

z/OS
3.2

DFSORT Installation and Customization



Note

Before using this information and the product it supports, read the information in [“Notices” on page 223](#).

This edition applies to IBM® z/OS® 3.2 (5655-ZOS) and to all subsequent releases and modifications until otherwise indicated in new editions.

Last updated: 2025-09-30

© **Copyright International Business Machines Corporation 1973, 2025.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Figures.....	vii
Tables.....	ix
About this document.....	xi
How to use this document.....	xi
Required product knowledge.....	xii
Referenced documents.....	xii
z/OS information.....	xiii
How to provide feedback to IBM.....	xv
Summary of changes.....	xvii
Summary of changes for z/OS 3.2.....	xvii
Summary of changes for z/OS 3.1.....	xvii
Part 1. Planning for installation.....	1
Chapter 1. Getting ready to install DFSORT.....	3
DFSORT on the Web.....	3
What is on the tape?.....	3
Storage administrator examples.....	3
Chapter 2. System planning considerations.....	5
Where DFSORT fits on your system.....	5
Software requirements.....	5
Driving system requirements.....	5
Target system requirements.....	5
Required storage devices.....	6
Machine requirements.....	6
Choosing where to install DFSORT.....	6
Installing DFSORT.....	6
Required distribution libraries.....	6
Required target libraries.....	7
Should you run DFSORT resident?.....	8
Testing DFSORT.....	8
Making the target libraries accessible.....	8
Allocating DFSORT libraries.....	9
Verifying that DFSORT was installed correctly.....	9
ICEJCLJ and ICEINVJ verification.....	9
ICETOOLJ verification.....	10
ICECSRTJ verification.....	11
Deleting private libraries after testing.....	11
Verifying the SVC was installed correctly.....	11
DFSORT storage considerations.....	12
Required main storage.....	12
Effects of varying main storage.....	13
Required storage for running DFSORT resident.....	13
Required hiperspace storage.....	13

Required memory object storage.....	13
Tailoring main storage.....	14
Tailoring storage for Hipersorting, memory object sorting and dataspace sorting.....	25
System-managed storage.....	28
Installing DFSORT in permanent libraries	28
Replacing IEBGENER with the ICEGENER facility.....	28
Verifying ICEGENER installation.....	30
Discontinuing use of ICEGENER as a replacement for IEBGENER.....	31
Part 2. Customizing DFSORT.....	33
Chapter 3. Planning for customization.....	35
Customizing the standard DFSORT procedures	35
Changing the DFSORT installation defaults.....	35
Cultural environment considerations.....	36
Unicode.....	37
Unicode Environment Considerations.....	37
Invoking an EFS routine automatically	38
Using initialization and termination exits	38
Initialization exits.....	38
Termination exits.....	38
Modifying message texts	39
For the base product.....	39
Chapter 4. Changing the installation defaults.....	43
Migrating to DFSORT from other sort products.....	45
Tailoring main storage.....	45
Tailoring hiperspace, memory object and dataspace storage.....	46
Using ICEPRMxx members in PARMLIB.....	46
Coding options in ICEPRMxx members.....	46
Activating ICEPRMxx members.....	47
Activating ICEPRMxx members for different LPARS.....	48
Merging PARMLIB and ICEMAC options.....	48
Using multiple START ICEOPT commands.....	50
Using the ICEMAC macro.....	51
Installation options syntax diagram.....	52
Installation options	56
Examples of ICEMAC macros.....	108
ICEMAC JCL - batch direct invocation defaults.....	108
ICEMAC INV - batch program invocation defaults.....	109
ICEMAC TSO - TSO direct invocation defaults.....	110
ICEMAC TSOINV - TSO program invocation defaults.....	110
ICEMAC TD1 - first time-of-day defaults.....	111
Example of an ICEPRMxx member.....	112
ICEPRM01 installation default overrides.....	112
IBM-supplied installation defaults.....	113
Listing installation defaults.....	113
Chapter 5. Controlling resources using ICEIEXIT.....	121
What run-time options can ICEIEXIT change?.....	121
What installation- and run-time information can ICEIEXIT examine?.....	121
Installation-time information.....	121
Run-time information.....	122
Writing and installing ICEIEXIT.....	122
Interface points for ICEIEXIT.....	122
Contents of general register 1.....	123
Contents of the installation-time options list.....	123

Contents of the run-time options list.....	124
A sample ICEIEXIT routine.....	127
Chapter 6. Collecting statistical data.....	131
Why use DFSORT's SMF record?.....	131
Why use an ICETEXIT?.....	131
Collecting data from the DFSORT SMF record.....	131
Controlling what DFSORT passes to SMF.....	132
When DFSORT cannot pass a record to SMF.....	132
SMF record structure.....	132
Mapping the SMF record.....	133
Verifying supported record sections.....	133
Accessing multiple input data set sections.....	133
Accessing multiple OUTFIL data set sections.....	134
Identifying what release produced your SMF record.....	134
Collecting data using an ICETEXIT.....	134
Installing an ICETEXIT.....	137
Writing an ICETEXIT routine.....	137
Controlling what DFSORT passes to ICETEXIT.....	138
Mapping ICETEXIT statistics.....	138
ICETEXIT examples.....	138
A nonexecutable ICETEXIT.....	139
An executable ICETEXIT.....	143
Chapter 7. Improving tape processing with DFSMSrmm or ICETPEX.....	151
Information passed by DFSMSrmm or ICETPEX.....	151
Installing an ICETPEX routine.....	151
Writing an ICETPEX routine.....	151
A sample ICETPEX routine.....	152
Appendix A. Sample jobs	155
Customizing DFSORT.....	155
Procedures.....	155
Usermods.....	155
Using DFSORT and ICETOOL.....	155
Appendix B. ICEOPT messages and return codes.....	157
ICEOPT messages.....	157
ICEOPT return codes.....	160
Appendix C. ICEMAC and message module messages.....	161
Error messages.....	161
ICEMAC error messages.....	161
Message module error messages	168
Informational messages.....	169
ICEMAC informational messages	169
Appendix D. SMF Type-16 record.....	175
Product section.....	177
Data section.....	177
ZSORT statistics section.....	185
Record-length distribution section.....	185
Input data set section.....	186
SORTOUT data set section.....	188
OUTFIL data set section.....	189
Footnotes.....	191

Appendix E. Termination exit (ICETEXIT) parameters.....	193
SMF statistics.....	193
General statistics.....	193
Options statistics.....	195
Sort/merge statistics.....	199
Virtual storage statistics.....	201
Phase-timing statistics.....	202
SORTIN statistics.....	204
SORTOUT statistics.....	209
Work data set statistics.....	214
Work data set entry.....	214
Hipersorting statistics.....	216
Sorting with data space statistics.....	216
Memory object sorting statistics.....	217
Footnotes.....	218
Appendix F. Accessibility.....	221
Notices.....	223
Terms and conditions for product documentation.....	224
IBM Online Privacy Statement.....	225
Policy for unsupported hardware.....	225
Minimum supported hardware.....	225
Programming interface information.....	226
Trademarks.....	226
Index.....	227

Figures

1. Expected PRTAFTER SORTOUT output from ICEJCLJ and ICEINVJ.....	10
2. Expected TOOLRUN DEPTSP output from ICETOOLJ.....	10
3. Expected TOOLRUN LIST1 output from ICETOOLJ.....	11
4. Expected TOOLRUN DEPTOT output from ICETOOLJ.....	11
5. Expected CSORT SORTOUT Output from ICECSRTJ.....	11
6. Storage comparisons for SIZE and REGION.....	18
7. Effect of the TMAXLIM option on storage.....	20
8. Effects of MAXLIM Option on Storage.....	22
9. Effects of RESALL Option on Storage.....	24
10. Suggested translations for two ICEMSGs messages.....	40
11. The two original ICEMSGs messages.....	40
12. The two ICEMSGs messages changed to French.....	41
13. Coding the ICEMAC Macro for ICEAM1.....	108
14. Coding the ICEMAC Macro for ICEAM2.....	109
15. Coding the ICEMAC Macro for ICEAM3.....	110
16. Coding the ICEMAC Macro for ICEAM4.....	111
17. Coding the ICEMAC Macro for ICETD1.....	111
18. An ICEPRM01 Member in PARMLIB.....	112
19. ICETOOL DEFAULTS job.....	113
20. Page 1 of the output from the ICETOOL DEFAULTS job showing IBM-supplied defaults.....	114
21. Page 2 of the output from the ICETOOL DEFAULTS job showing IBM-supplied defaults.....	115
22. Page 3 of the output from the ICETOOL DEFAULTS job showing IBM-supplied defaults.....	115
23. Page 4 of the output from the ICETOOL DEFAULTS job showing IBM-supplied defaults.....	116

24. Page 5 of the output from the ICETOOL DEFAULTS job showing IP-supplied defaults.....	117
25. Page 6 of the output from the ICETOOL DEFAULTS job showing IBM-supplied defaults.....	117
26. Output from an ICETOOL DEFAULTS job for changed defaults (Part 1 of 2).....	118
27. Output from an ICETOOL DEFAULTS job for changed defaults (Part 2 of 2).....	119
28. Sample ICEIEXIT routine (1 of 2).....	128
29. Sample ICEIEXIT routine (2 of 2).....	129
30. Information DFSORT passes to your ICETEXIT.....	135
31. A nonexecutable ICETEXIT (1 of 4).....	139
32. A nonexecutable ICETEXIT (2 of 4).....	140
33. A nonexecutable ICETEXIT (3 of 4).....	141
34. A nonexecutable ICETEXIT (4 of 4).....	142
35. An executable ICETEXIT (1 of 7).....	143
36. An executable ICETEXIT (2 of 7).....	144
37. An executable ICETEXIT (3 of 7).....	145
38. An executable ICETEXIT (4 of 7).....	146
39. An executable ICETEXIT (5 of 7).....	147
40. An executable ICETEXIT (6 of 7).....	148
41. An executable ICETEXIT (7 of 7).....	149
42. Sample ICETPEX routine.....	153

Tables

1. Related documents.....	xii
2. Referenced documents.....	xii
3. KB of virtual memory required in LPA for DFSORT for 3.2.....	13
4. How storage-management installation defaults can be overridden at runtime.....	14
5. Options that tailor main storage.....	15
6. Options that can ease migration.....	45
7. Example of DYNSPC primary space.....	70
8. SDB=LARGE for tape output data sets.....	94
9. Controlling the type of SMF record DFSORT passes to ICETEXIT.....	138
10. SMF Type-16 record.....	175
11. SMF Type-16 Record: Product section.....	177
12. SMF Type-16 Record: Data Section.....	177
13. SMF Type-16 record: ZSORT statistics section.....	185
14. SMF Type-16 record: Record-length distribution section.....	185
15. SMF Type-16 record: input data set section.....	186
16. SMF Type-16 record: SORTOUT data set section.....	188
17. SMF Type-16 record: OUTFIL data set section.....	189
18. DFSORT SMF statistics.....	193
19. DFSORT general statistics.....	193
20. DFSORT option statistics.....	195
21. DFSORT sort/merge statistics.....	199
22. DFSORT virtual storage statistics.....	201
23. DFSORT phase-timing statistics.....	202

24. DFSORT SORTIN statistics.....	204
25. SORTIN first volume or first data set information:24.....	206
26. Additional SORTIN information.....	208
27. DFSORT SORTOUT statistics.....	209
28. SORTOUT first volume or first data set information:26.....	210
29. SORTOUT multivolume Information:19,28.....	213
30. DFSORT work data set statistics.....	214
31. DFSORT work data set entry.....	214
32. DFSORT Hipersorting statistics.....	216
33. DFSORT Sorting with data space statistics.....	216
34. DFSORT memory object sorting statistics.....	217

About this document

Planning is the task of making fundamental decisions about the options a program offers. These decisions guide, set limits for, and identify requirements for the tasks of installation, customization, administration, application programming, and diagnosis. Installation is the task of making a program ready to do useful work. This task includes adding the materials on the IBM distribution tape to your system, initializing the program, and applying PTFs to the program. When you install a product, you are carrying out decisions you made in the planning step. Customization, an optional step, gives you the opportunity to tailor the program to the needs of the users at your site.

This document is designed to be used with the [z/OS Install/Migration page \(www.ibm.com/systems/z/os/zos/installation\)](http://www.ibm.com/systems/z/os/zos/installation) for planning before installation. This document is also designed to be used for customization after installation. The [z/OS Install/Migration page \(www.ibm.com/systems/z/os/zos/installation\)](http://www.ibm.com/systems/z/os/zos/installation) covers tasks to be done for installation. If you are installing from a z/OSMF portable software instance (ServerPac), installation tasks are covered in *z/OS Planning for Installation*.

The [z/OS Install/Migration page \(www.ibm.com/systems/z/os/zos/installation\)](http://www.ibm.com/systems/z/os/zos/installation) describes all of the installation materials and gives installation instructions specific to the product release level. It also identifies and gives the contents of the functional modification identifiers (FMIDs).

For information about the accessibility features of z/OS, for users who have a physical disability, see Appendix F, “Accessibility,” on page 221.

Note: The terms “option” and “parameter” are used interchangeably in this document.

How to use this document

Read Part 1, “Planning for installation,” on page 1 first. It helps you to understand all of the decisions and steps involved in installing DFSORT. If DFSORT was provided by z/OSMF portable software instance (ServerPac), follow the steps in *z/OS Planning for Installation* to install DFSORT. If DFSORT was not provided by ServerPac, follow the steps in the [z/OS Install/Migration page \(www.ibm.com/systems/z/os/zos/installation\)](http://www.ibm.com/systems/z/os/zos/installation) to install DFSORT. If you then want to customize DFSORT, read the pertinent chapters in Part 2, “Customizing DFSORT,” on page 33.

The information in this document is organized into the following sections:

- [Part 1, “Planning for installation,” on page 1](#), prepares you to install DFSORT.
 - [Chapter 1, “Getting ready to install DFSORT,” on page 3](#), provides useful introductory information.
 - [Chapter 2, “System planning considerations,” on page 5](#), discusses such considerations as how DFSORT relates to various operating systems, what storage devices are required, how to choose distribution and target libraries, and how much storage to reserve for DFSORT.
- [Part 2, “Customizing DFSORT,” on page 33](#), discusses how to customize DFSORT and what you need to think about before you do so.
 - [Chapter 3, “Planning for customization,” on page 35](#), discusses items you need to think about before customizing standard DFSORT procedures, modifying DFSORT defaults, and changing the message texts.
 - [Chapter 4, “Changing the installation defaults,” on page 43](#), describes the DFSORT installation options.
 - [Chapter 5, “Controlling resources using ICEIEXIT,” on page 121](#), describes how you can use a customized initialization exit (ICEIEXIT) to examine and change certain installation and run-time options that control resource usage.
 - [Chapter 6, “Collecting statistical data,” on page 131](#), explains how to use the DFSORT SMF type-16 record and a termination exit (ICETEXIT) to collect statistical data.

- Chapter 7, “Improving tape processing with DFSMSrmm or ICETPEX,” on page 151, describes how DFSORT can use information from DFSMSrmm or a tape exit routine (ICETPEX) to improve the way it processes tape data sets.
- The appendixes contain detailed customization information.
 - Appendix A, “Sample jobs,” on page 155 lists the sample jobs shipped with DFSORT.
 - Appendix B, “ICEOPT messages and return codes,” on page 157 lists the messages issued by the START ICEOPT command and the return codes from the command.
 - Appendix C, “ICEMAC and message module messages,” on page 161 lists messages issued when assembling ICEMAC and message modules.
 - Appendix D, “SMF Type-16 record,” on page 175 illustrates the format of the SMF type-16 record.
 - Appendix E, “Termination exit (ICETEXIT) parameters,” on page 193 illustrates the format of the information passed to the termination exit, ICETEXIT.

Required product knowledge

To use this document effectively, you should be familiar with the following information:

- SMP/E
- Job control language (JCL)
- DFSORT and ICETOOL control statement syntax
- Data management
- Tape and disk hardware

You should also be familiar with the information presented in the following related documents:

<i>Table 1. Related documents</i>
Documentation link
z/OS DFSORT Application Programming Guide
z/OS SMP/E Messages, Codes, and Diagnosis
z/OS SMP/E Commands
z/OS SMP/E Reference
z/OS SMP/E User's Guide

Referenced documents

This document refers to the following documents:

<i>Table 2. Referenced documents</i>
Documentation links
z/OS Planning for Installation
z/OS DFSMSdfp Utilities
z/OS ISPF Dialog Tag Language Guide and Reference
z/OS ISPF Planning and Customizing
z/OS ISPF Services Guide
z/OS Language Environment Customization
z/OS MVS Initialization and Tuning Guide

<i>Table 2. Referenced documents (continued)</i>
Documentation links
<i>z/OS MVS Initialization and Tuning Reference</i>
<i>z/OS MVS JCL Reference</i>
<i>z/OS MVS System Management Facilities (SMF)</i>
z/OS Install/Migration page (www.ibm.com/systems/z/os/zos/installation)

[*z/OS DFSORT Installation and Customization*](#) is a part of a more extensive DFSORT library. These documents can help you work with DFSORT more effectively.

Documents that can help you work with DFSORT more effectively	
Task	Documentation link
Learning to Use DFSORT	<i>z/OS DFSORT: Getting Started</i>
Tuning DFSORT	<i>z/OS DFSORT Tuning Guide</i>
Diagnosing Failures and Interpreting Messages	<i>z/OS DFSORT Messages, Codes and Diagnosis Guide</i>
Application Programming	<i>z/OS DFSORT Application Programming Guide</i>

z/OS information

This information explains how z/OS references information in other documents and on the web.

When possible, this information uses cross-document links that go directly to the topic in reference using shortened versions of the document title. For complete titles and order numbers of the documents for all products that are part of z/OS, see *z/OS Information Roadmap*.

How to provide feedback to IBM

We welcome any feedback that you have, including comments on the clarity, accuracy, or completeness of the information. For more information, see [How to send feedback to IBM](#).

Summary of changes

This information includes terminology, maintenance, and editorial changes. Technical changes or additions to the text and illustrations for the current edition are indicated by a vertical line to the left of the change.

Note: IBM z/OS policy for the integration of service information into the z/OS product documentation library is documented on the z/OS Internet Library under [IBM z/OS Product Documentation Update Policy](http://www.ibm.com/docs/en/zos/latest?topic=zos-product-documentation-update-policy) (www.ibm.com/docs/en/zos/latest?topic=zos-product-documentation-update-policy).

Summary of changes for z/OS 3.2

The following content is new, changed, or no longer included in z/OS 3.2.

New

The following content is new.

September 2025 release

- None.

Changed

The following content is changed.

September 2025 release

- Various release-specific updates are made.

Deleted

The following content is deleted.

September 2025 release

- None.

Summary of changes for z/OS 3.1

The following content is new, changed, or no longer included in z/OS 3.1.

New

The following content is new.

September 2023 release

- None.

Changed

The following content is changed.

September 2023 release

- References to *Serverpac: Using the Installation Dialog* are replaced with [z/OS Planning for Installation](#).

Deleted

The following content was deleted.

September 2023 release

- Removed all references to the DFSORT FTP site.

Part 1. Planning for installation

Chapter 1. Getting ready to install DFSORT

z/OS DFSORT Installation and Customization is a valuable resource for system programmers installing DFSORT. Besides providing information on storage amounts, prerequisite products, and so on, this manual provides all of the information necessary for you to customize DFSORT for the needs of your site.

Getting ready to install DFSORT consists of 2 phases:

Planning the Installation. You should complete a thorough installation plan before installing DFSORT. This involves using the *z/OS Install/Migration* page (www.ibm.com/systems/z/os/zos/installation) in conjunction with this document to understand all of the decisions and steps involved in the installation, ensuring that system and storage requirements have been met, and making any decisions regarding private and public libraries.

Planning for Customization. To install DFSORT, you must modify some aspects of its operation, either during or after installation, to meet user requirements. This document will help you customize DFSORT for your site.

DFSORT on the Web

For articles, online documents, news, tips, techniques, examples, and more, visit the *DFSORT Home Page* (www.ibm.com/storage/dfsor).

What is on the tape?

The DFSORT product tape contains:

- SMP/E control statements
- DFSORT modules
- DFSORT macros
- Source code for the modules that contain installation defaults and message texts
- Sample jobs for installing DFSORT

DFSORT is distributed on a tape that you can install using IBM SMP/E for z/OS. This tape contains functional modification identifiers (FMIDs) that identify DFSORT to SMP/E. See *z/OS Install/Migration* page (www.ibm.com/systems/z/os/zos/installation) for details on the installation of DFSORT. See *z/OS SMP/E Reference* and *z/OS SMP/E User's Guide* for information on SMP/E.

Storage administrator examples

DFSORT provides a set of sample jobs that demonstrate techniques of interest to storage administrators and others who analyze data collected from products such as DFSMSHsm, DFSMSrmm, DCOLLECT, and SMF. These sample jobs can be found in the ICESTGEX member of the SICESAMP library after DFSORT is installed. These sample jobs show some of the many ways DFSORT features such as ICETOOL and OUTFIL can be used to analyze data and generate reports:

DCOLEX1

DCOLLECT Example 1: VSAM report

DCOLEX2

DCOLLECT Example 2: Conversion reports

DCOLEX3

DCOLLECT Example 3: Capacity planning analysis and reports

DFHSMEX1

DFHSM Example 1: Deciphering Activity Logs

Getting Ready to Install DFSORT

DFHSMEX2

DFHSM Example 2: Recover a DFHSM CDS with a broken index

RMSEX1

DFSMSrmm Example 1: SMF audit report

RMSEX2

DFSMSrmm Example 2: Create ADDVOLUME commands

Chapter 2. System planning considerations

This section supplements the step-by-step installation procedures in the [z/OS Install/Migration page \(www.ibm.com/systems/z/os/zos/installation\)](http://www.ibm.com/systems/z/os/zos/installation) by providing information about the decisions you need to make before and during installation.

There are four main items you should consider before you install DFSORT:

- DFSORT's relationship to the operating system
- Required program products and storage devices
- Selection of distribution and target libraries
- Storage allocation.

This section also reviews these topics:

- The questions you should answer before deciding whether to run DFSORT resident or nonresident
- The storage requirements for main, central and auxiliary storage during testing and use
- The available options in tailoring the amount of main storage
- The procedures for verifying that DFSORT was installed correctly
- Considerations for replacing IEBGENER with ICEGENER

Where DFSORT fits on your system

DFSORT can be installed under the z/OS® operating system.

Additionally, DFSORT runs under z/OS when it is running as a guest under z/VM®.

Because DFSORT uses the operating system's data management facilities:

- DFSORT must be invoked according to operating system conventions.
- Any data sets used by DFSORT must be defined according to operating system standards.

DFSORT is intended to run in problem state and in a user key (that is, key 8 or higher).

Software requirements

There are two sets of software requirements for DFSORT. The first set of products is required to install DFSORT. Those products must be installed on the driving system, that is, the system used to perform the installation. The second set of products is required to run DFSORT. Those products must be installed on the target system, that is, the same system on which DFSORT is installed.

Driving system requirements

To install DFSORT, you must have a supported release of z/OS available.

Target system requirements

To use DFSORT, you must have z/OS 3.2 available.

Additionally:

- Enabling support from the SAS Institute, Inc. is required to use DFSORT's Performance Booster for the SAS System.
- The IBM Double-Byte Character Set Ordering Support Program (DBCS Ordering) Release 2, 5665-360, is required to process Japanese (Kanji) character data. Alternatively, you can use DFSORT locale processing for minimal DBCS ordering.

Required storage devices

To install DFSORT, you need:

- A tape drive if you ordered DFSORT on tape media.
- A disk device for the distribution and target libraries (storage estimates are given under “[DFSORT storage considerations](#)” on page 12). You can choose any disk device supported by z/OS.

To use DFSORT, you need:

- Devices supported by QSAM or VSAM for input or output.
- Disk or tape devices for work storage. You can choose any disk device or tape device supported by z/OS

For work data set storage during DFSORT processing, high-speed disk devices such as the 3390 offer the best performance. For more information on how different devices affect performance, see [z/OS DFSORT Application Programming Guide](#).

Machine requirements

DFSORT can run in any hardware environment that supports the software described in “[Software requirements](#)” on page 5.

Choosing where to install DFSORT

You should install DFSORT in the same target and distribution library zone as your operating system.

Before installing DFSORT, you must allocate space for the distribution and target libraries. After installation, the distribution libraries contain a copy of DFSORT for maintenance purposes and the target libraries contain a copy for production.

Installing DFSORT

The [z/OS Install/Migration page \(www.ibm.com/systems/z/os/zos/installation\)](http://www.ibm.com/systems/z/os/zos/installation) provides access to step-by-step instructions for installing DFSORT.

After you have allocated space for your libraries, you can use SMP/E to install DFSORT using the following commands:

- **RECEIVE.** SMP/E RECEIVE processing loads products and service from tape. Products are stored in temporary data sets on disk, and PTFs are stored in the SMP/E SMPPTS data set.
- **APPLY.** SMP/E APPLY processing loads the elements of products and PTFs that have been received into the target libraries you defined in DDDEFs or on DD statements.
- **ACCEPT.** SMP/E ACCEPT processing loads the elements of products and PTFs you have received into the distribution libraries you defined in DDDEFs or on DD statements.

The required distribution and target libraries and the considerations for selecting the target libraries are discussed in the following sections.

Required distribution libraries

The required distribution libraries are:

DDname
Description

AICELIB
All DFSORT modules.

AICESRCE
The installation-default and message-text source statements.

AICESAMP

The sample jobs.



Attention: Be sure to use the sample jobs provided for **this** release, and not for any other release.

AICEUSER

The DFSORT macros. The installation-default and message-text macros are intended for use in changing installation defaults and in changing the DFSORT message text. The programming interface mapping macros are intended to be used with the DFSORT function where referenced. All of these macros should not be used for any other purpose.

Required target libraries

The required target libraries are:

DDname**Description****SIEALNKE**

The LNKE-eligible DFSORT ZSORT modules.

SICELINK

The DFSORT Blockset and Peervale technique modules not eligible to be placed in LPA.

SORTLPA

The LPA-eligible DFSORT Blockset technique modules.

SICELPA

The DFSORT SVC.

SORTLIB

The DFSORT Conventional technique modules not eligible to be placed in LPA.

SICESRCE

The installation-default and message-text source statements.

SICESAMP

The sample jobs.



Attention: Be sure to use the sample jobs provided for **this** release, and not for any other release.

SICEUSER

The DFSORT macros. The installation-default and message-text macros are intended for use in changing installation defaults and in changing the DFSORT message text. The programming interface mapping macros are intended to be used with the DFSORT function where referenced. All of these macros should not be used for any other purpose.

SICEPROC

DFSORT procedures

If you plan to run DFSORT resident, you must place the SORTLPA library modules in LPA (LPA list or Dynamic LPA). For more information, see [“Should you run DFSORT resident?” on page 8](#).

If you plan to use standard DFSORT catalogued procedures and you use DFSORT frequently, you should consider taking one of the following actions to make the procedures generally accessible:

- Place the SICEPROC library in your JES procedure library concatenation.
- Copy the procedures to another procedure library already in the concatenation (such as PROCLIB).

See [“Customizing the standard DFSORT procedures ” on page 35](#) for information on the standard DFSORT cataloged procedures.

To permanently modify the DFSORT procedures, you can either:

- Modify them in the SICEPROC library. This method has the advantage of letting you identify the changes with an SMP/E usermod. However, it has the disadvantage of requiring you to repeat this task each time you reinstall or replace DFSORT.
- Copy them from the SICEPROC library to another library and modify the copies. The target library should be one that is already in the JES procedure library concatenation or one that you plan to add to that concatenation. Because the DFSORT procedures are not often changed, this method has the advantage of reducing your installation and migration workload. However, it has the disadvantage that you must remember to check for changes to the DFSORT procedures each time you reinstall or replace DFSORT.

If you have SORT or SORTD cataloged procedures from a prior release of DFSORT in a procedure library in the JES procedure library concatenation (such as PROCLIB), you should delete them. Make sure you do this before you add SICEPROC to the procedure library concatenation or copy the new DFSORT procedures into PROCLIB. (If you want to keep the old procedures while you do this, you must move them to another library.)

For testing, you can allocate the SICEPROC library using the JCLLIB JCL statement.

If you plan to record DFSORT SMF type-16 records or use IBM cached disk devices, you must add the SVC module in SICELPA to LPA or MLPA.

You can combine SORTLIB with SICELINK. If you combine these libraries, you will only need to supply a SORTLIB DD statement for tape work data set sort or Conventional merge applications that use dynamic linkedit of user exits. You will have to specify SORTLIB=SYSTEM as an installation default. Combining the libraries requires more SICELINK space.

Should you run DFSORT resident?

Your production requirements determine whether you should run DFSORT resident (that is, with the link library for the reentrant Blockset modules in LPALST) or nonresident (that is, with the link library for the reentrant Blockset modules in LNKLST, or in a private library specified in a JOBLIB or STEPLIB DD statement at run time). For information on the storage required when you run DFSORT resident, see [“Required storage for running DFSORT resident” on page 13](#).

Testing DFSORT

Before migrating a new release of DFSORT to your production systems, you should test it. If you have installed DFSORT on a copy of your operating system, perhaps with other products, you should IPL the operating system on a test image to do this testing. This approach is recommended.

Test DFSORT in the same execution mode you plan to run it in. For example, if you will run DFSORT resident in production, you should test it running resident. To test DFSORT running resident, you need a test system because only one level of DFSORT's modules can be used in LPA at a time. You can test nonresident DFSORT either by using a test system (recommended) or by calling its modules from a TASKLIB, STEPLIB, or JOBLIB.

Making the target libraries accessible

After you have installed DFSORT, you must make the target libraries accessible on the system.

To run DFSORT resident, you must:

- Load the modules from SORTLPA into LPA or Dynamic LPA
- Add SICELINK to the link list and to the APF-table (member PROGxx of PARMLIB), if you use LNKAUTH=APFTAB in IEASYSxx.

To run DFSORT nonresident, you can do one of the following:

- Add SICELINK and SORTLPA to the link list and to the APF-table (member PROGxx of PARMLIB), if you use LNKAUTH=APFTAB in IEASYSxx
- Use a JOBLIB or STEPLIB for SICELINK and SORTLPA (in that order).

If you plan to record DFSORT SMF type-16 records or use IBM cached disk devices (such as the 3990 Model 3), the DFSORT SVC must be accessible:

- Load the DFSORT SVC from SICELPA into LPA or MLPA.

If you want to assign a new user SVC number to the DFSORT SVC, you will have to IPL to make the changes to the IEASVCxx member of parmlib effective.

For information about adding modules to LPA, Dynamic LPA, and the link list, and for information about assigning user SVC numbers, see *z/OS MVS Initialization and Tuning Guide* and *z/OS MVS Initialization and Tuning Reference*.

If you use DFSORT macros, you should add SICEUSER to the assembler SYSLIB concatenation.

Allocating DFSORT libraries

The ICEALLOC sample job allocates space for all DFSORT target and distribution libraries. For information about the libraries that are required, see the *z/OS Install/Migration page* (www.ibm.com/systems/z/os/zos/installation) and the comments in the ICEALLOC sample job.

You can combine SORTLIB with SICELINK. If you combine these libraries, you will only need to supply a SORTLIB DD statement for tape work data set sort or Conventional merge applications that use dynamic linkedit of user exits. You will have to specify SORTLIB=SYSTEM as an installation default. Combining libraries requires more SICELINK space.

Verifying that DFSORT was installed correctly

You should use the appropriate installation verification jobs listed later in this section to ensure that DFSORT is installed correctly. All of these jobs reside in the SICESAMP target library after APPLY processing.

Review each job and modify it appropriately before you run it (for example, if you are testing resident DFSORT, remove the STEPLIB DD statements from each job). You should run all of the installation verification jobs for which you have the needed programs to make sure the installation was successful, even if you do not plan to use all the functions they test. The jobs are:

- ICEJCLJ, to invoke DFSORT directly. Sorts and copies a data set.
- ICEINVJ, to invoke DFSORT from an assembler program. Sorts and copies a data set. To run this job, an Assembler must be available.
- ICETOOLJ, to invoke ICETOOL directly. Performs multiple operations, which include listing your installation defaults, copying and sorting data sets, displaying statistics and printing reports.
- ICECSRTJ, if you use Locale Processing at your site, to verify that data is sorted correctly using LOCALE=DA_DK. To run this job, you must have the SCEERUN library available for the job to use.

ICEJCLJ and ICEINVJ verification

Verify that the ICEJCLJ and ICEINVJ jobs each ran correctly by:

1. Checking that the condition code for each step is 0.
2. Comparing the SORTOUT output of step PRTAFTER to [Figure 1 on page 10](#), which shows what the first 20 records and last 10 records of the 360 output records look like.


```

*****000001*****ABCDEF GHIJ KLMNOP*****
*****000037*****ABCDEF GHIJ KLMNOP*****
*****000073*****ABCDEF GHIJ KLMNOP*****
*****000109*****ABCDEF GHIJ KLMNOP*****
*****000145*****ABCDEF GHIJ KLMNOP*****
*****000181*****ABCDEF GHIJ KLMNOP*****
*****000217*****ABCDEF GHIJ KLMNOP*****
*****000253*****ABCDEF GHIJ KLMNOP*****
*****000289*****ABCDEF GHIJ KLMNOP*****
*****000325*****ABCDEF GHIJ KLMNOP*****
*****000002*****BCDEF GHIJ KLMNOPQ*****
*****000038*****BCDEF GHIJ KLMNOPQ*****
*****000074*****BCDEF GHIJ KLMNOPQ*****
*****000110*****BCDEF GHIJ KLMNOPQ*****
*****000146*****BCDEF GHIJ KLMNOPQ*****
*****000182*****BCDEF GHIJ KLMNOPQ*****
*****000218*****BCDEF GHIJ KLMNOPQ*****
*****000254*****BCDEF GHIJ KLMNOPQ*****
*****000290*****BCDEF GHIJ KLMNOPQ*****
*****000326*****BCDEF GHIJ KLMNOPQ*****
.
.
.
*****000036*****9ABCDEF GHIJ KLMNO*****
*****000072*****9ABCDEF GHIJ KLMNO*****
*****000108*****9ABCDEF GHIJ KLMNO*****
*****000144*****9ABCDEF GHIJ KLMNO*****
*****000180*****9ABCDEF GHIJ KLMNO*****
*****000216*****9ABCDEF GHIJ KLMNO*****
*****000252*****9ABCDEF GHIJ KLMNO*****
*****000288*****9ABCDEF GHIJ KLMNO*****
*****000324*****9ABCDEF GHIJ KLMNO*****
*****000360*****9ABCDEF GHIJ KLMNO*****

```

Figure 1. Expected PRTAFTER SORTOUT output from ICEJCLJ and ICEINVJ

ICETOOLJ verification

Verify that the ICETOOLJ job ran correctly by:

1. Checking that the condition code for each step is 0.
2. Checking that 3.2 is shown in the heading of the DFLTS output of step TOOLRUN.
“Listing installation defaults” on page 113 shows an example of how the complete DFLTS output will look if you have not changed any DFSORT installation options. If you have changed any, the output should reflect the options that you selected. In either case, dates in your listing will differ from those in the example output.
3. Comparing the DEPTSP output of step TOOLRUN to [Figure 2 on page 10](#).
4. Comparing the LIST1 output of step TOOLRUN to [Figure 3 on page 11](#). (Carriage control characters are shown in position 1; do not be concerned if you do not see them.)
5. Comparing the DEPTOT output of step TOOLRUN to [Figure 4 on page 11](#). (Carriage control characters are shown in position 1; do not be concerned if you do not see them.)

