Short Order
User Reference Guide

High Performance Sort Utility for Wang/VS Computers

Ninth Edition, March 2000

Copyright

Trademarks

Software License Agreement

Address comments and questions to:

usi@overeasy.com

Table of Contents

1

Introduction

2

Installation

3

General Information

4

Getparm Reference

5

Return Codes

6

Getparm Quick Reference

1 INTRODUCTION

Short Order Version 3

Short Order is a high-performance sort utility for the Wang VS. When installed as a replacement for the Wang-supplied Sort, it provides immediate and dramatic performance improvements for all programs and reports which use the sort utility.

In general, here's what you can expect from Short Order version 3:

Increased Speed Now **4-8 times faster** than the Wang Sort.

Instant Results Starts working the minute you load it – not a single change to any programs or procedures required.

Smaller Workfiles Uses up to **90% less disk space** for workfiles.

Reduced Disk I/O Requires up to **80% less disk I/O** than Wang's Sort.

Reduced CPU Time Uses up to **66% less CPU** time.

New Y2K field Easier handling of two digit year date fields.

The best way to test Short Order is simply to load it on your VS, and then run your daily reports, applications, month-end processing, etc. as you normally would. Notice right away how much faster your biggest reports and DP jobs run. The bigger and more sort-intensive the job, the more dramatic your timesavings will be.

A second, more subtle improvement you may notice after installing Short Order is an improvement in general system performance, especially at peak processing times. Short Order uses significantly less CPU time and Disk I/O as it delivers better performance, leaving more system resources available for other competing system tasks.

Short Order version 3.0 is Y2K compliant; it has no Y2K-specific dependencies. However, many of the applications that use Short Order, and the data that Short Order manipulates, are Y2K sensitive. This sensitivity is due to the use of two digits to represent the year portion of dates. It is now easier to make applications Y2K compliant by the use of Short Orders' *sliding window* technique for processing dates. This 'sliding window' algorithm treats two digits years greater than 50 as dates in the 20th century, and years less than 50 as dates in the 21st century.

2 INSTALLATION

Preparation

Loading Short Order is a one-step process managed by the INSTALL utility. You must have system Security Administrator rights when installing Short Order.

Requirements

Short Order runs on any Wang VS with VS/OS 6 or greater. It occupies approximately 60 blocks of disk space on your system IPL volume

Run INSTALL

Run the program INSTALL in library SHORTO on volume SHORTO.

INSTALL replaces the Wang Sort or an older Short Order version with the copy of Short Order on the diskette. The Wang Sort remains in your system library under the name WANGSORT.

After you’ve successfully installed Short Order, the INSTALL program presents you with two additional options – “Set installation defaults”, and “Re-install Wang Sort”. Choose “Set installation defaults” to customize Short Order for your environment as described below. Choose the option to “Re-install Wang Sort” to put the Wang Sort back into the system library at the end of your test or 30 day demo period. These two options can be set after installation via the SORTCNFG utility, which is installed in the system library along with Short Order.

Setting Defaults

The default settings of Short Order are designed to provide maximum compatibility with the Wang Sort utility. However, in two cases, we recommend different settings than the defaults.

To change defaults, run SORTCNFG in library @SYSTEM@ on your system IPL volume and select “Set installation defaults”.

Default options are described below. On-line help text is also available to assist you in making your selections.

What is the default value for MEMORY?

The default is AUTO. Default memory size is the amount of memory Short Order will use if the MEMORY parameter is not supplied in the OPTIONS getparm. Specify MAX to instruct Short Order to use the maximum available. Specify an integer of 32 or greater to fix the value at the number of kilobytes. We recommend the setting of AUTO, which instructs Short Order to automatically use the most efficient amount of memory.

Should AUTO memory size override any program specified value?

The default is NO to maintain compatibility with the Wang Sort. We recommend changing to YES. Specifying YES enables Short Order to use the AUTO memory size regardless of any numeric MEMORY specification made by a user or procedure. This enables Short Order to take advantage of larger memory sizes now available on most VS systems in spite of restrictive memory values that may be specified by old programs and procedures. Note that the MIN and MAX values may still be used to accommodate extreme cases.

Should bypass of invalid records be allowed?

The default is YES. Bypass specifies whether or not Short Order is to allow the operator to bypass invalid records that are encountered. Specifying YES allows bypass of invalid records and is compatible with Wang Sort. However, we feel that allowing bypass presents a real integrity problem to production data processing systems. We recommend a value of NO, which causes Short Order to notify the operator of an invalid record and terminate after acknowledgment.

How many workfiles should be used?

The default is 2. This is the number of workfiles Short Order will attempt to use during a sort operation. A greater number of workfiles does not usually make the sort any faster, but it does make work space easier to allocate for large sorts. We recommend increasing this value if you frequently have trouble finding sort work space. The minimum value is 2, the maximum is 5. The value specified here can be overridden for a specific sort using the FILES keyword of the OPTIONS getparm.

Should KEYES be required to match the number of keys specified?

The default is NO. Use this feature if you wish to have Short Order verify that the value of the KEYS keyword matches the number of key fields specified in the KEYS getparm. Wang Sort requires that the number of keys specified be equal to or greater than the value of the KEYS keyword. Some interactive users of sort have found that a frequent error is to specify several keys in the table but forget to increase the value of the KEYS keyword. Specifying YES will force a re-specification in the above situation, but may cause some older programs using CALLSORT to fail. We recommend NO to remain compatible with Wang Sort.

Should tape I/O errors be ignored?

The default is NO. Specifying YES allows the operator to bypass errors encountered on an input tape, which is useful in rare situations where losing a few records is not that important. For example, losing records may be acceptable in a mailing list application, but not in a financial or payroll application. We recommend a default of NO, which is compatible with Wang Sort.

3 GENERAL INFORMATION

This chapter provides information about Short Order's compatibility with, and differences from, the Wang Sort utility. It also provides and explanation of enhancements and performance characteristics.

Compatibility

Short Order is getparm and keyword compatible with the Wang Sort utility provided with the VS operating system. Procedures and programs that previously invoked the Wang utility should run without change when Short Order is substituted.

Some changes to the way errors are detected and handled during the program run have been made and should not have any impact on existing programs and procedures that use Wang Sort. Those chances are described in the next section.

The process of testing Short Order included many identical runs between it and the Wang Sort program. Some of the differences that were revealed were subsequently confirmed to be problems in the Wang Sort program. In those cases where the Wang program is definitely wrong, we do not try to be bug compatible.

Differences

For the purposes of this section, differences are defined as those functions that are performed differently from the Wang Sort utility as opposed to the additional features of Short Order.

Differences have been knowingly allowed only in handling exceptional conditions. In some cases, the architecture of Short Order requires that the options available be different, and in other cases our solution to the problem just seems to make more sense.

Invalid Numeric Keys – The Wang Sort halts when an invalid numeric key is encountered. The address of the record in memory is presented to the intrepid user who wants to jot down the particulars using the debugger and search the input file(s) for the record. Short Order validates numeric keys during input and will issue a getparm identifying the record number and file of each record that has an invalid key. The user has the option of omitting the record and continuing, or aborting the run.

Note that in addition to the different method of error presentation, there is a subtle difference in the possibility of *detection* of the error. When the invalid field is a minor key, the Wang Sort will complete if the more significant keys are unique. The invalid field will only be used, and thus detected, if *equal* higher significance keys are presented.

Short Order will always detect the invalid key.

Short Records – Both sort programs screen variable length input records to insure that they are long enough to contain all of the fields that are required to process the records.

When Short Order encounters a short record, it issues the RECORD getparm identifying the file and record number of the offending record. The operator may then cancel the program or, if not disallowed by an installation option, bypass the offending record and continue.

Wang Sort differentiates this problem into one of three categories which are listed below, along with the response options that are provided.

> When a record is too short to perform the record selection, a getparm called RECORD is issued which identifies the file containing the record. The users must select or reject the record. The address of the record in memory is provided, presumably to aid in making the decision.
>
> When the record is too short to contain one of the sort keys, a getparm called EOJ is issued which identifies the record by its memory address and requests that the user acknowledge the problem after which the program terminates.
>
> The third case occurs when the record is too short to contain one of the reformat source fields. No error is issued. The documentation does not say what is supposed to happen in this case, but the program seems to use hex 00 as the source field.

While the handling of these situations may seem arbitrary, our philosophy is that these problems are typically presented to an operator who has no idea how to examine a record in memory, much less decide if it should be selected, bypassed or whatever. Short Order provides an installation option, which removes the choices and forces program abort, removing the opportunity for operator error.

Invalid Numeric Selection Keys – The two programs handle this problem in basically the same way. A getparm called RECORD is issued to allow the user to select or reject the record manually. Short Order identifies the record number in addition to identifying the file. Wang Sort gives the main memory address of the record. Short Order provides a PF16 abort, which we believe, should be the only choice. Short Order provides an installation option, which removes the choices and forces program abort, removing the opportunity for operator error.

Underestimated Record Count – When tape input is specified, the number of records contained in the file must be supplied. Both sort programs use this number to allocate the resources necessary to perform the sort. If the specified number or records proves to be low, Wang Sort will stop and offer the option of continuing to read the files so as to determine the actual number of records, and the restart itself with the revised count.

Short Order will continue the sort without interruption. An additional workfile and associated tables are allocated to increase the sort capacity. As required, more workfiles are opened to handle severely underestimated sort files until either memory or open file count limitations make it impossible to continue. Each successive extension increases previous capacity by 25%. After the fifth additional workfile is opened, the compound effect is that the sort capacity is more than 300% of the original specification.

Return Codes – The return codes from Short Order are the same as those found in the Wang Sort documentation, where applicable. Where no corresponding value can be found, code 20 – “Unexpected Program Check” is used. We feel that using and expected value will cause fewer problems for linking programs and procedures than assigning new values. Another consideration is that Wang will probably assign new values in future versions that might conflict with our assignments. The return codes are explained in Chapter 5.

