

## What is DFSORT/VSE?

### DFSORT/VSE Web Site

For papers, online books, examples and more, visit the DFSORT/VSE website at URL:

<http://www.ibm.com/storage/dfsorvse>



---

# Contents

<b>What is DFSORT/VSE?</b>	<b>1</b>
Introduction	1
<b>How does DFSORT/VSE increase your efficiency?</b>	<b>1</b>
Using dataspace sorting	2
Using getvis sorting	2
Expanding system capacity	2
Improving system availability	2
Exploiting system resources	2
Enhancing system space management	3
<b>What tasks can you perform using DFSORT/VSE?</b>	<b>4</b>
Rearrange records	4
Combine files	4
Duplicate files	5
Produce reports	5
Analyze data	6
Process files	7
<b>What kind of flexibility does DFSORT/VSE offer you?</b>	<b>8</b>
Customizing DFSORT/VSE	8
Invoking DFSORT/VSE	9
Invoking the ICETOOL utility	9



---

# What is DFSORT/VSE?

---

## Introduction

DFSORT/VSE offers three basic functions - sorting, merging, and copying. These functions provide the basis for a host of features offered in DFSORT/VSE. Using DFSORT/VSE, you can:

- Sort using data space and GETVIS areas.
- Sort using virtual disk space rather than DASD.
- Specify secondary allocation amounts for SAM ESDS work files which allows more efficient use of work space.
- Sort, merge, and transform a wide variety of dates with two-digit years according to a specified sliding or fixed century window.
- Create reports quickly and easily using DFSORT/VSE's ICETOOL utility. ICETOOL also enables you to do a variety of utility-type functions such as data analysis and validation.
- Specify subsets of data using the SKIPREC and STOPAFT options.
- Control which records to keep in the final output file by using the INCLUDE and OMIT control statements.
- Rearrange or eliminate fields in your record by using the INREC and OUTREC control statements.
- Sum numeric information into totals or eliminate duplicate records by using the SUM control statement.
- Control the use of DFSORT/VSE's STXIT exit for abend recovery processing.
- Invoke DFSORT/VSE from an application program written in a high level language such as COBOL, PL/I, or RPG II with the Auto-Report Feature.
- Use EBCDIC or ISCI/ASCII collating sequences, specify your own collating sequence, or use locale processing.
- Call user-written exit routines to expand the capabilities of DFSORT/VSE.

---

## How does DFSORT/VSE increase your efficiency?

Performance improvements with DFSORT/VSE help reduce batch window times and increase system throughput. Reduced input/output times and work space requirements are also possible using DFSORT/VSE. It is designed specifically to optimize efficiency and processing speed by:

- Using data space
- Using virtual storage (GETVIS areas)
- Using extended addressing
- Supporting the COBOL/VSE FASTSRT compile-time option

## Using dataspace sorting

Dataspace sorting is a DFSORT/VSE capability that uses data space available with VSE/ESA systems (ESA supervisor mode) in place of intermediate work space to improve the performance of sort applications. A data space is an area of contiguous storage backed by processor or auxiliary storage, whichever is necessary as determined by the system.

With dataspace sorting, data space is used as an extension of main storage for storing records during a sort. Dataspace sorting reduces CPU time and elapsed time by using data space in place of DASD work files.

You can use the DSPSIZE installation or run-time option to enable dataspace sorting.

Dataspace sorting can be used with both fixed-length records (FLR) and variable-length records (VLR). If you currently use partition program area sorting for your applications, using dataspace sorting should provide significant CPU and elapsed time performance improvements.

## Using getvis sorting

Getvis sorting is a DFSORT/VSE capability that uses partition virtual storage for sort applications. Getvis sorting is similar to dataspace sorting except that it uses the partition GETVIS area for virtual storage instead of dataspace.

You can use the GVSIZE installation or run-time option to enable getvis sorting.

If you currently use partition program sorting for your applications, getvis sorting should provide significant CPU and elapsed time performance improvements. Also, using the partition GETVIS area exploits the available partition virtual storage above 16M virtual.

## Expanding system capacity

With 31-bit addressing, DFSORT/VSE takes advantage of the additional virtual storage available above 16-megabyte virtual. 31-bit addressing gives the system more capacity for calling programs and user exits resulting in better performance for DFSORT/VSE and for the overall system. The additional virtual storage means more sort applications can run with less work space, thereby improving elapsed time, CPU time, and SIOs.

## Improving system availability

By moving its program modules and data buffers above 16-megabyte virtual, DFSORT/VSE eases the virtual storage constraints on your system. With more available storage, your system can handle more sort applications and process data quicker, resulting in better overall system performance and throughput.