JOHN	BURT	IS IN DEPARTMENT J69
ANDY	GELLAI	IS IN DEPARTMENT J82
PAUL	LEE	IS IN DEPARTMENT J69
MIGUEL	MADRID	IS IN DEPARTMENT J69
JANICE	MEAD	IS IN DEPARTMENT J69
LEE	TOWNSEND	IS IN DEPARTMENT J82
WILLIAM	WARREN	IS IN DEPARTMENT J82
FRANK	YAEGER	IS IN DEPARTMENT J69
HOLLY	YAMAMOTO-SMITH	IS IN DEPARTMENT J69

Figure 2. Expected TOOLRUN DEPTSP output from ICETOOLJ

1KEYS AND TOTALS	- 1 -	
KEY	PD_TOTAL	ZD_TOTAL
-----	-----	-----
ABCDXYZ123	1041	579
BCDXYZ123A	-42	290
CDXYZ123AB	142	314
DEXYZ123ABC	326	338
XYZ123ABCD	-615	363
YZ123ABCDX	-339	399
Z123ABCDXY	-63	435
123ABCDXYZ	213	471
23ABCDXYZ1	489	507
3ABCDXYZ12	765	543
MINIMUM	-615	290

Figure 3. Expected TOOLRUN LIST1 output from ICETOOLJ

1(45,3,CH)	VALUE COUNT
J62	0000000000000001
J69	0000000000000006
J82	0000000000000003
L92	0000000000000005

Figure 4. Expected TOOLRUN DEPTOT output from ICETOOLJ

ICECSRTJ verification

Verify that the ICECSRTJ job ran correctly by:

1. Checking that the condition code for each step is 0.
2. Comparing the SORTOUT output of step CSORT to [Figure 5 on page 11](#).

```
a
A
b
B
c
C
d
D
e
E
```

Figure 5. Expected CSORT SORTOUT Output from ICECSRTJ

Deleting private libraries after testing

When installation verification has been successfully completed, you can run the sample job ICEDEL to delete the temporary distribution, target, and SMP/E libraries allocated by ICEALLOC and ICEASMPE. ICEDEL is provided to prepare DFSORT for permanent installation by removing libraries temporarily installed for testing purposes.

Verifying the SVC was installed correctly

The DFSORT SVC is called to write SMF type-16 records and to process data sets on cached disk devices.

If SMF=SHORT (write short SMF type-16 records) or SMF=FULL (write full SMF type-16 records) is in effect, the correct DFSORT SVC for this release must be loaded in LPA or MLPA.

If the SVC is not properly installed, DFSORT applications may encounter one of the following problems:

- A system abend (56D or Fnn) when writing the SMF type-16 record
- Degraded performance when using data sets on cached disk devices

In either case, message ICE187I is issued if the SVC is installed at the wrong level. If you are using a cached disk device, message ICE191I is issued as a warning that performance might be degraded. However, the run continues successfully if there are no other errors. You can specify a SORTDIAG DD statement to cause an additional message, ICE816I, to be issued. This message shows the abend code resulting from an attempted SVC call.

You can use the abend code in ICE816I, or the code resulting from the attempt to write an SMF type-16 record, to determine the reason DFSORT failed to call the SVC successfully.

DFSORT storage considerations

The following sections provide information about the storage required to run DFSORT, and discuss how to tailor storage to meet the requirements at your site. For additional information on tuning storage, see [z/OS DFSORT Tuning Guide](#).

Required main storage

Although DFSORT requires a minimum of 88 KB (below 16 MB virtual), the minimum amount of main storage required depends on the application.

Recommendation: For best performance, use significantly more than the minimum amount of main storage.

You will generally need more main storage if you use:

- Spanned records
- COBOL user exit routines
- CHALT or SMF options
- ALTSEQ, INCLUDE, OMIT, SUM, OUTREC, or INREC control statements
- Very large blocks or logical records
- VSAM data sets
- An Extended Function Support (EFS) program
- An ICETEXIT routine
- A large ICEIEXIT routine
- OUTFIL control statements (especially if many OUTFIL data sets are specified or if the data sets have large block sizes)
- Locale processing
- A large number of JCL or dynamically allocated work data sets.
- SORT/MERGE of Unicode data

You can limit the amount of main storage that DFSORT allocates by using the SIZE installation option or the MAINSIZE run-time option, or by installing an initialization exit (ICEIEXIT), in which you specify the maximum amount of storage DFSORT is to allocate under certain conditions. See [Chapter 5, “Controlling resources using ICEIEXIT,” on page 121](#) for more information. This limit applies to the total storage allocated both above and below 16 MB virtual. Storage above 16 MB virtual is used for Blockset sort, copy, and merge applications.

OUTFIL processing is the exception to these limits. It is subject only to system storage limits, such as IEFUSI, and to DFSORT's ODMAXBF limit, but not to other DFSORT storage limits, such as SIZE, MAINSIZE, MAXLIM, TMAXLIM, and so on. OUTFIL data set buffers can be placed above 16 MB virtual for sort, copy and merge applications. You can see how much storage is used exclusively for OUTFIL processing, both above and below 16 MB virtual, in message ICE231I.

Note: In some cases, this release might use more storage than prior releases. This can affect the operation of some jobs. For example, some jobs that previously ran as in-storage sorts (with no SORTWKdd data sets) might not run in-storage under this release.

Effects of varying main storage

When you choose higher storage limits for DFSORT:

- EXCPs are reduced.
- Processor time generally decreases for large files.
- Elapsed time might increase for a heavily loaded system because DFSORT might be swapped out more often.
- Processor time for small sorts might remain stable or increase.

When you choose lower storage limits for DFSORT:

- EXCPs are increased.
- Elapsed time increases for most sorts.
- Processor time decreases for very small files, but increases for larger files.

Required storage for running DFSORT resident

The size of LPA will be increased if you:

- Run DFSORT resident
- Place installation-wide exits in LPA
- Place Extended Function Support (EFS) programs in LPA.

Table 3 on page 13 shows approximate kilobytes (KB) of virtual memory required in the LPA for DFSORT 3.2.

Table 3. KB of virtual memory required in LPA for DFSORT for 3.2.

	Below	Above
Running DFSORT resident	162 KB	1014 KB

These approximations are based on the initial shipment of the distribution tape for 3.2. The size required in the LPA might increase with the installation of a program temporary fix (PTF) applied to DFSORT 3.2.

Required hiperspace storage

DFSORT requires hiperspace backed by central storage in order to use Hipersorting. DFSORT will use Hipersorting when a sufficient amount of central storage is available. Hipersorting greatly reduces elapsed time and work device I/O processing for sort applications. The more storage available on the system, the more Hipersorting DFSORT is able to do. You should carefully evaluate the overall effect on system performance before changing your central storage allocations.

See the discussion of the HIPRMAX option in [“Installation options”](#) on page 56 for a more detailed discussion of Hipersorting.

Required memory object storage

DFSORT requires storage that is above the bar in order to use memory object sorting. A memory object is a data area in virtual storage allocated above the bar and backed by central storage. DFSORT determines the size (in megabytes) of an appropriate memory object for memory object sorting. Because a memory object is backed by central storage, I/O processing is reduced for memory object sorting. Elapsed time, EXCP counts and channel usage are also reduced.

See the discussion of the MOSIZE option in [“Installation options”](#) on page 56 for a more detailed discussion of memory object sorting.

Tailoring main storage

You can tailor the way DFSORT uses main storage to meet the requirements at your site. This section describes the default settings you can establish by using installation options (see [Chapter 4, “Changing the installation defaults,”](#) on page 43).

At runtime, you can override several of your installation-determined defaults with:

- An ICEIEXIT routine
- An EXEC statement
- An OPTION program control statement
- An EFS program.

The ICEIEXIT routine is described in [Chapter 5, “Controlling resources using ICEIEXIT,”](#) on page 121. The EXEC statement, the OPTION statement, and using an EFS program are described in [z/OS DFSORT Application Programming Guide](#)

[Table 4 on page 14](#) summarizes how storage-related installation defaults can be overridden at runtime.

Table 4. How storage-management installation defaults can be overridden at runtime

Installation Option	ICEIEXIT	EXEC PARM	OPTION Statement
SIZE ¹	•	•	. ¹
OVERRGN			
MAXLIM			
TMAXLIM			
MINLIM			
RESALL	•	•	•
ARESALL		•	•
RESINV	•		•
ARESINV			•

Note: ¹ The MAINSIZE option of the OPTION statement is equivalent to the SIZE installation option.

The primary system storage limits for an application are set by your site’s IEFUSI exit, or by default to 1 MB below and 32 MB above 16 MB virtual.

Main storage for DFSORT jobs is limited primarily by the SIZE installation option. While SIZE cannot override the system limits, it specifies the total amount of storage (within those limits) to be used by DFSORT. In most cases, DFSORT will use storage both above and below 16 MB virtual, but will minimize its use of storage below 16 MB virtual.

Normally, REGION only limits the total storage a DFSORT job can obtain below 16 MB virtual. The value set at your site for OVERRGN can allow up to 16128 KB of additional storage over the value set for REGION. These system limits apply to storage DFSORT can obtain below 16 MB virtual, not to the total main storage available both above and below that value.

[Table 5 on page 15](#) summarizes the DFSORT installation defaults you can set.

Table 5. Options that tailor main storage

Type	Option Name	Controls	Dependencies
Options that limit storage used by DFSORT.	SIZE	Specifies total storage that DFSORT uses above and below 16 MB virtual.	If SIZE=MAX is in effect, limited by the TMAXLIM or DSA installation values when Blockset is selected, or by the MAXLIM installation value when Blockset is not selected.
	OVERRGN	Specifies amount of below 16 MB virtual storage above the REGION value that DFSORT can use.	Limited by IEFUSI exit, or 64 KB by default.
	MAXLIM	Sets upper limit to amount of storage available below 16 MB virtual.	Always used for Blockset sort applications. Used for copy and Blockset merge applications unless the use of more storage below 16 MB virtual should improve performance. If Blockset is not selected, used when SIZE=MAX is in effect.
	TMAXLIM	Sets upper limit to the total amount of storage both above and below 16 MB virtual.	Only used when SIZE=MAX is in effect.
	MINLIM	Specifies a lower limit for the SIZE value.	Only used when SIZE=n is less than MINLIM.
Options that reserve storage for system and invoking program use.	RESALL	Specifies amount of storage reserved for system use below 16 MB virtual.	Only used when SIZE=MAX is in effect.
	ARESALL	Specifies amount of storage reserved for system use above 16 MB virtual.	None.
	RESINV	Specifies amount of storage reserved for the invoking program below 16 MB virtual.	Only used when SIZE=MAX is in effect.
	ARESINV	Specifies amount of storage reserved for the invoking program above 16 MB virtual.	None.

There are general guidelines for storage allocation that will help you when setting your site's limits.

1. Your system limit is always the maximum you can get. You can set lower limits for storage, but you cannot raise them. If the limits in effect for the DFSORT options are higher than the system limits, you will get the lower system limits.
2. The maximum amounts of storage available above and below 16 MB virtual can be limited by the IEFUSI exit. For more information about IEFUSI and limiting the amount of virtual storage that can be requested, see the appropriate documents for your system.
3. DFSORT gets better performance for larger data sets if a large amount of virtual storage is available, provided there is sufficient central storage to back this amount of virtual storage. With smaller amounts of storage available, these jobs take longer to complete, and can therefore tie up system resources for longer periods. Unless your site has severe storage constraints, the trade-off for setting larger or smaller limits is between (1) better system-throughput for the jobs, and (2) being able to process more jobs at a time without excessive paging or swapping-out.
4. Several DFSORT options are available to prevent DFSORT from using all the storage in an address space, regardless of the value of SIZE in effect. RESALL, RESINV, ARESALL, and ARESINV will reserve storage for system use, and for your user exits running along with DFSORT.
5. UTFIL processing is subject only to system storage limits, such as IEFUSI, and to DFSORT's ODMAXBF limit, but not to other DFSORT storage limits, such as SIZE, MAINSIZE, MAXLIM, TMAXLIM, and so on. UTFIL data set buffers can be placed above 16 MB virtual for sort, copy and merge applications.

Note: If you are unsure of how much storage you are getting for a job, examine messages ICE156I, ICE231I, and ICE092I, or ICE093I. These messages tell how much storage was used above 16 MB

virtual, and what the total amount of storage was. Use these values to calculate how much storage was used above and below 16 MB virtual, and compare that amount to the amount you requested.

Options that allocate main storage

Use the SIZE installation option to specify the amount of main storage DFSORT should attempt to use. Specify the amount of storage with SIZE=*n* (where *n* is the number of bytes). You can also allow DFSORT to determine the amount of storage by specifying SIZE=MAX (the default and the recommended setting for this option).

Use the OVERRGN installation option to limit the amount of storage above the REGION value that will be obtained by a Blockset job. If OVERRGN is less than the system's region limit value (64 KB greater than the REGION by default), the extra virtual storage is available for other uses in the address space.

At run time, you can specify SIZE as an option in the EXEC statement to override the value SIZE installation option. MAINSIZE, which is the equivalent of SIZE, can be specified on the OPTION program control statement, and an equivalent option can also be specified in the parameter list of an invoking program.

Options that limit storage used by DFSORT

Use the options MAXLIM, TMAXLIM, and MINLIM to set limits to the amounts of maximum and minimum storage available. TMAXLIM and MAXLIM both specify upper limits to the amount of main storage available to DFSORT. Storage above 16 MB virtual is used for Blockset sort, copy, and merge applications. TMAXLIM is the limit for the total storage above and below 16 MB virtual, when SIZE=MAX is in effect. MAXLIM is the limit for the storage below 16 MB virtual. MAXLIM is always used for Blockset sort applications, and is used for copy and Blockset merge applications unless the use of more storage below 16 MB virtual should improve performance. If Blockset is not selected, MAXLIM is used when SIZE=MAX is in effect. By increasing the values for MAXLIM and TMAXLIM, you can automatically increase the storage available for all jobs that have SIZE=MAX in effect.

MINLIM is a safeguard against jobs running with too little storage. MINLIM specifies a lower limit to the SIZE value, and overrides SIZE when SIZE is too low. DFSORT always attempts to use an amount of storage equivalent to the value set for MINLIM. The MINLIM option is used only when SIZE=*n* is in effect. If MINLIM is greater than the SIZE value, DFSORT uses storage up to the MINLIM value.

You can increase or decrease the setting of MINLIM from the default of 450560 bytes (440 KB), depending on the requirements of your site. Lowering MINLIM below 440 KB can degrade performance for some applications. This performance degradation can become more pronounced and affect more applications the more MINLIM is lowered below 440 KB.

Options that reserve storage for system use

Because DFSORT always attempts to obtain as much storage as possible when running a job, you might want to reserve some storage for system use, especially if you want to use a user exit in your job. The options RESALL, ARESALL, RESINV, and ARESINV work together to reserve storage for system use.

The RESALL option is used only when SIZE=MAX is in effect. It directs DFSORT to reserve some of the storage it has allocated for use by the system, while DFSORT is running. With RESALL, the storage reserved and released to the system is always below 16 MB virtual.

If you set RESALL to 0 (zero), or do not use SIZE=MAX, you may need to allow room in your region for system storage requirements by setting the SIZE value less than the REGION value. Alternatively, you can set the REGION value greater than the MAXLIM value, but this method may not always work for copy or merge applications.

ARESALL also reserves storage for use by the system, but the storage reserved by ARESALL is above 16 MB virtual.

The RESINV option is used only when DFSORT is program invoked and SIZE=MAX is in effect. It directs DFSORT to reserve some of the storage it has allocated for user exits controlled by the invoking program

(for example, a COBOL input procedure), while DFSORT is running. RESINV is very similar to RESALL, and reserves storage in the same manner.

Because the storage reserved by RESINV is always reserved for program invoked jobs when SIZE=MAX is in effect, and is reserved for all DFSORT processing phases, you should not set the value higher than is required for your average application. The recommended way to reserve storage for user exit routines that are not controlled by the invoking program is to use the exit length value in the MODS statement.

You can change the setting of RESINV from the default of 0 bytes, if necessary.

ARESINV has the same relationship to RESINV as ARESALL has to RESALL. ARESINV reserves storage above 16 MB virtual for user exits controlled by the invoking program. Normally the site limit to total storage is sufficient. There is storage above 16 MB virtual for user exits above that which DFSORT uses.

Storage considerations

The following discussion focuses on storage options for Blockset sort, copy, and merge applications to illustrate the use of storage above and below 16 MB virtual. However, many of the concepts illustrated also apply when Blockset is not selected and only storage below 16 MB virtual is used.

SIZE

If you specify SIZE=*n*, DFSORT uses as much storage as it can obtain both above and below 16 MB virtual up to the value of *n*. If you specify SIZE=MAX, DFSORT uses as much storage as it can obtain both above and 16 MB virtual, up to the value you specify for TMAXLIM.

Lowering SIZE below 6 MB can degrade performance of some applications. This performance degradation can become more pronounced and affect more applications the more SIZE is lowered below 6 MB.

The MAXLIM and REGION (plus OVERRGN) values limit the amount of storage used below 16 MB virtual. Any storage that cannot be obtained below 16 MB virtual will be obtained above 16 MB virtual by DFSORT. See [Figure 6 on page 18](#) for examples of how SIZE affects DFSORT storage.

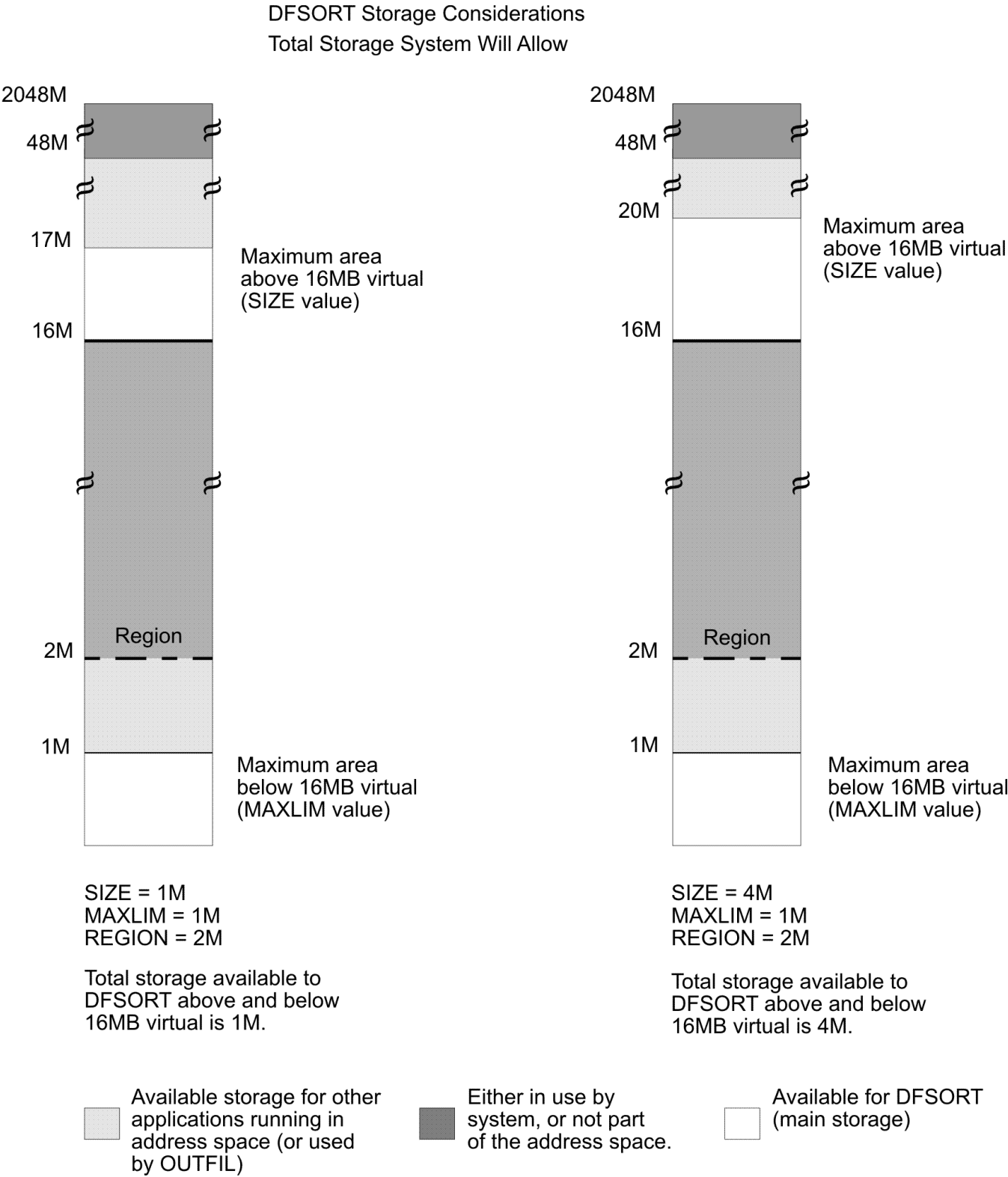


Figure 6. Storage comparisons for SIZE and REGION

TMAXLIM

The TMAXLIM option applies only when SIZE=MAX is in effect. It sets the upper limit to the total amount of storage both above and below 16 MB virtual.

You can increase or decrease the setting of this option from the default of 6291456 (6 MB), depending on the requirements of your site. Lowering TMAXLIM below 6 MB, can degrade performance for some applications. This performance degradation can become more pronounced and affect more applications the more TMAXLIM is lowered below 6 MB.

The total storage, shown in message IEF374I, might be more than TMAXLIM. However, after determining the storage available, DFSORT uses only the amount of storage specified by TMAXLIM. See [Figure 7 on page 20](#) for examples of how TMAXLIM affects DFSORT storage.

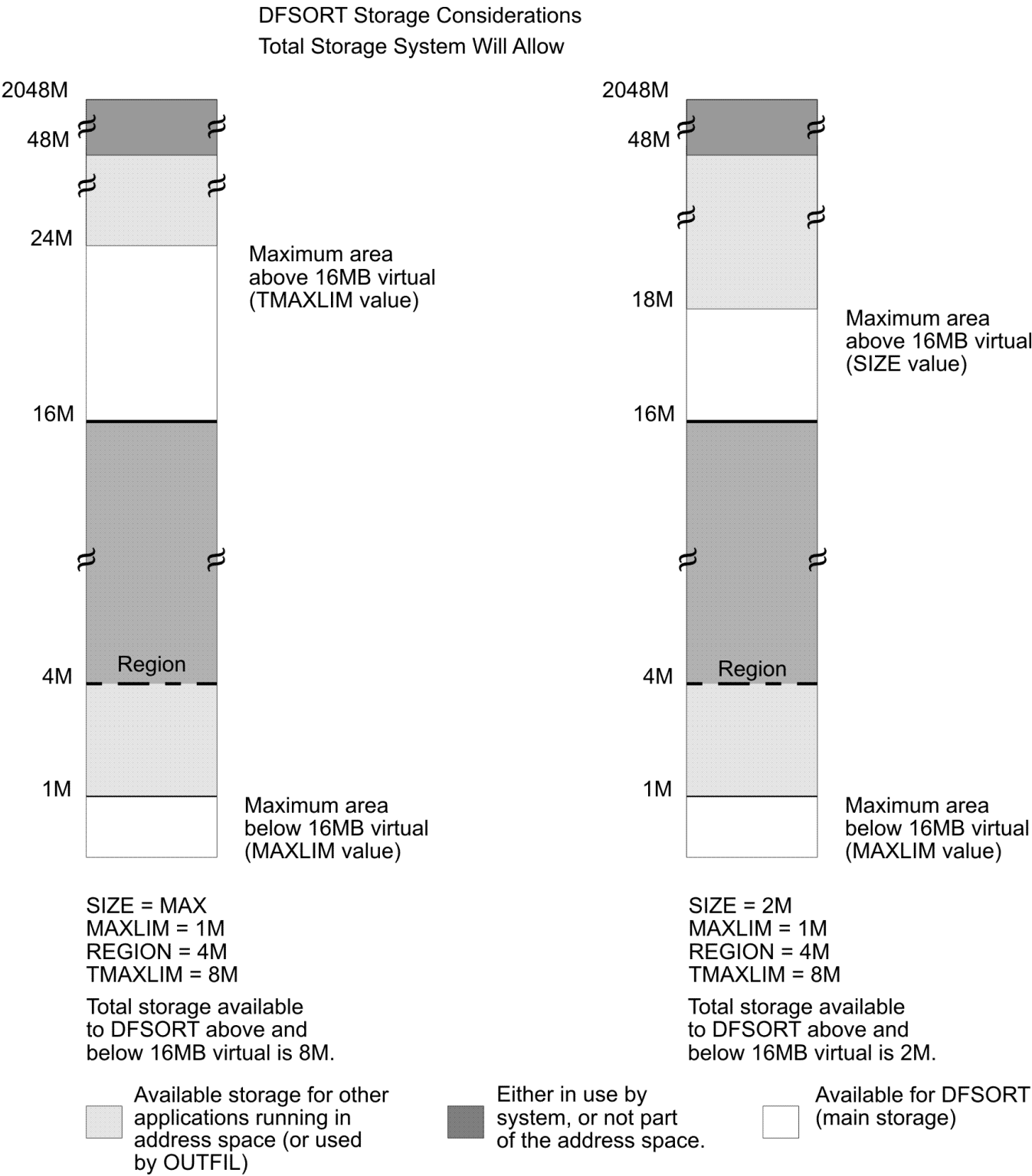


Figure 7. Effect of the TMAXLIM option on storage

TMAXLIM is ignored when SIZE is not equal to MAX.

MAXLIM

If MAXLIM is greater than the REGION size, DFSORT uses storage below 16 MB virtual only up to the size of REGION plus OVERRGN. You can increase or decrease the setting of this option from the default of 1048576 bytes (1 MB), depending on the requirements of your site. Lowering MAXLIM

below 1 MB can degrade performance for some applications. This performance degradation can become more pronounced and affect more applications the more MAXLIM is lowered 1 MB. See [Figure 8 on page 22](#) for examples of how MAXLIM affects DFSORT storage.

Normally, DFSORT uses less than 1 MB of storage below 16 MB virtual, so this option might have no effect. However, if your application uses E15 or E35 user exits, DFSORT attempts to use storage up to the MAXLIM value below 16 MB virtual.

Note: The MAXLIM limit is not observed for a copy or Blockset merge application when the use of more storage below 16 MB virtual should improve performance.

DFSORT Storage Considerations Total Storage System Will Allow

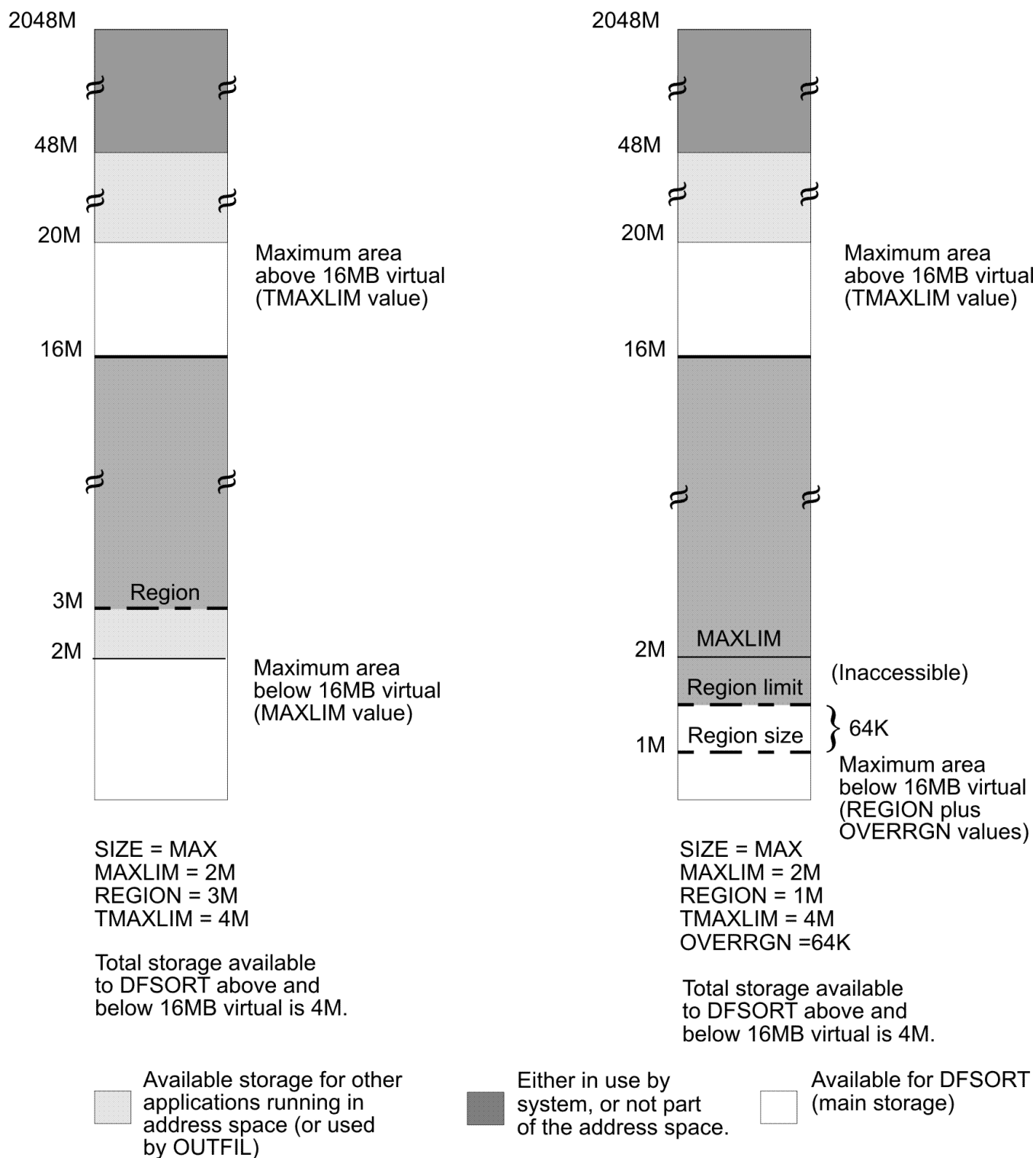


Figure 8. Effects of MAXLIM Option on Storage

When REGION is greater than MAXLIM, DFSORT uses storage below 16 MB virtual up to the value of MAXLIM (or, in some cases, for copy and merge applications, up to the lesser of REGION or SIZE).
When REGION is less than MAXLIM, DFSORT uses storage up to the value of REGION plus OVERRGN.

RESALL

RESALL reserves storage below 16 MB virtual for system use, when SIZE=MAX is in effect. The amount for RESALL is released below 16 MB virtual, and an equivalent amount can then be taken above 16 MB virtual.

The default of 4096 bytes (4 KB) is usually sufficient, but you can increase the setting of this option if necessary.

RESALL can be used to reserve storage for user exit routines. However, because the storage reserved by RESALL is always reserved when SIZE=MAX is in effect, and is reserved for all DFSORT processing phases, you should not set the value higher than is required for your average application. See [Figure 9 on page 24](#) for an example of how RESALL affects DFSORT storage.

DFSORT Storage Considerations
Total Storage System Will Allow

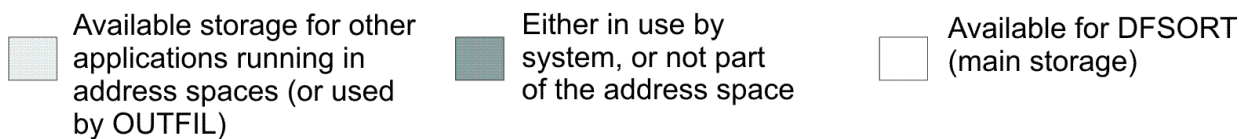
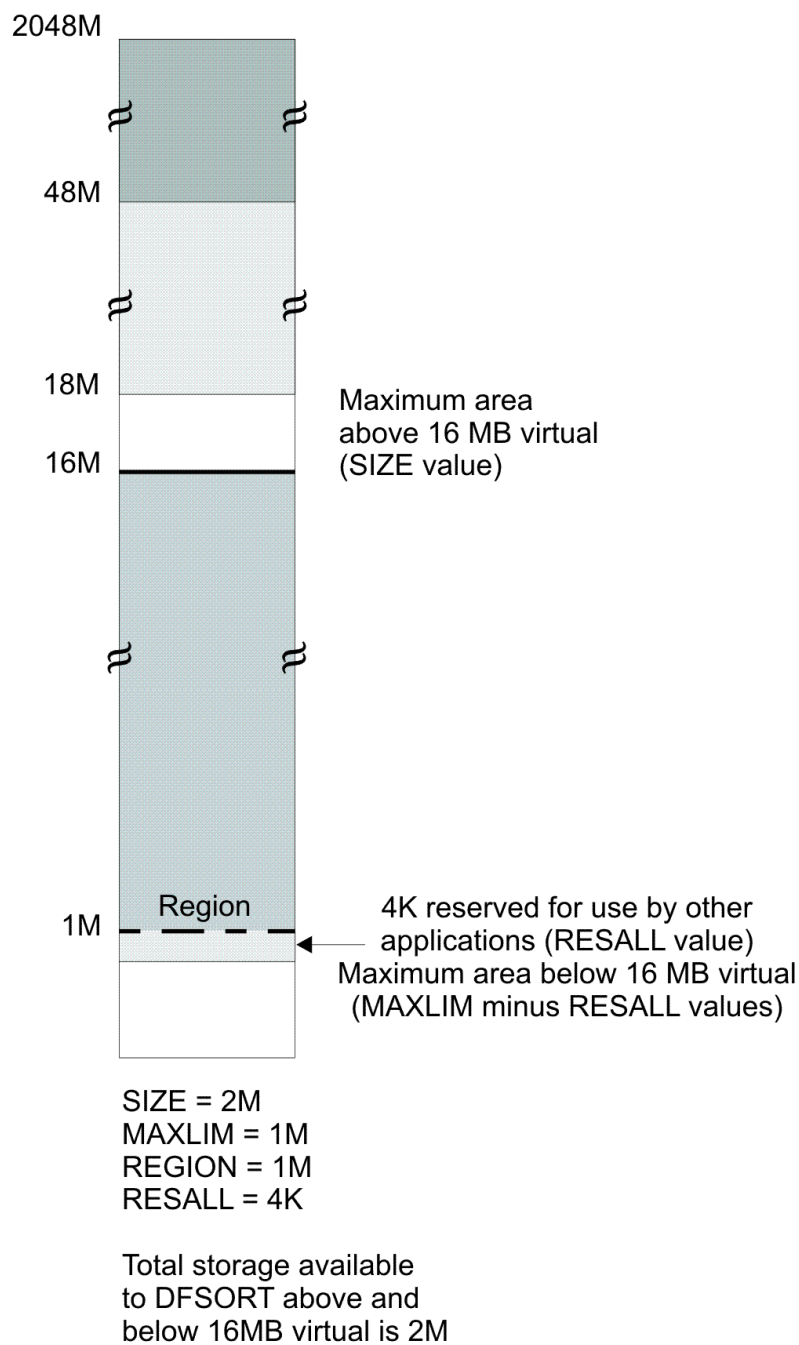


Figure 9. Effects of RESALL Option on Storage

RESINV

RESINV reserves storage below 16 MB virtual for the invoking program, when DFSORT is program-invoked and SIZE=MAX is in effect. RESINV storage is released in the same way as RESALL storage, and, if both are specified, the storage released is the sum of their values.