Enhancements

In addition to the greatly improved performance of Short Order over the Wang Sort utility, there are some added features that enhance functionality and/or make it easier to realize improved sorting performance. This section explains what those enhancements are.

No Input File Limit – The Wang Sort program restricts the number of sort input files to 10 and the number of merge files to 20. Short Order places no restrictions on the number of input files. The maximum number is determined by the maximum open file count allowed by the operating system and the availability of task GETHEAP space to save the file UFBs and related data.

New MEMORY Options – New responses have been added for the MEMORY keyword in the OPTIONS getparm. The Wang program requires an integer value. Short Order will accept an integer value but also accepts MIN, MAX, or AUTO. The valid responses are explained in the OPTIONS getparm section of Chapter 4.

Short Order is delivered with the MEMORY default set to AUTO, since this setting often provides the best performance. This default may be changed via the SORTCNFG program, as explained in Chapter 2. Individual Sort calls that request a specific MEMORY value may override the default.

Uppercase Fold Over – Wang introduced alternate collating sequences with the release 7 sort program. Short Order provides an additional preset table that causes lower-case alphabetic characters to collate as their upper-case equivalents. It is selected by specifying ALTSEQ=UPPER in the OPTIONS getparm.

Short Floating-Point Keys – Short Order allows the use of short (4 byte) floating point fields as sort keys.

COPY Function – Short Order provides a COPY function in addition to the standard SORT and MERGE functions. With COPY, any number of files may be copied to an output file using Short Order’s record selection and reformatting capabilities. The COPY function works on a record basis and is not as efficient as the COPY utility program, which works on a block basis. However, when value added processing such as record selection, reformatting or file concatenation is needed, Short Order will do the job very efficiently.

Y2K Compliant - Short Order version 3.0 is Y2K compliant, it has no Y2K-specific dependencies. It does not have any logical or mathematical inconsistencies when dealing with dates, including those beyond 1999.

Y2K Support - Many of the applications that use Short Order, and the data that Short Order manipulates, are Y2K sensitive. This sensitivity is due to the use of two digits to represent the year portion of dates. It is now easier to make applications Y2K compliant by the use of Short Orders' *sliding window* technique for processing dates. This 'sliding window' algorithm treats two digits years greater than 50 as dates in the 20th century, and years less than 50 as dates in the 21st century.

Short Order version 3.02 introduces a data type identifier '**Y**' (Y2K) to implement this new feature. This new data type behaves exactly as the data type '**C**' (character) except that, if the first character is a number, it is sorted as follows: 5, 6, 7, 8, 9, 0, 1, 2, 3, 4.

Performance Information

Short Order is a high performance sorting utility. It is designed to sort files as quickly and efficiently as possible. That translates into drastically reduced elapsed times with reduced utilization of all of the system's critical resources. This section provides information that can be used in estimating the improvements that will be realized in your environment.

These improvements can be expected for medium and large sorts that currently run from several minutes to several hours on your system.

Disk Workfile Requirements – Short Order's workfile requirements are about 110% to 115% of the *compressed* size of the file(s) to be sorted. Wang Sort is always about 200% of the *uncompressed* file sizes. That means you can sort more information with less disk workfile space using Short Order.

When record reformatting is used and some of the input record fields are not used in the output record, the sort size is reduced. If only half of the input record is included in the output record, Short Order will require half the workfile space (or sort internally). The Wang Sort program does not economize in this way.

CPU Utilization – Short Order typically uses 33% to 66% fewer CPU cycles than the Wang Sort. The range is in part a function of the significantly different performance characteristics of the various VS processors.

Disk I/O Operations – The average decrease in disk I/Os for external sorting is between 25% and 90%. In general, there is more improvement as the ratio between sort size and memory size increases.

Elapsed Time – Elapsed time improvement is the bottom line for most shops. Short Order continues to be improved with that as the primary goal. Average reductions between 40% and 80% can be expected for medium and large sorts.

Performance Tuning

The performance improvements described in the previous section can be expected by simply installing Short Order as provided and going about business as usual. However, there may be a few particularly long sorts that would be worth spending the extra time to fine tune. This section presents a few tips on how to tune such sorts.

Note that these techniques will not necessarily produce the same effect with the Wang Sort.

Use REFORMAT to Reduce the Size of the Sort – The time required to sort is directly proportional to the size of the sort in bytes. Reducing the size of the records will speed up the sort. When record reformatting is used with Short Order, any part of the input record that is not used in the output record or as a sort key is eliminated during input.

Of course, actually reformatting the sorted records will require changes to the input record definitions of the programs that use the sorted files. However, there is a way to use reformatting to reduce the size of the sort without changing any programs. Specify an output record length equal to the input record length but specify only those fields that will be used.

For example: Consider a file of 1000 byte records that consist of 200 bytes of customer data and 800 bytes of sales history. Only the customer data portion of the sorted file will be used for reports. Specify reformatting with 1000 byte records and a single 200-byte field corresponding to the customer data portion. The effect is that of clearing the sales history. The size or the resulting sort will be reduced by 80%, probably reducing the sort time by 60% or more.

Allocate the Workfiles – The number of workfiles as well as their location and amount of contention for the disks can have a significant effect on sort performance. Short Order, as provided, uses two workfiles. They are usually both allocated on the current WORKVOL of the task. You may override the default allocation by re-specifying the FILES keyword on the OPTIONS getparm. To change the default number of workfiles, see the section in Chapter 2 – Setting Defaults.

The workfiles should be placed on separate volumes when possible. Those volumes should have little or no other activity and, for best performance, should not contain the input or output files.

Increase Memory Size – There is a considerable advantage to using very large memory sizes for very large sorts. If your system has many megabytes of physical memory and you are running the VS/OS release 7 or above, try using several megabytes (2048 or more).

Specifying large memory sizes does not necessarily help for small sorts. In fact, it can have the effect of slightly increasing sort time. For large sorts it is significantly better. Experiment to find the best size.

Check Efficiency of COBOL Sorts – When sort is invoked from COBOL, the records are transferred to and from sort with intermediate files. Whenever the records are not processed during input, try a USING clause instead of an INPUT PROCEDURE clause. Similarly, when the sorted records are not processed, except to write them to a file, use a GIVING clause instead of an OUTPUT PROCEDURE clause.

Use the MERGE Function – When multiple input files are already sorted, merging is much faster than sorting them together. However, if the files have to be copied to setup for a merge, as with multiple tapes, then sorting together is probably the better choice.

Indexed File Output

With the introduction of the VS operating system 7.20, Wang introduced Sort enhancements designed to allow COBOL/85 compliance with ANSI standards. Short Order contains all of those enhancements, which include the ability to create indexed and relative output files.

Indexed file output is obtained by specifying FILEORG=I in the OUTPUT getparm. The primary key position and length are set to the position and length of the first sort key as specified in the KEYS getparm. If the second and subsequent sort keys are adjacent to and to the right of the first key, the indexed file primary key length will be extended to include those keys as well. An additional getparm called FILEKEY is issued to solicit alternate index information.

Several new return codes have been added to handle exceptional conditions that are unique to indexed file output. Return code 40 indicates that the primary key value is out of sequence. There are several possible causes for this condition:

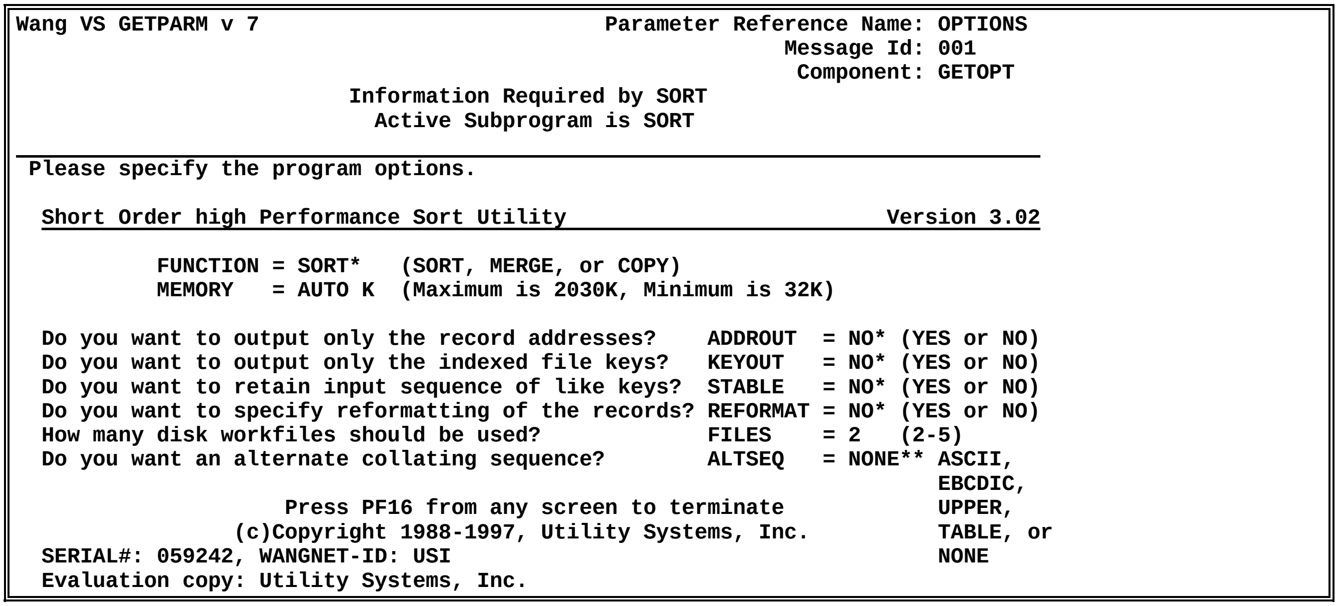
- The primary key is not unique and a duplicate primary has been encountered.
- The sort key type is numeric and does not collate in the same sequence as DMS indexed file keys.
- The sort key type is C and an alternate collating sequence is used.

Indexed file output is not allowed when the ADDROUT, KEYOUT or REFORMAT keywords are set to YES in the OPTIONS getparm.

