



SORT V2 ECO Patch Kit

Updates and Bug Fixes

Publication Date: April 2025

Operating Systems: VSI OpenVMS Alpha Version 8.4-2L1 or higher
VSI OpenVMS IA-64 Version 8.4-1H1 or higher
VSI OpenVMS x86-64 Version 9.2-2 or higher

Table of Contents

1. Introduction	3
2. Common Updates (SORTSHR and HYPERSORT)	3
3. HYPERSORT Updates	5
3.1. Performance Improvements	5
3.2. Bug Fixes	5
4. SORTSHR Updates	6
4.1. Future Support	6
4.2. Known Issues	6
4.3. Bug Fixes	6

1. Introduction

The SORT V2 ECO patch kit provides enhancements and bug fixes for the SORT operation (both SORTSHR and HYPERSORT methods)¹.

The SORT V2 ECO patch kits for all VSI OpenVMS architectures provide equivalent functionality. The table below details which kits can be applied to which versions of VSI OpenVMS.

Kit Name	Operating System
VMS842L2A_SORT-V0200	VSI OpenVMS Alpha V8.4-2L1 and V8.4-2L2
VMS842L3I_SORT-V0200	VSI OpenVMS IA-64 V8.4-1H1, V8.4-2, V8.4-2L1, and V8.4-2L3
VMS923X_SORT-V0200	VSI OpenVMS x86-64 V9.2-2 and V9.2-3

All changes to SORT described in this document will be directly included in the next versions of VSI OpenVMS without the requirement to install the SORT V2 ECO kit.

2. Common Updates (SORTSHR and HYPERSORT)

The following changes affect both the SORTSHR and the HYPERSORT methods of the SORT operation:

- The data type of several metrics reported by the **SORT/STATISTICS** command was changed from longword to quadword to avoid overflows when working with very large files. The following metrics were affected:
 - Records read
 - Records sorted
 - Records output
 - Work file allocation
- The output format for the values displayed by the **SORT/STATISTICS** command has been adjusted to allow for larger values (all output fields have the same names and are presented in the same order as before). See the following example:

```

                                OpenVMS Sort/Merge Statistics

Records read:                    8139332           Input record length:           244
Records sorted:                  8139332           Internal length:                246
Records output:                  8139332           Output record length:          244
Working set:                     2068480           Sort tree size:                1392312
Virtual memory:                  43617            Number of initial runs:         4
Direct I/O:                      79515            Maximum merge order:           4
Buffered I/O:                     46             Number of merge passes:         1
Page faults:                     44826            Work file alloc:                4176962
Elapsed time:                    0 00:02:56.35       Elapsed CPU:                    0 00:01:37.14

```

¹Every SORT change described in this document is also applicable to the MERGE operation. MERGE has the equivalent functionality as SORT but can provide better overall performance in cases when multiple partially sorted input files are combined into one sorted output file.

Note

The first digit of the `Elapsed time` and `Elapsed CPU` values now indicates the number of days that the operation took.

HYPERSORT uses the same output format as above, although note that it does not compute the following metrics:

- Internal length
- Output record length
- Sort tree size
- Number of initial runs
- Maximum merge order
- Number of merge passes
- The work file allocation value is now provided by HYPERSORT.
- Any program that uses the callable SORT routines can request the statistical values using the `SOR$STAT` routine as documented in the *VSI OpenVMS Utility Routines Manual* [<https://docs.vmssoftware.com/vsi-openvms-utility-routines/>].

To avoid breaking legacy programs, the following `SOR$STAT` item codes remain unchanged in this kit and return longword values (truncated if the total exceeds 32 bits):

- `SOR$K_REC_INP`
- `SOR$K_REC_OUT`
- `SOR$K_REC_SOR`
- `SOR$K_WRK_ALQ`

To obtain the complete 64-bit quadword metric values, use the following new item codes:

- `SOR$K_REC_INP_64`
- `SOR$K_REC_OUT_64`
- `SOR$K_REC_SOR_64`
- `SOR$K_WRK_ALQ_64`
- To allow DCL command procedures that use the `SORT/STATISTICS` command to capture relevant statistical information, the following DCL symbols are now created in the local symbol table after the `SORT/STATISTICS` command completes:
 - `SORT$CPU_TIME`