The default for RESINV is 0 bytes.

If invoking programs or their user exits require storage for data buffers that is not available otherwise, a RESINV value of 16384 (16 KB) is usually sufficient, although a larger value might be required in certain cases. However, because the storage reserved by RESINV is always reserved when DFSORT is program-invoked and SIZE=MAX is in effect, and is reserved for all DFSORT processing phases, you should not set the value higher than is required for your average application. The concepts illustrated in [Figure 9 on page 24](#) for RESALL also apply to RESINV.

ARESALL

The ARESALL option specifies the amount of storage to be reserved for system use above 16 MB virtual. ARESALL works the same way as RESALL, but the storage reserved by ARESALL is always above 16 MB virtual. This option is normally not needed because of the large amount of storage available above 16 MB virtual.

The default for ARESALL is 0 bytes.

ARESINV

ARESINV is similar to RESINV, except that it applies only to storage above 16 MB virtual. The ARESINV option is used only when DFSORT is program invoked. It specifies the amount of storage to be reserved for the invoking program above 16 MB virtual. This option is normally not needed because of the large amount of storage available above 16 MB virtual.

The default for ARESINV is 0 bytes.

OVERRGN

OVERRGN specifies the additional storage DFSORT should try to obtain above the REGION. The system's default region limit allows 64 KB of additional storage to be obtained; this default can be changed in an IEFUSI exit.

The default and recommended setting for this option for directly invoked jobs is 65536 bytes (64 KB). To avoid problems caused by incorrect use of the RESALL and RESINV options, the default setting for program invoked jobs is 16384 bytes (16 KB).

Recommendation: Increase the OVERRGN value for program invoked applications to 65536 bytes, if your site uses RESALL and RESINV correctly or is installing DFSORT for the first time. This is especially important if you are running benchmarks.

Tailoring storage for Hipersorting, memory object sorting and dataspace sorting

Hipersorting, memory object sorting and dataspace sorting are all methods of exploiting central storage to reduce work data set I/O and improve DFSORT efficiency. DFSORT selects the method which best fits the characteristics of the sort being executed and the available resources at the time of execution. You can tailor the way DFSORT uses storage for Hipersorting, memory object sorting, and dataspace sorting applications to meet the requirements of your site. DFSORT provides several options to control the use of Hipersorting, memory object sorting, and data space sorting. This section describes the default settings you can establish for these installation options (see [Chapter 4, “Changing the installation defaults,”](#) on [page 43](#)).

Hipersorting is a DFSORT feature that uses Hiperspaces as a way to store intermediate data in central storage instead of using disk work data sets. Central storage is used by Hipersorting applications to back the data written to their Hiperspaces. DFSORT will not write new data to a Hiperspace unless there is available storage to back it.

Memory object sorting is a DFSORT feature that uses memory objects as a way to store intermediate data. A memory object is a data area in virtual storage that is allocated above the bar and backed by central storage.

Dataspace sorting is a DFSORT feature that uses data space as a way to store intermediate data in central storage. A data space is a large (up to 2 GB) contiguous area of virtual storage that is backed by central storage.

Available storage is the storage used to back new Hiperspace, memory object, or dataspace data and consists of two types:

1. Free storage is storage that is not being used by any application.
2. Old storage is storage that is being used by other applications, but whose data has been unreferenced for a sufficiently long period such that the system considers it eligible to be paged out to auxiliary storage to make room for new Hiperspace, memory object or dataspace data.

The following DFSORT installation options are available to control the use of Hiperspace, memory object, and data space storage:

EXPMAX

Specifies the maximum total amount of available storage to be used at any one time by all Hipersorting, memory object sorting, and dataspace sorting applications

EXPOLD

Specifies the maximum total amount of old storage to be used at any one time by all Hipersorting, memory object sorting, and dataspace sorting applications

EXPRES

Specifies the minimum amount of available storage to be reserved for use by non-Hipersorting, non-memory object sorting, and non-data space sorting applications

HIPRMAX

Specifies the maximum amount of Hiperspace to use for a single Hipersorting application

TUNE

Specifies how DFSORT should manage the use of central storage by concurrent sorting applications.

MOSIZE

Specifies the maximum amount of memory object storage to use for a single memory object sorting application

DSPSIZE

Specifies the maximum amount of data space storage to use for a single dataspace sorting application.

At run time, you can override your HIPRMAX, MOSIZE, or DSPSIZE installation default with:

- An ICEIEXIT routine
- An EXEC statement
- An OPTION program control statement
- An EFS program.

The ICEIEXIT routine is described in [Chapter 5, “Controlling resources using ICEIEXIT,” on page 121](#). The EXEC statement, the OPTION statement, and using an EFS program are described in [z/OS DFSORT Application Programming Guide](#).

Anything that limits the amount of Hiperspace an application can create also limits how much storage that application can use. Hiperspace, memory object, and data space usage are primarily limited by a system default that can be changed in an IEFUSI exit.

Hiperspace usage by individual DFSORT applications can be limited with the HIPRMAX installation option. While HIPRMAX cannot override the system limits, it specifies the maximum amount of Hiperspace (within those limits) to be used by a single Hipersorting application.

Data space usage by individual DFSORT applications can be limited with the DSPSIZE installation option. While DSPSIZE cannot override the system limits, it specifies the maximum amount of data space to be used by a single dataspace sorting application.

Anything that limits the size of a memory object an application can allocate also limits how much storage that application can use. You can set a limit on usage of memory objects for a specific application in JCL through the MEMLIMIT option on the JOB or EXEC statement (see [z/OS MVS JCL Reference](#) for more details).

Memory object usage by individual DFSORT applications can be limited with the MOSIZE installation option. While MOSIZE cannot override the system limits, it specifies the maximum size of a memory object (within these limits) to be used by a single memory object sorting application.

Since central storage is a system resource, it is important to consider DFSORT's use of storage in light of total system activity, especially that of other concurrent Hipersorting applications. Throughout a Hipersorting run, DFSORT determines both the available storage and the amount of storage needed by other concurrent Hipersorting applications. Based on this information, DFSORT can decide to switch from using Hiperspace to using disk work data sets if there is insufficient available storage to back new Hiperspace data.

System-wide storage usage by one or more concurrent Hipersorting, memory object sorting, or database sorting applications is limited by the EXPMAX, EXPOLD, EXPRES, and TUNE installation options. EXPMAX limits the total amount of available storage used by all Hipersorting, memory object sorting, and data space sorting applications in a system. EXPOLD limits the total amount of old storage used by all Hipersorting, memory object sorting, and dataspace sorting applications in a system. EXPRES reserves a specified amount of available storage for use by non-Hipersorting, non-memory object sorting, and non-dataspace sorting applications. TUNE controls how DFSORT balances the use of available central storage by concurrent sorting applications. DFSORT can allocate storage in increments to balance central storage usage of concurrent sorting applications and dynamically adjust storage controls based on available resources, or it can allow each sort to allocate all it needs on a "first come, first served" basis.

The following are general guidelines for setting the DFSORT installation options that limit Hiperspace, memory object, and data space usage:

1. Avoid setting the defaults to anything other than HIPRMAX=OPTIMAL, MOSIZE=MAX and DSPSIZE=MAX at installation time. It is more efficient to limit total storage usage by all Hipersorting, memory object sorting, and dataspace sorting applications (through EXPMAX, EXPOLD, and EXPRES) than it is to impose a limit on each individual sorting application.
2. When there is a sudden increased need for storage by non-DFSORT applications, a small value for EXPRES can help to ensure there is sufficient available storage to meet the demand.
3. Setting EXPOLD to 0 or to a small percent of the configured storage is recommended if you are unsure whether your auxiliary storage paging subsystem is adequately sized to support an increased number of page outs from central storage. If your paging subsystem is sized to meet the peak needs of your workload, then EXPOLD should be set to the default value of 50%.
4. If you want to restrict DFSORT to using no more than a specified amount of storage for all Hipersorting, memory object sorting, and dataspace sorting activity, then EXPMAX is the only way to accomplish it with DFSORT options. This is a much better solution than trying to use the system to do storage isolation for DFSORT applications, because EXPMAX still allows full use of storage by non-DFSORT applications.
5. Use the shipped default of TUNE=STOR to allow DFSORT to dynamically adjust controls as available resources change. This will cause DFSORT to be more conservative in reducing disk work space allocations to compensate for potential changes in available central storage that could cause an unexpected increase in the disk work space requirements. If available disk work space is constrained, consider using TUNE=DISK which will cause each sort to obtain all the storage it needs on a "first come, first served" basis but allow DFSORT to be more aggressive in reducing disk work allocations based on expected central storage usage.
6. Each of the installation options EXPMAX, EXPOLD, EXPRES, TUNE, HIPRMAX, MOSIZE and DSPSIZE fulfills a different role, and each works somewhat independently of the others. Setting EXPMAX=MAX, EXPOLD=MAX, EXPRES=0, HIPRMAX=OPTIMAL, MOSIZE=MAX and DSPSIZE=MAX (the defaults)

permits the maximum amount of Hipersorting, memory object sorting and dataspace sorting subject to available storage and other concurrent DFSORT sorting activity.

Setting EXPMAX=0 or EXPRES=MAX disables Hipersorting, memory object sorting and dataspace sorting.

Setting HIPRMAX=0 disables only hipersorting.

Setting MOSIZE=0 disables only memory object sorting.

Setting DSPSIZE=0 disables only dataspace sorting.

HIPRMAX, MOSIZE and DSPSIZE can be overridden at run time. In general, setting EXPMAX, EXPOLD, HIPRMAX, MOSIZE and DSPSIZE to large values (and EXPRES to a small value) allows a generous amount of DFSORT storage exploitation, while setting EXPMAX, EXPOLD, HIPRMAX, MOSIZE or DSPSIZE to a small value (or EXPRES to a large value) allows only a limited amount of DFSORT storage exploitation.

System-managed storage

See *z/OS DFSORT Tuning Guide* for important information about system-managed storage, ACS routines, and why software that reduces disk space allocations should not be allowed to reduce DFSORT work data set space.

Installing DFSORT in permanent libraries

If you have installed DFSORT in private libraries for testing and are now ready to install DFSORT in permanent libraries, repeat the installation steps in the *z/OS Install/Migration* page (www.ibm.com/systems/z/os/zos/installation), this time modifying the sample jobs to specify your permanent libraries.

Replacing IEBGENER with the ICEGENER facility

You can achieve more efficient processing for jobs set up to use the IEBGENER system utility by using the ICEGENER facility of DFSORT. Qualifying IEBGENER jobs are processed by the equivalent (though not identical), but more efficient, DFSORT copy function. If, for any reason, the DFSORT copy function cannot be used (for example, when IEBGENER control statements are specified), control is automatically transferred to the IEBGENER system utility. DFSORT transfers control to IEBGENER using the name for IEBGENER from the invocation installation environment (ICEAM2 or ICEAM4) or time-of-day installation environment (ICETD1-4) used for the run. The ICEGENER facility can be used either for selected IEBGENER jobs or, automatically, for all IEBGENER jobs.

ICEGENER, like IEBGENER, will use an SDB=value parameter you supply using PARM='SDB=value', when appropriate. The valid SDB=value parameters are SDB=LARGE, SDB=YES, SDB=SMALL, SDB=INPUT and SDB=NO, as explained in *z/OS DFSORT Application Programming Guide*. If you supply an invalid SDB=value parameter, ICEGENER will transfer control to IEBGENER, which will terminate due to the invalid parameter. If you do not supply an SDB=value parameter, ICEGENER will use your site's DFSORT installation default for SDB, when appropriate (the IBM-supplied default is SDB=INPUT). If ICEGENER transfers control to IEBGENER, IEBGENER will use the SDB=value parameter you supply, if any, or its normal default for SDB.

ICEGENER will also recognize DFSORT parameters other than SDB=value you supply using PARM='parameter' that are valid on DFSORT's OPTION statement as explained in *z/OS DFSORT Application Programming Guide*. However, IEBGENER does not recognize any parameters other than the valid SDB=value forms, so if DFSORT must transfer control to IEBGENER, IEBGENER will not recognize DFSORT's parameters and will terminate. Likewise, if you supply a DFSORT parameter using PARM='parameter' that is not valid on DFSORT's OPTION statement, DFSORT will transfer control to IEBGENER and IEBGENER will terminate due to the invalid parameter.

For example, if you specify:

```
//S1 EXEC PGM=ICEGENER,PARM='SIZE=2000,MAINSIZE=2000K'
```


ICEGENER will accept SIZE=2000 and MAINSIZE=2000K as valid DFSORT OPTION parameters, which specify an exact file size of 2000 records and a limit of 2000K bytes of storage, respectively. If DFSORT copy can be used, these parameters are used. But if DFSORT must transfer control to IEBGENER, IEBGENER terminates because it treats SIZE=2000 and MAINSIZE=2000K as invalid options.

As another example, if you specify:

```
//S2 EXEC PGM=ICEGENER,PARM='SIZE=2000K'
```

ICEGENER will treat SIZE=2000K as an invalid DFSORT OPTION parameter and will transfer control to IEBGENER which will terminate because it treats SIZE=2000K as an invalid parameter.

Thus, you can pass PARM parameters to ICEGENER that are valid as DFSORT OPTION parameters, but you must be aware that if ICEGENER transfers control to IEBGENER, those parameters will cause IEBGENER to terminate. PARM parameters that are not valid as DFSORT OPTION parameters (even if they are valid as DFSORT PARM parameters) will cause ICEGENER to transfer control to IEBGENER which will terminate.

ICEGENER can transfer control to IEBGENER due to DFSPARM or SORTCNTL statement errors or other errors detected by DFSORT. Therefore, you should use DFSORT copy rather than ICEGENER for applications that cannot be handled by ICEGENER. For example, consider an ICEGENER application with an INCLUDE statement in DFSPARM. While ICEGENER can process the INCLUDE statement, IEBGENER cannot. If ICEGENER transfers control to IEBGENER for any reason, IEBGENER will complete with a return code of 0, but the results will not be as intended because the INCLUDE statement will be ignored by IEBGENER. DFSORT copy should be used instead so the INCLUDE statement will not be ignored.

However, if you know that ICEGENER will use DFSORT copy, you can use a DFSPARM data set with ICEGENER to pass control statements and parameters to DFSORT. For example, if you specify:

```
//DFSPARM DD *  
    OPTION SPANINC=RC0  
/*
```

and ICEGENER uses DFSORT copy, any incomplete spanned records DFSORT detects in a variable spanned input data set are eliminated.

ICEGENER can benefit from running DFSORT resident through improved performance, allowing more jobs to use DFSORT copy, and eliminating storage-related abends when storage is limited.

You can invoke ICEGENER in one of the following ways:

- For selective use, substitute the name ICEGENER (or the alias SORTGENR) for the name IEBGENER in any jobs you choose.
- To make ICEGENER the system's default "IEBGENER", install an SMP/E usermod to create an alias of "IEBGENER" for ICEGENER. With slight modification, you can use the ICEGAREC and ICEGAAPP sample jobs in SICESAMP to create this alias. Then place the ICEGENER load module ahead of IEBGENER in the system's search order for programs. For example, you can place the ICEGENER load module with its new alias in LPA. All jobs and programs that call IEBGENER will then execute ICEGENER instead. If ICEGENER needs to call IEBGENER, it will do so automatically. For information about placing modules in LPA and Dynamic LPA, and about placing libraries ahead of the LINKLIB library in the link list, see [z/OS MVS Initialization and Tuning Reference](#).

Note:

1. Whether ICEGENER is invoked from a program or not, DFSORT is invoked from ICEGENER using an extended parameter list. Therefore, the installation options for the program-invoked environment (that is, ICEAM2 or ICEAM4 or an ICETDx environment activated for the ICEAM2 or ICEAM4 environment) apply, and SORTCNTL or DFSPARM can be used to provide additional control statements for the copy application, such as OPTION. However, ICEGENER can transfer control to IEBGENER due to DFSPARM or SORTCNTL statement errors or other errors detected by DFSORT. Therefore, DFSORT copy should be used directly rather than ICEGENER if DFSORT processing statements such as INCLUDE, OUTREC, SUM and so on are required.

2. For most error conditions that prevent the use of DFSORT copy, control is transferred to the IEBGENER system utility. DFSORT messages are not printed unless a SORTDIAG DD statement is supplied. Use the SORTDIAG DD statement to determine why DFSORT copy was not used.
3. If DFSORT copy is used, its operation and messages are equivalent to a directly called DFSORT copy application. If an unrecoverable error is encountered (such as an I/O error), a DFSORT return code of 16 is changed by ICEGENER to a return code of 12 to emulate the return code from a failing IEBGENER application.
4. DFSORT copy can perform some functions not provided by IEBGENER, such as certain padding and truncation operations. ICEGENER processing is not identical to IEBGENER processing in all cases, because DFSORT copy uses methods to enhance performance (EXCP, for example) that IEBGENER does not use.
5. In some cases, IEBGENER terminates when the SYSUT2 LRECL is different from the SYSUT1 LRECL. ICEGENER takes one of three actions depending on the value you select for GNPAD (for LRECL padding) or GNTRUNC (for LRECL truncation).

If you want ICEGENER to transfer control to IEBGENER when the SYSUT2 LRECL is larger than the SYSUT1 LRECL, use GNPAD=IEB. If you want ICEGENER to handle LRECL padding, use GNPAD=RC0 (the supplied default) or GNPAD=RC4.

If you want ICEGENER to transfer control to IEBGENER when the SYSUT2 LRECL is smaller than the SYSUT1 LRECL, use GNTRUNC=IEB. If you want ICEGENER to handle LRECL truncation, use GNTRUNC=RC0 (the supplied default) or GNTRUNC=RC4.

6. For a call to ICEGENER, or to IEBGENER as an alias for ICEGENER, register 1 must point to a valid parameter list consisting of three addresses as follows:

Bytes 1 through 4
Address of the Option List
Address of the Alternate DDname List
Address of the Page Number List

Methods of calling ICEGENER which generate a valid parameter list will allow ICEGENER to use DFSORT's copy feature, whereas methods of calling ICEGENER which generate an invalid parameter list will cause ICEGENER to transfer control to IEBGENER. For example:

call *(icegener)

on the TSO command line generates a valid parameter list, whereas:

icegener

on the TSO command line generates an invalid parameter list.

Verifying ICEGENER installation

You can use the sample job ICEGENJ supplied with the licensed program to verify the installation of the ICEGENER facility. Before performing the verification, you should review the comments in the sample job.

Examine the results of each step in the verification job to ensure that the data has been copied correctly and that the SYSOUT output was produced by the appropriate program (DFSORT copy or the IEBGENER utility).

If message ICE054I appears, showing a nonzero number of records in and records out, then DFSORT did perform a copy application. You should be able to recognize the difference between the output produced by IEBGENER and any output produced by DFSORT because there is a distinct difference in the style between the two types of output.

The following steps each exercise a function of copying. Each step copies from the generated data set to a printed output (JES spool) file. If you see the data printed, you know a copy function was performed.

1. The job step called GEN creates (using the IEBDG utility program) the data to be copied by the rest of the steps.
2. The GCOPY1 step copies and prints the data created by GEN. Because the EXEC statement specifies PGM=ICEGENER, the ICEGENER facility is used and it selects DFSORT copy to perform the copy application. If the application runs correctly, the DFSORT messages will indicate that DFSORT copied 360 records.
3. The BCOPY1 step copies and prints the data in a different manner to verify that the different invocation path works. Because the EXEC statement specifies PGM=IEBGENER, the method used to perform the copy application depends on how you have installed the ICEGENER facility:
 - If you chose selective use of ICEGENER, the IEBGENER utility performs the copy application.
 - If you chose automatic use of ICEGENER, the ICEGENER facility selects DFSORT copy to perform the copy application.
4. The GCOPY2 step also copies and prints the data. Because the EXEC statement specifies PGM=ICEGENER, the ICEGENER facility is used. Because the SYSIN data set contains IEBGENER control statements, ICEGENER selects the IEBGENER utility to perform the copy application.

Either of the following results indicates that the ICEGENER facility was not installed correctly (see instructions described previously in this section):

- You receive an ABEND 806 and message CSV003I (for IEBGENR) on the console. Ensure that you have installed the DFSMS or DFP PTF that creates the IEBGENR alias for IEBGENER correctly.
- You receive an ABEND 163 and message ICE163A on the console. Ensure that you use GENER=IEBGENR (the default). If you have an alias of IEBGENR for ICEGENER, remove it.

Discontinuing use of ICEGENER as a replacement for IEBGENER

If you made ICEGENER the system's default "IEBGENER" and you want to discontinue its use for this purpose, RESTORE the SMP/E usermod you used to create the "IEBGENER" alias of ICEGENER. With slight modification, you can use the ICEGAAPP sample job in SICESAMP to do this RESTORE.

Part 2. Customizing DFSORT

Chapter 3. Planning for customization

This chapter describes how you can use supplied services and facilities to customize DFSORT to the needs of your site. Customization is the task of enhancing or extending a program by using built-in facilities. This chapter helps you decide what to customize and shows you how to do the customization; the rest of the chapters in this document discuss these topics in more detail. It describes planning considerations for:

- SORT and SORTD cataloged procedures
- Changing installation defaults
- Cultural Environment
- Extended Function Support (EFS) programs
- Initialization and termination exits
- Message texts.

Customizing the standard DFSORT procedures

Two cataloged procedures, SORT and SORTD, are delivered with DFSORT. These standard procedures are for running in as many environments as possible. Sample jobs ICEPROC and ICEPROCD install SORT and SORTD into SYS1.PROCLIB during installation.

These procedures can simplify the use of DFSORT by making the libraries accessible and by reducing the number of DD statements and parameters that must be specified in each application.

You should review the JCL in these procedures and modify them to meet your system requirements. For example, you might want to add parameters on the EXEC statement, modify the DD statements to allocate your own library names, delete DD statements your installation does not normally use, or add DD statements for data sets your installation uses regularly.

Specifically, consider the following modifications:

- You could specify EXEC PARM parameters to be passed to the DFSORT program. However, many of these parameters can instead be set as installation defaults .
- If DFSORT's libraries are not in the link list and/or LPA list, you could specify a STEPLIB DD for these libraries in the procedures.
- If you use private libraries for the Conventional technique modules, you could specify a SORTLIB DD for these libraries in the procedures. You can also specify a SORTLIB DD in the procedures if you use dynamic linkedit of user exits for tape work data set sort or Conventional merge applications.
- If your installation uses COBOL E15 or COBOL E35 exit routines frequently, you could concatenate the Language Environment® run-time library to the STEPLIB DD statement.
- The SYSPRINT, SYSLMOD, SYSLIN, and SYSUT1 DD statements in the SORTD procedure are used only for dynamic link-edit. They can be deleted if your installation uses only pre-link-edited user exit routines with DFSORT.
- If you are allocating temporary work data sets in the procedures, see the explanation of WRKREL and WRKSEC in [“Installation options”](#) on page 56.
- If you want the DFSORT diagnostic messages printed for each run, add a SORTDIAG DD DUMMY statement to the procedures. SORTDIAG is normally used for debugging only.

Changing the DFSORT installation defaults

The IBM-supplied installation defaults provided are given in [Chapter 4, “Changing the installation defaults,”](#) on page 43. These defaults are in effect if you install DFSORT without specifying your own defaults.

Any or all of the defaults can be changed at any time after DFSORT is installed either with ICEPRMxx members of concatenated PARMLIB or with the ICEMAC macro. You might want to first install DFSORT with the given defaults, then tailor them to your requirements after you have run DFSORT for a while.

You can list the installation defaults currently in effect at any time using the ICETOOL DEFAULTS operator, as shown in [Figure 19 on page 113](#).

You can override the IBM-supplied installation defaults in several ways. You can use ICEPRMxx members of PARMLIB or the ICEMAC macro to change the defaults for the eight different environments provided (ICEAM1-4 and ICETD1-4) (see Chapter 4, “Changing the installation defaults,” on page 43 for details). Some of the defaults can also be overridden by using an EFS program or an ICEEXIT routine.



Attention: You must use the ICEAM1-4 and ICETD1-4 modules shipped with this version of DFSORT, or recompiled with the ICEMAC macro shipped with this release of DFSORT. You cannot use ICEAM1-4, ICETD1-4 or ICEMAC from a previous release of DFSORT.

Many of the defaults can be overridden by the application programmer at run-time by use of the JCL EXEC statement, a parameter list, or program control statements. For full override details, see [z/OS DFSORT Application Programming Guide](#)

For additional information on ICEPRMxx members of PARMLIB and how to code and activate them, see “Using ICEPRMxx members in PARMLIB” on page 46.

For additional information on the ICEMAC macro and instructions on how to code and use it, see “Using the ICEMAC macro” on page 51.

Cultural environment considerations

DFSORT's collating behavior can be modified according to your cultural environment. Your cultural environment is defined to DFSORT using the X/Open locale model. A *locale* is a collection of data grouped into categories that describes the information about your cultural environment.

The collate category of a locale is a collection of sequence declarations that defines the relative order between collating elements (single character and multi-character collating elements). The sequence declarations define the collating rules.

The cultural environment is established by selecting the active locale. The active locale affects the behavior of locale sensitive functions. In particular, the active locale's collating rules affect DFSORT's SORT, MERGE, INCLUDE, and OMIT processing as follows:

- Sort and Merge

DFSORT will produce sorted or merged records for output according to the collating rules defined in the active locale. This provides sorting and merging for single- or multi-byte character data based on defined collating rules which retain the cultural and local characteristics of a language.

- Include and Omit

DFSORT will include or omit records for output according to the collating rules defined in the active locale. This provides inclusion or omission for single- or multi-byte character data based on defined collating rules which retain the cultural and local characteristics of a language.

The DFSORT option, LOCALE, specifies whether locale processing is to be used and, if so, designates the active locale. Only one locale can be active at a time for any DFSORT application. The LOCALE option can be specified:

- as an installation option
- in the PARM field of the EXEC statement
- in the DFSPARM data set
- on the OPTION control statement of an extended parameter list

To use an IBM-supplied locale, DFSORT must have access to the Language Environment run-time library. This library, SCEERUN, contains dynamically loadable routines, including the IBM-supplied locales. You can place the SCEERUN library in the link list, place heavily-used modules in LPA, or use a JOBLIB or

STEPLIB. For more information about making the SCEERUN library available for use, see [z/OS Language Environment Customization](#).

Locale processing for DFSORT's SORT, MERGE, INCLUDE, and OMIT functions can improve performance relative to applications which perform pre- and/or post-processing of data to produce the desired collating results. However, locale processing should only be used when required since it can show degraded performance relative to collation using character encoding values of the code page defined for your data.

Unicode

The Unicode Standard provides a single character set that covers the languages of the world, and a small number of machine-friendly encoding forms and schemes to fit the needs of existing applications and protocols. It is designed for best interoperability with both ASCII and ISO-8859-1, the most widely used character sets, to make it easier for Unicode to be used in applications and protocols.

A special number is assigned to every single character. You can refer to [The Unicode Consortium \(www.unicode.org\)](#) to find out more about Unicode. You can find the code charts, which show you which number is assigned to which character within the Unicode standard on the [Unicode Character Code Charts \(www.unicode.org/charts\)](#). If you prefer to have a list of all available characters on your workstation so that you do not have to be online every time you want to find a specific character in the Unicode code page, you can download the [Unibook Character Browser \(www.unicode.org/unibook\)](#).

Currently, the following three forms of Unicode encoding are supported:

UTF8:

Unicode **T**ransformation **F**ormat in 8 bits that uses 1 to 4 bytes, depending on the character.

UTF16:

Unicode **T**ransformation **F**ormat in 16 bits that uses either 2 or 4 bytes to represent a character.

UTF32:

Unicode **T**ransformation **F**ormat in 32 bits.

Unicode Environment Considerations

DFSORT's Unicode Collation behavior can be modified according to your cultural environment. A new COLLKEY parameter allows specifying the Unicode standard character suite for national collation sequences. Your Unicode environment is defined to DFSORT using the COLLKEY=value

The COLLKEY option can be specified:

- as an installation option
- in the PARM field of the EXEC statement
- in the DFSPARM data set
- on the OPTION control statement of an extended parameter list

Collation supports customization, which means that collation service might behave according to some specific collation rules. Collation rules can be specified using a Locale or User Collation Rules (UCR).

The active collation rules affects the behavior of locale-sensitive functions. In particular, the active locale's collating rules affect DFSORT's SORT and MERGE processing as follows:

Sort and Merge

DFSORT produces sorted or merged records for output according to the collating rules defined in the active locale. This provides sorting and merging for single-or multi-byte character data based on defined collating rules that retain the cultural and local characteristics of a language.

For supported Locales settings, see Appendix E. Locales for collation in [z/OS Unicode Services User's Guide and Reference](#).

Invoking an EFS routine automatically

You can write and activate your own Extended Function Support (EFS) programs to do tasks such as those described in the *z/OS DFSORT Application Programming Guide*. There is an interface available in DFSORT to accept and use EFS programs that you write yourself. IBM does not supply an EFS program with DFSORT.

To activate an EFS program, you must use the EFS=name option. This option can be specified:

- as an installation option
- in the PARM field of the EXEC statement
- in the DFSPARM data set
- on the OPTION control statement of an extended parameter list.

The EFS program can reside in the same library as the DFSORT modules or in a separate library. The library with the EFS program must be accessible (for example, the library can be placed in the link list or LPA list). If an EFS program resides in LPA, it must be reentrant.

Using initialization and termination exits

You can use user-written, installation-wide initialization and termination exits (ICEIEXIT and ICETEXIT) to perform a variety of functions, such as overriding the options currently in effect and collecting statistical data. (The installation-wide initialization exit is subsequently referred to as an initialization exit or ICEIEXIT throughout this document; the installation-wide termination exit is referred to as a termination exit or ICETEXIT.)

You can write and install your own ICEIEXIT and ICETEXIT to do tasks such as those described in [Chapter 5, “Controlling resources using ICEIEXIT,” on page 121](#) and [Chapter 6, “Collecting statistical data,” on page 131](#).

Initialization exits

You can use sample jobs ICEIXREC and ICEIXAPP in the SICESAMP library to install an ICEIEXIT using SMP/E. The sample SMP/E usermod in ICEIXREC will place your ICEIEXIT in the SORTLPA library.

ICEIXREC is a sample ICEIEXIT. Another example of an ICEIEXIT is shown in [Figure 28 on page 128](#).

Choose the library you want your ICEIEXIT to reside in before you run ICEIXREC to RECEIVE the SMP/E usermod. Your ICEIEXIT can reside in the same library as the DFSORT modules or in a separate library. The library in which your ICEIEXIT resides must be accessible (for example, the library can be placed in the link list or LPA list). If your ICEIEXIT resides in LPA, it must be reentrant.

To change the library in which SMP/E will store your ICEIEXIT, change NAME=SortLPA to NAME=dsname on the EXEC statement of the IXLINK step in the JCLIN.

After you have received the SMP/E usermod, you can use the ICEIXAPP sample job to APPLY it. SMP/E will assemble and link your ICEIEXIT into the library you selected.

After you have applied the SMP/E usermod and made the library with your ICEIEXIT accessible, use the IEXIT=YES installation option to activate your ICEIEXIT, as described in [Chapter 4, “Changing the installation defaults,” on page 43](#).

Termination exits

You can use sample jobs ICEIXREC and ICEIXAPP in the SICESAMP library to install an ICETEXIT using SMP/E. The sample SMP/E usermod in ICEIXREC will place your ICETEXIT in the SORTLPA library.

Two examples of ICETEXITs are provided in [“ICETEXIT examples” on page 138](#).

Choose the library you want your ICETEXIT to reside in before you run ICEIXREC to RECEIVE the SMP/E usermod. Your ICETEXIT can reside in the same library as the DFSORT modules or in a separate library.

The library in which your ICETEXIT resides must be accessible (for example, the library can be placed in the link list or LPA list). If your ICETEXIT resides in LPA, it must be reentrant.

To change the library in which SMP/E will store your ICETEXIT, change NAME=SortLPA to NAME=dsname on the EXEC statement of the IXLINK step in the JCLIN.

After you have received the SMP/E usermod, you can use the ICEIXAPP sample job to APPLY it. SMP/E will assemble and link your ICETEXIT into the library you selected.

After you have applied the SMP/E usermod and made the library with your ICETEXIT accessible, use the TEXT=YES installation option to activate your ICETEXIT, as described in [Chapter 4, “Changing the installation defaults,”](#) on page 43.

Modifying message texts

You can change the text of DFSORT messages. For example, you might want to translate them into another language.

The text for DFSORT messages is shipped in member ICEMSGS of the AICEUSER distribution library and in member ICEMSGS of the SICEUSER target library.

To make changes to any or all DFSORT messages, you can use an SMP/E usermod modeled on sample job ICEMSREC in the SICESAMP library.

For the base product

You can modify the message texts in the ICEMSGS macro by installing an SMP/E usermod system modification, shipped in the ICEMSREC sample job, under SMP/E. However, you can modify the messages only for the Blockset and Peerage/Vale techniques. If you increase the length of a message past about 84 characters (subject to the restrictions of your system), the message can be truncated when written to the master console. A copy of the ICEMSGS macro appears in the AICEUSER and SICEUSER libraries.