Indexed files are loaded using standard DMS record output. For large files or files with alternate indexes, there are more expedient ways to create indexed files. The OverEasy utility, also from USI, can create indexed and indexed+ files 10 to 15 times faster than DMS. It is usually much faster to create a consecutive file with Sort and to then load the indexed file with OverEasy.

4 GETPARM REFERENCE

This chapter describes the Short Order getparm requests and associated keywords. A summarization table is provided in Chapter 6 for quick reference.



```
Wang VS GETPARM v 7                           Parameter Reference Name: OPTIONS
                                                            Message Id: 001
                                                             Component: GETOPT
                          Information Required by SORT
                            Active Subprogram is SORT

 Please specify the program options.

  Short Order high Performance Sort Utility                        Version 3.02

           FUNCTION = SORT*   (SORT, MERGE, or COPY)
           MEMORY   = AUTO K  (Maximum is 2030K, Minimum is 32K)

  Do you want to output only the record addresses?    ADDROUT  = NO* (YES or NO)
  Do you want to output only the indexed file keys?   KEYOUT   = NO* (YES or NO)
  Do you want to retain input sequence of like keys?  STABLE   = NO* (YES or NO)
  Do you want to specify reformatting of the records? REFORMAT = NO* (YES or NO)
  How many disk workfiles should be used?             FILES    = 2   (2-5)
  Do you want an alternate collating sequence?        ALTSEQ   = NONE** ASCII,
                                                                        EBCDIC,
                    Press PF16 from any screen to terminate             UPPER,
                (c)Copyright 1988-1997, Utility Systems, Inc.           TABLE, or
  SERIAL#: 059242, WANGNET-ID: USI                                      NONE
  Evaluation copy: Utility Systems, Inc.
```

OPTIONS Getparm

This is the first getparm issued by Short Order. Use it to specify the following parameters.

FUNCTION Specify the type of sequencing operation you want to perform. Valid responses are SORT, MERGE, and COPY. The default is SORT.

Sort combines one or more randomly sequenced files into a single ordered output file.

MERGE combines one or more previously sequenced files into a single ordered output file.

COPY combines one or more files into a single output file. The input files are processed serially and the records are written in the same sequence that they are read. This function may be used to simply combine several files into a larger file or, when used in conjunction with the SELECT and/or REFORMAT options, to produce subsets of files and restructure the record contents.

MEMORY The amount of memory used for a sort or merge operation is one of the most important performance parameters. Valid responses are an integer value within the program-specified range, MIN, MAX, or AUTO. The default is AUTO.

Integer Value – The number of K (1024) bytes or segment-2 memory used during the run. The valid range for a particular run will have been determined by the program and displayed as part of the OPTION getparm.

MIN – Use the minimum amount of memory available. Equivalent to specifying 32K.

MAX – Use the maximum amount of memory available. The maximum is computed by the program and is a function of the amount of physical memory available as well as the amount of virtual memory available to the program.

AUTO – A program computed value, which attempts to assign a large value without causing an overall system memory shortage. The recommended value for MEMORY is AUTO.

ADDROUT Use to specify that the output file is to be a file of 3-byte records that contain the binary relative record numbers of the records in the required sorted sequence. ADDROUT files are typically used as input to RPG II programs. Valid responses are YES and NO. The default is NO.

ADDROUT is mutually exclusive with the KEYOUT and REFORMAT keywords. It may only be specified for SORT operations and only one input file is permitted.

KEYOUT Used to specify that the output file is to be a file containing the primary keys of the indexed file input file. Valid responses are YES and NO. The default is NO.

KEYOUT is mutually exclusive with the ADDROUT and REFORMAT keywords. It may only be specified for SORT operations. The single input file must be an indexed file.

STABLE Use to specify that records with equal keys be output in the same order that they were input. Valid responses are YES and NO. When STABLE=NO is selected, record with equal keys will be output in random sequence. The default is NO.

REFORMAT Specify to reformat the records during the sort. Valid responses are YES and NO. With REFORMAT=YES, subsequent getparms will allow specification of the output record length and format on a field by field basis. The default is NO.

Reformatting may be used to produce shorter or longer output records. It may also be used to significantly improve sort performance without changing other programs. Consult “Performance Considerations” in Chapter 3 for specific ways to improve sort performance using reformat.

FILES Use to specify the number of workfiles used during the sort. Valid values are 2 through 5. The default value may be reset by running the SORTCNFG utility.

Specifying more workfiles does not usually make the sort faster, but it can make workfile space easier to allocate for larger sorts. Increase the number of workfiles if you have trouble find sort workfile space for large sorts.

ALTSEQ This option allows for specification of an alternate collating sequence for character-type keys. This option affects the way that character keys are compared but does not alter the contents of the records in any way. Valid responses are NONE, ASCII, EBCDIC, UPPER, and TABLE. The default is NONE. Selection of any option other than NONE causes the TABLE getparm to be issued for confirmation and/or modification of the alternate collating sequence table.

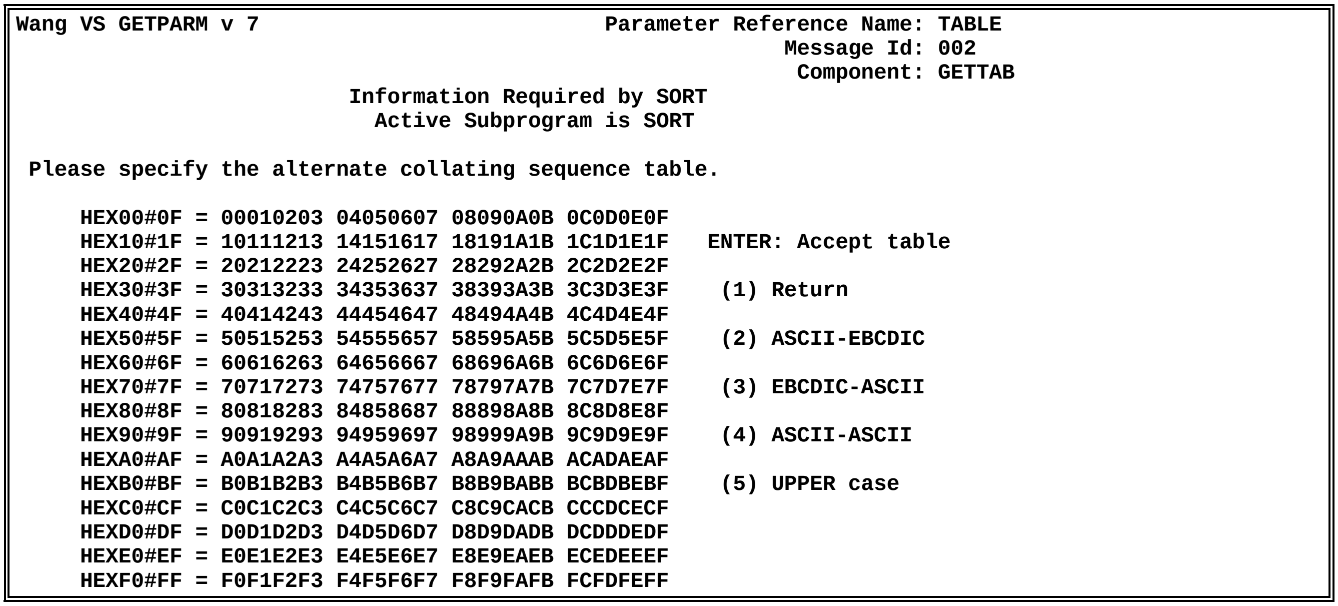
NONE – Specifies that no alternate collating sequence is to be used.

TABLE – Selects and ASCII to ASCII table for subsequent modification via the TABLE getparm.

ASCII – Selects an EBCDIC to ASCII table

EBCDIC – Selects and ASCII to EBCDIC table.

UPPER – Selects a table, which collates lower case alphabetic characters as their uppercase equivalents. This is a particularly useful option when sorting names. For example, it will cause MacDonald to collate after Macaris.



```
Wang VS GETPARM v 7                          Parameter Reference Name: TABLE
                                                           Message Id: 002
                                                            Component: GETTAB
                          Information Required by SORT
                            Active Subprogram is SORT

 Please specify the alternate collating sequence table.

     HEX00#0F = 00010203 04050607 08090A0B 0C0D0E0F
     HEX10#1F = 10111213 14151617 18191A1B 1C1D1E1F   ENTER: Accept table
     HEX20#2F = 20212223 24252627 28292A2B 2C2D2E2F
     HEX30#3F = 30313233 34353637 38393A3B 3C3D3E3F    (1) Return
     HEX40#4F = 40414243 44454647 48494A4B 4C4D4E4F
     HEX50#5F = 50515253 54555657 58595A5B 5C5D5E5F    (2) ASCII-EBCDIC
     HEX60#6F = 60616263 64656667 68696A6B 6C6D6E6F
     HEX70#7F = 70717273 74757677 78797A7B 7C7D7E7F    (3) EBCDIC-ASCII
     HEX80#8F = 80818283 84858687 88898A8B 8C8D8E8F
     HEX90#9F = 90919293 94959697 98999A9B 9C9D9E9F    (4) ASCII-ASCII
     HEXA0#AF = A0A1A2A3 A4A5A6A7 A8A9AAAB ACADAEAF
     HEXB0#BF = B0B1B2B3 B4B5B6B7 B8B9BABB BCBDBEBF    (5) UPPER case
     HEXC0#CF = C0C1C2C3 C4C5C6C7 C8C9CACB CCCDCECF
     HEXD0#DF = D0D1D2D3 D4D5D6D7 D8D9DADB DCDDDEDF
     HEXE0#EF = E0E1E2E3 E4E5E6E7 E8E9EAEB ECEDEEEF
     HEXF0#FF = F0F1F2F3 F4F5F6F7 F8F9FAFB FCFDFEFF
```

TABLE Getparm

The TABLE getparm is presented when an ALTSEQ choice other than NONE is specified in the OPTIONS getparm. Use of an alternate collating sequence will in no way affect the contents of any records or reduce the capacity of the sort program.