- SORT\$ELAPSED_TIME
- SORT\$RECORDS_OUTPUT
- SORT\$RECORDS_READ
- SORT\$RECORDS_SORTED

Consider the following example of the data returned by these DCL symbols:

```
SORT$CPU_TIME = "0-00:00:31.10"  
SORT$ELAPSED_TIME = "0-00:00:33.57"  
SORT$RECORDS_OUTPUT = "8139332"  
SORT$RECORDS_READ = "8139332"  
SORT$RECORDS_SORTED = "8139332"
```

Notes:

- The values for the time fields are defined in the correct DCL Delta Time format to allow DCL time value manipulations.
 - The DCL symbols are created as character strings in the local symbol table. This prevents any issues with overflow of DCL integer symbols, which are limited to 32 bits.
-

3. HYPERSORT Updates

3.1. Performance Improvements

- HYPERSORT has been modified to use the P2 process address space for the sort data instead of the P0 process space. This means that the limit for virtual address space that can be requested has changed from 1 gigabyte (P0) to 1 terabyte (P2).² The larger address space available means that HYPERSORT can retain much more data in memory and incur a much reduced need for or size of work files. This may provide a significant performance increase for HYPERSORT operations.

Predicting SORT performance behavior is complicated by the starting order of the data, the size of the file(s) involved, and the available resources that the operation can use. Benchmarks conducted by VSI showed a range from no improvement for very small files to up to a 33% reduction in elapsed time for a very large file, with no special tuning required. Your results will vary but, for reasonably large SORT files, they are likely to be noticeably better compared to the previous version of HYPERSORT.

- HYPERSORT can now sort files up to 2 terabytes in size.
- HYPERSORT now uses larger RMS multiblock count (MBC) and multibuffer count (MBF) limits (255 and 8 respectively). This provides better overall file system performance for HYPERSORT I/O operations.

3.2. Bug Fixes

- HYPERSORT temporary files now have a .TMP filename extension if not specifically designated by the work file logical names (SORTWORK*n*). Previously, no filename extension was provided.

²Note that neither of these values is the actual amount that can be obtained due to other uses of the address space as well as process and system quotas and limitations.

- HYPERSORT now correctly sorts files in the Variable with Fixed Control (VFC) record format. Previously, some records could be lost if the control header for a record spanned internal buffers. In some cases this could result in an access violation and SORT termination.
- Conversions between fixed- and variable-length record sizes are now handled correctly.
- HYPERSORT now supports the **SORT/OVERLAY** and **SORT/COLLATING=MULTI** commands.

4. SORTSHR Updates

4.1. Future Support

While VSI will continue to provide support for SORTSHR, no large-scale performance updates are planned for it, as SORTSHR is inherently much less efficient than HYPERSORT.

HYPERSORT is the preferred mechanism for better SORT performance, especially for sorting large files. If you have not used HYPERSORT before or have had problems with HYPERSORT in the past, VSI encourages you to try using it after installing the SORT V2 ECO kit.

4.2. Known Issues

When starting a sort for a large file, SORTSHR may issue multiple error messages of the following form:

```
%SORT-W-SYSERROR, system service error  
-LIB-F-INSVIRMEM, insufficient virtual memory
```

This is normal behavior for SORTSHR as it attempts to determine how much P0 space memory it can allocate. The sort operation will continue after the messages are output. If sufficient resources are available, SORTSHR will eventually complete the operation. The most likely failure case for SORTSHR when sorting large files is insufficient work file space.

4.3. Bug Fixes

- SORTSHR can now sort files with more than 4,294,967,295 records. Previously, a %SOR-BADLOGIC error was returned if the file exceeded that number of records.
- SORTSHR can now be installed as a resident image for better overall system performance in SORT-intensive environments. Previously, this was not possible on x86-64 and Alpha systems.

For details on installing images as resident images, see *VSI System Management Utilities Reference Manual, Volume I* [https://docs.vmssoftware.com/vsi-openvms-system-management-utilities-reference-manual-volume-i-a-1/#INSTALL_CH].