The format of the ICEMSGS macro is:

```
MACRO
ICEMSGS
ICEMSET 1,'text1...'
ICEMSET 2,'text2...'
      .
      .
      .
END
```

Each message macro consists of a macro call name (ICEMSET), a message number, and a text string. The text string consists of fixed characters and insert fields. The insert fields supply variable information in these forms:

- Character information represented by **?** characters (one **?** for each character of the maximum-length field).
- Fixed-length numeric information represented by **%** characters (one **%** for each character of the maximum-length field). Leading zeros are printed.
- Variable-length numeric information represented by **!** characters (one **!** for each character of the maximum-length field). Leading zeros are not printed.

Coding the changes to ICEMSGS

To make changes to DFSORT messages, you must create an SMP/E usermod to modify the ICEMSGS macro. You can model your SMP/E usermod on sample job ICEMSREC, an IEBUPDTE update, which can be found in the SICESAMP library. Make sure that the modifications in your SMP/E usermod use appropriate sequence numbers relative to ICEMSGS in the SICEUSER library.

All changes must follow Assembler coding rules, including:

- Coding an ampersand (&) in a text string as two ampersands (&&).

- Coding an apostrophe (') in a text string as two apostrophes (').
- Using standard conventions for continuing macro calls.

Also, you must observe these restrictions:

- Message macros must not be added.
- Macro call names (ICEMSET) and message numbers must not be changed.
- Fixed characters can be changed, added, deleted, or relocated.
- Insert field characters (?, %, !) must not be changed, added, or deleted.
- Insert fields must not be made contiguous.
- Because the characters ?, %, and ! have special meanings in this context, you must not use them in new message texts.
- Insert fields can be relocated. However, the position of insert fields relative to each other must not be changed.
- Each text string must not exceed 110 bytes.
- If you increase the length of a text string past about 84 characters (subject to the restrictions of your system), a message written to the master console might be truncated.

Note: If your translation causes the message texts to expand by more than about 20 percent, you might cause the reassembly of one or more message CSECTs to fail.

Figure 10 on page 40 shows how the text of messages ICE134I and ICE142I can be translated into French.

ICE134I:

```
NUMBER OF BYTES SORTED: n
NOMBRE D'OCTETS TRIÉS: n
```

ICE142I:

```
ddname NOT FOUND - SYSOUT USED
ddname NON TROUVE - SYSOUT UTILISE
```

Figure 10. Suggested translations for two ICEMSGs messages

Figure 11 on page 40 shows the original text for the ICE134I and ICE142I messages in ICEMSGs:

```
.
.
.
ICEMSET 134, 'NUMBER OF BYTES SORTED: !!!!!!!!!!!!!!!!!!!!!!!'
.
.
.
ICEMSET 142, '???????? NOT FOUND - SYSOUT USED'
.
.
.
```

Figure 11. The two original ICEMSGs messages

Figure 12 on page 41 shows the changed text for the ICE134I and ICE142I messages in ICEMSGs:


```

.
.
.
ICEMSET 134,'NOMBRE D''OCTETS TRIES: !!!!!!!!!!!!!!!!!!!!!!!'
.
.
.
ICEMSET 142,'??????? NON TROUVE - SYSOUT UTILISE'
.
.
.

```

Figure 12. The two ICEMSGs messages changed to French

Receiving and applying SMP/E usermods

Two sample jobs are used to create an SMP/E usermod to modify your message texts. Modify these jobs to conform to your system requirements.

To RECEIVE your modifications to ICEMSGs, run ICEMSREC. To APPLY your modifications to ICEMSGs, run ICEMSAPP.

If you violate any of the rules for modifying messages, you render that message invalid, and an appropriate MNOTE message is issued when the message CSECTs are reassembled during the APPLY step. (Explanations of the MNOTE module errors are in [“Message module error messages”](#) on page 168.) You can recover any message that you render invalid by using the copy of ICEMSGs in the AICEUSER distribution library.

Correct the statement containing the error. Run SMP/E RESTORE and REJECT commands to remove the SMP/E usermod, and rerun the ICEMSREC and ICEMSAPP jobs to re-RECEIVE the corrected SMP/E usermod and APPLY it.

Chapter 4. Changing the installation defaults

DFSORT allows you to maintain separate sets of installation defaults for eight different environments as follows:

- JCL/ICEAM1: Batch direct invocation environment.
- INV/ICEAM2: Batch program invocation environment
- TSO/ICEAM3: TSO direct invocation environment.
- TSOINV/ICEAM4: TSO program invocation environment.
- TD1/ICETD1: First time-of-day environment.
- TD2/ICETD2: Second time-of-day environment.
- TD3/ICETD3: Third time-of-day environment.
- TD4/ICETD4: Fourth time-of-day environment.

The set of defaults used for a particular run is determined by the invocation environment for the run (ICEAM1-4), the time-of-day of the run (that is, day and time) and the defaults in effect for each installation environment (ICEAM1-4 and ICETD1-4).

Each invocation environment can specify time-of-day environments which can be used for that invocation environment when appropriate. If a time-of-day environment is not used, then the invocation environment will be used.

For example, you could set up the ICETD1 environment to use larger storage values, such as 48 MB for DSA, 8 MB for TMAXLIM, and so on. You could use these ICETD1 defaults for program-invoked batch runs on Saturday and Sunday from 8:00am to 4:59pm and use the regular ICEAM2 defaults for program-invoked batch runs during any other time.

More specifically, the eight installation environments are as follows:

- ICEAM1 (JCL) is the batch direct invocation installation environment. This set of defaults is used at run time when DFSORT is invoked directly (that is, not through programs) by batch jobs, provided that an enabled time-of-day installation environment (ICETDx) is not activated.
- ICEAM2 (INV) is the batch program invocation installation environment. This set of defaults is used at run time when DFSORT is invoked through batch programs, provided that an enabled time-of-day installation environment (ICETDx) is not activated.
- ICEAM3 (TSO) is the TSO direct invocation installation environment. This set of defaults is used at run time when DFSORT is invoked directly (that is, not through programs) by foreground TSO users, provided that an enabled time-of-day installation environment (ICETDx) is not activated.
- ICEAM4 (TSOINV) is the TSO program invocation installation environment. This set of defaults is used at run time when DFSORT is invoked through programs by foreground TSO users, provided that an enabled time-of-day installation environment (ICETDx) is not activated.
- ICETD1 (TD1) is the first time-of-day installation environment. This set of defaults is used at run time when it is activated for the time-of-day of the run, provided it is enabled by the invocation installation environment (ICEAMx) in effect.
- ICETD2 (TD2) is the second time-of-day installation environment. This set of defaults is used at run time when it is activated for the time-of-day of the run, provided it is enabled by the invocation installation environment (ICEAMx) in effect.
- ICETD3 (TD3) is the third time-of-day installation environment. This set of defaults is used at run time when it is activated for the time-of-day of the run, provided it is enabled by the invocation installation environment (ICEAMx) in effect.
- ICETD4 (TD4) is the fourth time-of-day installation environment. This set of defaults is used at run time when it is activated for the time-of-day of the run, provided it is enabled by the invocation installation environment (ICEAMx) in effect.

You can change any of the IBM-supplied DFSORT defaults in one of two ways:

1. With ICEPRMxx members in concatenated PARMLIB. This is the recommended method. You specify the defaults you want to change for any or all environments in one or more ICEPRMxx members in PARMLIB and use the supplied ICEOPT started task to activate those members. For example, to activate ICEPRM01 and ICEPRM05, you would use this START ICEOPT command:

```
START ICEOPT,ICEPRM=(01,05)
```

You can activate different ICEPRMxx members for different LPARs automatically at IPL time by including a START ICEOPT command in an appropriate COMMNDxx member in PARMLIB, or at any time by issuing a START ICEOPT command from the console.

ICEPRMxx changes to the installation defaults are merged with and override the specified or IBM-supplied ICEMAC defaults.

For additional information on ICEPRMxx members of PARMLIB and how to code and activate them, see [“Using ICEPRMxx members in PARMLIB” on page 46](#).

2. With the ICEMAC macro. You specify the defaults you want to change for each environment with an SMP/E usermod. ICEMAC changes to the defaults are merged with and override the IBM-supplied ICEMAC defaults.

For additional information on the ICEMAC macro and instructions on how to code and use it, see [“Using the ICEMAC macro” on page 51](#).

The specific value used for each installation default in each environment is determined from the following hierarchy:

1. The last value specified in an activated ICEPRMxx member if any
2. The value specified by the ICEMAC macro if any
3. The IBM-supplied default.

You can choose to establish separate defaults for any of the eight installation environments. The defaults for each installation environment are completely independent of the defaults for any other installation environment. Thus if you enable an ICETDx environment for one or more ICEAMx environments, be sure the ICETDx environment includes the same default changes as the ICEAMx environment, when appropriate. For example, if ICEAM2 specifies SVC=252 and enables ICETD1, you would most likely want ICETD1 to specify SVC=252 as well.

You can choose to override many of the installation defaults at run time using an EXEC parameter, a program control statement, an EFS program (see [z/OS DFSORT Application Programming Guide](#) for details of these override methods), or with ICEIEXIT (see [Chapter 5, “Controlling resources using ICEIEXIT,” on page 121](#)), as appropriate.

You can list the installation defaults selected for all of the ICEAMx and ICETDx environments at any time using the ICETOOL DEFAULTS operator, as shown in [Figure 19 on page 113](#). DEFAULTS produces a three-part report showing:

1. The merged PARMLIB/ICEMAC installation default values for ICEAM1-4 and ICETD1-4 that will be used at run-time.
2. The specified PARMLIB ICEPRMxx member option values for ICEAM1-4 and ICETD1-4 (for reference).
3. The ICEMAC installation default values for ICEAM1-4 and ICETD1-4 (for reference).

This chapter explains how to change installation defaults for the eight installation environments using ICEPRMxx members in PARMLIB or the ICEMAC macro and shows several examples for each method. This chapter also shows the syntax for all of the installation parameters, explains the parameters in detail, and lists the IBM-supplied defaults.

Migrating to DFSORT from other sort products

If you are migrating to DFSORT, you should review the IBM-supplied installation defaults and change them as appropriate to correspond to equivalent settings for your previous sort product. In particular, the options shown in the table that follows can make DFSORT operate more like other sort products, thus making migration easier. The installation options, described later in this Chapter, can be used to change the way DFSORT works globally by default. The run-time options, described in *z/OS DFSORT Application Programming Guide*, can be used to override the installation options for specific jobs.

Table 6. Options that can ease migration	
Installation Options	Run-Time Options
ABCODE=MSG/n	
DYNALOC=(d,n)	DYNALLOC=(d,n)
DYNAPCT=x/OLD	DYNAPCT=x/OLD
DYNAUTO=YES/IGNWKDD/NO	DYNALLOC=(d,n)
DYNSPC=n	DYNSPC=n
EQUALS=YES/NO/VBLKSET	EQUALS/NOEQUALS
EXITCK=STRONG/WEAK	EXITCK=STRONG/WEAK
FSZEST=YES/NO	FILSZ=n/En/Un
NOMSGDD=QUIT/ALL/CRITICAL/NONE	
PARMDDN=ddname	
RESET=YES/NO	RESET/NORESET
SORTLIB=SYSTEM/PRIVATE	
SZERO=YES/NO	SZERO/NOSZERO
VLLONG=YES/NO	VLLONG/NOVLLONG
VLSCMP=YES/NO	VLSCMP/NOVLSCMP
VSAMEMT=YES/NO	VSAMEMT/NVSAMEMT
VSAMIO=YES/NO	VSAMIO/NOVSAMIO
ZDPRINT=YES/NO	ZDPRINT/NZDPRINT

Tailoring main storage

You can use the following installation options to tailor DFSORT's use of storage to fit the requirements of your site:

- SIZE
- MAXLIM
- TMAXLIM
- MINLIM
- RESALL
- ARESALL
- RESINV
- ARESINV
- OVERRGN

- ODMAXBF

Study the values for these options carefully before you decide on the installation defaults you want to use. These options are described in this chapter. To see how the options relate to planning decisions, see [“Tailoring main storage” on page 14](#).

Tailoring hiperspace, memory object and dataspace storage

You can use installation options to tailor the way DFSORT uses central storage for Hipersorting, memory object sorting and dataspace sorting to fit the requirements of your site. These options control DFSORT Hipersorting, memory object and data space storage decisions:

- EXPMAX
- EXPOLD
- EXPRES
- TUNE,
- HIPRMAX
- MOSIZE
- DSPSIZE

Study the values for these options carefully before you decide on the installation defaults you want to use. These options are described in this chapter. To see how these options relate to planning decisions, see [“Tailoring storage for Hipersorting, memory object sorting and dataspace sorting” on page 25](#).

Using ICEPRMxx members in PARMLIB

This is the recommended method for changing installation defaults.

See [“Installation options syntax diagram” on page 52](#) for the syntax of the installation options you can specify with ICEPRMxx members in PARMLIB. Note that the TABLE value for ALTSEQ is only valid and needed for the ICEMAC macro, not for ICEPRMxx members.

Coding options in ICEPRMxx members

You can specify installation options for any or all of the eight environments in an ICEPRMxx member of concatenated PARMLIB. You must precede each group of options with one of the following keywords to indicate the installation environment those options apply to: JCL for ICEAM1, INV for ICEAM2, TSO for ICEAM3, TSOINV for ICEAM4, TD1 for ICETD1, TD2 for ICETD2, TD3 for ICETD3 or TD4 for ICETD4. The keyword must end with a blank or a comma.

The rules for coding the options in an ICEPRMxx member of PARMLIB are as follows:

- An option with a single value can be coded as KEYWORD=value or KEYWORD=(value) or KEYWORD(value) and must end with a blank or a comma. An option with more than one value can be coded as KEYWORD=(list) or KEYWORD(list) and must end with a blank or a comma.
- Each option (except ALTSEQ) must be coded on a single line anywhere in columns 1 to 71 and must not contain embedded blanks. Columns 72 to 80 are set to blanks and ignored.
- ALTSEQ can be continued across multiple lines by ending after a comma on one line and continuing on the next line. Example:

```
ALTSEQ=(C1F1,C2F2,  
C3F3)
```

- **Characters after the blank or comma that end an option are ignored.** Example:
DSA=128,ABCODE=10 or DSA=128 ABCODE=10 will be interpreted as DSA=128. Specify DSA=128 and ABCODE=10 on separate lines.
- A line that starts with an * in column 1, or with /* as the first nonblank characters anywhere in columns 1 to 70, is treated as a comment line and ignored. */ is ignored; it is not required to end a comment.

- A line with blanks in columns 1 to 71 is ignored.
- A numeric value can be 1-15 digits with leading zeros. However, the value must not be less than the minimum allowed or greater than the maximum allowed.
- Characters can be entered as uppercase, lowercase or mixed case; they will be converted to uppercase.
- ICEPRMxx member(s) of PARMLIB can be shared between systems running different releases of DFSORT. Any installation options specified in the ICEPRMxx member that are only supported in the newer release of DFSORT will be ignored on the system using the lower release of DFSORT.

Here is an example of three different ways to code an ICEPRM01 member that illustrates three different styles for specifying the same installation options and comment lines. All three are equivalent and interchangeable.

Style 1: KEYWORD=value, KEYWORD=(list) and ending blank

```
* Changes to JCL (ICEAM1) installation environment options
JCL
  ABCODE=MSG
  RESALL=20480
  EQUALS=YES      Keep original order of duplicates
  DYNALOC=(3390,8)
  ALTSEQ=(C1F1,C2F2,
          C3F3,C4F4,
          C5F5)
```

Style 2: KEYWORD=(value), KEYWORD=(list) and ending comma

```
/* Changes to JCL (ICEAM1) installation environment options */
JCL,
  ABCODE=(MSG),
  RESALL=(20480),
  EQUALS=(YES), /* Keep original order of duplicates */
  DYNALOC=(3390,8),
  ALTSEQ=(C1F1,C2F2,
          C3F3,C4F4,
          C5F5),
```

Style 3: KEYWORD(value), KEYWORD(list) and ending blank

```
* Changes to JCL (ICEAM1) installation environment options
JCL
  ABCODE(MSG)
  RESALL(20480)
  EQUALS(YES)      Keep original order of duplicates
  DYNALOC(3390,8)
  ALTSEQ(C1F1,C2F2,
          C3F3,C4F4,
          C5F5)
```

See [“Example of an ICEPRMxx member” on page 112](#) for a more complete example.

Activating ICEPRMxx members

To change the installation defaults with one ICEPRMxx member in PARMLIB, issue a START ICEOPT command to activate that member as follows:

```
START ICEOPT,ICEPRM=xx
```

where xx is the suffix of the ICEPRMxx member you want to use.

For example, to activate ICEPRM01, use:

```
START ICEOPT,ICEPRM=01
```

To change the installation defaults with more than one ICEPRMxx member in PARMLIB, issue a START ICEOPT command to activate those members as follows:


```
START ICEOPT,ICEPRM=(x1,...,xn)
```

where $x1, \dots, xn$ are the suffixes of the ICEPRMxx members you want to use.

For example, to activate ICEPRMMN and ICEPRMS3, type the following on the console:

```
START ICEOPT,ICEPRM=(MN,S3)
```

You can use a START ICEOPT command at any time to change your installation defaults by activating up to 10 ICEPRMxx members.

You can issue a START ICEOPT command from the console, or include it in an appropriate COMMNDxx member in PARMLIB so it will be executed automatically at IPL time. However, we recommend that you test your ICEPRMxx members by activating them from the console before activating them automatically with a COMMNDxx member. ICEOPT issues messages to the console to indicate the status of your installation default changes as well as any errors it detects. Be sure to review these console messages and the return code from ICEOPT. The ICEOPT messages and return codes are explained in [Appendix B, “ICEOPT messages and return codes,”](#) on page 157.

After you execute a START ICEOPT command, you can run an ICETOOL DEFAULTS report to verify that your installation defaults are as expected. See [“Listing installation defaults”](#) on page 113 for more information.

Activating ICEPRMxx members for different LPARS

You can activate a different ICEPRMxx member or combination of ICEPRMxx members for different LPARs at IPL time by including a START ICEOPT command in an appropriate COMMNDxx member in PARMLIB, or at any time by issuing a START ICEOPT command from the console. You might want to set up a “base” ICEPRMxx member with installation options you use for all of your LPARs and a “specific” ICEPRMxx member for each LPAR. For example, you might want to use this START ICEOPT command for system1:

```
START ICEOPT,ICEPRM=(MN,01)
```

and this START ICEOPT command for system2:

```
START ICEOPT,ICEPRM=(MN,02)
```

Merging PARMLIB and ICEMAC options

When you activate ICEPRMxx members with a START ICEOPT command, the ICEPRMxx changes to the defaults are merged with and override the specified or IBM-supplied ICEMAC defaults. The specific value used for each installation default in each environment is determined from the following hierarchy:

1. The last value specified in an activated ICEPRMxx member if any
2. The value specified by the ICEMAC macro if any
3. The IBM-supplied default.

If an option for a specific installation environment is specified more than once in the same activated ICEPRMxx member or in multiple activated ICEPRMxx members, the last instance of the option found is used. For example, if:

```
START ICEOPT,ICEPRM=(01,02)
```

is used and the ICEPRMxx members have the following options:

```
ICEPRM01:
JCL
  ABCODE=10
INV
  DSA=128
JCL
  ABCODE=20
```



```
ICEPRM02:
```

```
  INV
    DSA=256
```

ABCODE=20 is used for the JCL environment and DSA=256 is used for the INV environment.

A value for a particular option in a particular environment at a higher level in the hierarchy completely overrides a value for that same option in that same environment at a lower level in the hierarchy, but does not effect any other options or environments. For example, RESALL=20480 for JCL in an activated ICEPRMxx member overrides RESALL=10240 for JCL in the ICEMAC macro, but does not affect any other options in the JCL environment and does not affect the RESALL value in the other environments.

If you specify a day installation option in an activated ICEPRMxx member, other unspecified day installation options are set to their default of NONE. For example, if you have ICEMAC TD1,WKEND=ALL, ALL is used for Saturday and Sunday. But if you use THU=ALL for TD1 in an activated ICEPRMxx member, ALL is used for Thursday and NONE is used for the other days.

If you don't specify all of the values for an option that allows more than one value, unspecified values are set to their defaults. For example, if you have ICEMAC JCL,DYNALOC=(3390,6) and use DYNALOC=(,12) for JCL in an activated ICEPRMxx member, DYNALOC=(SYSDA,12) is used. Or if you use START ICEOPT,ICEPRM=(05,06), and ICEPRM05 has SVC=(109,ALT) for INV and ICEPRM06 has SVC=109 for INV, SVC=109 is used.

Here is an example to illustrate merged PARMLIB/ICEMAC options when the following is used:

```
START ICEOPT,ICEPRM=05
```

```
ICEPRM05:
  JCL
    DSA=128
    RESALL=20480
    DYNALOC=(3390,6)
```

```
  INV
    ENABLE=TD1
    RESINV=50000
    SMF=SHORT
```

```
  TD1
    WKEND=ALL
    DSA=256
    SMF=FULL
```

```
ICEMAC:
  ICEMAC JCL,SVC=(109,ALT),RESALL=8192,DSA=64
  ICEMAC INV,ENABLE=(TD1,TD2),SVC=(109,ALT),RESINV=4096,      *
    DYNALOC=(SYSDA,8)
  ICEMAC TD1,SVC=(109,ALT),WKDAYS=ALL,SMF=SHORT
```

The following options would be used for the JCL installation environment: DSA=128, RESALL=20480, DYNALOC=(3390,6), SVC=(109,ALT) and the IBM-supplied defaults for all of the other installation options.

The following options would be used for the INV installation environment: ENABLE=TD1, RESINV=50000, SMF=SHORT, SVC=(109,ALT), DYNALOC=(SYSDA,8) and the IBM-supplied defaults for all of the other installation options.

The following options would be used for the TD1 installation environment: SUN=ALL, SAT=ALL, DSA=256, SMF=FULL, SVC=(109,ALT) and the IBM-supplied defaults for all of the other installation options.

For the TSO, TSOINV, TD2, TD3 and TD4 installation environments, the IBM-supplied defaults would be used for all of the installation options.

You can run an ICETOOL DEFAULTS report to list the merged PARMLIB/ICEMAC defaults. See [“Listing installation defaults”](#) on page 113 for more information.

Using multiple START ICEOPT commands

The ICEPRMxx members you activate with a START ICEOPT command stay in effect throughout the IPL unless you issue another successful START ICEOPT command. Each successful execution of ICEOPT activates the ICEPRMxx members you specify, superceding ICEPRMxx members from the previous execution of ICEOPT. So be sure to specify all ICEPRMxx members you need with the START ICEOPT command. As a simple example, if you have the following ICEPRMxx members and ICEMAC macro:

```
ICEPRMMN:
  JCL
    RESALL=20480
    DYNALOC=(SYSDA,8)
    ABCODE=MSG

ICEPRM01:
  JCL
    DYNALOC=(,12)
    DSA=128

ICEPRM02:
  JCL
    DSA=96
    ABCODE=10

ICEMAC:
  ICEMAC JCL,RESALL=10240,ABCODE=20,DYNALOC=(3390,7),DSA=64,SMF=SHORT
```

and you use:

```
START ICEOPT,ICEPRM=(MN,01)
```

the merged PARMLIB/ICEMAC options for the JCL installation environment are: DYNALOC=(SYSDA,12), DSA=128, RESALL=20480, ABCODE=MSG, SMF=SHORT and the IBM-supplied installation defaults for all of the other installation options.

If you then use:

```
START ICEOPT,ICEPRM=02
```

the merged PARMLIB/ICEMAC options for the JCL installation environment are: DSA=96, ABCODE=10, RESALL=10240, DYNALOC=(3390,7), SMF=SHORT and the IBM-supplied installation defaults for all of the other installation options.

All of the previous ICEPRMMN and ICEPRM01 options are cleared, and only the ICEPRM02 options and ICEMAC options are used. However, if instead you use:

```
START ICEOPT,ICEPRM=(MN,02)
```

the merged PARMLIB/ICEMAC options for the JCL installation environment are: DSA=96, ABCODE=10, RESALL=20480, DYNALOC=(SYSDA,8), SMF=SHORT and the IBM-supplied defaults for all of the other installation options.

All of the previous ICEPRMMN and ICEPRM01 options are cleared, but the ICEPRMMN, ICEPRM02 and ICEMAC options are used.

You can clear all of the previous ICEPRMxx options without setting any new options by specifying an ICEPRMxx member with no options in a START ICEOPT command. For example, if ICEPRMCL has:

```
/* Clear options */
```

and you use:

```
START ICEOPT,ICEPRM=CL
```


all of the ICEMAC options are used regardless of any previously activated ICEPRMxx members.

Using the ICEMAC macro

Although the use of ICEPRMxx members in PARMLIB is the recommended way to change installation defaults (see “Using ICEPRMxx members in PARMLIB” on page 46), you can also change installation defaults using the ICEMAC macro.

See “Installation options syntax diagram” on page 52 for the syntax of the installation options you can specify with the ICEMAC macro.

The rules for coding the ICEMAC macro are the same as those for coding IBM assembler language macro statements. ICEMAC must be preceded and followed by at least one blank, and options must be separated by commas. For more complete rules, refer to your assembler language manual.

You must specify one and only one of the following keywords to indicate the installation environment you are modifying: JCL for ICEAM1, INV for ICEAM2, TSO for ICEAM3, TSOINV for ICEAM4, TD1 for ICETD1, TD2 for ICETD2, TD3 for ICETD3 or TD4 for ICETD4. Options must be coded as shown in the syntax diagram, for example, ABCODE=MSG but not ABCODE=(MSG) or ABCODE(MSG). Individual options (such as ABCODE) cannot be repeated.

The ICEMAC macro can be found in the SICEUSER library. The source statement for any installation environment can be found in the AICESRCE library.

To change the IBM-supplied defaults with the ICEMAC macro:

1. Create an SMP/E usermod to modify the ICEMAC macro call in the source statement for any installation environment (ICEAM1, ICEAM2, ICEAM3, ICEAM4, ICETD1, ICETD2, ICETD3, or ICETD4) you want to change in your source distribution library (AICESRCE or a private library). The SMP/E usermod is an IEBUPDTE update. For an SMP/E usermod example, see the ICEOPREC sample job provided on the distribution tape.

Except for the first line of the statements, you can create your own sequence numbers for your update statements, since the statements result from your source code rather than the IBM-supplied default.

Your update statements must conform to the syntax requirements of IEBUPDTE, assembler language, and the ICEMAC macro. See *z/OS DFSMSdfp Utilities* for a description of the IEBUPDTE syntax. Examples and rules for coding the ICEMAC macro are given in “Examples of ICEMAC macros” on page 108.

2. Modify (as instructed in the comments) and run the following sample jobs:
 - a. ICEOPREC to RECEIVE the changes.
 - b. ICEOPAPP to APPLY the changes.

MNOTE messages are produced by the assembler program while expanding the ICEMAC statement during the APPLY step. If you violate any of the rules for coding ICEMAC, an appropriate error message will be issued when ICEAM1, ICEAM2, ICEAM3, ICEAM4, ICETD1, ICETD2, ICETD3, or ICETD4 is reassembled during the APPLY step. The ICEMAC error messages (MNOTEs) are explained in [Appendix C, “ICEMAC and message module messages,”](#) on page 161.

If you receive an error message:

- a. Correct and recode the statement containing the error.
- b. Run an SMP/E RESTORE to remove the SMP/E usermod. (See *z/OS SMP/E Reference* for details).
- c. Run an SMP/E REJECT to remove the SMP/E usermod from the global zone, or add a later REWORK date to the updated SMP/E usermod.
- d. Rerun the RECEIVE and APPLY jobs ICEOPREC and ICEOPAPP.

Installation options syntax diagram

The syntax diagram for the installation options you can specify in ICEPRMxx members in PARMLIB and with the ICEMAC macro is shown later in this section.

You must specify one of the following keywords to indicate the installation environment you are modifying: JCL for ICEAM1, INV for ICEAM2, TSO for ICEAM3, TSOINV for ICEAM4, TD1 for ICETD1, TD2 for ICETD2, TD3 for ICETD3 or TD4 for ICETD4. **All other parameters are optional.**

The ENABLE option can be used with JCL, INV, TSO or TSOINV to indicate one or more time-of-day environments to be associated with an ICEAMx environment. One or more day parameters can be used with TD1, TD2, TD3 and TD4 to indicate the days and times when an ICETDx environment can be activated.

Notes on using the Syntax Diagram:

- Each option consists of a keyword and one or more values stacked vertically. If you specify an option, you must select one of its values. Example: MSG or n for ABCODE.
- The keyword must be coded exactly as shown. Example: ABCODE.
- If the value is shown in uppercase letters, it must be coded exactly as shown. Example: MSG.
- If the value is shown in lowercase letters, it represents specific information that you supply. Example: n requires a number.
- For the ENABLE option, TD1, TD2, TD3, TD4 or any combination of those values separated by a comma can be used. Example: ENABLE=(TD1,TD3).
- For the ALTSEQ option, multiple ffft values separated by a comma can be used. Example: ALTSEQ=(C1F1,C2F2).

Example of ICEPRMxx member:

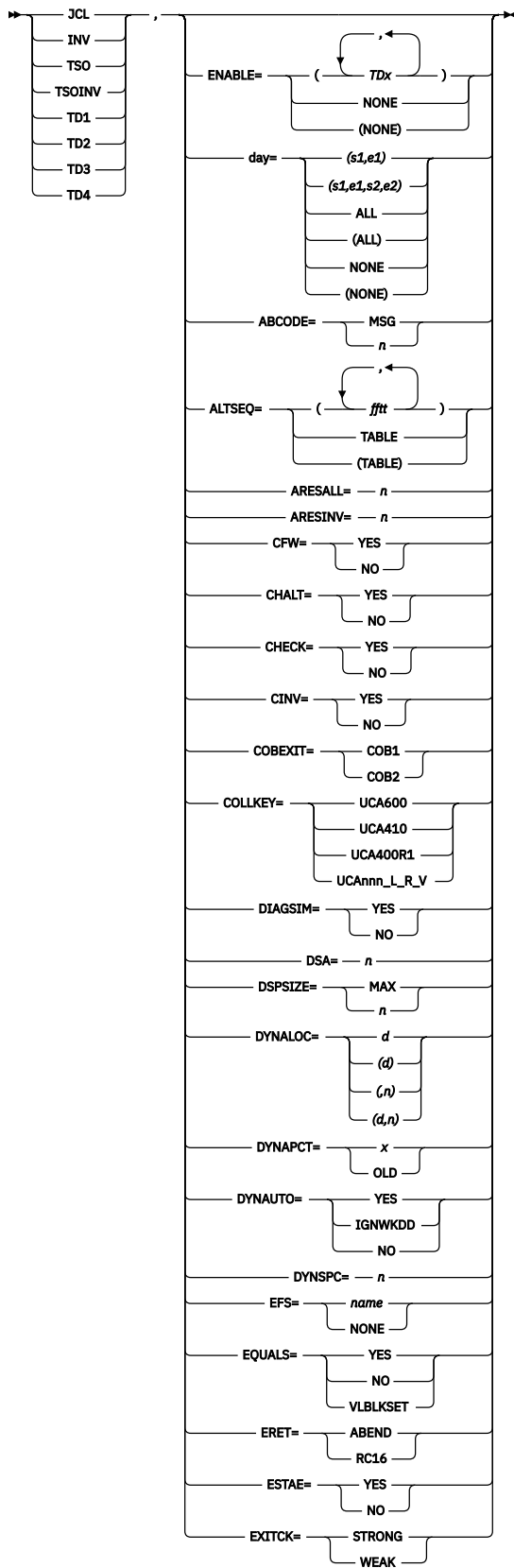
```
JCL
  DSA=128
  DYNALOC=( , 8)
  RESALL=20480
  EQUALS=YES
INV
  DSA=128
  DYNALOC=( , 8)
  RESINV=40960
  EQUALS=YES
```

See [“Using ICEPRMxx members in PARMLIB” on page 46](#) for the specific syntax rules that apply to ICEPRMxx.

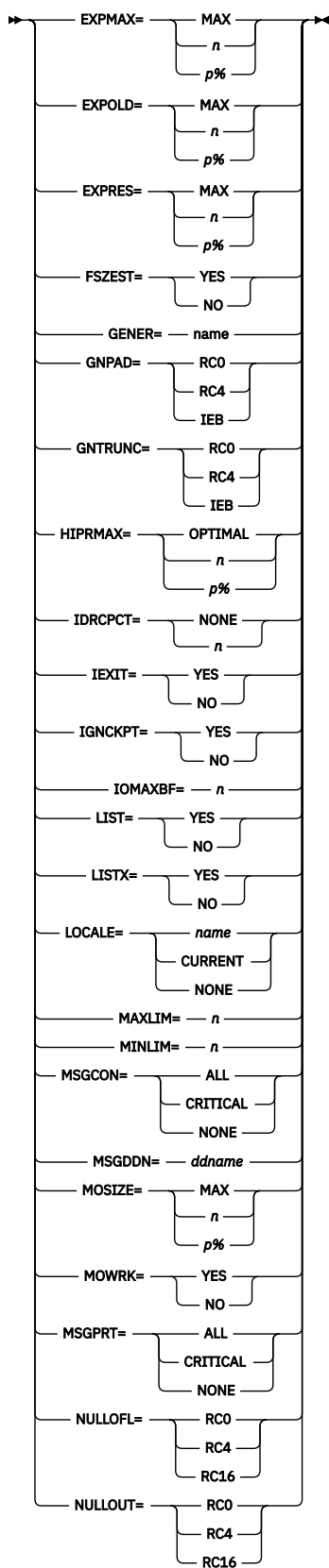
Example of ICEMAC macro:

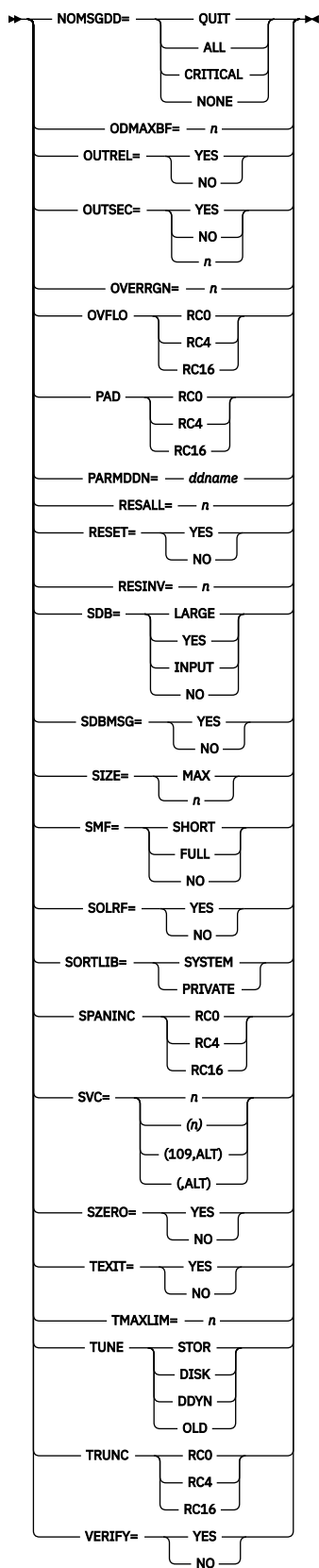
```
ICEMAC JCL, DSA=128, DYNALOC=( , 8), RESALL=20480, EQUALS=YES
```

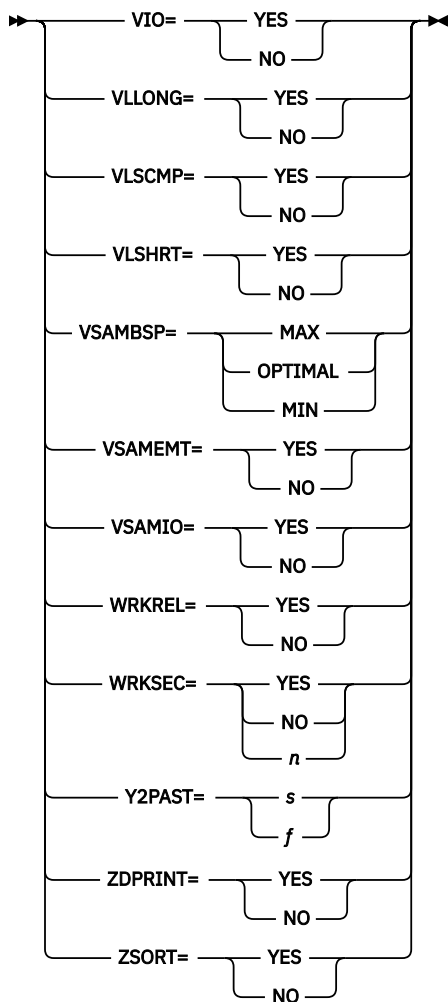
See [“Using the ICEMAC macro” on page 51](#) for the specific syntax rules that apply to ICEMAC.