A table of 256 hexadecimal codes is presented which represents the alternate collating values of the 256 possible character values of each byte in a character key. The table may be accepted as is, changed to a different standard table, modified byte by byte for a custom alternate collating sequence, or rejected by returning to the OPTIONS getparm via PF1.

The table will contain the appropriate values for the alternate collating sequence, which was specified in the ALTSEQ parameter of the OPTIONS getparm. The table can be accepted as displayed by responding with the ENTER key. PF keys 2 through 5 may be used to change the table to other default values as follows:

PF2 ASCII to EBCDIC (collate character-data like an IBM 370)

PF3 EBCDIC to ASCII (collate IBM 370 character data in ASCII sequence)

PF4 ASCII –ASCII (standard sequence table for subsequent modification)

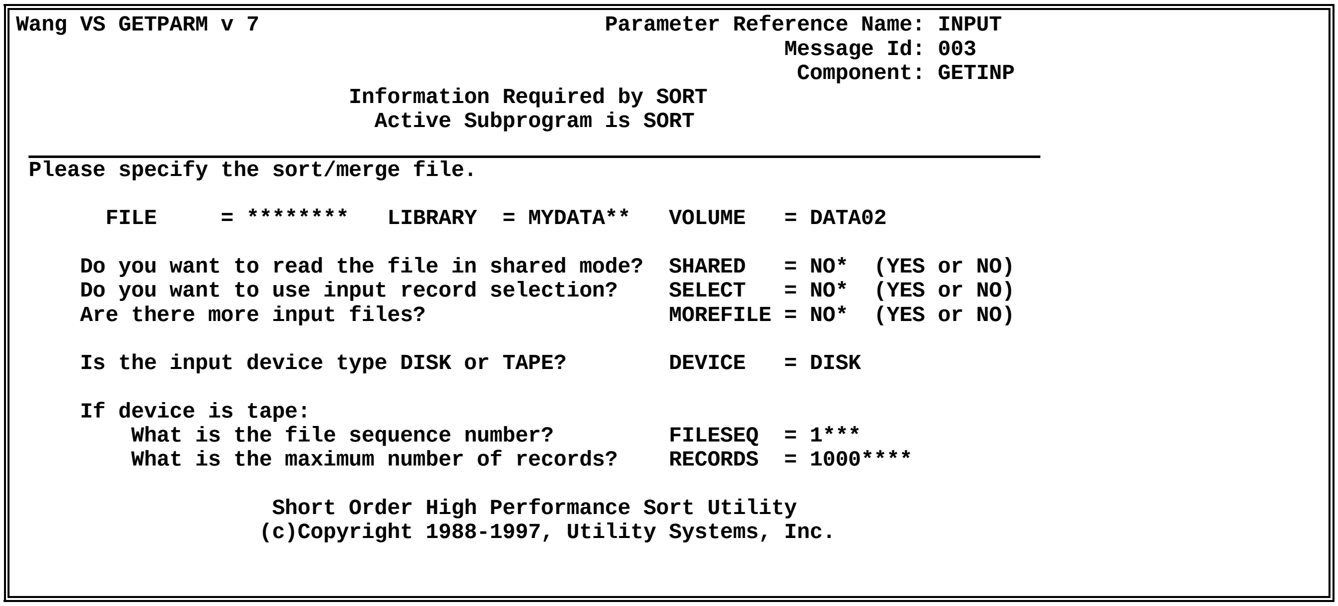
PF5 UPPER case (collates lower-case alphabetics as their upper-case equivalents)

The table contains 256 hexadecimal values, which represent the alternate collating values of the 256 possible characters in each position of a character field. Individual bytes in the table may be changed. For example, if a space is to collate greater than all other characters, a value of X’FF’ might be inserted into table position X’20’ (the ASCII value for a space).

If Short Order is being run as an interactive task, the table modifications can be made directly on the screen. However, for production-invoked runs, the getparm KEYWORDS should be used to modify the table.

There are 16 keywords, each of which are used to specify 16 alternate collating characters. The keyword names contain the range of 16 ASCII character values represented by that table entry. The value specified for keyword HEX40#4F represents the alternate collating values for the ASCII characters with the values from X’40’to X’4F’. The keyword values are specified as four substrings of eight hexadecimal digits separated by a space. To enter the value from a procedure, the entire 35-character value must be specified. For the above example, changing space (ASCII X’20’) to collate highest (X’FF’), the procedure statement would be as follows:

```
ENTER TABLE HEX20#2F=“FF212223 24252627 28292A2B 2C2D2E2F”
```



```
Wang VS GETPARM v 7                           Parameter Reference Name: INPUT
                                                            Message Id: 003
                                                             Component: GETINP
                          Information Required by SORT
                            Active Subprogram is SORT
 ______________________________________________________________________________
 Please specify the sort/merge file.

       FILE     = ********   LIBRARY  = MYDATA**   VOLUME   = DATA02

     Do you want to read the file in shared mode?  SHARED   = NO*  (YES or NO)
     Do you want to use input record selection?    SELECT   = NO*  (YES or NO)
     Are there more input files?                   MOREFILE = NO*  (YES or NO)

     Is the input device type DISK or TAPE?        DEVICE   = DISK

     If device is tape:
         What is the file sequence number?         FILESEQ  = 1***
         What is the maximum number of records?    RECORDS  = 1000****

                    Short Order High Performance Sort Utility
                   (c)Copyright 1988-1997, Utility Systems, Inc.
```

INPUT Getparm

The INPUT getparm is used to specify the files to be used as input to the sort, merge or copy operation. Various input file processing options such as shared file access and record selection are also requested through the INPUT getparm.

Input files may be located on disk or tape volumes. For disk files, FILE, LIBRARY and VOLUME must be specified and the program ignores FILESEQ and RECORDS. When tape files are specified, FILE and LIBRARY are optional, VOLUME is required, and the FILESEQ and RECORDS keywords must be specified.

Disk files may be processed in shared input mode. Shared mode allows other programs to access and update the file while it is being read by Short Order. When SHARED=YES is specified, the LOCK getparm is issued to allow selection of various additional file sharing options.

When KEYOUT or ADDROUT is specified in the OPTIONS getparm for sort, only one input file is permitted and the MOREFILE keyword is not presented.

MOREFILE=YES is assumed for merger operations.

The keywords are described individually below:

FILE Specify the name of the input file. This parameter is required for disk files and optional for tape files.

LIBRARY Specify the name of the input library. This parameter is required for disk files and optional for tape files. Library defaults to the user specified INLIB.

VOLUME Specify the name of the disk or tape input volume. This parameter is required. Default value is the user specified INVOL.

SHARED Specify if the input file is to be processed in shared mode. Valid responses are YES and NO. The default is NO.

Shared input mode allows other programs to update the file while it is being read by Short Order. Selection of shared mode usually increases input processing time and should be carefully considered.

When SHARED=YES is specified for a file, the LOCK getparm is immediately issued to allow selection of various shared file processing options.

SELECT Specify if record selection criteria is to be applied to the input records. Valid responses are YES and NO. The default is NO.

When SELECT=YES is specified, a subsequent SELECT getparm is issued to solicit the actual record selection criteria.

The SELECT keyword is only presented on the first INPUT getparm and the response applies equally to all input files, i.e. either no record selection is performed on any file (NO) or the same record selection criteria is applied to all records of all files (YES).

MOREFILE Specifies that additional files are to be included in a sort or copy operation. Valid responses are YES and NO. The default is NO.

This keyword is not presented for merge operations and is assumed to be YES. It is also not presented for the ADDROUT or KEYOUT sorts and is assumed to be NO.

When MOREFILE=YES is specified or implied, additional INPUT getparms are issued allowing specification of more input files. End of input file specification is then indicated by responding ENTER to an INPUT getparm with DEVICE=DISK and no FILE (the default values).

Sorts and copies will process the input files in the order specified via the INPUT getparms. Merges will process all the files concurrently but will resolve duplicate key situations between files by considering the order in which the files were specified. For example, when equal keys are encountered on two or more input files, all records with that key value from the first file will precede all records with that key from the second file, which will precede all records with that key from the third file, etc.

DEVICE Specify the type of device for the input file. Valid responses are DISK and TAPE. The default is DISK.

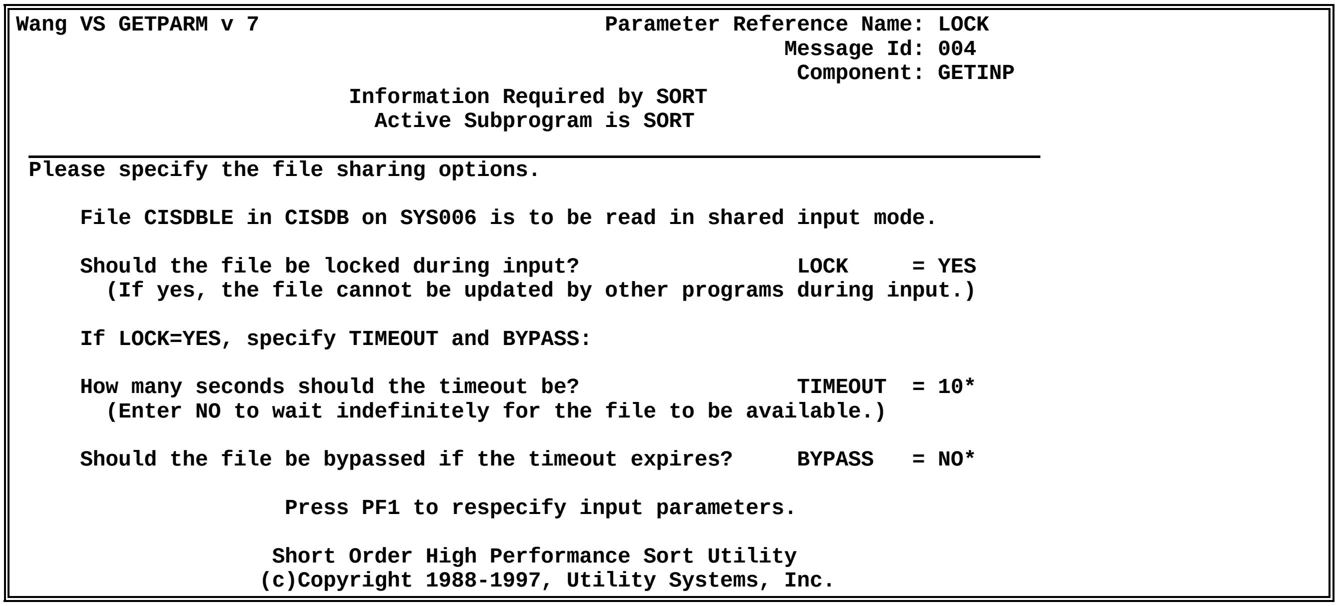
When TAPE is specified, the FILESEQ and RECORDS keywords must be specified.