## Exploiting system resources

DFSORT/VSE is typically used many times each day at large sites. Because sorting can occupy much of a site's CPU time and I/O and channel resources, even small improvements in performance can translate directly into significant dollar and machine resource savings. DFSORT/VSE automatically selects processing methods designed to give significant overall performance benefits while balancing:

- Elapsed time used by the program (elapsed time is always important and is often the primary consideration when evaluating the efficiency of very large sorts)
- CPU time used by the program
- I/O utilization (device connect time, channel usage, and SIOs)

DFSORT/VSE's dataspace sorting capability uses data spaces available on VSE/ESA. By using dataspace sorting, more sort applications can be processed without work space, and, those applications that still require work space can be processed with a smaller amount.

In both cases, dataspace sorting can improve your batch window. By avoiding or reducing I/O to DASD, DFSORT/VSE's dataspace sorting improves elapsed time, CPU time, SIOs, and channel usage. When DFSORT/VSE determines that dataspace sorting is possible and practical, the performance benefits are achieved automatically and require no changes to your sorting applications.

You can get similar benefits even when there is not enough data space available. If dataspace sorting is not possible, DFSORT/VSE will attempt to use the getvis sorting feature in addition to work space. The getvis sorting feature simulates dataspace sorting by using job resources instead of system resources. Getvis sorting can also improve elapsed and CPU time, decrease SIOs, and reduce work space requirements on systems with ESA/370 and ESA/390 architecture.

When you install DFSORT/VSE, dataspace sorting and getvis sorting are not enabled. Use the DSPSIZE installation or run-time option to enable dataspace sorting. Use the GVSIZE installation option to enable getvis sorting.

## Enhancing system space management

Even though 31-bit addressing, dataspace sorting and getvis sorting reduce work space requirements, some sort applications with extremely large files may still require the use of work space. DFSORT/VSE provides for the secondary allocation of SAM ESDS work files with the VSE/VSAM Space Management for SAM feature. This means that sort processing will continue even for cases where the primary allocation has been exhausted. By allowing DFSORT/VSE to calculate and allocate the secondary work space, you will no longer need to do precise primary allocation calculations yourself, thus saving you valuable time. This means less work space will be wasted and more jobs will be able to use the allocated space. Secondary work allocation also means fewer jobs will terminate due to an incorrect primary work space allocation and, therefore, fewer jobs will need to be re-run. You can see how better space management can help you increase your system throughput.

If you have a File Management System installed at your site and set the new DFSORT/VSE FMS installation option to YES, the File Management System provides additional assistance with space management. File Management Systems provide benefits such as:

- Dynamic logical and physical device assignment
- Dynamic primary and secondary extent allocation
- Output file truncation of unused space

Getvis and dataspace sorting provide the same performance, and both allow you to exploit the storage areas above 16-megabyte virtual for sorting. Getvis sorting uses the storage resources of the partition and is preferable when an application should not depend on the availability of system virtual storage. Dataspace sorting uses global system resources and is preferable when an application can take extensive advantage of system virtual storage.

Since processor storage plays an important role in the use of getvis and dataspace sorting, DFSORT/VSE bases its use of its very efficient sorting methods on the amount of processor storage it can use without causing excessive paging on the system. This allows DFSORT/VSE to dynamically control the storage areas used by dataspace sorting and getvis sorting. Monitoring of paging activity of the system is performed by DFSORT/VSE only when getvis and data space size are set to the maximum value.

---

## What tasks can you perform using DFSORT/VSE?

You can perform many different types of tasks with DFSORT/VSE.

### Rearrange records

The primary function of DFSORT/VSE is sorting files, that is, rearranging records from one sequence to another. The SORT control statement activates the sort function. You can use additional control statements to:

- Include or exclude records.

You can create subsets of files based on relational conditions that:

- Compare two fields
- Compare a field and a constant
- Search for a constant within a field value
- Search for a field value within a constant
- Test the state (on or off) of selected bits in a field
- Compare fields and constants based on EBCDIC sequence, ISCII/ASCII sequence, a sequence you define or locale processing

You can combine relational conditions with AND and OR operators to create complex logical expressions.

- Reformat records before or after other processing.

You can edit, delete and rearrange fields, align fields in columns and insert strings, blanks or zeros before, between or after fields.

- Sum numeric values in your records.

You can use DFSORT/VSE's summing capabilities to produce totals or subtotals, or summaries of records based on combinations of numeric or character control fields. You can also use these capabilities to eliminate duplicate records.

- Change the sequence for control fields.