Installation Defaults

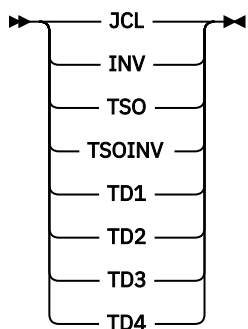






Installation options

JCL|INV|TSO|TSOINV|TD1|TD2|TD3|TD4



Specifies the invocation installation environment (ICEAMx where x is 1, 2, 3 or 4) or time-of-day installation environment (ICETDx where x is 1, 2, 3 or 4) for which this set of installation defaults is to be used. JCL, INV, TSO, TSOINV, TD1, TD2, TD3, or TD4 must be specified first to indicate to which of the eight possible installation environments the options that follow apply.

The ENABLE parameter can be specified with JCL, INV, TSO, or TSOINV to enable one or more time-of-day installation environments (ICETDx) for the selected invocation installation environment (ICEAMx). One or more of the ICETDx environments can be enabled for any ICEAMx environment. For example, ICETD1 and ICETD4 can be enabled for ICEAM1, and ICETD4 can be enabled for ICEAM2.

The day parameter can be specified with TD1, TD2, TD3, or TD4 to specify the days and times for which the selected time-of-day installation environment (ICETDx) is to be activated.

JCL

Specifies that this set of installation defaults is for ICEAM1, the batch direct invocation environment. ICEAM1 is used at run time when DFSORT is invoked directly (that is, not through programs) by batch jobs, provided that an enabled time-of-day environment (ICETDx) is not activated. All of the installation options can be specified with JCL. However, options ARESINV, RESINV, GENER, GNPAD, and GNTRUNC are not meaningful for direct invocation and are ignored.

INV

Specifies that this set of installation defaults is for ICEAM2, the batch program invocation environment. ICEAM2 is used at run time when DFSORT is invoked through batch programs, provided that an enabled time-of-day installation environment (ICETDx) is not activated. All of the installation options can be specified with INV.

TSO

Specifies that this set of installation defaults is for ICEAM3, the TSO direct invocation environment. ICEAM3 is used at run time when DFSORT is invoked directly (that is, not through programs) by foreground TSO users, provided that an enabled time-of-day installation environment (ICETDx) is not activated. All of the installation options can be specified with TSO. However, options ARESINV, RESINV, GENER, GNPAD, and GNTRUNC are not meaningful for direct invocation and are ignored.

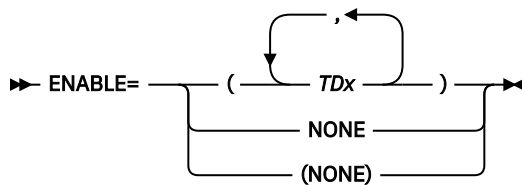
TSOINV

Specifies that this set of installation defaults is for ICEAM4, the TSO program invocation environment. ICEAM4 is used at run time when DFSORT is invoked through programs by foreground TSO users, provided that an enabled time-of-day installation environment (ICETDx) is not activated. All of the installation options can be specified with TSOINV.

TDx (x=1, 2, 3 or 4)

Specifies that this set of installation defaults is for ICETDx, the indicated time-of-day installation environment (TD1=first, TD2=second, TD3=third, TD4=fourth). ICETDx is used at run time when it is activated for the time-of-day of the run, provided it is enabled by the invocation environment (ICEAMx) in effect. All of the installation options can be specified with TDx. However, options ARESINV, RESINV, GENER, GNPAD, and GNTRUNC are not meaningful for direct invocation and are ignored if the enabling installation environment is ICEAM1 or ICEAM3.

ENABLE



Specifies whether time-of-day installation environments (ICETDx) are to be used if activated for this invocation environment (ICEAMx). One or more of the ICETDx environments can be enabled for any ICEAMx environment. For example, ICETD1 and ICETD4 can be enabled for ICEAM1, and ICETD4 can be enabled for ICEAM2.

(TDx...,TDx)

Specifies the ICETDx installation environments in effect for this ICEAMx invocation environment, in the order in which they are to be used if activated. Any or all of TD1, TD2, TD3, and TD4 can be specified as ENABLE values, but the same TDx value must not be specified more than once. For example, ENABLE=(TD4,TD1,TD2) is valid, but ENABLE=(TD4,TD1,TD4) is not.

NONE | (NONE)

Specifies that ICETDx installation environments are not to be used for this ICEAMx installation environment.

Default: NONE

ENABLE is ignored when it is specified with TD1, TD2, TD3, or TD4.

ICEPRMxx example

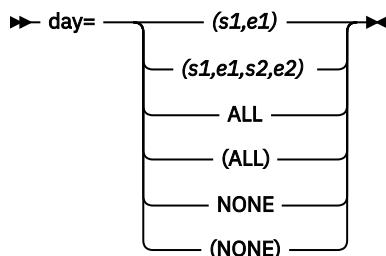
```
INV
ENABLE=(TD3,TD2)
```

ICEMAC example

```
ICEMAC INV,ENABLE=(TD3,TD2)
```

Enables ICETD3 and ICETD2, in that order, for ICEAM2. When DFSORT is invoked through batch programs, the ICETD3 defaults are used if ICETD3 is activated for the time and day of the run. If ICETD3 is not activated, the ICETD2 defaults are used if ICETD2 is activated. If ICETD3 is not activated and ICETD2 is not activated, the ICEAM2 defaults are used.

day



Specifies the time ranges for each day of the week when this time-of-day installation environment (ICETDx) is to be activated. The time ranges can be specified for all days, the weekend days, the weekdays, or each day of the week; any combination may be used. However, the same day must not be specified twice (for example, either weekdays or Monday can be specified, but not both).

To be activated, an ICETDx environment must:

- be specified by the ENABLE parameter of the invocation installation environment (ICEAMx) in effect for the runtime environment (for example, ICEAM1 for a batch direct run) and
- specify a day and time range that includes the day and time of the run (as listed in DFSORT's ICE000I message).

The following operands can be used for day:

SUN, MON, TUE, WED, THU, FRI, SAT

these operands for the individual days can be specified in any combination.

WKEND

this is equivalent to specifying the SUN and SAT operands (weekend days). If WKEND is specified, SUN and SAT cannot be specified.

WKDAYS

this is equivalent to specifying the MON, TUE, WED, THU, and FRI operands (weekdays). If WKDAYS is specified, MON, TUE, WED, THU, and FRI cannot be specified.

ALLDAYS

ALLDAYS is equivalent to specifying the SUN, MON, TUE, WED, THU, FRI, and SAT operands. If ALLDAYS is specified, SUN, MON, TUE, WED, THU, FRI, SAT, WKEND, and WKDAYS cannot be specified.

The following operands can be used for any day operand:

s1,e1

Specifies a time range, consisting of a starting time (s1) and ending time (e1), during which this time-of-day installation environment can be activated for the specified day. s1 and e1 must each be specified as *hhmm* using a 24-hour clock where *hh* is in hours (00-23) and *mm* is in minutes

(00-59). Leading zeros can be omitted for *hhmm* values. The same value must not be specified for *s1* and *e1*.

hhmm must be a valid time between midnight (0) and 11:59 PM (2359).

For ICEPRMxx, a time range cannot cross midnight, that is, an *s1* value must not be greater than the *e1* value. Here is an example of time ranges on Sunday, Friday, and Saturday that do not cross midnight.

```
TD2
SUN=(630,1959)
FRI=(2200,2359)
SAT=(0,1159)
```

For this example, ICETD2 can be activated if the day and time of the run (as shown in DFSORT's ICE000I message) is between 6:30 AM and 7:59 PM (630,1959) on Sunday, between 10:00 PM and 11:59 PM (2200,2359) on Friday, or between midnight and 11:59 AM (0,1159) on Saturday.

For ICEMAC, a time range can cross midnight. To indicate a time range that crosses midnight (that is, starts on the specified day and ends on the next day), specify an *s1* value that is greater than the *e1* value. Here is an example of a time range on Sunday that does not cross midnight and a time range on Friday that does:

```
ICEMAC TD2,SUN=(630,1959),FRI=(2200,1159)
```

For this example, ICETD2 can be activated if the day and time of the run (as shown in the ICE000I message) is between 6:30 AM and 7:59 PM (630,1959) on Sunday, between 10:00 PM and 11:59 PM (2200,2359) on Friday, or between midnight and 11:59 AM (0,1159) on Saturday.

Note that *e1* is interpreted as meaning "up to and including the *hhmm* value given". *SUN=(630,1959)* means on Sunday from 6:30:00 am to 7:59:59 PM. Thus, for the *SUN=(630,1959)* time range:

- If ICE000I shows "00:00 ON SUN" through "06:29 ON SUN", ICETD2 will not be activated.
- If ICE000I shows "06:30 ON SUN" through "19:59 ON SUN", ICETD2 can be activated.
- If ICE000I shows "20:00 ON SUN" through "23:59 ON SUN", ICETD2 will not be activated.

Be careful to specify *mm* as 59 (for example, 1959) rather than as 00 (for example, 2000) if you do not want your ICETDx environment activated for the first minute of the next hour.

s2,e2

Specifies a second time range, consisting of a starting time (*s2*) and ending time (*e2*), during which this time-of-day installation environment can be activated for the specified day. See *s1,e1* described previously for details.

ALL | (ALL)

Specifies that this time-of-day environment is always to be activated on the specified day. This is equivalent to a time range of midnight to 11:59 PM (0,2359). For ICEMAC, ALL overrides any *s1,e1* or *s2,e2* times that cross midnight of the previous day. (Note that time ranges cannot cross midnight for ICEPRMxx).

Here's an example:

```
ICEMAC TD1,WKDAY=(1200,729),SAT=ALL
```

For this example, ICETD1 can be activated if the day and time of the run is between noon and 11:59 PM (1200,2359) on Monday through Friday, between midnight and 7:29 AM (0,729) on Tuesday through Friday or between midnight and 11:59 PM (0,2359) on Saturday (that is, any time on Saturday).

NONE | (NONE)

Specifies that this time-of-day environment is not to be activated on the specified day. For ICEMAC, *s1,e1* or *s2,e2* times that cross midnight of the previous day override NONE (either specified or defaulted). (Note that time ranges cannot cross midnight for ICEPRMxx).

Installation Defaults

Here's an example:

```
ICEMAC TD3,FRI=(2000,1240)
```

For this example, ICETD3 can be activated if the day and time of the run is between 8:00 PM and 11:59 PM (2000,2359) on Friday or between midnight and 12:40 PM (0,1240) on Saturday. FRI=(2000,1240) overrides the IBM-supplied default of SAT=NONE.

Default: ALLDAYS=NONE (that is, NONE for each day of the week)

day is ignored when it is specified with JCL, INV, TSO, or TSOINV.

Example 1

IDFPRMxx example

```
INV
  ENABLE=(TD3)
  HIPRMAX=20
TD3
  WKDAYS=(0,759,1800,2359)
  HIPRMAX=OPTIMAL
  DSA=48
```

ICEMAC example

```
ICEMAC INV,ENABLE=(TD3),HIPRMAX=20
END
ICEMAC TD3,WKDAYS=(0,759,1800,2359),HIPRMAX=OPTIMAL,DSA=48
END
```

The ICEPRMxx and ICEMAC options shown previously are equivalent ways of changing the installation defaults.

When DFSORT is invoked through batch programs, the ICETD3 set of defaults consisting of HIPRMAX=OPTIMAL, DSA=48 and the IBM-supplied defaults for all other installation options, is used on Monday through Friday from midnight to 7:59 AM (0,759) and from 6:00 PM to 11:59 PM (1800,2359).

The ICEAM2 set of defaults consisting of HIPRMAX=20 and the IBM-supplied defaults for all other installation options, will be used for all other times.

Example 2

ICEPRMxx

```
JCL
  ENABLE=(TD2)
  SVC=252
TD2
  WKDAYS=(1000,1115,2130,2359)
  SAT=(1205,1559,1800,2359)
  SUN=ALL
  SVC=252
  DSA=48
  TMAXLIM=33554432
```

ICEMAC

```
ICEMAC JCL,ENABLE=(TD2),SVC=252
END
ICEMAC TD2,WKDAYS=(1000,1115,2130,2359),
          SAT=(1205,1559,1800,2359),
          SUN=ALL,SVC=252,DSA=48,TMAXLIM=33554432
END
```

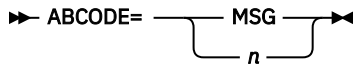
The ICEPRMxx and ICEMAC options shown previously are equivalent ways of changing the installation defaults.

When DFSORT is invoked directly by batch jobs, the ICETD2 set of defaults consisting of SVC=252, DSA=48, TMAXLIM=33554432 and the IBM-supplied defaults for all other installation options, will be used as follows:

- Sunday: midnight-11:59 PM (ALL = 0,2359)
- Monday-Friday: 10:00 AM-11:15 AM (1000,1115) and 9:30 PM-11:59 PM (2130,2359)
- Saturday: 12:05 PM-3:59 PM (1205,1559) and 6:00 PM-11:59 PM (1800,2359)

The ICEAM1 set of defaults consisting of SVC=252 and the IBM-supplied defaults for all other installation options, will be used for all other times.

ABCODE



Specifies whether to abend with a user completion code equal to the critical error message number, or with a specific number between 1-99.

MSG

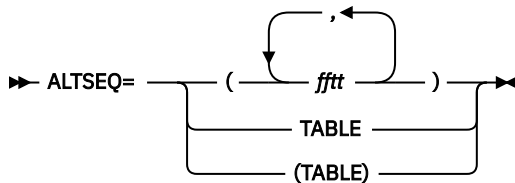
Abend with user completion code equal to the critical error message number.

n

Abend with a specific code between 1-99.

Default: MSG

ALTSEQ



Specifies changes to the alternate translation table (ALTSEQ table). Any modifications you specify are applied to the standard EBCDIC translation table.

The ALTSEQ table can be used in two ways as follows:

- To apply an alternate collating sequence for SORT, MERGE, INCLUDE, or OMIT fields, or for INREC, OUTREC, or OUTFIL IFTHEN WHEN=(logexp) fields, with format AQ (or format CH with CHALT in effect). In this case, the ALTSEQ table is used to change only the order in which data is collated, not the data itself.

For example, if you want to specify that the character \$ (X'5B') is to collate at position X'EA', after uppercase Z (X'E9'), you should specify ALTSEQ=(5BEA).

- To convert characters for INREC, OUTREC, or OUTFIL OUTREC fields with TRAN=ALTSEQ. In this case, the ALTSEQ table is used to change the actual data.

For example, if you want to change the character \$ (X'5B') to the character * (X'5C'), you should specify ALTSEQ=(5B5C).

(fftt...,fftt)

Specifies the characters whose positions are to be changed. You can change the positions of up to 50 characters with this form of the ALTSEQ option. (If this limit is exceeded, then the macro processor issues an error message.)

ff

Specifies, in hexadecimal, the character whose position is to be changed in the ALTSEQ table.

tt

Specifies, in hexadecimal, the new position the character is to occupy in the ALTSEQ table.

See the description of the ALTSEQ Control Statement in [z/OS DFSORT Application Programming Guide](#) for additional details and examples.

TABLE | (TABLE)

For ICEMAC only, specifies that you are supplying a complete 256-byte ALTSEQ translation table that will override all 256 characters of the EBCDIC translation table. You must use this form of the ALTSEQ option if you need to change the positions of more than 50 characters with ICEMAC.

Since

```
ALTSEQ=(fhtt...,
        ...,fhtt)
```

can be used to change up to 256 characters with ICEPRMxx, ALTSEQ=TABLE is not needed or allowed for ICEPRMxx.

When using ALTSEQ=(TABLE) with ICEMAC, TABLE must be the only value. If there is more than one value, for example, ALTSEQ=(TABLE,C1F2), MNOTE "ICESM1083 TABLE IS INVALID WITH OTHER ALTSEQ VALUES" is issued. Your 256-byte ALTSEQ table must start on the line following the last line of the ICEMAC statement. It is your responsibility to ensure that the 256-byte table is correct.

For example, if you want to completely reverse the standard EBCDIC collating sequence, you should specify the following:

```
ICEMAC JCL,ALTSEQ=TABLE
DC X' FFFDFDFCFBFAF9F8F7F6F5F4F3F2F1F0'
DC X' EEEEEDECEBEAE9E8E7E6E5E4E3E2E1E0'
DC X' DFDEDDDCDBDAD9D8D7D6D5D4D3D2D1D0'
DC X' CFCECDCCBCAC9C8C7C6C5C4C3C2C1C0'
DC X' BFBEBDBCBBBAB9B8B7B6B5B4B3B2B1B0'
DC X' AFAEADACABAAA9A8A7A6A5A4A3A2A1A0'
DC X' 9F9E9D9C9B9A99989796959493929190'
DC X' 8F8E8D8C8B8A89888786858483828180'
DC X' 7F7E7D7C7B7A79787776757473727170'
DC X' 6F6E6D6C6B6A69686766656463626160'
DC X' 5F5E5D5C5B5A59585756555453525150'
DC X' 4F4E4D4C4B4A49484746454443424140'
DC X' 3F3E3D3C3B3A39383736353433323130'
DC X' 2F2E2D2C2B2A29282726252423222120'
DC X' 1F1E1D1C1B1A19181716151413121110'
DC X' 0F0E0D0C0B0A09080706050403020100'
END
```

Default: The normal EBCDIC translation table.



Attention: Use of ALTSEQ can degrade performance.

ARESALL

➡ ARESALL= *n* ➡

Specifies the amount of main storage reserved above 16 MB virtual for system use while DFSORT is running. Normally, this parameter is not needed.

RESALL specifies the amount of storage reserved below 16 MB virtual. For more information, see ["Tailoring main storage" on page 14](#).

n

Specifies the number of bytes as a decimal value within the range of 0 to 16515072 (16128 KB). If the value specified is not a multiple of 8, it is rounded up to the next multiple of 8.

Default: 0 bytes

ARESINV

➤ ARESINV= — *n* ➤

Specifies the amount of main storage reserved above 16 MB virtual, while DFSORT is running, for an invoking program's user exit routines that reside above or use space above 16 MB virtual. The reserved space is not meant to be used for the invoking program's executable code. For more information, see [“Tailoring main storage” on page 14](#). Normally, this parameter is not needed.

RESINV specifies the amount of storage reserved below 16 MB virtual. For more information, see [“Tailoring main storage” on page 14](#).

n

Specifies the number of bytes as a decimal value within the range of 0 to 16515072 (16128 KB). If the value specified is not a multiple of 8, it is rounded up to the next multiple of 8.

Default: 0 bytes

ARESINV is ignored when it is specified with the JCL or TSO parameter.

CFW

➤ CFW= — YES —
 NO —

Specifies whether DFSORT can use cache fast write when processing SORTWKdd data sets that reside on devices connected to storage subsystems that support it. (DFSORT's Blockset technique uses cache fast write for SORTWKdd data sets, when possible, to improve performance.)

YES

Specifies that DFSORT can use cache fast write when processing SORTWKdd data sets.

NO

Specifies that DFSORT cannot use cache fast write.

Default: YES

CHALT

➤ CHALT= — YES —
 NO —

Specifies whether format CH fields are translated by the alternate collating sequence.

YES

Specifies that DFSORT translates character control fields and compare fields with format CH using the alternate collating sequence.

NO

Specifies that format CH fields are not translated.

Default: NO

CHECK

➤ CHECK= — YES —
 NO —

Installation Defaults

Specifies whether record count should be checked for applications that use the E35 user exit routine without an output data set.

YES

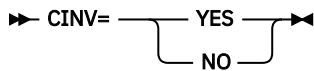
Specifies that record count should be checked.

NO

Specifies that record count should not be checked.

Default: YES

CINV



Specifies whether DFSORT can use control interval access for VSAM data sets. (DFSORT's Blockset technique uses control interval access for VSAM input data sets, when possible, to improve performance.)

YES

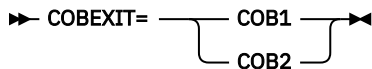
DFSORT can use control interval access for VSAM data sets.

NO

DFSORT cannot use control interval access.

Default: YES

COBEXIT



Specifies the library for COBOL E15 and E35 routines. See [z/OS DFSORT Application Programming Guide](#) for details on the use of COBEXIT and runtime libraries with COBOL E15 and E35 routines.

COB1

Specifies that COBOL E15 and E35 routines are run with the OS/VS COBOL runtime library or, in some cases, with no COBOL runtime library.

COBEXIT=COB1 is obsolete, but is still available for compatibility reasons.

Note that Language Environment is the only runtime library for COBOL supported by IBM service.

COB2

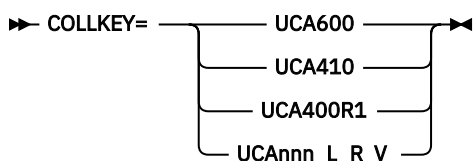
Specifies that COBOL E15 and E35 routines are run with either the VS COBOL II runtime library or the Language Environment runtime library.

Note that Language Environment is the only runtime library for COBOL supported by IBM service.

Default: COB2

The DFSORT documents only discuss the Language Environment runtime library, and assume that COBEXIT=COB2 is in effect. Although it is possible to run with older runtime libraries, and with COBEXIT=COB1, these are not recommended or discussed, and are not supported by IBM service.

COLLKEY



Specifies the collation rules.

DFSORT's Unicode Collation behavior can be modified according to your cultural environment. The COLLKEY parameter allows specifying the Unicode standard character suite for national collation sequences. Your Unicode environment is defined to DFSORT using the COLLKEY=value

UCA600

This collation version supports the Unicode Standard character suite 6.0.0 and uses Normalization Service under 6.0.0 Unicode character suite.

UCA410

This collation version supports the Unicode Standard character suite 4.1.0 and uses Normalization Service under 4.1.0 Unicode character suite.

UCA400R1

This collation version supports the Unicode Standard character suite 4.0.0 and uses Normalization Service under 4.0.1 Unicode character suite.

UCA_{nnn}_L_R_V

Specifies a collation version, where specific collation rules will modify any of the default Unicode Collation tables specified (UCA400R1, UCA410, or UCA600). Collation versions are set when you specify the following fields:

L

- Language (Specify a language for desired locale.)

R

- Region (Specify a region for desired locale.)

V

- Variant (Specify a variant for desired locale.)

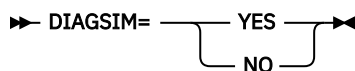
Default: UCA600

Note:

- For supported collation version settings (Language/Region/Variant), see Appendix E. Locales for collation in *z/OS Unicode Services User's Guide and Reference*.
- If there is no collation information, installation default collation version will be set as default without any change.
- If records are being sorted/merged using UTF8/UTF16/UTF32 data formats, then the following operands are not supported.
 - LOCALE
 - EFS
 - SUM
 - E61 exit
 - VLSHRT
 - INREC
 - OUTREC

Unicode Locales repository data set name SYS1.SCUNLOCL contains a set of locales documented in Locales for collation support. All of those locales contain a section for Collation rules.

DIAGSIM



Specifies whether all DFSORT control statements and messages, including diagnostic messages (ICE800I through ICE999I), are printed. This option enables you to simulate a SORTDIAG DD statement for DFSORT applications. For full details on the SORTDIAG DD statement and the use of the message data set, see *z/OS DFSORT Application Programming Guide*.

YES

Specifies that a SORTDIAG DD statement is to be simulated for DFSORT applications.

Messages and control statements are written to the message data set. If a message data set (alternate or SYSOUT, as appropriate) is not present, the specified NOMSGDD value determines the action to be taken. For example, if NOMSGDD=QUIT is specified, DFSORT terminates the application with a return code of 20. See the NOMSGDD parameter discussion for complete details.

NO

Specifies that a SORTDIAG DD statement is not to be simulated for DFSORT applications.

Default: NO

DSA

►► DSA= — *n* ◄◄

Specifies the maximum amount of storage available to DFSORT for dynamic storage adjustment of a Blockset sort application when SIZE/MAINSIZE=MAX is in effect. If you specify a DSA value greater than the TMAXLIM value, you allow DFSORT to use more storage than the TMAXLIM value if doing so should improve performance. The amount of storage DFSORT uses is subject to the DSA value as well as system limits such as region size. However, whereas DFSORT always tries to obtain as much storage as it can up to the TMAXLIM value, DFSORT only tries to obtain as much storage as needed to improve performance up to the DSA value.

The performance improvement from dynamic storage adjustment usually provides a good tradeoff against the increased storage used by DFSORT. On storage constrained systems, however, the DSA value should be set low enough to prevent unacceptable paging.

n

Specifies that DFSORT can dynamically adjust storage to improve performance, subject to a limit of *n* MB. *n* must be a value between 0 and 2000.

If the DSA value specified in ICEMAC is less than the TMAXLIM value specified in ICEMAC, the DSA value is set to 0 in the ICEAMx or ICETDx module. The DSA value specified in an ICEPRMxx member is not adjusted based on the TMAXLIM value.

If the DSA value in effect at run time is less than the TMAXLIM value in effect at run time, the DSA value is set to 0 at run time .

Default: 128

DSPSIZE

►► DSPSIZE= — MAX — ◄◄
 └── *n* ─┘

Specifies the maximum amount of data space to be used with dataspace sorting. A data space is an area of contiguous storage that is backed by real and auxiliary storage, whichever is necessary as determined by the system. Because DFSORT is able to sort large pieces of data using data space, CPU time and elapsed time are reduced.

There are several factors that can limit the size of a data space used by an application:

1. The IEFUSI exit can limit the total amount of Hiperspace, memory objects and data space available to an application.
2. DSPSIZE can limit the size of a data space available to an application, as detailed later in this section.
3. Sufficient available central storage must be present to back DFSORT's data space. The amount of available central storage changes constantly, depending on current system activity. Consequently, DFSORT checks the amount of available central storage throughout a dataspace sorting run and switches from using a data space to using disk work data sets if the available central storage is too low.
4. Other concurrent Hipersorting, memory object sorting and dataspace sorting applications further limit the amount of available storage. A dataspace sorting application keeps track of the storage needs of all other Hipersorting, memory object sorting and dataspace sorting applications on the system, and does not attempt to back its data space with storage needed by another Hipersorting, memory object sorting or dataspace sorting application. This prevents overcommitment of storage resources in the event of multiple large concurrent Hipersorting, memory object sorting and dataspace sorting applications starting at similar times on the same system.
5. The installation options EXPMAX, EXPOLD, and EXPRES can also be used to further limit the amount of storage available to dataspace sorting applications. EXPMAX limits the total amount of available storage that can be used at any one time to back DFSORT Hiperspaces, memory objects and data spaces. EXPOLD limits the total amount of old storage that can be used at any one time to back DFSORT hiperspaces, memory objects and data spaces. EXPRES sets aside a specified amount of available storage for use by non-Hipersorting, non-memory object sorting and non-dataspace sorting applications.

Some of these limits depend on system and other DFSORT Hipersorting and memory object sorting activity during the time that a dataspace sorting application runs. Consequently, the size of the data space that a dataspace sorting application uses can vary from run to run.

If the amount of data space DFSORT decides to use is sufficient, DFSORT sorts your data in main storage and does not require additional temporary work space. If the amount of data space is not sufficient, DFSORT uses disk as temporary work space. When dataspace sorting is enabled, DFSORT changes DYNAUTO=NO to DYNAUTO=YES in some cases. Hiperspace and memory objects are not used when dataspace sorting is used.

MAX

Specifies that DFSORT dynamically determines the maximum amount of data space that is to be used for dataspace sorting. In this case, DFSORT bases its data space usage on the size of the file being sorted and the paging activity of the system.

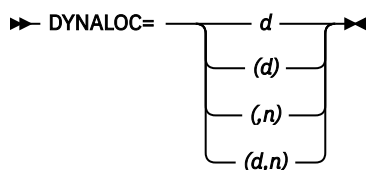
n

Specifies the maximum amount, in MB, of data space to be used for dataspace sorting. n must be a value between 0 and 9999. The actual amount used does not exceed n, but may be less, depending on the size of the file being sorted and the paging activity of the system.

If n is zero, dataspace sorting is not used.

Default: MAX

DYNALOC



Specifies the default values to be used when dynamic allocation is requested using the DYNAUTO installation parameter or the DYNALOC runtime parameter. DYNALOC does not request dynamic allocation of needed work space. It merely supplies default values for d and n.

Even if you specify DYNAUTO=NO, DFSORT can enable automatic dynamic allocation in some cases. Therefore, you should always specify the defaults for the device (d) and number of work data sets (n) for use with dynamic allocation.

d

Specifies the default device name for work data sets to be dynamically allocated. You can specify any IBM disk device or tape device supported by your operating system in the same way you would specify it in the JCL UNIT parameter. You can also specify a group name, such as DISK or SYSDA.

n

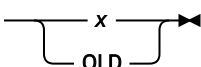
Specifies the default maximum number of work data sets to be dynamically allocated, from 1 to 255. If you specify more than 255, the default is set to 255. If you specify 1 and the Blockset technique is selected at run time, a maximum of 2 data sets is used. If you specify more than 32 and the Blockset technique is not selected at run time, a maximum of 32 data sets is used.

For optimum allocation of resources such as virtual storage, avoid specifying a large number of work data sets unnecessarily.

Default: The default for *d* is SYSDA. If at run time one of the Blockset techniques is chosen, the default for *n* is set to 4. If any other technique is chosen, the default for *n* is set to 3.

In rare cases where DFSORT is unable to dynamically allocate the required space using the value supplied by *d*, allocation may be attempted with a more generic unit name (such as 3390) in an effort to allocate the required space on volumes from a different storage group. See [System-managed storage](#) in *z/OS DFSORT Tuning Guide* for recommendations on controlling data set allocation.

DYNAPCT

➡ DYNAPCT= 

Specifies additional work data sets to be dynamically allocated with zero space. DFSORT only extends these data sets when necessary to complete a sort application. The availability of additional work data sets can help avoid out of space ABENDs.

x

Specifies the number of additional work data sets (*y*) as a percentage of the maximum number of dynamically allocated work data sets (DYNALLOC/DYNALOC *n* value) in effect. *y* will be set to $n * x\%$. The total number of dynamically allocated work data sets will be $n + y$. For example, if DYNALLOC=(SYSDA,20) and DYNAPCT=20 are in effect, 4 additional work data sets will be allocated for a total of 24.

The value *x* must be between 0 and 254. The minimum value for *y* is 1 and the maximum value for *y* is 254. The maximum value for $n + y$ is 255; if *x* results in a value for $n + y$ greater than 255, *y* will be set to $255 - n$.

OLD

Specifies additional work data sets should only be allocated when DFSORT cannot determine the file size. When DFSORT can determine the file size, additional work data sets are not allocated ($y=0$), and the total number of work data sets is *n*.

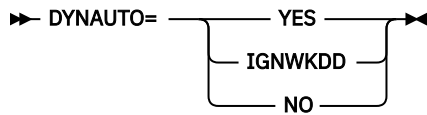
Default: 10

Note: When message ICE118I is issued indicating that DFSORT cannot determine the file size, *y* is set as follows:

- For DYNAPCT=OLD, *y* is set to $n * 50\%$

- For DYNAPCT=x with $x \leq 50$, y is set to $n * 50\%$
- For DYNAPCT=x with $x > 50$, y is set to $n * x\%$

DYNAUTO



Specifies whether work space should automatically be dynamically allocated at run time.

YES

Specifies that work data sets should automatically be dynamically allocated using the DYNALLOC/ DYNALOC values in effect when no SORTWKdd DD statement is present at run time.

IGNWKDD

Specifies that work data sets should automatically be dynamically allocated using the DYNALLOC/ DYNALOC values in effect even if SORTWKdd DD statements are present at run time. SORTWKdd data sets are deallocated before sorting begins, thus freeing unneeded temporary work space. DYNAUTO=IGNWKDD provides the same automatic dynamic allocation capability as DYNAUTO=YES regardless of whether or not SORTWKdd DD statements are present.

Tape SORTWKdd data sets are not deallocated.

OPTION USEWKDD may be specified in DFSPARM or the extended parameter list to allow SORTWKdd data sets to be used for individual jobs (that is, deallocation does not take place).

NO

Specifies that work data sets should not automatically be dynamically allocated.

Even if you specify DYNAUTO=NO, DFSORT can enable automatic dynamic allocation in some cases. Therefore, you should always use DYNALOC to specify the defaults for the device and number of work data sets for use with dynamic allocation.