Selection of the ADDROUT or KEYOUT sort options implies disk input, therefore the DEVICE keyword is not presented.

FILESEQ Specify the tape file sequence number. Non-zero positive integers are valid. The default is one. This parameter is not required when DEVICE=DISK.

RECORDS Specify the maximum number of input records in the tape file. Non-zero positive integers are valid. The default is 1000. This parameter is not required when DEVICE=DISK.

This record estimation should be as accurate as possible. Moderate underestimation is acceptable. Gross underestimation could cause the sort to fail.



```
Wang VS GETPARM v 7                              Parameter Reference Name: LOCK
                                                              Message Id: 004
                                                               Component: GETINP
                          Information Required by SORT
                            Active Subprogram is SORT
 ________________________________________________________________________________
 Please specify the file sharing options.

     File CISDBLE in CISDB on SYS006 is to be read in shared input mode.

     Should the file be locked during input?                LOCK     = YES
       (If yes, the file cannot be updated by other programs during input.)

     If LOCK=YES, specify TIMEOUT and BYPASS:

     How many seconds should the timeout be?                TIMEOUT  = 10*
       (Enter NO to wait indefinitely for the file to be available.)

     Should the file be bypassed if the timeout expires?     BYPASS   = NO*

                     Press PF1 to respecify input parameters.

                    Short Order High Performance Sort Utility
                  (c)Copyright 1988-1997, Utility Systems, Inc.
```

LOCK Getparm

LOCK will be requested immediately after each INPUT getparm that specifies SHARED=YES. It solicits specific file-sharing options to be used in processing the file. The specified options are used in processing the most recently specified input file only and may be different for each file that is specified as input to the program.

The file may be locked open in shared input mode until all of the records have been read. When locking is requested, a timeout interval may be specified to return control in the event that the file does not become available in the specified time.

An escape is provided, via PF1, which returns to the previously specified INPUT getparm for re-specification.

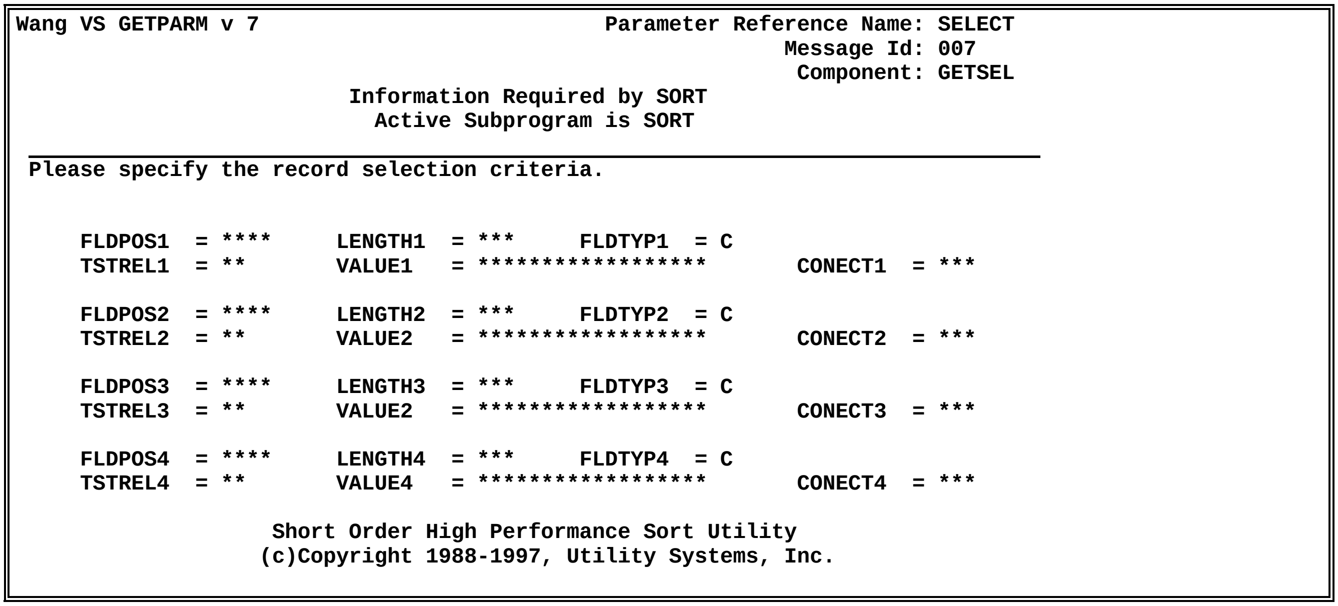
The keywords are described below:

LOCK Specify if the file is to be locked for exclusive use of this program until it has been completely processed. Locking prohibits other programs from updating records in the file. Valid responses are YES and NO. The default is YES.

TIMEOUT Specify NO to wait indefinitely for the file to become available for locked processing. Specify an integer value between 0 and 255 to return to the LOCK getparm if the file is not available after that number of seconds.

Valid responses are NO and the integers between 0 and 255.

BYPASS Used to request that the program skip this file rather than return control to the LOCK getparm when the TIMEOUT interval expires. Valid responses are YES and NO. The default is NO.



```
Wang VS GETPARM v 7                         Parameter Reference Name: SELECT
                                                          Message Id: 007
                                                           Component: GETSEL
                          Information Required by SORT
                            Active Subprogram is SORT
 ______________________________________________________________________________
 Please specify the record selection criteria.


     FLDPOS1  = ****     LENGTH1  = ***     FLDTYP1  = C
     TSTREL1  = **       VALUE1   = ******************       CONECT1  = ***

     FLDPOS2  = ****     LENGTH2  = ***     FLDTYP2  = C
     TSTREL2  = **       VALUE2   = ******************       CONECT2  = ***

     FLDPOS3  = ****     LENGTH3  = ***     FLDTYP3  = C
     TSTREL3  = **       VALUE2   = ******************       CONECT3  = ***

     FLDPOS4  = ****     LENGTH4  = ***     FLDTYP4  = C
     TSTREL4  = **       VALUE4   = ******************       CONECT4  = ***

                    Short Order High Performance Sort Utility
                   (c)Copyright 1988-1997, Utility Systems, Inc.
```

SELECT Getparm (and SELECT2 – SELECT8)

Record selection is requested by specifying SELECT=YES in the first input getparm. This SELECT getparm is then issued to request specification of the criteria that is to be used to select records during input. Records that do not satisfy the selection criteria are rejected at input time

Up to 32 individual tests may be specified, the results of which are evaluated using logical connectives to determine overall acceptance or rejection of the record.

Each individual test allows comparison of an input record field to a constant or another field in the same input record. The input record field is described by its position and length within the input record and by its data type. It is compared to a specified constant or to another field in the input record which is the same length and data type. The results of the comparison are evaluated based on satisfaction of the specified relational operator (EQ, NE, GT, etc.).

In evaluating the selection criteria, the results of each individual test are combined using the connective operators specified. AND operators take precedence over OR operations. That is to say that all individual tests linked by AND connectives in any one group of tests linked by OR connectives must be true to select the record.

The end of the selection program is determined by a blank CONNECTn field. Each getparm allows specification of up to four individual tests. Additional getparms, named SELECT2 through SELECT8, are issued when necessary to allow specification of up to 32 individual tests.

Each individual test is specified by five keywords. All but the 32nd test also has a connective keyword. The keywords in each test set are identical except for a test group number appended to make each one unique.

The program will apply the selection tests serially beginning with the first. It will skip ahead to the next test following an OR connective when a false condition is encountered, and it will stop as soon as the record can be selected.

If an invalid numeric field is encountered during selection processing, a getparm will be issued identifying the file and record number and presenting the options of accepting the record, rejecting the record, or terminating the program. A similar getparm is presented if a record is too short to contain all of the specified selection fields.

FLDPOSn Starting position of the selection field in the input record relative to one. Valid responses are non-zero positive integers less than or equal to the record size.

LENGTHn Size of the selection field in bytes. This length combined with FLDPOS cannot specify a field which exceeds the record length. Also, the valid length values depend on the field type. There is no default value. The following table gives the allowable length values for the various permissible FLDTYP values.

FLDTYP		LENGTH	
C	character	1 to 16 or 1 to 256[1]	
B	binary	2 or 4	
P	packed decimal	1 to 16	
D	external decimal	1 to 16	
L	zoned decimal	1 to 16	(leading sign)
Z	zoned decimal	1 to 16	(trailing sign)
Y	Y2K character[2]	1 to 16 or 1 to 256[1]	

[1] The maximum length for character fields is 16 when a constant comparator is specified. When comparing to another field in the input record, the maximum is 256.

[2] The Y2K data type behaves exactly as the data type C (character) except that, if the first character is a number, it is sorted as follows: 5, 6, 7, 8, 9, 0, 1, 2, 3, 4.

FLDTYPn Specify the field data type. Refer to the above table for allowable values. The default is C.

TSTRELn Specify the acceptable relation for this test. There is no default. Valid responses are:

EQ	Field values must be equal
NE	Field values must be not-equal
GT	Field must be greater than specified value
LT	Field must be less than specified value
GE	Field must be greater than or equal to specified value
LE	Field must be less than or equal to specified value

VALUEn Specify the value to be compared with the above field. Value may be specified as a constant enclosed in single (‘’) or double (“”) quotation marks or as a starting position in the record of a comparison field of the same length and type specified for the primary field.