You can use the standard EBCDIC collating sequence, the ISCII/ASCII collating sequence, a collating sequence that you define, or locale processing.

- Skip records.

You can use the SKIPREC option to specify the number of records you want to skip before starting to sort the input files.

- Accept records.

You can use the STOPAFT option to specify the maximum number of records you want accepted for sorting.

### Combine files

With DFSORT/VSE's merge function, you can combine up to 9 previously sorted input files. The MERGE control statement activates the merge function. As with a sort, you can also use additional control statements to:

- Include or exclude records.
- Reformat records after other processing.
- Sum numeric values in your records or eliminate duplicate records.



- Specify the sequence of your output file.

## Duplicate files

With DFSORT/VSE's copy function, you can:

- Create an output file that duplicates the information in your input file.
- Specify copying using either the SORT or MERGE control statement.
- Use additional control statements to include or exclude records and reformat records after other processing.
- Specify the number of records you want to skip before starting to copy the input files using the SKIPREC option.
- Specify the maximum number of records you want accepted before starting to copy the input files using the STOPAFT option.

## Produce reports

With DFSORT/VSE's ICETOOL utility, you can produce reports quickly and easily. You can produce detailed reports with section headings and statistics, page headings and statistics, and a summary page with headings and statistics. You can produce reports that identify and count unique values, duplicate values, or values that occur a specific number of times. You can also use ICETOOL to create a report showing the DFSORT/VSE installation defaults selected at your site.

Here's an example of a complete ICETOOL job. It uses the DISPLAY operator to produce a monthly revenue report:

```
// JOB EXAMP JOBA,PROGRAMMER
// LIBDEF PHASE,SEARCH=(PRD2,PRSMPROD)
// DLBL VSESPUC,'VSESP.USER.CATALOG',VSAM
// DLBL IN2,'SORT.BRANCH',VSAM,DISP=(OLD,KEEP),CAT=VSESPUC
// EXEC ICETOOL,SIZE=128K
  DEFINE NAME(IN2) TYPE(F) LENGTH(33)
* Print monthly revenue report
  DISPLAY FROM(IN2) LIST(LST) DATE -
    TITLE('Monthly Revenue Report') PAGE -
    HEADER('Location') ON(1,15,CH) -
    HEADER('Revenue') ON(22,6,PD,A1) -
    HEADER('Profit/Loss') ON(28,6,PD,A1) -
    TOTAL('Totals') AVERAGE('Averages') BLANK
/*
/&
```

Here's the report in SYSLST that might be produced by this DISPLAY operator:

10/27/95 Monthly Revenue Report -1-

Location	Revenue	Profit/Loss
Los Angeles	22,530	-4,278
San Francisco	42,820	6,832
Fort Collins	12,300	-2,863
Sacramento	42,726	8,276
...	...	...
Totals	329,637	50,665
Averages	27,469	4,222

Notice that the numbers have commas and negative signs. The numbers look like this because the "A1" editing mask was specified for both the revenue and the profit/loss data. ICETOOL has 33 different editing masks encompassing many of the numeric notations used throughout the world.

Here's another example of ICETOOL control statements. In this example, ICETOOL's OCCUR operator is used to identify userids that appear more than 4 times in a data set as possible system intruders:

```
OCCURS FROM(FAILURS) LIST(LST) BLANK -  
PAGE TITLE('Possible System Intruders) DATE(DM4.) -  
HEADER('Userid') HEADER('Logon Failures') -  
ON(23,8,CH) ON(VALCNT) -  
HIGHER(4)
```

Here's the report in SYSLST that might be produced by this OCCUR operator:

```
- 1 - Possible System Intruders 27.10.1995  
  
Userid Logon Failures  
-----  
B7234510 5  
D9853267 11  
...
```

## Analyze data

With DFSORT/VSE, you can use the ICETOOL utility to perform data analysis. You can create multiple copies or create different subsets of an input file in a single job step. You choose the selection criteria that gives you the information you need. For example, you can decide to:

- Keep the first or last record with each unique field.
- Keep the records with duplicate values.
- Keep all the records that occur a specific number of times.
- Keep all the records that have values within a certain range.

With ICETOOL, you can also identify and display invalid decimal values in a readable format and show the location of the record in the file. You can conditionally process a group of dependent operators; if an operation fails, the remaining operators are not processed. Or, you can use ICETOOL to unconditionally process a group of independent operators; each operator is processed even though other operators succeed or fail.

## Process files

With DFSORT/VSE control statements, you direct how your files are processed. You can use control statements to provide information about the control fields and collating sequences, and to describe the input data. In addition, you can specify whether the order of equally collating records is to be preserved from input to output, and whether any user-written routines are to be included at run time.