Default: YES

DYNSPC

➡ DYNAPCT= — n ➡

Specifies the total default primary space allocation for all of the dynamically allocated work data sets when the input file size is unknown, that is, when DFSORT cannot determine the input file size for a sort application and the number of records is not supplied by a FILSZ or SIZE value. Generally, DFSORT can automatically determine the input file size. However, in a few cases, such as when an E15 supplies all the input records, when information about a tape data set is not available from a tape management system or when Blockset is not selected, DFSORT cannot determine an accurate file size. In these cases, if the number of records is not supplied by a FILSZ or SIZE runtime option, and dynamic allocation of work data sets is used, DFSORT uses the DYNSPC value in effect as the approximate amount of primary space. DFSORT uses 20% of the primary space as secondary space. Although the primary space is always allocated, secondary space (up to 15 extents) is only allocated as needed.

You may want to increase or decrease the default DYNSPC value depending on the amount of disk space available for DFSORT work data sets. As a guideline, [Table 7 on page 70](#) shows the approximate primary space in cylinders allocated on a 3390 for various DYNSPC values when Blockset sorts an unknown number of 6000-byte records.

Table 7. Example of DYNSPC primary space	
DYNSPC value (megabytes)	Primary space (cylinders)
32	48
64	93
128	183
256	366
512	732

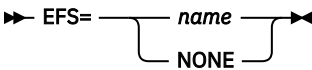
The larger your DYNSPC value, the more data DFSORT can sort when the file size is unknown. For example, in a test using the dynamically allocated work space (no Hiperspace, data space, or memory object) with the primary space shown in [Table 7 on page 70](#), and all of the corresponding secondary space, Blockset is able to sort approximately 150 megabytes with DYNSPC=32 and approximately 1200 megabytes with DYNSPC=256. If Hiperspace, data space, or a memory object can be used along with dynamically allocated work space, the amount of data DFSORT can sort increases according to the amount of Hiperspace, data space, or memory object storage available.

n
Specifies the total default primary space, in megabytes, to be allocated for all dynamically allocated work data sets (n is not the primary space for each data set). n must be a value between 1 and 65535.

Do not specify a value which exceeds the available disk space, because this causes dynamic allocation to fail for sort applications that use this default.

Default: 256

EFS



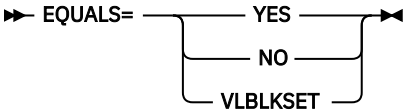
Specifies the name of a user-written Extended Function Support (EFS) program to be called by DFSORT. For full details on writing an EFS program, see [z/OS DFSORT Application Programming Guide](#).

name
Is the name of the EFS program to be called by DFSORT.

NONE
Means no EFS program is to be called.

Default: NONE

EQUALS



Specifies whether the order of records that collate identically for sort or merge applications should be preserved from input to output. For full details, see [z/OS DFSORT Application Programming Guide](#).

YES
Means the order should be preserved.

NO

Means the order need not be preserved.

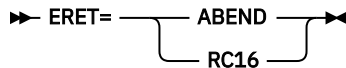
VLBLKSET

Means the order should be preserved for variable-length (VLR) Blockset applications, but need not be preserved for other applications. In other words, VLBLKSET is equivalent to EQUALS=YES for VLR Blockset applications and to EQUALS=NO for all other applications.

Default: VLBLKSET



Attention: Performance can be degraded for an application if EQUALS is in effect at run time, but is not needed. If EQUALS=YES or EQUALS=VLBLKSET is used as the installation default, specify the NOEQUALS runtime option for such applications for best performance.

ERET

Specifies the action to be taken if DFSORT encounters a critical error.

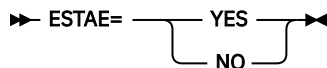
ABEND

Specifies that if your sort, merge, or copy is unsuccessful, DFSORT abends with a user completion code equal to the appropriate message number (with ABCODE=MSG), or with a specific number (n) between 1 and 99 (with ABCODE=n).

RC16

Specifies that DFSORT terminates with a return code of 16.

Default: RC16

ESTAE

Specifies whether DFSORT can use its ESTAE recovery routine for the entire run. At the beginning of each run, DFSORT establishes an ESTAE recovery routine to trap system abends or user exit routine abends for Blockset and Peerage/Vale, and to perform certain required initialization functions for the program.

If an abend occurs when DFSORT's ESTAE recovery routine is in effect, the system passes control to the recovery routine. This routine attempts to perform the following functions, as applicable, before terminating the application:

- Print additional abend information
- Continue a sort, merge, or copy application after successful SORTOUT output
- Call the EFS program at Major Calls 4 and 5 for cleanup or housekeeping
- Write an SMF record
- Call the termination exit (ICETEXIT).

If an abend occurs when DFSORT's ESTAE recovery routine is not in effect (that is, if the recovery routine was deleted), DFSORT does not attempt to perform the functions listed previously in this section.

YES

Specifies that DFSORT can use its ESTAE recovery routine for the entire run.

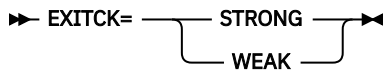
NO

Specifies that DFSORT is to delete its ESTAE recovery routine at a point early in its processing. If DFSORT terminates or abends before this point is reached, it does not delete its ESTAE recovery routine (that is, ESTAE=NO is not in effect).

Default: YES

See *z/OS DFSORT Application Programming Guide* for more information on the DFSORT ESTAE recovery routine.

EXITCK



Specifies whether DFSORT terminates or continues when it receives certain invalid return codes from E15 or E35 user exit routines. For full details of the return codes affected by this parameter, see [z/OS DFSORT Application Programming Guide](#).

STRONG

Specifies that DFSORT issues an error message and terminates when it receives an invalid return code from an E15 or E35 user exit routine.

WEAK

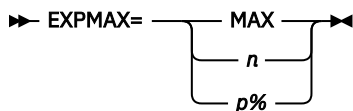
Specifies that DFSORT interprets certain invalid return codes from E15 and E35 user exit routines as valid and continue processing. Use of EXITCK=WEAK can make it difficult to detect errors in the logic of E15 and E35 user exit routines.

Default: STRONG

EXITCK=WEAK is treated like EXITCK=STRONG when:

- Tape work data sets are specified for a sort application.
- The Blockset technique is not selected for a merge application.

EXPMAX



Specifies the maximum total amount of available central storage to be used at any one time by all Hipersorting, memory object sorting and dataspace sorting applications. See the discussions of the HIPRMAX, MOSIZE, and DSPSIZE parameters for more information about these methods of sorting in memory.

Available storage is the storage that is used to back new Hiperspace, memory object and dataspace data. It includes "free" storage not currently in use by any application as well as "old" storage that is being used by other applications, but whose data has been unreferenced for a sufficiently long period of time that the system considers it eligible to be paged out to auxiliary storage to make room for new hiperspace, memory object or data space data. EXPMAX can be used to restrict Hipersorting, memory object sorting and dataspace activity to a specified portion of available storage. In order for total Hipersorting, memory object and dataspace sorting usage to reach the limit specified by EXPMAX, sufficient storage must be available on the system. When significant amounts of storage are being actively used by other applications, total usage of storage by Hipersorting, memory object and data space sorting applications can be less than that specified by EXPMAX.

EXPMAX=n specifies a fixed value for EXPMAX.

EXPMAX=p% specifies a value for EXPMAX that varies based on either the configured central storage or the available resources at run time .

- When TUNE=OLD is in effect, EXPMAX=p% causes the EXPMAX value to vary as a percentage of the configured central storage on the system at run time. If the storage on a system changes, EXPMAX=p% will cause a corresponding change in the EXPMAX value selected by DFSORT, whereas EXPMAX=n will not. When sharing DFSORT installation options between systems, such as in a sysplex, EXPMAX=p% can be used to tailor the EXPMAX value to the system selected for the application, providing a more dynamic EXPMAX value than EXPMAX=n.
- When TUNE=STOR, TUNE=DISK or TUNE=DDYN is in effect, EXPMAX=p% causes the EXPMAX value to be dynamically calculated as a percentage of the available central storage at run time. As workloads change and available resources increase or decrease, EXPMAX=p% will cause a corresponding change in the EXPMAX value whereas EXPMAX=n will not.

For more information, see [“Tailoring storage for Hipersorting, memory object sorting and dataspace sorting”](#) on page 25.

MAX

specifies that DFSORT determines dynamically the maximum total amount of available storage to be used at any one time by all Hipersorting, memory object sorting and dataspace sorting applications on the system.

n

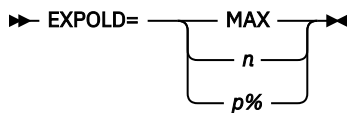
specifies that DFSORT determines dynamically the maximum total amount of available storage to be used at any one time by all Hipersorting, memory object sorting and dataspace sorting applications on the system, subject to a maximum of n MB. n must be a value between 0 and 8388606. If n is 0, Hipersorting, memory object sorting and dataspace sorting are not used.

p%

specifies that DFSORT determines dynamically the maximum total amount of available storage to be used at any one time by all Hipersorting, memory object sorting and dataspace sorting applications on the system, subject to a maximum of p percent of either the configured central storage (if TUNE=OLD is in effect) or the available resources at run time (if TUNE=OLD is not in effect). p must be a value between 0 and 100. If p is 0, Hipersorting, memory object sorting and dataspace sorting are not used. The value calculated for p% is limited to 8388606 MB.

Default: MAX

EXPOLD



Specifies the maximum total amount of old central storage to be used at any one time by all Hipersorting, memory object sorting and dataspace sorting applications. See the discussion of the HIPRMAX, MOSIZE, and DSPSIZE parameters for more information about these methods of sorting in memory.

Old storage includes only the storage that is being used by other applications, but whose data has been unreferenced for a sufficiently long period of time that the system considers it eligible to be paged out to auxiliary storage to make room for new hiperspace, memory object or data space data. EXPOLD limits the total amount of such paging at any one time caused by all Hipersorting, memory object sorting and data space sorting applications on the system. It should be set to a value other than MAX if this paging causes an unacceptable impact on the system.

EXPOLD=n specifies a fixed value for EXPOLD.

EXPOLD=p% specifies a value for EXPOLD that varies based on either the configured central storage or the amount of old storage in use by other applications at run time .

- When TUNE=OLD is in effect, EXPOLD=p% causes the EXPOLD value to vary as a percentage of the configured central storage on the system at run time. If the storage on a system changes, EXPOLD=p% will cause a corresponding change in the EXPOLD value selected by DFSORT, whereas EXPOLD=n will not. When sharing DFSORT installation options between systems, such as in a sysplex, EXPOLD=p% can be used to tailor the EXPOLD value to the system selected for the application, providing a more dynamic EXPOLD value than EXPOLD=n.
- When TUNE=STOR, TUNE=DISK or TUNE=DDYN is in effect, EXPOLD=p% causes the EXPOLD value to be dynamically calculated as a percentage of the old storage currently in use by other applications. As workloads change and the amount of old storage increases or decreases, EXPOLD=p% will cause a corresponding change in the EXPOLD value whereas EXPOLD=n will not.

As old data is paged out and replaced by data from Hipersorting, memory object sorting or data space sorting applications, the corresponding storage becomes free when the Hipersorting, memory object sorting or dataspace sorting applications terminate. At that point, more old storage can be paged out as a result of new Hipersorting, memory object sorting or dataspace sorting applications. It is possible that, over a sufficient period of time, all old data can be paged out to auxiliary storage as a result of continuous Hipersorting, memory object sorting or dataspace sorting activity, though never more at one time than the amount specified by EXPOLD.

For more information, see [“Tailoring storage for Hipersorting, memory object sorting and dataspace sorting”](#) on page 25.

MAX

specifies that DFSORT can use up to all of the old storage for Hipersorting, memory object sorting and dataspace sorting.

n

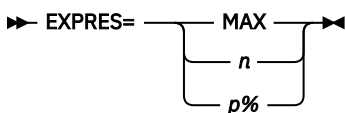
specifies that DFSORT limits the total amount of old storage to be used at any one time by all Hipersorting, memory object sorting and dataspace sorting applications on the system to a maximum of n MB. n must be a value between 0 and 8388606. If n is 0, old storage is not used for Hipersorting, memory object sorting or dataspace sorting.

p%

specifies that DFSORT limits the total amount of old storage to be used at any one time by all Hipersorting, memory object sorting and dataspace sorting applications on the system to a maximum of p percent of either the configured central storage (if TUNE=OLD is in effect) or the amount of old storage currently in use by other applications at run time (if TUNE=OLD is not in effect). p must be a value between 0 and 100. If p is 0, old storage is not used for Hipersorting, memory object sorting or dataspace sorting. The value calculated for p% is limited to 8388606 MB.

Default: 50%

EXPRES



Specifies the minimum amount of available central storage to be reserved for use by non-Hipersorting, non-memory object sorting, and non-dataspace sorting applications. This is storage that would otherwise be available for Hipersorting, memory object sorting, and dataspace sorting applications. See the discussion of the HIPRMAX, MOSIZE, and DSPSIZE parameters for more information about these methods of sorting in memory.

EXPRES defines a threshold amount of available storage below which all currently running Hipersorting, memory object sorting, and dataspace sorting applications switch from using Hiperspace, memory objects, or data space to using disk work data sets. In addition, no new DFSORT applications use Hipersorting, memory object sorting, or dataspace sorting until such time as available storage exceeds the value specified in EXPRES.

EXPRES can be used to make DFSORT more sensitive to the future needs of non-DFSORT applications. Regardless of the setting of EXPRES, DFSORT is always aware of the future needs of other concurrent Hipersorting, memory object sorting, and dataspace sorting applications, and never interferes with storage that is being actively used by non-DFSORT applications. Sudden increases in the use of storage by non-DFSORT applications, however, cannot be anticipated by DFSORT and can cause a temporary overcommitment of storage. EXPRES can be used to set up a buffer of available storage to prevent this type of overcommitment.

Setting EXPRES to a value other than 0 has the effect of decreasing Hipersorting, memory object sorting, and dataspace sorting activity on the system. This is especially noticeable when either available storage is small or EXPRES is large. In effect, EXPRES lowers the priority of DFSORT applications' access to storage versus that of other applications. The priority lowering becomes more pronounced as EXPRES is set to larger values.

EXPRES=*n* specifies a fixed value for EXPRES.

EXPRES=*p*% specifies a value for EXPRES that varies based on either the configured central storage or the available resources at run time.

- When TUNE=OLD is in effect, EXPRES=*p*% causes the EXPRES value to vary as a percentage of the configured central storage on the system at run time. If the storage on a system changes, EXPRES=*p*% will cause a corresponding change in the EXPRES value selected by DFSORT, whereas EXPRES=*n* will not. When sharing DFSORT installation options between systems, such as in a sysplex, EXPRES=*p*% can be used to tailor the EXPRES value to the system selected for the application, providing a more dynamic EXPRES value than EXPRES=*n*.
- When TUNE=STOR, TUNE=DISK or TUNE=DDYN is in effect, EXPRES=*p*% causes the EXPRES value to be dynamically calculated as a percentage of the available central storage at run time. As workloads change and the available resources increase or decrease, EXPRES=*p*% will cause a corresponding change in the EXPRES value whereas EXPRES=*n* will not.

For more information, see [“Tailoring storage for Hipersorting, memory object sorting and dataspace sorting”](#) on page 25.

MAX

specifies that DFSORT reserves all available storage for use by non-Hipersorting, non-memory object sorting, or non-dataspace sorting applications. This disables Hipersorting, memory object sorting, and dataspace sorting.

n

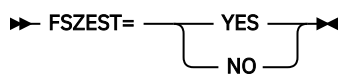
specifies that DFSORT reserves *n* MB of available storage for use by non-Hipersorting, non-memory object sorting, or non-dataspace sorting applications. *n* must be a value between 0 and 8388606. If *n* is 0, DFSORT considers all available storage eligible for Hipersorting, memory object sorting, and dataspace sorting applications.

p%

specifies that DFSORT reserves *p* percent of either the configured central storage (if TUNE=OLD is in effect) or the available resources at run time (if TUNE=OLD is not in effect) for use by non-Hipersorting, non-memory object sorting, or non-dataspace sorting applications. *p* must be a value between 0 and 100. If *p* is 0, DFSORT considers all available storage eligible for Hipersorting, memory object sorting, and dataspace sorting applications. The value calculated for *p*% is limited to 8388606 MB.

Default: 10%

FSZEST



Specifies whether DFSORT is to treat exact filesize values as estimates.

YES

Specifies that DFSORT is to treat exact filesize values as estimates. If FILSZ/SIZE=*n* is specified at runtime, *n* is treated as an estimated value (as if FILSZ/SIZE=*En* was specified). This prevents DFSORT applications from terminating when the number of records is not exactly equal to *n*.

NO

Specifies that DFSORT is not to treat exact filesize values as estimates. If FILSZ/SIZE=*n* is specified at runtime, *n* is treated as an exact value. This causes DFSORT applications to terminate when the number of records is not exactly equal to *n*.

1. FSZEST has no effect on FILSZ/SIZE=*Un* processing.
2. The FSZEST option is intended to ease migration to DFSORT from other sort products and is not recommended for any other purpose.

Default: NO

GENER

►► GENER= — *name* ►►

Specifies the name that ICEGENER is to use to transfer control to the IEBGENER system utility. The default, GENER=IEBGENR, corresponds to the "standard" alias for IEBGENER. You should change it only if you have a special reason for doing so.

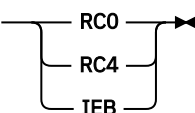
name

Specifies the name or alias of your IEBGENER system utility module (1-8 characters).

Default: IEBGENR

GENER is ignored when it is specified with the JCL or TSO parameter.

GNPAD

►► GNPAD= —  ►►

Specifies the action to be taken by ICEGENER when the SYSUT2 LRECL is larger than the SYSUT1 LRECL (LRECL padding).

In some cases, IEBGENER terminates when the SYSUT2 LRECL is larger than the SYSUT1 LRECL. ICEGENER handles LRECL padding in one of three ways depending on the GNPAD value you set.

RC0

Specifies that ICEGENER should issue message ICE202I, set a return code of 0, and continue processing when the SYSUT2 LRECL is larger than the SYSUT1 LRECL.

RC4

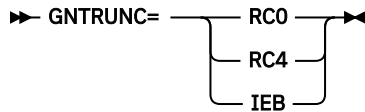
Specifies that ICEGENER should issue message ICE202I, set a return code of 4, and continue processing when the SYSUT2 LRECL is larger than the SYSUT1 LRECL.

IEB

Specifies that ICEGENER should transfer control to IEBGENER when the SYSUT2 LRECL is larger than the SYSUT1 LRECL.

Default: RC0

1. GNPAD is ignored when it is specified with the JCL or TSO parameter.
2. The return code of 0 or 4 set for LRECL padding can be overridden by a higher return code set for some other reason.
3. If SYSUT1 or SYSUT2 is not present, or is a VSAM data set, ICEGENER does not check for LRECL padding.

GNTRUNC

Specifies the action to be taken by ICEGENER when the SYSUT2 LRECL is smaller than the SYSUT1 LRECL (LRECL truncation).

In some cases, IEBGENER terminates when the SYSUT2 LRECL is smaller than the SYSUT1 LRECL. ICEGENER handles LRECL truncation in one of three ways depending on the GNTRUNC value you set.

RC0

Specifies that ICEGENER should issue message ICE202I, set a return code of 0, and continue processing when the SYSUT2 LRECL is smaller than the SYSUT1 LRECL.

RC4

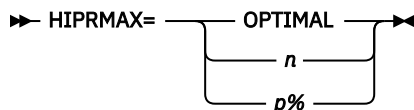
Specifies that ICEGENER should issue message ICE202I, set a return code of 4, and continue processing when the SYSUT2 LRECL is smaller than the SYSUT1 LRECL.

IEB

Specifies that ICEGENER should transfer control to IEBGENER when the SYSUT2 LRECL is smaller than the SYSUT1 LRECL.

Default: RC0

1. GNTRUNC is ignored when it is specified with the JCL or TSO parameter.
2. The return code of 0 or 4 set for LRECL truncation can be overridden by a higher return code set for some other reason.
3. If SYSUT1 or SYSUT2 is not present, or is a VSAM data set, ICEGENER does not check for LRECL truncation.

HIPRMAX

Specifies the maximum amount of Hiperspace to be committed for Hipersorting. A Hiperspace is a high-performance data space that resides in central storage. Because I/O processing is reduced for Hipersorting, elapsed time, EXCP counts, and channel usage are also reduced.

There are several factors that can limit the amount of Hiperspace used by an application:

1. The IEFUSI exit can limit the total amount of Hiperspace, memory objects, and data space available to an application.
2. HIPRMAX can limit the amount of Hiperspace available to an application, as detailed in the following.
3. Sufficient available storage must be present to back DFSORT's Hiperspaces. *Available storage* is the storage that is used to back new Hiperspace data, and consists of the following two types:
 - a. *Free storage* is storage that is not being used by any application.
 - b. *Old storage* is storage that is being used by another application, but whose data has been unreferenced for a sufficiently long period such that the system considers it eligible to be paged out to auxiliary storage to make room for new Hiperspace data.

The amount of available storage changes constantly, depending on current system activity. Consequently, DFSORT checks the available storage level throughout a Hipersorting application and switches from Hiperspace to work data sets if the available storage level gets too low.

4. Other concurrent Hipersorting, memory object sorting, and dataspace sorting applications further limit the amount of available storage. A Hipersorting application is aware of the storage needs of every other Hipersorting, memory object sorting, and dataspace sorting application on the system, and does not attempt to back its Hiperspace data with storage needed by another Hipersorting, memory object sorting, or dataspace sorting application. This prevents overcommitment of storage resources in the event of multiple large concurrent Hipersorting, memory object sorting, and dataspace sorting applications starting at similar times on the same system.
5. The installation options EXPMAX, EXPOLD, and EXPRES can also be used to further limit the amount of storage available to Hipersorting applications. EXPMAX limits the total amount of available storage that can be used at any one time to back DFSORT Hiperspaces, memory objects, and data spaces. EXPOLD limits the total amount of old storage that can be used at any one time to back DFSORT Hiperspaces, memory objects, and data spaces. EXPRES sets aside a specified amount of available storage for use by non-Hipersorting, non-memory object, and non-data space sorting applications.

Some of these limits depend on system and other DFSORT activity during the time a Hipersorting application runs. Consequently, the amount of Hiperspace a Hipersorting application uses can vary from run to run.

HIPRMAX=n specifies a fixed value for HIPRMAX. HIPRMAX=p% specifies a value for HIPRMAX that varies as a percentage of the configured central storage on the system at run time. If the storage on a system changes, HIPRMAX=p% will cause a corresponding change in the HIPRMAX value selected by DFSORT, whereas HIPRMAX=n will not. When sharing DFSORT installation options between systems, such as in a sysplex, HIPRMAX=p% can be used to tailor the HIPRMAX value to the system selected for the application, providing a more dynamic HIPRMAX value than HIPRMAX=n.

For additional information, see [“Tailoring storage for Hipersorting, memory object sorting and dataspace sorting”](#) on page 25.

If the amount of Hiperspace available for Hipersorting is insufficient for temporary storage of the records, intermediate disk storage is used along with Hiperspace. If the amount of Hiperspace is too small to improve performance, Hipersorting is not used. When Hipersorting is enabled, DFSORT changes DYNAUTO=NO to DYNAUTO=YES in some cases.

Hipersorting might cause a small CPU time degradation. When CPU optimization is a concern, you can use HIPRMAX=0 to suppress Hipersorting.

OPTIMAL

Specifies that DFSORT determines dynamically the maximum amount of Hiperspace to be used for Hipersorting.

n

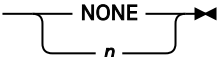
Specifies that DFSORT determines dynamically the maximum amount of Hiperspace to be used for Hipersorting, subject to a limit of n MB. n must be a value between 0 and 32767. If n is 0, Hipersorting is not used.

p%

Specifies that DFSORT determines dynamically the maximum amount of Hiperspace to be used for Hipersorting, subject to a limit of p percent of the configured central storage. p must be a value between 0 and 100. If p is 0, Hipersorting is not used. The value calculated for p% is limited to 32767 MB, and is rounded down to the nearest MB.

Default: OPTIMAL

IDRCPCT

► IDRCPCT=  NONE
n

Specifies a percentage which represents the approximate amount of data compaction achieved by using the Improved Data Recording Capability feature of IBM tape devices that support compaction. DFSORT can estimate the file size for input tape data sets managed by DFSMSrmm, but uses the

IDRPCT value to estimate the file size for compacted input tape data sets that are not managed by DFSMSrmm.

The specified IDRPCT percentage is relative to the standard (non-compacted) capacity of a single volume of tape of that particular type. The standard capacity of 3480, 3490, and 3590 tapes are 200 MB, 800 MB, and 4000 MB, respectively. DFSORT will compute the amount of data per volume of compacted tape as follows:

$$\text{standard capacity} * n = \text{compacted capacity}$$

where n is the IDRPCT percentage.

The IDRPCT parameter is designed to be an installation-wide safety net, to help DFSORT avoid dynamic work data set allocation problems for the majority of tape jobs in an installation. Failure of individual jobs as a result of uncharacteristic compaction ratios for particular data sets should be addressed by specifying a SIZE or FILSZ value for those jobs (see [z/OS DFSORT Application Programming Guide](#) for details).

NONE

Specifies that the compaction ratio is unknown. DFSORT will be unable to estimate the input file size for any compacted SORTIN tape data sets.



Attention: This can cause termination with message ICE046A when DYNALLOC is in effect.

n

Specifies the compaction factor percentage as a decimal value within the range of 100 (uncompacted) to 1000 (a compaction ratio of 10 to 1).

Default: NONE

Example 1

```
IDRPCT=100
```

This has the effect of treating a compacted tape exactly the same as a non-compacted tape with regard to estimating the amount of data on that tape. Whenever a compacted tape is used as input, DFSORT will assume that it contains 200 MB (3480), 800 MB (3490), or 4000 MB (3590) of data (the same amount as is assumed to reside on a non-compacted tape of that type). The capacity computations are as follows:

for 3480s --	200MB	*	100 % =	200MB
for 3490s --	800MB	*	100 % =	800MB
for 3590s --	4000MB	*	100 % =	4000MB

Example 2

```
IDRPCT=350
```

This has the effect of treating a single volume of compacted tape as 3.5 volumes of standard tape. The percentage of 350 indicates that the approximate ratio of compaction achieved by IDRC was 3.5 to 1. The capacity computations are as follows:

for 3480s --	200MB	*	350 % =	700MB
for 3490s --	800MB	*	350 % =	2800MB
for 3590s --	4000MB	*	350 % =	14000MB

A suggested method for determining the appropriate IDRPCT value is to use the data gathered from the tape library's conversion to IDRC tapes. For example, if a library consisting of 1000 standard 3480 cartridges contained only 200 compacted 3480 cartridges after conversion, the approximate compaction ratio is 5 to 1, and IDRPCT=500 is a good starting value.

Typically, compaction ratios fall between 2 to 1 and 3 to 1. If no tape library conversion data is available, a good starting value for IDRPCT would be between 200 and 300.

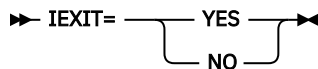
The starting IDRCPCT value selected may need to be tuned up or down according to the under- or over-allocation of dynamically allocated work data set space observed for DFSORT jobs with compacted tape input.

Note:

1. The IDRCPCT percentage is only an estimate of the amount of compaction that has occurred installation wide; it can be overridden for an individual job by specifying a FILSZ or SIZE value. Any over-estimates or under-estimates of the input file size due to the provided IDRCPCT value can be corrected in this manner for individual jobs.
2. For concatenated tape data sets, DFSORT will only determine the actual compaction status of the first data set in the concatenation. This is a result of opening only the first data set (to avoid unnecessary tape mounts).

All subsequent tape data sets in the concatenation will be treated as compacted only if the device that is allocated to the particular data set supports IDRC (compaction). Therefore, it is possible that DFSORT will make an incorrect estimate when a non-compacted tape resides on an IDRC device.
3. DFSORT will always assume full tape volumes when estimating the work data set space to allocate. For example, if a 4 MB data set resides on a standard 3490 cartridge, the amount of work space will be over-allocated two-hundred-fold, since the capacity of a 3490 is 800 MB. This type of over-allocation increases with the use of IDRCPCT and compacted tape data sets. For example, if the 4 MB data set resides on a compacted 3490 and IDRCPCT=350 is specified, the amount of over-allocation is seven-hundred-fold, since DFSORT assumes that the tape contains 2800 MB (3.5*800 MB) of data.

IEXIT



Specifies whether DFSORT is to pass control to an initialization exit (ICEIEXIT) as described in Chapter 5, “Controlling resources using ICEIEXIT,” on page 121.

YES

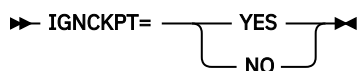
Specifies that DFSORT is to pass control to the ICEIEXIT routine.

NO

Specifies that DFSORT is not to pass control to an ICEIEXIT routine.

Default: NO

IGNCKPT



Specifies the action to be taken if checkpoint/restart is requested for a sort application at run time and the Blockset technique (which does not support checkpoint/restart) can be used.

YES

directs DFSORT to use the Blockset technique and ignore the checkpoint/restart request.

NO

directs DFSORT not to use the Blockset technique and honor the checkpoint/restart request.

Default: YES

IOMAXBF

►► IOMAXBF= *n* ◄◄

Specifies the maximum buffer space DFSORT can use for SORTIN, SORTINnn, and SORTOUT data sets on disk. This, in turn, will limit the amount of data that can be transferred in a single I/O operation. The actual amount of buffer space used for the SORTIN, SORTINnn, and SORTOUT data sets is subject to the IOMAXBF limit as well as the system storage limits (for example, IEFUSI and REGION) and the DFSORT storage limits (that is, SIZE, MAXLIM and TMAXLIM). DFSORT attempts to use storage above 16 MB virtual for SORTIN, SORTINnn, and SORTOUT data set processing whenever possible.

DFSORT uses the IOMAXBF value as:

- the limit for the SORTIN buffers (sort and copy applications)
- the limit for the total of all SORTINnn buffers (merge applications)
- the limit for the SORTOUT buffers (sort, copy, and merge applications).

For example, if IOMAXBF is set to 8388608 (8 MB), then at run time DFSORT uses:

- Up to 8 MB for the SORTIN buffers for sort and copy applications
- Up to 8 MB for all of the SORTINnn buffers for merge applications
- Up to 8 MB for the SORTOUT buffers for sort, copy, and merge applications.

The IOMAXBF value can be adjusted up or down to fit the needs of your devices and system load. At a transfer rate of 17 MB per second, the IBM shipped default of 35651584 (34 MB) for IOMAXBF engages the channel for a maximum of about 2 seconds at any one time. This is well below the IBM supplied missing interrupt handler (MIH) value of 30 seconds for most disk devices. Exceeding this MIH value can result in I/O terminations of DFSORT applications. With higher MIH values, or higher transfer rates, it is possible to use higher values of IOMAXBF without risk of MIH termination.

The IOMAXBF value can be reduced or increased using the following formula as a guideline:

$$\text{IOMAXBF} = (\text{transfer rate per second}) * (\text{channel seconds allowed})$$

For example, if you have a channel transfer rate of 17 MB per second, and you wish to limit DFSORT to a maximum channel connect time of 3 seconds, set IOMAXBF to 53477376 (51 MB).

Lowering IOMAXBF below the default value of 35651584 (34 MB) can cause performance degradation.

n

Specifies the maximum bytes of storage to be used for the SORTIN, SORTINnn, and SORTOUT data set buffers as a decimal value within the range of 8388608 (8 MB) to 268435456 (256 MB).

Default: 35651584 (34 MB)

Example 1

```
IOMAXBF=53477376
```

At run time, DFSORT uses up to 51 MB for the SORTIN buffers for sort and copy applications, up to 51 MB for all of the SORTINnn buffers for merge applications, and up to 51 MB for the SORTOUT buffers for sort, copy, and merge applications.

At a transfer rate of 17 MB per second, this IOMAXBF value limits DFSORT to a maximum connect time of 3 seconds.

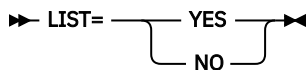
Example 2

```
IOMAXBF=20971520
```

At run time, DFSORT uses up to 20 MB for the SORTIN buffers for sort and copy applications, up to 20 MB for all of the SORTINnn buffers for merge applications, and up to 20 MB for the SORTOUT buffers for sort, copy, and merge applications.

At a transfer rate of 4 MB per second, this IOMAXBF value limits DFSORT to a maximum connect time of 5 seconds.

LIST



Specifies whether DFSORT program control statements are printed. For full details on use of the message data set, see [z/OS DFSORT Messages, Codes and Diagnosis Guide](#).

YES

Specifies that DFSORT control statements are printed (on the message data set).

NO

Specifies that DFSORT control statements are not printed.

Default: YES

LISTX



Specifies whether to print control statements returned by an EFS program. For full details on use of the message data set, see [z/OS DFSORT Messages, Codes and Diagnosis Guide](#).

YES

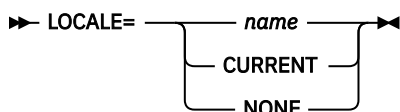
Specifies that program control statements returned by an EFS program are printed (on the message data set).

NO

Specifies that control statements returned by an EFS program are not printed.

Default: YES

LOCALE



Specifies whether locale processing is to be used and if so, designates the active locale.

DFSORT's collating behavior can be modified according to your cultural environment. Your cultural environment is defined to DFSORT using the X/Open locale model. A locale is a collection of data grouped into categories that describes the information about your cultural environment.

The collate category of a locale is a collection of sequence declarations that defines the relative order between collating elements (single character and multi-character collating elements). The sequence declarations define the collating rules.

If locale processing is to be used, the active locale will affect the behavior of DFSORT's SORT, MERGE, INCLUDE, and OMIT functions. For SORT and MERGE, the active locale will only be used to process character (CH) control fields. For INCLUDE and OMIT, the active locale will only be used to process character (CH) compare fields, and character and hexadecimal constants compared to character (CH) compare fields.

Note: Locale processing is not used for IFTHEN WHEN constants or compare fields.

name

specifies that locale processing is to be used and designates the name of the locale to be made active during DFSORT processing.

The locales are designated using a descriptive name. For example, to set the active locale to represent the French language and the cultural conventions of Canada, specify `LOCALE=FR_CA`. Up to 32 characters can be specified for the descriptive locale name. The locale names themselves are not case sensitive. See "Locale naming conventions" in *z/OS XL C/C++ Programming Guide* for a complete description of the naming conventions for locales.

You can use IBM-supplied and user-defined locales.

The state of the active locale prior to DFSORT being entered will be restored when DFSORT completes processing.

CURRENT

specifies that locale processing is to be used, and the current locale active when DFSORT is entered will remain the active locale during DFSORT processing.

NONE

specifies that locale processing is not to be used. DFSORT will use the binary encoding of the code page defined for your data for collating and comparing.