When specifying a constant value, be sure to use quotation marks to signify that the value is a constant. An unquoted integer value will be interpreted as the starting position of a comparison field in the input record. When specifying a constant using Procedure language, and additional set of quotes must be used because the procedure interpreter strips away the outer set of quotes. For example, to specify the numeric value –120 in a procedure, the keyword would have to be specified as:

VALUE1=”’-120’”, …

the net result to the program would be a specification of:

VALUE1 = ‘-120’

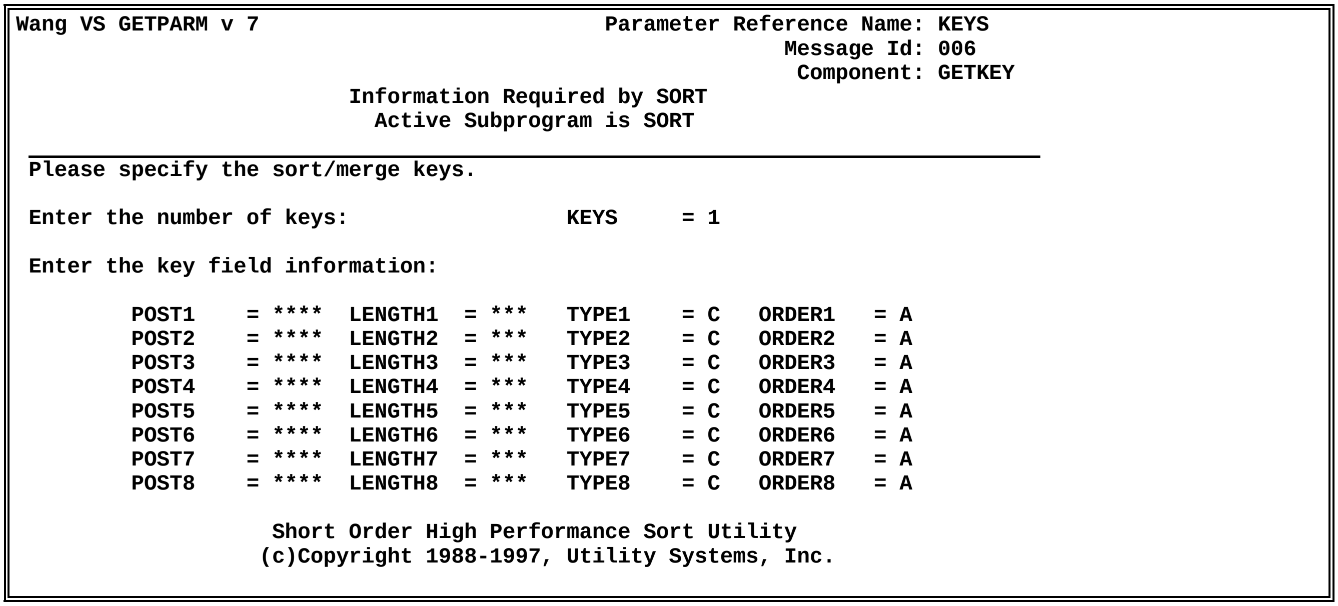
The same would apply to a character constant. Specify:

VALUE2=”’MA’”, … to obtain: VALUE2=’MA’

The field size is 18 characters which, allowing for the quotation marks, provides for up to 16 digits in a numeric constant. However, the program will not allow numeric constants that are greater than the maximum possible value for the primary field.

CONNECTn Specify a connective operator if more selection criteria are to be provided. Leaving this keyword blank signifies the end of the selection program.

The AND connective operator continues the current selection criteria. The OR connective operator signifies the start of a new alternative selection criteria.



```
Wang VS GETPARM v 7                            Parameter Reference Name: KEYS
                                                              Message Id: 006
                                                               Component: GETKEY
                          Information Required by SORT
                            Active Subprogram is SORT
 ________________________________________________________________________________
 Please specify the sort/merge keys.

 Enter the number of keys:                KEYS     = 1

 Enter the key field information:

         POST1    = ****  LENGTH1  = ***   TYPE1    = C   ORDER1   = A
         POST2    = ****  LENGTH2  = ***   TYPE2    = C   ORDER2   = A
         POST3    = ****  LENGTH3  = ***   TYPE3    = C   ORDER3   = A
         POST4    = ****  LENGTH4  = ***   TYPE4    = C   ORDER4   = A
         POST5    = ****  LENGTH5  = ***   TYPE5    = C   ORDER5   = A
         POST6    = ****  LENGTH6  = ***   TYPE6    = C   ORDER6   = A
         POST7    = ****  LENGTH7  = ***   TYPE7    = C   ORDER7   = A
         POST8    = ****  LENGTH8  = ***   TYPE8    = C   ORDER8   = A

                    Short Order High Performance Sort Utility
                   (c)Copyright 1988-1997, Utility Systems, Inc.
```

KEYS Getparm

The KEYS getparm is used to specify the fields that will be used by Short Order to sequence the records. A key field is identified by its position and length within the input records. The data type and collating order (ascending or descending) are also specified for each key field.

Up to eight key fields may be specified. The first key field is the most significant, the eighth is the least significant. In other words, the second key field is examined to determine the sequence of records whose first key fields are equal, the third those whose first and second key fields are equal, etc.

Key fields may be character of numeric fields.

Character fields are compared byte by byte as unsigned binary quantities. When an alternate collating sequence has been specified (via the keyword in the OPTIONS getparm), the translated values of character fields are compared. The only difference between C (character) and U (unsigned binary) fields is that character fields are translated when an alternate collating sequence is specified and unsigned binary fields are not. Therefore, when using an alternate collating sequence for a particular sort, specify U for the character fields to be compared without translation.

Numeric fields may be specified as any of a variety of data types.

Each key field may be specified as collating in either ascending or descending sequence.

Special note must be made of an aberration in the implementation of 8-byte binary numeric keys. Wang SORT has defined this operation as a comparison of two consecutive four byte signed binary fields. For a variety of reasons, we feel that this was a shortsighted change on Wang's part but have conformed to their standard.

The allowable length of a key field depends on the data type specified. The following table gives the allowable data types and the lengths which are permissible with each.

FLDTYP		LENGTH	
C	character	1 to 256	
U	unsigned binary	1 to 256	
B	binary numeric	2 or 4	
F	floating point	4 or 8	
P	packed decimal	1 to 16	
D	external decimal	1 to 16	
L	zoned decimal	1 to 16	(leading sign)
Z	zoned decimal	1 to 16	(trailing sign)
Y	Y2K character[1]	1 to 256	

[1] The Y2K data type behaves exactly as the data type C (character) except that, if the first character is a number, it is sorted as follows: 5, 6, 7, 8, 9, 0, 1, 2, 3, 4.

As there are no collating during a COPY operation, this getparm is not issued.

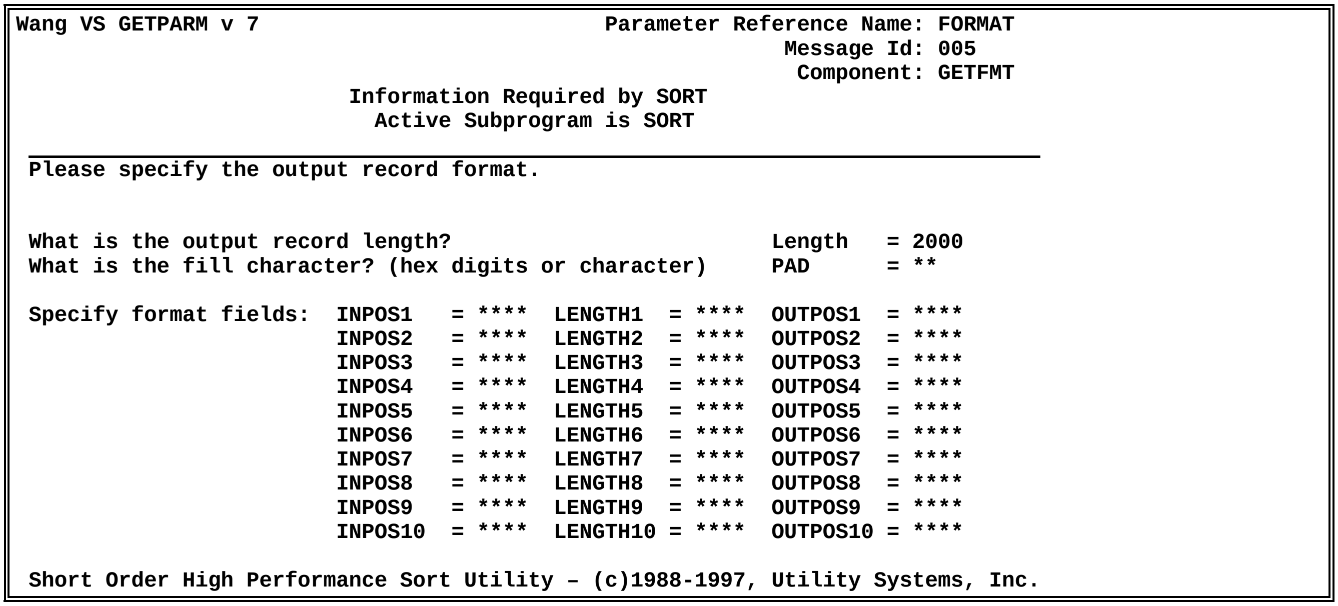
KEYS Specify the number of key fields that are to be used in sequencing the files. Valid responses are integers from 1 to 8. When the installation option "Should KEYS be required to match the number of keys specified" is YES (see Chapter 2, Setting Defaults), this value must agree with the number of key fields as specified below.

POSTn Specify the position of this key field in the input record, relative to one. There is no default.

LENGTHn Specify the length in bytes of the key field.

TYPEn Specify the data type. Refer to the table above for permissible data types. The default is C.