The following are required control statements in a DFSORT/VSE application:

- **SORT**

Describes control fields and the number of input files if you are coding a sort application, or specifies a copy application. Indicates whether you want ascending or descending order for the sort.

- **MERGE**

Describes control fields and the number of input files if you are coding a merge application, or specifies a copy application. Indicates whether you want ascending or descending order for the merge.

- **RECORD**

Describes record length and type information.

Other control statements give you more control over how your files are processed:

- **ALTSEQ**

Modifies the default collating sequence that was set at installation time.

- **ANALYZE**

Specifies that DFSORT/VSE is to analyze the control statements, perform optimization calculations, issue appropriate messages, and stop the run without actually sorting, merging, or copying.

- **INCLUDE**

Specifies that only those records whose fields meet certain criteria are included.

- **INPFIL**

Describes the input files and specifies the procedures that are to be followed when an input file is opened and closed.

- **INREC**

Specifies how records are reformatted before they are processed.

- **MODS**

Specifies the use of one or more user exit routines in a DFSORT/VSE application.

- **OMIT**

Specifies that any records whose fields meet certain criteria are deleted.

- **OPTION**

Overrides installation defaults and supplies optional information. The SKIPREC, STOPAFT, and STXIT functions are used with the OPTION control statement.

- **OUTFIL**

Describes the output file and specifies the procedures that are to be followed when an output file is opened and closed.

- **OUTREC**

Specifies how records are reformatted before they are output.

- **SUM**

Specifies that numeric summary fields in records with equal control fields are to be summarized in one record and that the other records are to be deleted. Also specifies that duplicate records are to be eliminated.

---

## What kind of flexibility does DFSORT/VSE offer you?

DFSORT/VSE offers you flexibility for customizing and invoking the product.

### Customizing DFSORT/VSE

You can customize the way you use DFSORT/VSE, and how it handles your data, with:

- Installation and run-time options
- User exit routines
- Alternate collating sequences

#### Installation and Run-Time Options

DFSORT/VSE installation defaults can be customized to suit your needs. The ILUINST macro can be used to modify the IBM-supplied DFSORT/VSE installation defaults. Many of these installation options have an impact on performance, including:

- Virtual storage limit
- Data space limit
- GETVIS area limit

At run-time, you can temporarily override most specifications made during installation by specifying:

- Control statements in SYSIPT for JCL-invoked applications
- Control statements in the parameter list for program-invoked applications

#### User Exit Routines

For selected jobs, you can direct DFSORT/VSE to transfer control to your own routines at predesignated points (user exits). User exit routines extend the flexibility of DFSORT/VSE beyond the many standard capabilities it provides, allowing you to:

- Open files
- Initialize files
- Insert, delete, or alter records in files
- Terminate DFSORT/VSE
- Summarize records
- Close files
- Complete housekeeping tasks

Like the other routines written at your site, user exit routines must follow standard linkage conventions. Routines can be written in any language that can pass the location or address of a record or parameter list in register 1. Assembler and PL/I are examples. PL/I users, however, must use the special PL/I SORT subroutine facilities.

In addition, user-written routines and programs that invoke DFSORT/VSE can:

- Reside above or below 16-megabyte virtual
- Run in 24-bit or 31-bit mode
- Pass data that resides above or below 16-megabyte virtual to DFSORT/VSE

### Alternate Collating Sequences

DFSORT/VSE usually determines the order of the output records using the standard EBCDIC or ISCII/ASCII collating sequence. However, you can use locale processing, the ALTSEQ installation option, or the ALTSEQ control statement to alter the EBCDIC collating sequence.

## Invoking DFSORT/VSE

With DFSORT/VSE, you have the flexibility of choosing how to invoke DFSORT/VSE processing. Depending on the needs of your business, you can invoke DFSORT/VSE processing in the following ways:

- With an EXEC job control statement in the input stream using SORT as the program name. This means that DFSORT/VSE is not initiated from another program but is **directly invoked**.
- With a program written in either basic assembler language, COBOL, PL/I, or RPG II with the Auto-Report Feature. This means that DFSORT/VSE is initiated from another program and is **program invoked**.

## Invoking the ICETOOL utility

As with DFSORT/VSE, you have the flexibility of choosing how to invoke the ICETOOL utility. Depending on the needs of your business, you can invoke ICETOOL either directly or from a program. ICETOOL allows operator statements (and comments) to be supplied in a file or in a parameter list passed by a calling program. For each operator supplied in the parameter list, ICETOOL puts information in the parameter list pertaining to that operation, thus allowing the calling program to use the information derived by ICETOOL.