Default: NONE

Note:

- 1.
2. `LOCALE=name` or `LOCALE=CURRENT` can cause DFSORT to run differently than it does with the IBM-supplied default `LOCALE=NONE`. In particular, locale processing may require more resources (for example storage and work space), handle fewer control fields, and degrade performance. As a result, you may only want to set `LOCALE=name` or `LOCALE=CURRENT` on an individual application, rather than installation-wide, basis.
- 3.
4. DFSORT locale processing may require an additional amount of storage that depends on the environment supporting the locale, as well as the locale itself. It may be necessary to specify a REGION of several MB or more for DFSORT applications that use locale processing.
5. To use an IBM supplied locale, DFSORT must have access to the Language Environment runtime library containing the dynamically loadable locales. For example, the library containing them might be called `SYS1.SCEERUN`. If you are unsure of the name of the library where the dynamically loadable locales are installed at your location, contact your system administrator. To use a user-defined locale, DFSORT must have access to the library where the dynamically loadable locale is contained.
6. If locale processing is used for SORT, MERGE, INCLUDE, or OMIT fields:
 - `VLSHRT` is not used for SORT or MERGE
 - `INREC` or an E61 user exit must not be used for SORT or MERGE.
 - `CHALT` or an EFS program must not be used.
7. Locale processing is not used for IFTHEN WHEN constants or compare fields.
8. Locale processing for DFSORT's SORT, MERGE, INCLUDE, and OMIT functions can improve performance relative to applications which perform pre- and/or post-processing of data to produce the desired collating results. However, locale processing should only be used when required since it can show degraded performance relative to collation using binary encoding of the code page defined for your data.

MAXLIM

➤ MAXLIM= *n* ➤

Specifies an upper limit to the total amount of main storage, below 16 MB virtual, available to DFSORT. When Blockset is selected, MAXLIM is used to limit the storage below 16 MB virtual, and SIZE/MAINSIZE is used to limit the total storage above and below 16 MB virtual. When Blockset is not selected, MAXLIM is only used when SIZE=MAX or MAINSIZE=MAX is in effect. For more information, see [“Tailoring main storage” on page 14](#).

n

Specifies the number of bytes as a decimal value within the range of 90112 (88 KB) to 16777208 (16 MB minus 8 bytes), but must not be greater than the TMAXLIM value. A value of at least 200 KB is recommended for MAXLIM (a lower value may cause jobs to fail due to insufficient storage below 16 MB virtual). Lowering MAXLIM below 1 MB can cause performance degradation for some applications. This performance degradation can become more pronounced and affect more applications the more MAXLIM is lowered below 1 MB. If the value specified is not a multiple of 8, it is rounded down to the next multiple of 8.

Default: 1048576 (1 MB)

Note:

1. The MAXLIM limit is not observed for a copy or Blockset merge application when the use of more storage below 16 MB virtual would improve performance.
2. The MAXLIM limit is not observed for UTFIL processing.

MINLIM

➡ MINLIM= — *n* ➡

Specifies a lower limit to the amount of main storage available to DFSORT when SIZE=*n* or MAINSIZE=*n* is in effect. If the requested minimum amount of main storage is not available: DFSORT might still run if at least 88 KB of storage is available below 16 MB virtual. For more information, see [“Tailoring main storage” on page 14](#).

n

Specifies the number of bytes as a decimal value. The minimum value is 90112 (88 KB). Lowering MINLIM below 440 KB can degrade performance for some applications. This performance degradation can become more pronounced and affect more applications the more MINLIM is lowered below 440 KB. If the value specified is not a multiple of 8, it is rounded down to the next multiple of 8.

If the MAXLIM value specified in ICEMAC is less than the MINLIM value specified in ICEMAC, the MINLIM value is set to the MAXLIM value in the ICEAMx or ICETDx module. The MINLIM value specified in an ICEPRMxx member is not adjusted based on the MAXLIM value.

If the MAXLIM value in effect at run time is less than the MINLIM value in effect at run time, the MINLIM value is set to the MAXLIM value at runtime .

Default: 450560 (440 KB)

MOSIZE

➡ MOSIZE= — MAX —
 — *n* —
 — *p%* — ➡

Specifies the total amount of memory object storage to be used for memory object sorting. A memory object is a data area in virtual storage that is allocated above the bar and backed by central storage. Because I/O processing is reduced for memory object sorting, elapsed time, EXCP counts, and channel usage are also reduced.

Note: The "bar" refers to the 2-gigabyte address within the 64-bit address space. The bar separates storage below the 2-gigabyte address called "below the bar", from storage above the 2-gigabyte address called "above the bar".

There are several factors that can limit the total memory object storage used by an application:

1. The MEMLIMIT parameter on the JOB or EXEC JCL statement can limit the total number of usable virtual pages above the bar in a single address space.
2. In addition to limiting the amount of Hiperspace and data space available to an application, the IEFUSI exit can also limit the total number of usable virtual pages above the bar in a single address space.
3. MOSIZE can limit the total amount of memory object storage available to an application, as detailed in the following.
4. Sufficient available central storage must be present to back DFSORT's memory object storage. The amount of available central storage changes constantly, depending on current system activity. Consequently, DFSORT checks the amount of available central storage throughout a memory object sorting run and switches from using a memory object to using disk work data sets if the available central storage is too low.
5. Other concurrent Hipersorting, memory object sorting, and dataspace sorting applications further limit the amount of available storage. A memory object sorting application keeps track of the storage needs of all other Hipersorting, memory object sorting, and dataspace sorting applications on the system, and does not attempt to back its memory object storage with storage needed by another Hipersorting, memory object sorting, or dataspace sorting application. This prevents overcommitment of storage resources in the event of multiple large concurrent Hipersorting, memory object sorting, and dataspace sorting applications starting at similar times on the same system.
6. The installation options EXPMAX, EXPOLD, and EXPRES can also be used to further limit the amount of storage available to memory object sorting applications. EXPMAX limits the total amount of available storage that can be used at any one time to back DFSORT Hiperspaces, memory objects, and data spaces. EXPOLD limits the total amount of old storage that can be used at any one time to back DFSORT Hiperspaces, memory objects, and data spaces. EXPRES sets aside a specified amount of available storage for use by non-Hipersorting, non-memory object sorting, and non-dataspace sorting applications.

Some of these limits depend on system and other DFSORT activity during the time that a memory object sorting application runs. Consequently, the total amount of memory object storage that a memory object sorting application uses can vary from run to run.

MOSIZE=n specifies a fixed value for MOSIZE. MOSIZE=p% specifies a value for MOSIZE that varies as a percentage of the available central storage on the system at run time. If the available central storage on a system changes, MOSIZE=p% will cause a corresponding change in the MOSIZE value selected by DFSORT, whereas MOSIZE=n will not. When sharing DFSORT installation options between systems, such as in a sysplex, MOSIZE=p% can be used to tailor the MOSIZE value to the system selected for the application, providing a more dynamic MOSIZE value than MOSIZE=n.

For additional information, see [“Tailoring storage for Hipersorting, memory object sorting and dataspace sorting”](#) on page 25.

If the total memory object storage available for memory object sorting is insufficient for temporary storage of the records, intermediate disk storage can be used along with the memory object. When memory object sorting is enabled, DFSORT changes DYNAUTO=NO to DYNAUTO=YES in some cases.

MAX

Specifies that DFSORT determines dynamically the maximum size of a memory object to be used for memory object sorting. In this case, DFSORT bases its memory object usage on the size of the file being sorted and the central storage usage activity.

n

Specifies that DFSORT determines dynamically the maximum size of a memory object to be used for memory object sorting, subject to a limit of n MB. n must be a value between 0 and

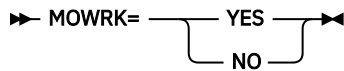
2147483646. The actual size used does not exceed n , but may be less, depending on the size of the file being sorted and the central storage usage activity on the system. If n is 0, memory object sorting is not used.

p%

Specifies that DFSORT determines dynamically the maximum size of a memory object to be used for memory object sorting, subject to a limit of p percent of the available central storage. p must be a value between 0 and 100. If p is 0, memory object sorting is not used. The value calculated for $p\%$ is limited to 2147483646 MB, and is rounded down to the nearest MB.

Default: MAX

MOWRK



Specifies whether the memory object storage available to DFSORT for memory object sorting can be used as intermediate work space. DFSORT has the capability of using memory object storage as intermediate work space (similar to the way Hiperspace is used but more efficient), or as an extension of main storage. Using memory object storage as intermediate work space is the preferred and recommended choice, but can be disabled, if appropriate.

YES

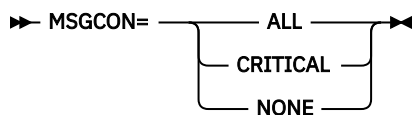
Specifies that memory object storage can be used as intermediate work space, or as an extension of main storage, as appropriate.

NO

Specifies that memory object storage can only be used as an extension of main storage.

Default: YES

MSGCON



Specifies the class of messages to be written to the master console. To learn how to set up DFSORT so that DFSORT writes to the master console, see [z/OS DFSORT Messages, Codes and Diagnosis Guide](#).

ALL

Specifies that all messages, except parameter-in-effect messages (ICE127I to ICE133I and ICE235I) and diagnostic messages (ICE800I to ICE999I) are written to the master console.

CRITICAL

Specifies that only critical messages are written to the master console.

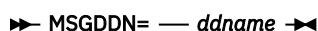
NONE

Specifies that messages are not written to the master console.

Default: NONE

Note: The value specified for MSGCON has no effect on messages ICE097I or ICE158A. These messages are always written to the master console.

MSGDDN



Specifies an alternate ddname for the message data set. MSGDDN must be in effect if:

- A program that invokes DFSORT uses SYSOUT and you do not want DFSORT messages mixed with program messages.
- The invoking program is written in COBOL and you do not want DFSORT messages mixed with COBOL messages.
- A program invokes DFSORT more than once and you want separate messages for each invocation of DFSORT.
- You have E15 or E35 user exit routines written in COBOL and you do not want DFSORT messages mixed with COBOL messages.

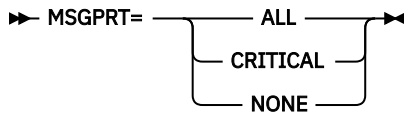
ddname

Specifies a name of 1 to 8 characters. The name must be unique within the job step. Do not use a name that is used by DFSORT (for example, SORTIN).

If the ddname specified is not available at run time, SYSOUT is used instead. For full details on use of the message data set, see [z/OS DFSORT Messages, Codes and Diagnosis Guide](#).

Default: SYSOUT

MSGPRT



Specifies the class of messages to be printed. Messages are written to the message data set. For full details on use of the message data set, see [z/OS DFSORT Messages, Codes and Diagnosis Guide](#).

ALL

Specifies that all messages except diagnostic messages (ICE800I to ICE999I) are printed. Control statements are printed only if LIST is in effect (that is, if LIST is the parameter actually chosen at run time by DFSORT). For override details, see [z/OS DFSORT Application Programming Guide](#)

CRITICAL

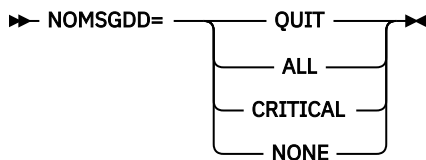
Specifies that only critical messages are printed. Control statements are printed only if LIST is in effect.

NONE

Specifies that messages and control statements are not printed.

Default: ALL

NOMSGDD



Specifies whether DFSORT terminates or continues when the message data set is required but not present. If DFSORT is to continue without using the message data set, this parameter also specifies the class of messages to be written to the master console. (For details about the message data set, see the description for MSGDDN earlier in this chapter. For details about console message classes, see the description for MSGCON earlier in this chapter.)

QUIT

Specifies that DFSORT issues an error message to the console and terminates.

ALL

Specifies that DFSORT continues processing using the equivalent of the MSGCON=ALL parameter.

CRITICAL

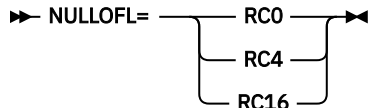
Specifies that DFSORT continues processing using the equivalent of the MSGCON=CRITICAL parameter.

NONE

Specifies that DFSORT continues processing using the equivalent of the MSGCON=NONE parameter.

Default: QUIT

NULLOFL



Specifies the action to be taken by DFSORT when there are no data records for an OUTFIL data set, as indicated by a DATA count of 0 in message ICE227I. OUTFIL report records have no effect on the action taken as a result of this option.

RC0

Specifies that DFSORT should issue message ICE174I, set a return code of 0, and continue processing when there are no data records for an OUTFIL data set.

RC4

Specifies that DFSORT should issue message ICE174I, set a return code of 4, and continue processing when there are no data records for an OUTFIL data set.

RC16

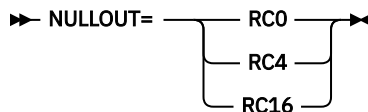
Specifies that DFSORT should issue message ICE209A, terminate, and give a return code of 16 when there are no data records for an OUTFIL data set.

Default: RC0

Note:

1. The return code of 0 or 4 set when there are no data records for an OUTFIL data set can be overridden by a higher return code set for some other reason.
2. If a NULLOFL value is not specified on an OUTFIL statement, the NULLOFL installation default applies to the OUTFIL data sets for that OUTFIL statement, that is, you can use the OUTFIL statement NULLOFL parameter to override the NULLOFL installation default for specific OUTFIL data sets.
3. NULLOUT applies to the SORTOUT data set. NULLOFL applies to OUTFIL data sets.

NULLOUT



Specifies the action to be taken by DFSORT when there are no records for the SORTOUT data set, as indicated by an OUT count of 0 in message ICE054I.

RC0

Specifies that DFSORT should issue message ICE173I, set a return code of 0, and continue processing when there are no records for the SORTOUT data set.

RC4

Specifies that DFSORT should issue message ICE173I, set a return code of 4, and continue processing when there are no records for the SORTOUT data set.

RC16

Specifies that DFSORT should issue message ICE206A, terminate, and give a return code of 16 when there are no records for the SORTOUT data set.

Default: RC0

Note:

1. The return code of 0 or 4 set when there are no records for the SORTOUT data set can be overridden by a higher return code set for some other reason.
2. NULLOUT does not apply when SORTOUT is not present, when tape work data sets are specified for a sort application, or when the Blockset technique is not selected for a merge application. DFSORT does not check if there are no records for the SORTOUT data set in these cases.
3. NULLOUT applies to the SORTOUT data set. NULLOFL applies to OUTFIL data sets
4. For an ICEGENER application, NULLOUT applies to the SYSUT2 data set if DFSORT copy is used. Note that ICEGENER passes back return code 12 instead of return code 16.

ODMAXBF

►► ODMAXBF= — *n* ◄◄

Specifies the maximum buffer space DFSORT can use for each OUTFIL data set. The actual amount of buffer space used for a particular OUTFIL data set will not exceed the ODMAXBF limit, but can be less than the limit. OUTFIL processing is supported by the Blockset technique for sort, copy and merge applications.

The storage used for OUTFIL processing is adjusted automatically according to the total storage available, the storage needed for non-OUTFIL processing and the number of OUTFIL data sets and their attributes (for example, block size). OUTFIL processing is subject to the ODMAXBF limit in effect and the system storage limits (for example, IEFUSI), but not to the DFSORT storage limits (that is, SIZE, MAXLIM and TMAXLIM). DFSORT attempts to use storage above 16 MB virtual for OUTFIL processing whenever possible.

The default value of 2097152 (2 MB) is recommended for ODMAXBF. Lowering ODMAXBF can cause performance degradation for some applications. Raising ODMAXBF can improve EXCPs for some applications, but can also increase the amount of storage needed.

n

Specifies the maximum bytes of buffer space to be used for each OUTFIL data set as a decimal value within the range of 262144 (256 KB) to 16777216 (16 MB).

Default: 2097152 (2 MB)

OUTREL

►► OUTREL= — YES — ◄◄
 NO —

Specifies whether unused temporary output data set space is to be released.

YES

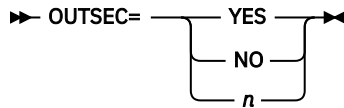
Specifies that unused space is to be released. This parameter is not used for a tape work data set sort or Conventional merge.

NO

Specifies that unused space is not to be released.

Default: YES

OUTSEC



Specifies whether DFSORT uses automatic secondary allocation for output data sets that are temporary or new. This parameter is not used for a tape work data set sort or Conventional merge.

YES

Specifies that automatic secondary allocation be used for temporary or new output data sets, and that 25 percent of the primary allocation be used as the secondary allocation.

NO

Specifies that automatic secondary allocation not be used for temporary or new output data sets.

n

Specifies that automatic secondary allocation be used for temporary or new output data sets, and that n percent of the primary allocation be used as secondary allocation. The value n must be between 1 and 1000.

Default: YES

Note: OUTSEC=0 is equivalent to OUTSEC=NO.

OVERRGN



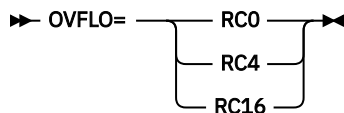
Specifies the amount of main storage, below 16 MB virtual, above the REGION value that Blockset attempts to obtain from the area defined by the difference between the system's IEFUSI region limit and region size values (by default, this difference is 64 KB). For more information, see ["Tailoring main storage" on page 14](#).

n

Specifies the number of bytes as a decimal value within the range 0 to 16515072 (16128 KB). If the value specified is not a multiple of 4096 (4 KB), it is rounded up to the next multiple of 4 KB.

Default: 65536 (64 KB) if the JCL, TSO, TD1, TD2, TD3, or TD4 parameter is specified; 16384 (16 KB) if the INV or TSOINV parameter is specified.

OVFLO



Specifies the action to be taken by DFSORT when BI, FI, PD or ZD summary fields overflow.

RC0

Specifies that DFSORT should issue message ICE152I (once), set a return code of 0, and continue processing when summary fields overflow. The pair of records involved in a summary overflow is left unsummed, and neither record is deleted. Summary overflow does not prevent further summation.

RC4

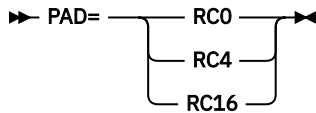
Specifies that DFSORT should issue message ICE152I (once), set a return code of 4, and continue processing when summary fields overflow. The pair of records involved in a summary overflow is left unsummed, and neither record is deleted. Summary overflow does not prevent further summation.

RC16

Specifies that DFSORT should issue message ICE195A, terminate, and give a return code of 16 when summary fields overflow.

Default: RC0

Note: The return code of 0 or 4 set for summary overflow can be overridden by a higher return code set for some other reason.

PAD

Specifies the action to be taken by DFSORT when the SORTOUT LRECL is larger than the SORTIN/SORTINnn LRECL, for the cases where DFSORT allows LRECL padding.

RC0

Specifies that DFSORT should issue message ICE171I, set a return code of 0, and continue processing when the SORTOUT LRECL is larger than the SORTIN/SORTINnn LRECL.

RC4

Specifies that DFSORT should issue message ICE171I, set a return code of 4, and continue processing when the SORTOUT LRECL is larger than the SORTIN/SORTINnn LRECL.

RC16

Specifies that DFSORT should issue message ICE196A, terminate, and give a return code of 16 when the SORTOUT LRECL is larger than the SORTIN/SORTINnn LRECL.

Default: RC0

Note:

1. The return code of 0 or 4 set for LRECL padding can be overridden by a higher return code set for some other reason.
2. For an ICEGENER application, the GNPAD value is used, and the PAD value is ignored.
3. For some LRECL padding situations (for example, a tape work data set sort), DFSORT issues message ICE043A and terminates with a return code of 16. The PAD value has no effect in these cases.
4. If SORTIN/SORTINnn or SORTOUT is not present, or is a VSAM data set, DFSORT does not check for LRECL padding.
5. DFSORT does not check OUTFIL data sets for LRECL padding.

PARMDDN

►► PARMDDN= — *ddname* ►►

Specifies an alternate ddname for the DFSORT DFSPARM data set. If this ddname is present at run time, DFSORT uses it to override parameters from other sources.

ddname

Specifies a name of 1 to 8 characters. The name must be unique within the job step. Do not use a name that is used by DFSORT (for example, SYSIN).

Default: DFSPARM

Note:

1. When PARMDDN=DFSPARM is specified or defaulted:
 - if a //DFSPARM DD data set is available at runtime , DFSORT will use it

- if a //DFSPARM DD data set is not available at runtime , DFSORT will use a //\$ORTPARM DD data set if available.

Thus with PARMDDN=DFSPARM, you can choose to specify either a //DFSPARM DD data set or a //\$ORTPARM DD data set for a particular DFSORT application.

2. When PARMDDN=ddname is specified:

- if a //ddname DD data set is available at runtime , DFSORT will use it
- if a //ddname DD data set is not available at runtime , DFSORT will use a //DFSPARM DD data set if available.

Thus with PARMDDN=ddname, you can choose to specify either a //ddname DD data set or a //DFSPARM DD data set for a particular DFSORT application.

RESALL

➡ RESALL= — *n* →

indicates the number of bytes to be reserved in a region, while DFSORT is running, when SIZE/MAINSIZE=MAX is in effect. Normally, only 4 KB (the default) of main storage has to be available in a region for system use. However, in a few cases this is not enough. For example, your site might not have the BSAM/QSAM data management modules resident, or you might have user exit routines that open data sets.

RESALL applies only to the amount of main storage below 16 MB virtual. ARESALL applies to the amount of main storage above 16 MB virtual.

For more information, see [“Tailoring main storage” on page 14.](#)

n

Specifies the number of bytes as a decimal value within the range of 4096 (4 KB) to 16515072 (16128 KB). If the value specified is not a multiple of 8, it is rounded up to the next multiple of 8.

Default: 4096 (4 KB)

Note: This specified amount of main storage is not used by DFSORT but can be logged as used by the system.

RESET

➡ RESET= — YES —
 NO —

Specifies whether DFSORT should process a VSAM output data set defined with REUSE as a NEW or MOD data set.

YES

Specifies that DFSORT processes a VSAM output data set defined with REUSE as a NEW data set. The high-used RBA is reset to zero and the output data set is effectively treated as an initially empty cluster.

NO

Specifies that DFSORT processes a VSAM output data set defined with REUSE as a MOD data set. The high-used RBA is not reset and the output data set is effectively treated as an initially non-empty cluster.

Default: YES

Note: A VSAM output data set defined without REUSE is processed as a MOD data set.

RESINV

➡ RESINV= — *n* ➡

indicates the number of bytes to be reserved for the invoking program, while DFSORT is running, when DFSORT is program invoked and SIZE/MAINSIZE=MAX is in effect. RESINV applies only to the amount of main storage below 16 MB virtual. ARESINV applies to the amount of main storage above 16 MB virtual. For more information, see [“Tailoring main storage” on page 14](#).

This extra space is usually required for data handling by the invoking program or user exits while DFSORT is running (as is the case with some PL/I- and all COBOL-invoked sort applications).

For example, when DFSORT is invoked from COBOL and the RESINV parameter is used, *n* specifies the number of bytes to be reserved for use with COBOL in its default or user-written input or output procedure. If your invoking program and its associated user exits do not process data sets, you do not need to specify this parameter.

The number and size of buffers you need depend upon the routines you have, how the data is stored, and which access methods you use. Normally, RESINV=16384 (16 KB) is sufficient. The reserved space is not meant to be used for the invoking program's executable code.

n

Specifies the number of bytes as a decimal value within the range of 0 to 16515072 (16128 KB). If the value specified is not a multiple of 8, it is rounded up to the next multiple of 8.

Default: 0 bytes

Note:

1. RESINV is always used if the program is to be called by a COBOL or PL/I program. It can also be needed if the program is to be called by programs written in other languages (refer to your specific language manual).
2. RESINV is ignored when it is specified with the JCL or TSO parameter.

SDB

➡ SDB= — LARGE — ➡

YES

INPUT

NO

Specifies whether DFSORT should use the system-determined optimum block size for output data sets when the block size is specified as zero or defaulted to zero. System-determined block size applies to both SMS-managed and non-SMS-managed data sets and results in the most efficient use of space for the device on which the output data set resides.

DFSORT can select system-determined optimum block sizes greater than 32760 bytes for tape output data sets.

If you want DFSORT to use system-determined block sizes for disk and tape output data sets, specify one of the following values:

- SDB=LARGE if you want to allow DFSORT to select tape output block sizes greater than 32760 bytes.
- SDB=YES (or its alias SDB=SMALL) if you want DFSORT to select tape output block sizes less than or equal to 32760 bytes.
- SDB=INPUT if you want to allow DFSORT to select tape output block sizes greater than 32760 bytes only when tape input data sets with block sizes greater than 32760 bytes are used.

DFSORT will not select a tape output block size greater than the BLKSZLIM in effect. In particular, if a default BLKSZLIM of 32760 is in effect, DFSORT will not select a tape output block size greater than 32760 bytes. Therefore, in order to allow DFSORT to select tape output block sizes greater

than 32760 bytes for particular jobs, you may need to ensure that your JCL or data class supplies appropriately large BLKSZLIM values (for example, 1GB) for those jobs.

If you don't want DFSORT to use system-determined block sizes, specify SDB=NO (not recommended as an installation option).

LARGE

Specifies that DFSORT is to use the system-determined optimum block size for an output data set when its block size is zero. SDB=LARGE allows DFSORT to select a block size greater than 32760 bytes for a tape output data set, when appropriate. A larger tape block size can improve elapsed time and tape utilization, but you must ensure that applications which subsequently use the resulting tape data set can handle larger block sizes.

DFSORT selects the system-determined optimum block size as follows:

- For a disk output data set, the optimum block size for the device used is selected based on the obtained or derived RECFM and LRECL for the output data set. The maximum block size for disk output data sets is 32760 bytes.
- For a tape output data set, the optimum block size is selected based on the obtained or derived RECFM and LRECL for the output data set, as shown in the figure that follows.

Table 8. SDB=LARGE for tape output data sets

RECFM	BLKSIZE is set to
F or FS	LRECL
FB or FBS	Highest possible multiple of LRECL that is less than or equal to the optimum block size for the device, subject to the BLKSZLIM in effect.
V, D, VS, or DS	LRECL + 4
VB, DB, VBS, or DBS	Optimum block size for the device, subject to the BLKSZLIM in effect.

DFSORT uses the system-determined optimum block size for the output data set in most cases when the block size is zero. However, the following conditions prevent DFSORT from using the system-determined block size:

- Output data set block size is available (that is, nonzero) in the JFCB (disk or tape) or format 1 DSCB (disk) or tape label (only for DISP=MOD with AL, SL, or NSL label, when appropriate)
- Output is a spool, dummy, VSAM, or unmovable data set, or a z/OS UNIX file
- The output data set is on tape with a label type of AL
- DFSORT's Blockset technique is not selected.

In these cases, DFSORT uses the specified block size, or determines an appropriate (though not necessarily optimum) block size for the output data set. The selected block size is limited to 32760 bytes.

YES

Specifies that DFSORT is to use the system-determined optimum block size for an output data set when its block size is zero, but is to limit the selected block size to a maximum of 32760 bytes. See the discussion of SDB=LARGE for more information; the only difference between SDB=LARGE and SDB=YES is that SDB=LARGE allows block sizes greater than 32760 bytes for tape output data sets, whereas SDB=YES does not.

INPUT

Specifies that DFSORT is to use the system-determined optimum block size for an output data set when its block size is zero, but is to limit the selected block size to a maximum of 32760 bytes if the input block size is less than or equal to 32760 bytes. Thus, SDB=INPUT works like SDB=LARGE if the input block size is greater than 32760 bytes (only possible for tape input data

sets) and works like SDB=YES if the input block size is less than or equal to 32760 bytes. See the discussions of SDB=LARGE and SDB=YES for more information.

NO

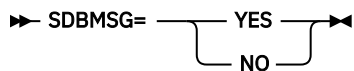
Specifies that DFSORT is not to use the system-determined optimum block size. When the output data set block size is zero, DFSORT selects an appropriate (though not necessarily optimum) block size for the output data set based on the obtained or derived output or input attributes. SDB=NO limits the selected block sizes to a maximum of 32760 bytes (only possible for tape input data sets). See the discussion of SDB=YES for more information.

Default: INPUT

Note:

1. SDB=LARGE and SDB=INPUT work like SDB=YES.
2. SDB=NO does not prevent the use of system-determined block size for the output data set at allocation or in other cases where the output data set block size is set before DFSORT gets control.
3. When DFSORT uses system-determined block size, the selected output data set block size may be different from the block size selected previously. SDB=NO can be selected or defaulted to eliminate the use of system-determined block size for all DFSORT applications, but this does not guarantee that a particular block size will be selected for any application. Applications that require a specific output data set block size should be explicitly changed to specify that block size.
4. SDB=SMALL can be used instead of SDB=YES.

SDBMSG



Specifies whether the system-determined optimum block size should be used for DFSORT message data sets and ICETOOL message and list data sets when the block size is specified as zero or defaulted to zero. System-determined block size applies to both SMS-managed and non-SMS-managed data sets and results in the most efficient use of space for the device on which the message or list data set resides.

SDBMSG applies to the following message and list data sets:

- DFSORT's SYSOUT message data set. This is also used for ICETOOL'S DFSMSG message data set and ICEGENER's SYSPRINT message data set
- ICETOOL's TOOLMSG message data set
- ICETOOL's list data sets. These are the data sets specified in the LIST(listdd) operand available with several of the ICETOOL operators.

YES

Specifies that DFSORT and ICETOOL are to use the system-determined optimum block size for message and list data sets when the block size is zero.

- For a disk message or list data set, the optimum block size for the device used is selected based on the RECFM and LRECL attributes for the data set.
- For a tape message or list data set, the optimum block size is selected based on the highest possible multiple of LRECL that is less than or equal to 32760.

The following conditions prevent DFSORT and ICETOOL from using the system-determined optimum block size for message and list data sets:

- Message or list data set block size is available (that is, nonzero)
- Not MVS/DFP 3.1.0 or later
- Message or list data set is a spool or dummy data set.

NO

Specifies that DFSORT and ICETOOL are NOT to use the system-determined optimum block size for message and list data sets. When the block size is zero for a message or list data set, DFSORT and ICETOOL are to select a block size equal to the LRECL.

SDBMSG=NO does not prevent the use of system-determined block size for message or list data sets at allocation or in other cases where the block size is set before DFSORT or ICETOOL gets control.

Default: NO

Note:

1. ICETOOL's TOOLMSG and list data sets will be affected by an SDBMSG value specified with the INV or TSOINV parameter, but will not be affected by an SDBMSG value specified with the JCL, TSO, TD1, TD2, TD3, or TD4 parameter.
2. When DFSORT or ICETOOL uses system-determined block size, the selected message or list data set block size may be different from the block size selected previously. SDBMSG=NO can be selected or defaulted to eliminate the use of system-determined block size for message and list data sets for all DFSORT and ICETOOL applications, but that will not guarantee that a particular block size will be selected for any application. Applications which require a specific message or list data set block size should be changed to specify that block size **explicitly**.

SIZE



Specifies the amount of main storage DFSORT attempts to use. If the amount you specify for SIZE exceeds the amount of real storage generally available for one initiator, excessive paging can occur. For more information, see [“Tailoring main storage” on page 14](#).

MAX

Specifies that DFSORT calculates the maximum amount of available storage as limited by the TMAXLIM or DSA value when Blockset is selected, or by the MAXLIM value when Blockset is not selected.

n

Specifies the number of bytes as a decimal value within the range of 90112 (88 KB) to 2097152000 (2000 MB). Lowering SIZE below 4 MB can degrade performance for some Blockset sort applications. This performance degradation can become more pronounced and affect more applications the more SIZE is lowered below 4 MB. If the value specified is not a multiple of 8, it is rounded down to the next multiple of 8.

If the SIZE value specified in ICEMAC is less than the MINLIM value specified in ICEMAC, the SIZE value is set to the MINLIM value in the ICEAMx or ICETDx module. The SIZE value specified in an ICEPRMxx member is not adjusted based on the MINLIM value.

If the SIZE value in effect at run time is less than the MINLIM value in effect at run time, the SIZE value is set to the MINLIM value at run time .

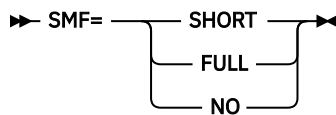
Default: MAX

Note:

1. SIZE is not observed for OUTFIL processing.
2. RESALL and RESINV are ignored unless SIZE=MAX is in effect.
3. Since SIZE=n does not distinguish between storage below 16 MB virtual and total storage, and SIZE=n does not permit use of RESALL and RESINV, it is strongly recommended that SIZE=MAX be specified, and MAXLIM, TMAXLIM, RESALL, and RESINV be used to customize the default storage for the type of application.

4. Setting SIZE to a very large value provides little or no performance benefit to DFSORT and can lead to DFSORT and system performance degradation due to excessive demands on processor storage.
5. Sufficient virtual storage must be available in the primary address space in order for DFSORT to use the amount of storage specified. Also, DFSORT and system performance can degrade if insufficient processor storage is available to back the specified amount of main storage.

SMF



Specifies whether DFSORT SMF records are to be produced as described in [Chapter 6, “Collecting statistical data,”](#) on page 131.

SHORT

DFSORT produces a short SMF type-16 record for each successful run. The short SMF record does not contain record-length distribution or data set sections.

FULL

DFSORT produces a full SMF type-16 record for each successful run. The full SMF record contains the same information as the short record, as well as record-length distribution and data set sections, as appropriate.

Note: SMF=FULL can degrade performance for variable-length record applications.

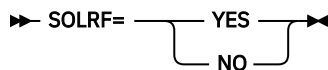
NO

DFSORT does not produce any SMF type-16 records.

Default: NO

Note: The DFSORT SVC is called to write SMF type-16 records. If SMF=SHORT or SMF=FULL is in effect, the correct DFSORT SVC for this release must be loaded in LPA or MLPA. The DFSORT SVC is not required if SMF=NO is in effect.

SOLRF



Specifies whether DFSORT should set the SORTOUT LRECL to the reformatted record length when the SORTOUT LRECL is unknown.