ORDERn Specify whether the records should be sequenced in ascending or descending order within this key field. The default is A (ascending).



```
Wang VS GETPARM v 7                          Parameter Reference Name: FORMAT
                                                           Message Id: 005
                                                            Component: GETFMT
                          Information Required by SORT
                            Active Subprogram is SORT
 ________________________________________________________________________________
 Please specify the output record format.


 What is the output record length?                        Length   = 2000
 What is the fill character? (hex digits or character)     PAD      = **

 Specify format fields:  INPOS1   = ****  LENGTH1  = ****  OUTPOS1  = ****
                         INPOS2   = ****  LENGTH2  = ****  OUTPOS2  = ****
                         INPOS3   = ****  LENGTH3  = ****  OUTPOS3  = ****
                         INPOS4   = ****  LENGTH4  = ****  OUTPOS4  = ****
                         INPOS5   = ****  LENGTH5  = ****  OUTPOS5  = ****
                         INPOS6   = ****  LENGTH6  = ****  OUTPOS6  = ****
                         INPOS7   = ****  LENGTH7  = ****  OUTPOS7  = ****
                         INPOS8   = ****  LENGTH8  = ****  OUTPOS8  = ****
                         INPOS9   = ****  LENGTH9  = ****  OUTPOS9  = ****
                         INPOS10  = ****  LENGTH10 = ****  OUTPOS10 = ****

 Short Order High Performance Sort Utility – (c)1988-1997, Utility Systems, Inc.
```

FORMAT Getparm

Record formatting is selected by specifying REFORMAT=YES in the OPTIONS getparm. This FORMAT getparm is subsequently issued to allow specification of the new record format.

Reformatted output records will be fixed length. Their length is specified in the FORMAT getparm. A pad character is specified that will be used to fill undefined areas of the output record.

The new record format is described by defining the input record position, field length, and output record position of the various fields that are to be included in the output record.

The FORMAT getparm allows for specification of up to ten such fields. When the tenth entry is used, another FORMAT getparm is issued to allow specification of more fields. Additional FORMAT getparms will continue to be issued until the tenth entry is left blank.

LENGTH Specify the length of the output records. Length may be any integer between 1 and 2048. The default value is the input record length.

LENGTH may only be specified in the first occurrence of FORMAT. In subsequent occurrences, the length is displayed but is not modifiable.

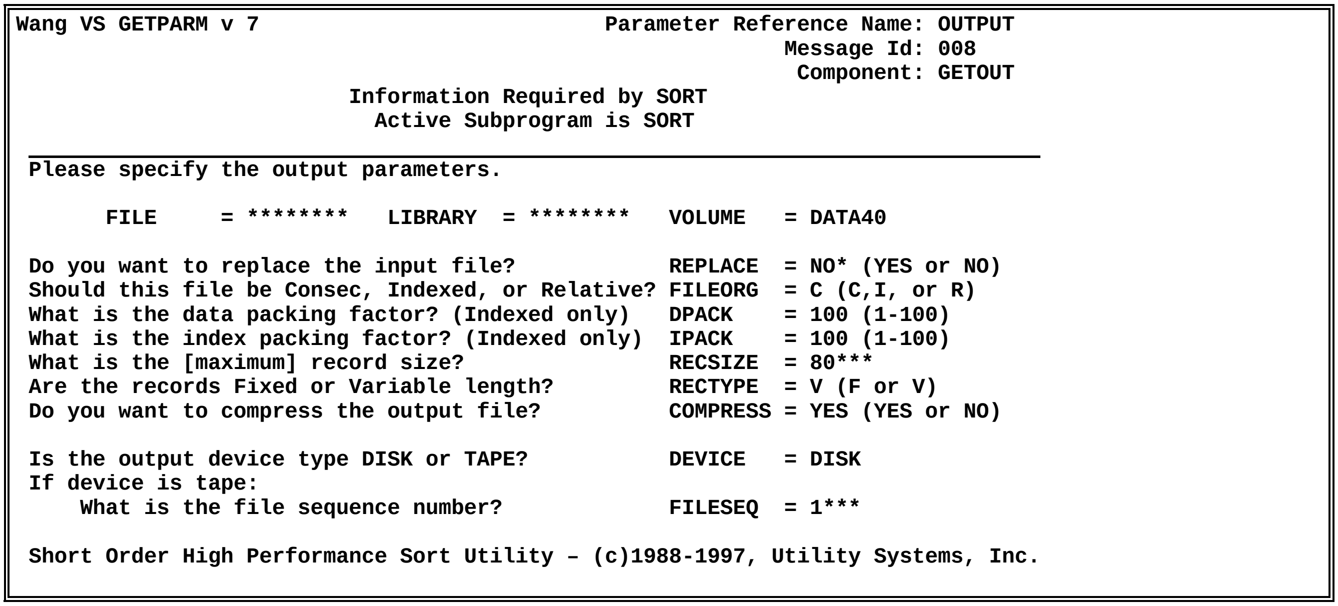
PAD Specify the pad character to be used to fill undefined areas of the output record. When two characters are specified, they are interpreted as a hexadecimal value. When one character is specified, it is assumed to be an ASCII character. The default value is the space character which is hex 20.

PAD may only be specified in the first occurrence of FORMAT. In subsequent occurrences, PAD is displayed but in not modifiable.

INPOSn Specify the input record position of a field. Valid responses are non-zero positive integers no greater than the input record length. There is no default.

LENGTHn Specify the field length. Valid responses are non-zero positive integers which, when combined with INPOS, do not exceed the input record length. There is no default.

OUTPOSn Specify the output record position for the field. Output fields are not allowed to overlap. Valid responses are non-zero positive integers which, when combined with LENGTH, do not overlap previously defined fields or extend beyond the output record length. There is no default.



```
Wang VS GETPARM v 7                          Parameter Reference Name: OUTPUT
                                                           Message Id: 008
                                                            Component: GETOUT
                          Information Required by SORT
                            Active Subprogram is SORT
 ______________________________________________________________________________
 Please specify the output parameters.

       FILE     = ********   LIBRARY  = ********   VOLUME   = DATA40

 Do you want to replace the input file?            REPLACE  = NO* (YES or NO)
 Should this file be Consec, Indexed, or Relative? FILEORG  = C (C,I, or R)
 What is the data packing factor? (Indexed only)   DPACK    = 100 (1-100)
 What is the index packing factor? (Indexed only)  IPACK    = 100 (1-100)
 What is the [maximum] record size?                RECSIZE  = 80***
 Are the records Fixed or Variable length?         RECTYPE  = V (F or V)
 Do you want to compress the output file?          COMPRESS = YES (YES or NO)

 Is the output device type DISK or TAPE?           DEVICE   = DISK
 If device is tape:
     What is the file sequence number?             FILESEQ  = 1***

 Short Order High Performance Sort Utility – (c)1988-1997, Utility Systems, Inc.
```

OUTPUT Getparm

The OUTPUT getparm describes the file to which the sorted or merged records are to be written. The file may reside on disk or tape.

When sorting, the OUTPUT getparm is not issued until all of the input files have been processed and the records are prepared for final output. When merging or copying, the OUTPUT getparm is issued during initialization, as the last getparm before processing begins.

A replace option is available which causes Short Order to scratch the input file before opening an output file of the same name.

FILE Specify the name of the output file. Required when DEVICE=DISK, optional when DEVICE=TAPE. There is no default.

LIBRARY Specify the name of the output library. Required when DEVICE=DISK, optional when DEVICE=TAPE. The default is that task specified OUTLIB.

VOLUME Specify the name of the tape or disk output volume. This keyword is always required. The default is the task specified OUTVOL.

REPLACE Specify to scratch the input file with the same name prior to opening the output file. YES may only be specified when DEVICE=DISK. The specified file, library, and volume must match the input file, library and volume. Valid responses are YES and NO. The default is NO.

The REPLACE keyword is only presented for sort operations. It will not be presented in cases where ADDROUT or KEYOUT was specified, more than one input file is used, the input file is indexed, or the input file is processed in shared mode.

FILEORG Specify the disk file organization. Valid responses are C (consecutive), I (indexed), and R (relative). The default is C.

When indexed is specified, the primary key will be the first sort key field. The indexed option is not allowed if REFORMAT=YES, KEYOUT=YES, or ADDROUT=YES. For more information about indexed output, see the discussion titled “Indexed File Output” in Chapter 3.

DPACK For FILEORG=I only, specify the data block packing percentage. The default value is 100.

IPACK For FILEORG=I only, specify the index block packing percentage. The default value is 100.

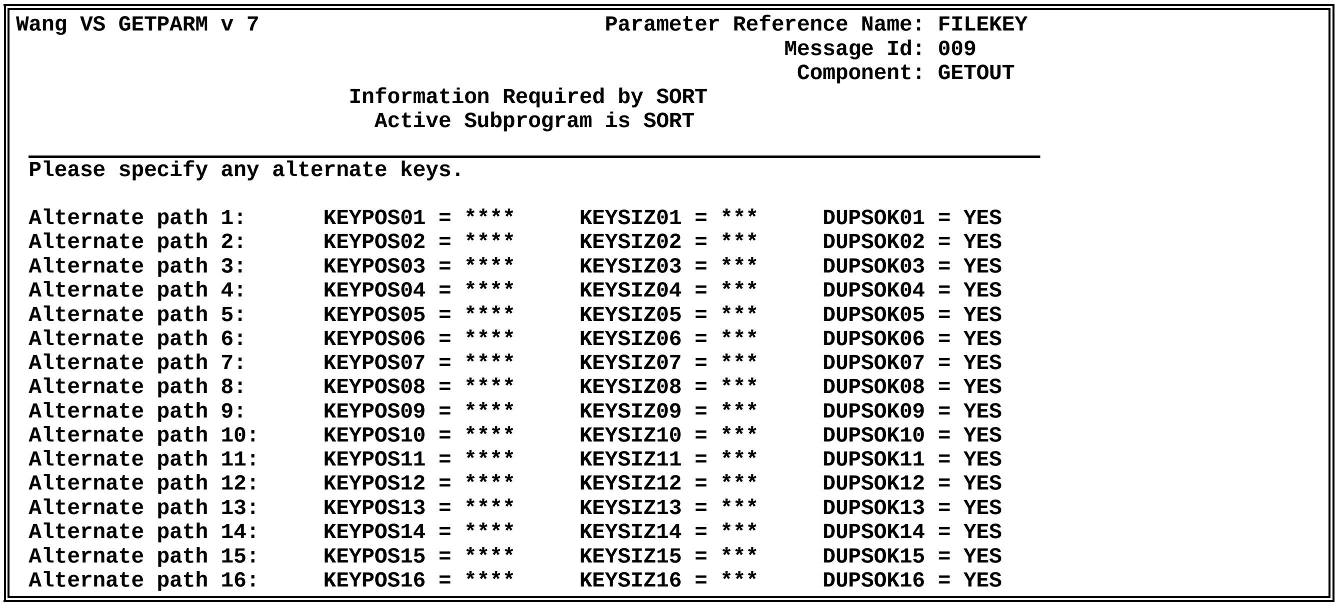
RECSIZE Specify the maximum size of the output record. When RECTYPE=V, records of excess length will be truncated or extended with binary zero to conform to this length. If REFORMAT=YES, the value defaults to the value specified for LENGTH in the FORMAT getparm. Otherwise, the value defaults to the largest record size of the input files.

