. There is no hardcopy output facility.
5. The assembly language linkage to the sort is available for compatibility, but programs using this linkage must not have destroyed the DOS function loader (07400-07777). It is recommended that programs call FASTSORT using the recursive CHAIN and the feature whereby a program can supply the next command line to EXIT$.
6. A tag sort on secondary records will yield tags to the secondary records only.
7. Under SORT, a secondary sort with the first key descending reversed the order of the primary blocks. Under FASTSORT, the blocks always remain in the original order.
8. FASTSORT may produce different results if the input file contains both compressed spaces and uncompressed multiple spaces. See the description of the 'S' parameter for the rules under FASTSORT. With SORT, each output record was an exact binary copy of the corresponding input record if none of the parameters 'C', 'I', 'K', 'L', or 'N' was specified.
9. If an attempt is made to run FASTSORT on a 2200, FASTSORT will automatically load and run SORT if it is present.

Manual Name______________________________

Manual Number______________________________

READER'S COMMENTS

Did you find errors in this manual? If so, specify by page.

____

____

____

____

____

Did you find this manual understandable, usable, and well-organized? Please make suggestions for improvement.

____

____

____

____

____

____

Name____________________________Date________________________

Organization____

Street____

City__________________State_________Zip Code____________________

All comments and suggestions become the property of Datapoint.

Fold Here

Fold Here and Staple

First Class
Permit
5774
San Antonio
Texas

BUSINESS REPLY MAIL

No Postage Necessary if mailed in the United States

Postage will be paid by:

DATAPOINT CORPORATION
DIRECTOR OF SOFTWARE SUPPORT
MS - N60
8550 DATAPOINT DRIVE
SAN ANTONIO, TEXAS 78284