RECTYPE Specify F (fixed) or V (variable) record format. When COMPRESS=YES, specification of F will force the logical records to conform to the value specified for RECSIZE, however the resultant file will have the variable attribute.

COMPRESS Specify whether to compress the output records. Valid responses are YES and NO. The default is YES for variable records and NO for fixed length records.

DEVICE Specify the output device type. Valid responses are DISK and TAPE. The default is DISK.

FILESEQ Specify the tape file sequence number. Required only when DEVICE=TAPE. Valid responses are non-zero positive integers. The default is one.



```
Wang VS GETPARM v 7                           Parameter Reference Name: FILEKEY
                                                            Message Id: 009
                                                             Component: GETOUT
                          Information Required by SORT
                            Active Subprogram is SORT

 Please specify any alternate keys.

 Alternate path 1:      KEYPOS01 = ****     KEYSIZ01 = ***     DUPSOK01 = YES
 Alternate path 2:      KEYPOS02 = ****     KEYSIZ02 = ***     DUPSOK02 = YES
 Alternate path 3:      KEYPOS03 = ****     KEYSIZ03 = ***     DUPSOK03 = YES
 Alternate path 4:      KEYPOS04 = ****     KEYSIZ04 = ***     DUPSOK04 = YES
 Alternate path 5:      KEYPOS05 = ****     KEYSIZ05 = ***     DUPSOK05 = YES
 Alternate path 6:      KEYPOS06 = ****     KEYSIZ06 = ***     DUPSOK06 = YES
 Alternate path 7:      KEYPOS07 = ****     KEYSIZ07 = ***     DUPSOK07 = YES
 Alternate path 8:      KEYPOS08 = ****     KEYSIZ08 = ***     DUPSOK08 = YES
 Alternate path 9:      KEYPOS09 = ****     KEYSIZ09 = ***     DUPSOK09 = YES
 Alternate path 10:     KEYPOS10 = ****     KEYSIZ10 = ***     DUPSOK10 = YES
 Alternate path 11:     KEYPOS11 = ****     KEYSIZ11 = ***     DUPSOK11 = YES
 Alternate path 12:     KEYPOS12 = ****     KEYSIZ12 = ***     DUPSOK12 = YES
 Alternate path 13:     KEYPOS13 = ****     KEYSIZ13 = ***     DUPSOK13 = YES
 Alternate path 14:     KEYPOS14 = ****     KEYSIZ14 = ***     DUPSOK14 = YES
 Alternate path 15:     KEYPOS15 = ****     KEYSIZ15 = ***     DUPSOK15 = YES
 Alternate path 16:     KEYPOS16 = ****     KEYSIZ16 = ***     DUPSOK16 = YES
```

FILEKEY Getparm

This getparm specifies the alternate indexes to be built. It is issued when FILEORG=I is specified in the OUTPUT getparm. If no alternate indexes are desired, simply leave the keywords blank and press ENTER.

Specifying alternate indexes may substantially increase the time required to create the output file. See the discussion in Chapter 3, “Indexed File Output” for more information about how to use the OverEasy® high-speed indexed file reorganization utility instead.

Up to 16 alternate indexes may be specified for the file. Use the corresponding KEYPOSnn, KEYSIZnn, and DUPSOKnn keywords described below to specify each alternate path.

KEYPOSnn Specify the position of this alternate key in the output record.

KEYSIZnn Specify the length of this alternate key.

DUSPOKnn Specify if duplicate key values are allowed for this alternate key. Valid responses are YES and NO. The default is YES.

When DUPSOKnn=NO, the program will terminate with return code 44 when a duplicate is encountered.

RETURN CODES

A return code from Short Order indicates the success or reason for failure of a particular sort run. To preserve compatibility with programs that invoke SORT, the same values are used. The following table contains the values and their associated conditions.

0 Successful completion of the sort, merge or copy.

4 No records were selected or the input file was null.

8 Insufficient memory to perform sort, merge or copy.

12 Record size is more than 2024 bytes.

16 Invalid numeric key was encountered.

20 User/operator abort via PF16, program expiration date exceeded, or program integrity halt. If present, print the halt file in library SORTDIAG on the system volume and contact USI for assistance.

24 Merge file records are out of sequence.

28 Record count was underestimated.

36 Out of space condition on output file.

40 Records out of order on write. Occurs when duplicate key encountered while creating an indexed output file.

44 Write failed on output for a reason other than those described by codes 36 and 40.

6 GETPARM QUICK REFERENCE

Getparm Parameters

This chapter is a quick reference guide for use when writing procedures. Detailed information about any of the getparms or keywords is provided in Chapter 4.

PRNAME	KEYWORDS	LENGTH	TYPE	DEFAULT	PERMITTED VALUES
OPTIONS	FUNCTION	5	AN	SORT	SORT, MERGE, COPY
	MEMORY	4	INT	AUTO	AUTO, MIN, MAX, or integer
	ADDROUT	3	AN	NO	YES or NO
	KEYOUT	3	AN	NO	YES or NO
	STABLE	3	AN	NO	YES or NO
	REFORMAT	3	AN	NO	YES or NO
	FILES	1	NUM	2	2 – 5
	ALTSEQ	6	AN	NONE	ASCII, EBCDIC, UPPER TABLE, or NONE
TABLE	HEX00#0F	35	AN	00100203 040506070 8090A0B 0C0D0E0F	
	HEX10#1F	35	AN	10111213 14151617 18191A1B 1C1D1E1F	
	…				
	HEXF0#FF	35	AN	F0F1F2F3 F4F5F6F7 F8F9FAFB FCFDFEFF	
INPUT	FILE	8	AN		
	LIBRARY	8	AN	INLIB	
	VOLUME	6	AN	INVOL	
	SHARED	3	AN	NO	YES or NO
	SELECT	3	AN	NO	YES or NO
	MOREFILE	3	AN	NO	YES or NO
	DEVICE	4	AN	DISK	DISK or TAPE
	FILESEQ	4	INT	1	1 – 9999
	RECORDS	8	INT	1000	1 – 99999999
LOCK	LOCK	3	AN	YES	YES or NO
	TIMEOUT	3	INT	10	1 – 999 or NO
	BYPASS	3	AN	NO	YES or NO
SELECT	FLDPOSn	4	INT		>0, <= record size
	LENGTHn	4	INT		1 – 16
	FLDTYPn	1	AN	C	C, B, P, D, L, Z, Y
	TSTRELn	2	AN		EQ,NE,GT,LT,GE,LE
	VALUEn	18	AN		Literal or field position
	CONECTn	3	AN		AND or OR
	(1 <= n <= 32)				

PRNAME	KEYWORDS	LENGTH	TYPE	DEFAULT	PERMITTED VALUES
SELECT2	(Same as SELECT1)	…	…	…	Issued when CONECT4 is not spaces.
SELECT3					Issued when CONECT8 is not spaces
SELECT4					Issuee when CONECT12 is not spaces.
SELECT5					Issued when CONECT16 is not spaces.
SELECT6					Issued when CONECT20 is not spaces
SELECT7					Issued when CONECT24 is not spaces.
SELECT8					Issued when CONECT28 is not spaces.
KEYS	KEYS	1	INT	1	1 – 8
	POSTn	4	INT		1 – record size
	LENGTHn	3	INT		Depends on TYPEn
	TYPEn	1	AN	C	C, U, B, D, F, L, P, Z, Y
	ORDERn	1	AN	A	A or D
	(1 <= n <= 8)				
FORMAT	LENGTH	4	INT		1–max for output FILEORG
	PAD	2	AN		Single char or 2 digit hex
	INPOSn	4	INT		1 – record size
	LENGTHn	4	INT		Within input record
	OUTPOSn	4	INT		Within output record
	(1 <= n <= 10)				
OUTPUT	FILE	8	AN		
	LIBRARY	8	AN	OUTLIB	
	VOLUME	6	AN	OUTVOL	
	FILEORG	1	AN	C	C, I, or R
	DPACK	3	INT	100	1 – 100
	IPACK	3	INT	100	1 – 100
	RECSIZE	5	INT	From input	1-max for output FILEORG
	RECTYPE	1	AN	From input	F or V
	REPLACE	3	AN	NO	YES or NO
	COMPRESS	3	AN	From input	YES or NO
	DEVICE	4	AN	DISK	DISK or TAPE
	FILESEQ	4	INT	1	1 - 9999
FILEKEY	KEYPOSn	4	INT		1 – record size or blank
	KEYSIZn	3	INT		1 –256 (blank if KEYPOSn is blank)
	DUPSOKn	3	AN	YES	YES or NO
	(01 <= n <= 16)				