YES

Specifies that DFSORT should use the reformatted record length for the SORTOUT LRECL when the SORTOUT LRECL is not specified or available. DFSORT will use one of the following for the SORTOUT LRECL, in the order listed:

1. The SORTOUT LRECL if available from the JFCB, format 1 DSCB, DFSMSrmm, or tape label
2. The L3 length if specified in the RECORD statement
3. The OUTREC length if the OUTREC statement is specified
4. The INREC length if the INREC statement is specified
5. The L2 length if specified in the RECORD statement provided an E15 user exit is present
6. The SORTIN or SORTINnn LRECL if available from the JFCB, format 1 DSCB, DFSMSrmm, or tape label
7. The L1 length if specified in the RECORD statement

NO

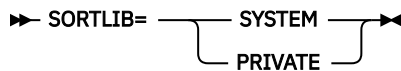
Specifies that DFSORT should not use the reformatted record length for the SORTOUT LRECL. DFSORT will use one of the following for the SORTOUT LRECL, in the order listed:

1. The SORTOUT LRECL if available from the JFCB, format 1 DSCB, DFSMSrmm, or tape label
2. The L3 length if specified in the RECORD statement provided an E35 exit, OUTREC statement or INREC statement is present
3. The L2 length if specified in the RECORD statement provided an E15 user exit is present
4. The SORTIN or SORTINnn LRECL if available from the JFCB, format 1 DSCB, DFSMSrmm, or tape label
5. The L1 length in the RECORD statement

Default: YES

Note: With SOLRF=YES (the default), DFSORT sets the SORTOUT LRECL to the INREC or OUTREC record length when appropriate, which is usually what you want when you use INREC or OUTREC. If you want DFSORT to use the input length for the SORTOUT LRECL even when INREC or OUTREC is present, you can use SOLRF=NO, but be aware that this can cause padding or truncation of the reformatted records, or termination.

SORTLIB



determines whether DFSORT searches LINKLIB or a private library separate from LINKLIB for the Conventional technique modules used for a tape work data set sort or Conventional merge application. Choose the value that reflects where the modules will reside.

SYSTEM

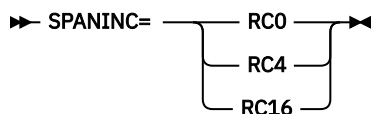
Specifies that the Conventional technique modules reside in LINKLIB. A SORTLIB DD statement is only required at run time for a tape work data set sort or Conventional merge application that uses dynamic linkedit of user exits.

PRIVATE

Specifies that the Conventional technique modules reside in a private library separate from LINKLIB. A SORTLIB DD statement is required at run time for a tape work data set sort or Conventional merge application.

Default: PRIVATE

SPANINC



Specifies the action to be taken by DFSORT when one or more incomplete spanned records are detected in a variable spanned input data set.

RC0

Specifies that DFSORT should issue message ICE197I (once), set a return code of 0, and eliminate all incomplete spanned records it detects. Valid records will be recovered.

RC4

Specifies that DFSORT should issue message ICE197I (once), set a return code of 4, and eliminate all incomplete spanned records that it detects. Valid records will be recovered.

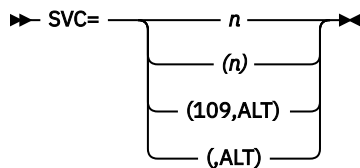
RC16

Specifies that DFSORT should issue message ICE204A, terminate, and give a return code of 16 when an incomplete spanned record is detected.

Default: RC16

Note:

1. The return code of 0 or 4 set for incomplete spanned records can be overridden by a higher return code set for some other reason.
2. In cases where a spanned record cannot be properly assembled (for example, it has a segment length less than 4 bytes), DFSORT issues message ICE141A and terminates with a return code of 16. The SPANINC value has no effect in these cases.

SVC

Specifies an SVC number that you choose for the DFSORT SVC routine.

n

Specifies your SVC number. n must be the decimal number 109 or a decimal number between 200 and 255. The SVC number you choose must be valid at your site. The form (n) is the same as n.

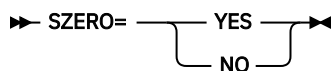
ALT

Specifies that alternate DFSORT module IGX00038 is used in conjunction with SVC 109 rather than default DFSORT module IGX00017. ALT applies only when you choose SVC number 109. The form (109,ALT) is the same as (,ALT).

You can also find more information on installing SVCs in [z/OS MVS Initialization and Tuning Guide](#).

Default: 109

Note: The DFSORT SVC is called to write SMF type-16 records and to process data sets on cached disk devices. If SMF=SHORT (write short SMF type-16 records) or SMF=FULL (write full SMF type-16 records) is in effect, the correct DFSORT SVC for this release must be loaded in LPA or MLPA.

SZERO

Specifies whether DFSORT should treat numeric -0 and +0 values as signed (that is, different) or unsigned (that is, the same) for collation, comparisons, editing, conversions, minimums, and maximums. The following DFSORT control statements are affected by this option: INCLUDE, INREC, MERGE, OMIT, OUTFIL, OUTREC, and SORT.

YES

Specifies that DFSORT should treat numeric zero values as signed. -0 and +0 are treated as different values, that is, -0 is treated as a negative value and +0 is treated as a positive value. SZERO=YES affects DFSORT processing of numeric values as follows:

- For collation of SORT and MERGE fields, -0 collates before +0 in ascending order and after +0 in descending order.
- For comparisons of INCLUDE, OMIT and OUTFIL compare fields and constants, -0 compares as less than +0.

- For editing and conversions of INREC, OUTREC and OUTFIL reformatting fields, decimal constants, and the results of arithmetic expressions, -0 is treated as negative and +0 is treated as positive.
- For minimums and maximums of OUTFIL TRAILERx fields, -0 is treated as negative and +0 is treated as positive.

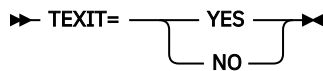
NO

Specifies that DFSORT should treat numeric zero values as unsigned. -0 and +0 are treated as the same value, that is, -0 and +0 are both treated as positive values. SZERO=NO affects DFSORT processing of numeric values as follows:

- For collation of SORT and MERGE fields, -0 collates equally with +0.
- For comparisons of INCLUDE, OMIT and OUTFIL compare fields and constants, -0 compares as equal to +0.
- For editing and conversions of INREC, OUTREC and OUTFIL reformatting fields, decimal constants, and the results of arithmetic expressions, -0 and +0 are treated as positive.
- For minimums and maximums of OUTFIL TRAILERx fields, -0 and +0 are treated as positive.

Default: YES

TEXTIT



Specifies whether DFSORT is to pass control to a termination exit (ICETEXIT) as described in [Chapter 6, "Collecting statistical data,"](#) on page 131.

YES

Specifies that DFSORT is to pass control to the site's ICETEXIT routine.

NO

Specifies that DFSORT does not pass control to an ICETEXIT routine.

Default: NO

TMAXLIM



Specifies an upper limit to the total amount of main storage, both above and below 16 MB virtual, available to DFSORT when SIZE/MAINSIZE=MAX is in effect and Blockset is selected. For more information, see ["Tailoring main storage" on page 14.](#)

n

Specifies the number of bytes as a decimal value. The maximum value is 2097152000 (2000 MB). Lowering TMAXLIM below 6 MB can degrade performance for some applications. This performance degradation can become more pronounced and affect more applications the more TMAXLIM is lowered below 6 MB. If the value specified is not a multiple of 8, it is rounded down to the next multiple of 8.

If the MAXLIM value specified in ICEMAC is greater than the TMAXLIM value specified in ICEMAC, the TMAXLIM value is set to the MAXLIM value in the ICEAMx or ICETDx module. The TMAXLIM value specified in an ICEPRMxx member is not adjusted based on the MAXLIM value.

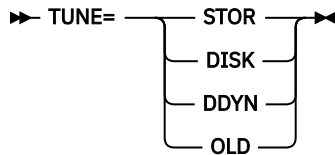
If the MAXLIM value in effect at run time is greater than the TMAXLIM value in effect at run time, the TMAXLIM value is set to the MAXLIM value at run time .

Default : 6291456 (6 MB)

Note:

1. TMAXLIM is not observed for OUTFIL processing.
2. Setting TMAXLIM to a very large value provides little or no performance benefit to DFSORT and can lead to DFSORT and system performance degradation due to excessive demands on processor storage.
3. Sufficient virtual storage must be available in the primary address space in order for DFSORT to use the amount of main storage specified. Also, DFSORT and system performance can degrade if insufficient processor storage is available to back the specified amount of main storage.

TUNE



Specifies whether DFSORT should favor optimization of central storage or disk work space for sort applications. Since these two resources are both critical to sort processing, DFSORT must synchronize their usage. When more central storage can be used, it may be possible to reduce the disk work space requirement. When central storage resources are constrained, additional disk work space may be required.

STOR

Specifies sort applications will allocate available central storage as needed in increments sized to balance usage when multiple sorts are executing on the same system. As a result, DFSORT's dynamic work data set allocations will be increased to account for increased disk space requirements in case central storage resources become constrained during execution of the sort.

DISK

Specifies sort applications will allocate all available central storage required at initialization. DFSORT's dynamic work data set allocations will be decreased based on the expected central storage usage to minimize the disk space requirements.

DDYN

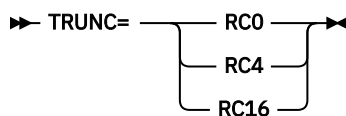
Specifies sort applications will allocate all of the available central storage required at initialization only when dynamic allocation is being used. DFSORT's dynamic work data set allocations will be decreased based on the expected central storage usage to minimize the disk space requirements. When work data sets have been preallocated in JCL or by an invoking program, sorts will allocate available central storage as needed in increments sized to balance resource usage when multiple sorts are executing on the same system.

OLD

Specifies sort applications will allocate central storage as needed using increments of a fixed size. DFSORT's dynamic work data set allocations will be decreased based on the expected central storage usage to minimize disk space requirements.

Default: STOR

TRUNC



Specifies the action to be taken by DFSORT when the SORTOUT LRECL is smaller than the SORTIN/ SORTINnn LRECL, for the cases where DFSORT allows LRECL truncation.

RC0

Specifies that DFSORT should issue message ICE171I, set a return code of 0, and continue processing when the SORTOUT LRECL is smaller than the SORTIN/SORTINnn LRECL.

RC4

Specifies that DFSORT should issue message ICE171I, set a return code of 4, and continue processing when the SORTOUT LRECL is smaller than the SORTIN/SORTINnn LRECL.

RC16

Specifies that DFSORT should issue message ICE196A, terminate, and give a return code of 16 when the SORTOUT LRECL is smaller than the SORTIN/SORTINnn LRECL.

Default: RC0

Note:

1. The return code of 0 or 4 set for LRECL truncation can be overridden by a higher return code set for some other reason.
2. For an ICEGENER application, the GNTRUNC value is used, and the TRUNC value is ignored.
3. For some LRECL truncation situations (for example, a tape work data set sort), DFSORT issues message ICE043A and terminates with a return code of 16. The TRUNC value has no effect in these cases.
4. If SORTIN/SORTINnn or SORTOUT is not present, or is a VSAM data set, DFSORT does not check for LRECL truncation.
5. DFSORT does not check OUTFIL data sets for LRECL truncation.

VERIFY



Specifies whether the sequence of output records is to be verified.

YES

Specifies that the sequence is to be verified.

NO

Specifies that the sequence is not to be verified.

Default: NO



Attention: VERIFY=YES can degrade performance, especially when sorting short records.

VIO



Specifies whether virtual allocations for sort work data sets are dynamically reallocated to real disk locations.

YES

If virtual allocation of sort work data sets takes place, DFSORT uses them without reallocating them to real disk locations.

NO

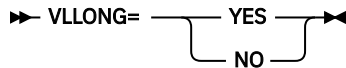
DFSORT dynamically reallocates any virtual allocation of sort work data sets to get real disk locations.

Default: NO

Note:

1. If virtual devices are used for sort work data sets, performance might be degraded.
2. With VIO=NO, if a SORTWKdd data set is specified on a virtual device, DFSORT carries out dynamic reallocation using the ddname SORTDKdd on a real device with the same device type as the virtual device. If a real device corresponding to the virtual device is not available in the system, DFSORT terminates.
3. In certain cases, DFSORT may have to ignore VIO=NO and use VIO work data sets to successfully process a sort application.
4. With VIO=YES, DFSORT will not dynamically allocate large physical sequential data sets as sort work data sets.

VLLONG



Specifies whether DFSORT is to truncate "long" variable-length output records. A long output record is one whose length is greater than the LRECL of the SORTOUT or OUTFIL data set it is to be written to.

VLLONG is not meaningful for fixed-length output record processing.

YES

Specifies that DFSORT truncates long variable-length output records to the LRECL of the SORTOUT or OUTFIL data set.

NO

Specifies that DFSORT terminates if a long variable-length output record is found.

Default: NO

Note:

1. VLLONG=YES should not be used unless you want the data at the end of long variable-length output records to be truncated for all of your DFSORT applications; inappropriate use of VLLONG=YES can result in unwanted loss of data. To truncate long records for specific DFSORT applications, specify the VLLONG runtime option for those applications and use VLLONG=NO (the IBM-supplied default) as your installation default.
2. VLLONG=YES can be used to truncate long OUTFIL data records, but has no effect on long OUTFIL header or trailer records.

VLSCMP



Specifies whether DFSORT is to pad "short" variable-length INCLUDE/OMIT compare fields with binary zeros. A short field is one where the variable-length record is too short to contain the entire field, that is, the field extends beyond the record. VLSCMP applies to the INCLUDE and OMIT statements and to the INCLUDE and OMIT parameters of the OUTFIL statement.

The compare fields are only padded temporarily for testing; they are not actually changed for output.

VLSCMP is not meaningful for fixed-length record processing.

The settings for VLSCMP and VLSHRT provide three levels of processing for short INCLUDE/OMIT fields in the following hierarchy:

1. VLSCMP=YES allows all of the INCLUDE/OMIT comparisons to be performed even if some fields are short. Since short fields are padded with binary zeros, comparisons involving short fields are false (unless a test against binary zero is relevant, as discussed in this section). Comparisons involving non-short fields can be true or false.

2. VLSCMP=NO and VLSHRT=YES treat the entire INCLUDE/OMIT logical expression as false if any field is short. Thus comparisons involving non-short fields are ignored if any comparison involves a short field.
3. VLSCMP=NO and VLSHRT=NO result in termination if any field is short.

To illustrate how this works, suppose the following INCLUDE statement is used:

```
INCLUDE COND=(6,1,CH,EQ,C'1',OR,70,2,CH,EQ,C'T1')
```

If a variable-length input record has a length less than 71 bytes, the field at bytes 70-71 is short and the following occurs:

- With VLSCMP=YES, the record is included if byte 6 of the input record is C'1' or omitted if byte 6 is not C'1'. The comparison of bytes 70-71 equal to C'T1' is false because bytes 70-71 contain either X'h00' (for a record length of 70 bytes) or X'0000' (for a record length of less than 70 bytes). The comparison involving the non-short field is performed even though a short field is present.
- With VLSCMP=NO and VLSHRT=YES, the record is omitted because any short field makes the entire logical expression false. The comparison involving the non-short field is not performed because a short field is present.
- With VLSCMP=NO and VLSHRT=NO, DFSORT terminates because any short field results in termination.

In general, comparisons involving short fields are false with VLSCMP=YES. However, if a binary zero value is relevant to the comparison, the use of binary zeros for padding might make the comparison true. For example, suppose the following INCLUDE statement is used:

```
INCLUDE COND=(21,2,CH,EQ,C'JX',OR,
              (55,2,CH,EQ,58,2,CH,AND,
               70,1,BI,LT,X'08'))
```

If a variable-length input record has a length less than 70 bytes, the field at byte 70 is short and is padded to X'00'. This makes the comparison of byte 70 less than X'08' true even though byte 70 is a short field and so probably irrelevant.

Likewise, if a variable-length record has a length less than 55 bytes, the fields at bytes 55-56 and 58-59 are short and are padded to X'0000' and the field at byte 70 is short and is padded to X'00'. This makes the comparison of bytes 55-56 equal to 58-59 true and the comparison of byte 70 less than X'08' true even though all three fields are short and probably irrelevant.

In such cases where padding of short fields with binary zeros may result in unwanted true comparisons, you can get the result you want by adding an appropriate check of the record length to the INCLUDE/OMIT logical expression, such as:

```
INCLUDE COND=(21,2,CH,EQ,C'JX',OR,
              (1,2,BI,GE,X'0046',AND,
               55,2,CH,EQ,58,2,CH,AND,
               70,1,BI,LT,X'08'))
```

Now the comparisons involving bytes 55-56, 58-59 and 70 can only be true for records that are 70 bytes (X'0046') or longer. Thus, the irrelevant comparisons involving short fields are eliminated.

Keep in mind that short compare fields are padded with zeros when VLSCMP is in effect and code your INCLUDE/OMIT logical expressions to allow for that or even take advantage of it.

YES

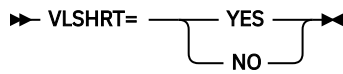
Specifies that short variable-length compare fields are padded with binary zeros.

NO

Specifies that short variable-length compare fields are not padded.

Default: NO

VLSHRT



Specifies whether DFSORT is to continue processing if a "short" variable-length SORT/MERGE control field, INCLUDE/OMIT compare field, or a SUM summary field is found. A short field is one where the variable-length record is too short to contain the entire field, that is, the field extends beyond the record. VLSHRT applies to the SORT, MERGE, INCLUDE, OMIT and SUM statements, and to the INCLUDE and OMIT parameters of the OUTFIL statement.

VLSHRT is not meaningful for fixed-length record processing.

VLSCMP and VLSHRT provide three levels of processing for short INCLUDE/OMIT fields. For details, see the discussion of the VLSCMP=YES/NO option.

For full details on VLSHRT, see the discussion of the VLSHRT and NOVLSHRT parameters of the OPTION statement in [z/OS DFSORT Application Programming Guide](#)

YES

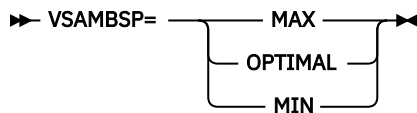
Specifies that DFSORT continues processing if a short control field, compare field, or summary field is found.

NO

Specifies that DFSORT terminates if a short control field, compare field, or summary field is found.

Default: NO

VSAMBSP



Specifies the number of VSAM buffers DFSORT can use when sorting using the Blockset technique. Additional buffers can reduce the elapsed time, CPU time, and EXCP counts needed for a sort application. The VSAMBSP option selected applies to all supported VSAM data set types: KSDS, ESDS, RRDS, and VRRDS.

MAX

Specifies that DFSORT can use the maximum number of VSAM buffers when sorting. MAX can provide the most significant performance benefits of the three options, but can result in excessive paging for storage-constrained systems.

OPTIMAL

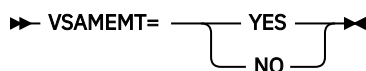
Specifies that DFSORT can use the optimal number of VSAM buffers when sorting. OPTIMAL can provide substantial performance benefits with minimal impact on system paging activity.

MIN

Specifies that DFSORT is to use the minimum number of VSAM buffers when sorting.

Default: OPTIMAL

VSAMEMT



Specifies whether DFSORT should accept an empty VSAM input data set.

YES

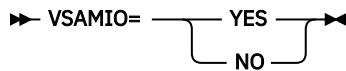
Specifies that DFSORT accepts an empty VSAM input data set and processes it as having zero records.

NO

Specifies that DFSORT terminates if an empty VSAM input data set is found.

Default: YES

VSAMIO



Specifies whether DFSORT should allow a VSAM data set defined with REUSE to be sorted in-place.

YES

Specifies that DFSORT can use the same VSAM data set for input and output when all of the following conditions are met:

- The application is a sort.
- RESET is in effect at run time.
- The VSAM data set was defined with REUSE.

These conditions ensure that the VSAM data set is processed as NEW for output and will contain the sorted input records, that is, it will be sorted in-place.

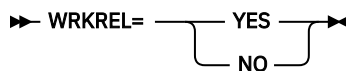
DFSORT terminates if the same VSAM data set is specified for input and output, and any of the previous conditions are not met.

NO

Specifies that DFSORT terminates if the same VSAM data set is specified for input and output

Default: NO

WRKREL



Specifies whether unused temporary SORTWKdd data set space is released.

YES

Specifies that unused space is released.

NO

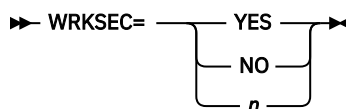
Specifies that unused space is not released.

Default: YES

Note:

1. If you have dedicated certain volumes for SORTWKdd data sets, and you do not want unused temporary space to be released, you should specify WRKREL=NO.
2. If WRKREL is in effect, DFSORT releases space for the SORTWKdd data sets just prior to termination. Space is released only for those SORTWKdd data sets that were used for the sort application.

WRKSEC



Specifies whether DFSORT uses automatic secondary allocation for temporary JCL SORTWKdd data sets for which a secondary allocation amount is not specified.

YES

Specifies that automatic secondary allocation for temporary JCL SORTWKdd data sets be used, and that 25 percent of the primary allocation be used as the secondary allocation.

NO

Specifies that automatic secondary allocation for temporary JCL SORTWKdd data sets not be used.

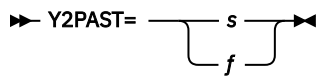
n

Specifies that automatic secondary allocation for temporary JCL SORTWKdd data sets be used, and that *n* percent of the primary allocation be used as the secondary allocation. The value *n* must be between 1 and 1000.

Default: YES

Note:

1. WRKSEC has no effect on dynamically allocated work data sets.
2. WRKSEC=0 is equivalent to WRKSEC=NO.

Y2PAST

specifies the sliding (s) or fixed (f) century window. The century window is used with DFSORT's Y2x formats to correctly interpret two-digit year data values as four-digit year data values.

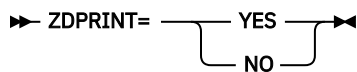
s

Specifies the number of years DFSORT is to subtract from the current year to set the beginning of the sliding century window. Since the Y2PAST value is subtracted from the current year, the century window slides as the current year changes. For example, Y2PAST=81 would set a century window of 1925-2024 in 2006 and 1926-2025 in 2007. *s* must be a value between 0 and 100.

f

Specifies the beginning of the fixed century window. For example, Y2PAST=1962 would set a century window of 1962-2061. *f* must be a value between 1000 and 3000.

Default: 80

ZDPRINT

Specifies whether positive zoned-decimal (ZD) fields resulting from summation must be converted to printable numbers (that is, whether the zone of the last digit should be changed from a hexadecimal C to a hexadecimal F).

YES

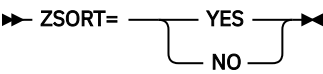
means convert positive ZD summation results to printable numbers. For example: change hexadecimal F3F2C5 (prints as 32E) to F3F2F5 (prints as 325).

NO

means do not convert positive ZD summation results to printable numbers.

Default: YES

ZSORT



Specifies IBM Integrated Accelerator for Z Sort can be used for sorting.

YES

IBM Integrated Accelerator for Z Sort is requested and eligible sort workloads will take advantage of sort accelerator.

NO

IBM Integrated Accelerator for Z Sort is disabled and the sorting is done using the traditional techniques.

Default: NO

Examples of ICEMAC macros

This section shows you examples of how the ICEMAC macro can be used to change the default values for several of the eight installation environments. See “Example of an ICEPRMxx member” on page 112 for an example of the recommended way to change the default values.

ICEMAC JCL - batch direct invocation defaults

Figure 13 on page 108 is a sample of the ICEMAC macro update statements to produce the ICEAM1/JCL installation defaults.

column	16	72 73	80
1			
ICEMAC JCL,		* 00550000	01
TMAXLIM=5242880,		* 00550100	02
MAXLIM=2097152,		* 00550200	03
MINLIM=524288,		* 00550300	04
LIST=NO,		* 00550400	05
EQUALS=YES,		* 00550500	06
SMF=FULL,		* 00550600	07
CHECK=NO,		* 00550700	08
IEXIT=YES,		* 00550800	09
TEXTIT=YES		00550900	10

Figure 13. Coding the ICEMAC Macro for ICEAM1

Line	Explanation
01	This ICEMAC call sets up your installation defaults for batch JCL directly invoked applications.
02	When Blockset is selected, DFSORT makes use of all available space (the default is SIZE=MAX) up to 5242880 bytes (5 MB) both above and below 16 MB virtual.
03	When Blockset is selected, DFSORT can use up to 2097152 bytes (2 MB) below 16 MB virtual. When Blockset is not selected, DFSORT makes use of all available space (the default is SIZE=MAX) up to 2097152 bytes (2 MB) below 16 MB virtual.
04	DFSORT always tries to use at least 524288 bytes (512 KB).
05	Control statements are not printed.

- 06**
The order of records that collate identically is preserved from input to output.
 - 07**
At the end of each application, a full SMF type-16 record is written to the SMF data set.
 - 08**
Record count is not checked for applications that use the E35 user exit routine.
 - 09**
DFSORT passes control to your initialization exit routine (ICEIEXIT).
 - 10**
DFSORT passes control to your termination exit routine (ICETEXIT).
- No other standard defaults for batch JCL directly invoked DFSORT applications are overridden in this example. See [Figure 20 on page 114](#), [Figure 21 on page 115](#), [Figure 22 on page 115](#), [Figure 23 on page 116](#), [Figure 24 on page 117](#), and [Figure 25 on page 117](#) for the IBM-supplied defaults for all of the installation options.

ICEMAC INV - batch program invocation defaults

[Figure 14 on page 109](#) is a sample of the ICEMAC macro update statements to produce the ICEAM2/INV installation defaults.

column				
1	16	72	73	80
ICEMAC INV,				
	DYNALOC=3390,	*	00550000	01
	IGNCKPT=NO,	*	00550100	02
	MSGDDN=MSGOUT,	*	00550200	03
	MSPRT=CRITICAL,	*	00550300	04
	OUTREL=NO,	*	00550400	05
	WRKREL=NO,	*	00550500	06
	LIST=YES	*	00550600	07
			00550700	08

Figure 14. Coding the ICEMAC Macro for ICEAM2

Line

Explanation

- 01**
This ICEMAC call sets up your installation defaults for batch program invoked DFSORT applications.
- 02**
This sets 3390 as the default device for dynamically allocated work data sets whenever DYNALLOC is specified at run time without a corresponding device name (for example, DYNALLOC or DYNALLOC=(,2)).

Because no value for the default maximum number of work data sets is specified, the standard default of 4 for Blockset or 3 for Peerage/Vale is used if DYNALLOC is specified at run time without a corresponding number of work data sets (for example, DYNALLOC or DYNALLOC=SYSDA).
- 03**
The Blockset technique is not selected if CKPT or CHKPT is specified on the OPTION, SORT, or MERGE statement.
- 04**
MSGOUT is to be used rather than SYSOUT as the ddname of the output data set for program messages. Note that if a MSGOUT DD statement is not found at run time, a SYSOUT DD statement is used, if found.
- 05**
Only error messages are printed.

- 06
Unused temporary SORTOUT space is not released.
- 07
Unused temporary SORTWKdd space is not released.
- 08
Control statements are printed. Note that LIST=YES is the standard default and need not be specified. It is shown here for purposes of illustration only.

No other standard defaults for batch program invoked applications are overridden in this example. See [Figure 20 on page 114](#), [Figure 21 on page 115](#), [Figure 22 on page 115](#), [Figure 23 on page 116](#), [Figure 24 on page 117](#), and [Figure 25 on page 117](#) for the IBM-supplied defaults for all of the installation options.

ICEMAC TSO - TSO direct invocation defaults

[Figure 15 on page 110](#) is a sample of the ICEMAC macro update statements to produce the ICEAM3/TSO installation defaults.

column				
1	16	72	73	80
ICEMAC TSO,		*	00550000	01
ENABLE=(TD1),		*	00550100	02
NOMSGDD=CRITICAL,		*	00550200	03
TMAXLIM=3145728,		*	00550300	04
DYNALOC=(3390,3)			00550400	05

Figure 15. Coding the ICEMAC Macro for ICEAM3

Line	Explanation
01	This ICEMAC call sets up your installation defaults for TSO directly invoked DFSORT applications.
02	Enables the ICETD1 time-of-day environment for directly invoked DFSORT applications. If a TSO directly invoked DFSORT application is run during a time-of-day for which ICETD1 is activated, the ICETD1 defaults will be used instead of the ICEAM3 defaults.
03	If the message data set is required but not present, DFSORT continues processing and writes critical messages to the master console.
04	When Blockset is selected, DFSORT makes use of all available space (the default is SIZE=MAX) up to 3145728 bytes (3 MB) both above and below 16 MB virtual.
05	DFSORT sets three 3390 devices as the default for dynamically allocated work data sets whenever dynamic allocation is used at run time.

No other standard defaults for TSO directly invoked applications are overridden in this example. See [Figure 20 on page 114](#), [Figure 21 on page 115](#), [Figure 22 on page 115](#), [Figure 23 on page 116](#), [Figure 24 on page 117](#), and [Figure 25 on page 117](#) for the IBM-supplied defaults or all of the installation options.

ICEMAC TSOINV - TSO program invocation defaults

[Figure 16 on page 111](#) is a sample of the ICEMAC macro update statements to produce the ICEAM4/TSOINV installation defaults.

column				
1	16	72	73	80
ICEMAC TSOINV,		*	00550000	01
RESINV=20480			00550100	02

Figure 16. Coding the ICEMAC Macro for ICEAM4

Line	Explanation
-------------	--------------------

01 This ICEMAC call sets up your installation defaults for TSO program-invoked DFSORT applications.

02 DFSORT reserves 20480 bytes (20 KB) for use by the invoking program while DFSORT is running.

No other standard defaults for TSO program-invoked applications are overridden in this example. See [Figure 20 on page 114](#), [Figure 21 on page 115](#), [Figure 22 on page 115](#), [Figure 23 on page 116](#), [Figure 24 on page 117](#), and [Figure 25 on page 117](#) for the IBM-supplied defaults for all of the installation options.

ICEMAC TD1 - first time-of-day defaults

[Figure 17 on page 111](#) is a sample of the ICEMAC macro update statements to produce the ICETD1/TD1 installation defaults.

column				
1	16	72	73	80
ICEMAC TD1,		*	00550000	01
WKEND=(ALL),		*	00550100	02
NOMSGDD=CRITICAL,		*	00550200	03
TMAXLIM=6291456,		*	00550300	04
DYNALOC=(3390,3),		*	00550400	05
DSA=48			00550500	06

Figure 17. Coding the ICEMAC Macro for ICETD1

Line	Explanation
-------------	--------------------

01 This ICEMAC call sets up your installation parameter defaults for the first time-of-day environment.

02 DFSORT uses these ICETD1 defaults on Saturday or Sunday whenever the ICEAMx environment in effect for the run enables ICETD1. For example, "ICEMAC TSO - TSO Direct Invocation Defaults" specified ENABLE=(TD1) for ICEAM3, so TSO directly invoked applications would use the following defaults when run on the indicated days:

- Saturday and Sunday: ICETD1 defaults
- Monday through Friday: ICEAM3 defaults

03 If the message data set is required but not present, DFSORT continues processing and writes critical messages to the master console.

04 When Blockset is selected, DFSORT makes use of all available space (the default is SIZE=MAX) up to 6291456 bytes (6 MB) both above and below 16 MB virtual.

05

DFSORT sets three 3390 devices as the default for dynamically allocated work data sets whenever dynamic allocation is used at run time.

06

DFSORT can use up to 48 MB of space for Blockset sort applications when doing so should improve performance.

No other standard defaults for the first time-of-day environment are overridden in this example. See [Figure 20 on page 114](#), [Figure 21 on page 115](#), [Figure 22 on page 115](#), [Figure 23 on page 116](#), [Figure 24 on page 117](#), and [Figure 25 on page 117](#) for the IBM-supplied defaults for all of the installation options.

Example of an ICEPRMxx member

This section shows you an example of one ICEPRMxx member, ICEPRM01, you could use instead of the multiple ICEMAC calls shown in [“Examples of ICEMAC macros” on page 108](#) to make the same changes to the installation defaults. You would activate the ICEPRMxx changes using the following START ICEOPT command from the console or in a COMMNDx member in PARMLIB:

```
START ICEOPT,ICEPRM=01
```

For information on ICEPRMxx members of PARMLIB and how to code and activate them, see [“Using ICEPRMxx members in PARMLIB” on page 46](#).

ICEPRM01 installation default overrides

[Figure 18 on page 112](#) is an example of an ICEPRM01 member in PARMLIB. See [“Examples of ICEMAC macros” on page 108](#) for an explanation of the specified options.

```
JCL
  TMAXLIM=5242880
  MAXLIM=2097152
  MINLIM=524288
  LIST=NO
  EQUALS=YES
  SMF=FULL
  CHECK=NO
  IEXIT=YES
  TEXTIT=YES
INV
  DYNALOC=3390
  IGNCKPT=NO
  MSGDDN=MSGOUT
  MSGPRT=CRITICAL
  OUTREL=NO
  WRKREL=NO
  LIST=YES
TSO
  ENABLE=(TD1)
  NOMSGDD=CRITICAL
  TMAXLIM=3145728
  DYNALOC=(3390,3)
TSOINV
  RESINV=20480
TD1
  WKEND=(ALL)
  NOMSGDD=CRITICAL
  TMAXLIM=6291456
  DYNALOC=(3390,3)
  DSA=48
```

Figure 18. An ICEPRM01 Member in PARMLIB

The specified options override the corresponding ICEMAC defaults for the corresponding environments. For example, LIST=NO under JCL overrides the ICEMAC LIST default for the ICEAM1/JCL environment, and DSA=48 under TD1 overrides the ICEMAC DSA default for the ICETD1/TD1 environment. Only the

specified options are overridden; the ICEMAC defaults are used for any options that are not specified. See [Figure 20 on page 114](#), [Figure 21 on page 115](#), [Figure 22 on page 115](#), [Figure 23 on page 116](#), [Figure 24 on page 117](#), and [Figure 25 on page 117](#) for the IBM-supplied defaults for all of the installation options.

IBM-supplied installation defaults

[Figure 20 on page 114](#), [Figure 21 on page 115](#), [Figure 22 on page 115](#), [Figure 23 on page 116](#), [Figure 24 on page 117](#), and [Figure 25 on page 117](#), which were produced using the job shown in [Figure 19 on page 113](#), list the IBM-supplied defaults for each installation option for the eight installation environments. The IBM-supplied defaults are used for each installation environment unless you change them. The defaults for each installation environment can be changed independently of the others.

Listing installation defaults

You can use a job similar to the one in [Figure 19 on page 113](#) to list the installation defaults in effect at your site.

```
//DEFAULTS JOB <Job Card Parameters>
//SHOWDEF EXEC PGM=ICETOOL,REGION=1024K
//TOOLMSG DD SYSOUT=A
//DFSMSG DD SYSOUT=A
//LIST1 DD SYSOUT=A
//TOOLIN DD *
          DEFAULTS LIST(LIST1)
/*
```

Figure 19. ICETOOL DEFAULTS job

DEFAULTS produces a three-part report showing:

1. The merged PARMLIB/ICEMAC installation default values for ICEAM1-4 and ICETD1-4 that will be used at run time.
2. The specified PARMLIB ICEPRMxx member option values for ICEAM1-4 and ICETD1-4 (for reference).
3. The ICEMAC installation default values for ICEAM1-4 and ICETD1-4 (for reference).

Note: Be sure that the environment for the DEFAULTS run is the same as the environment for your DFSORT runs. For example, use a JOBLIB or STEPLIB for the DEFAULTS run if you will be using it for your DFSORT runs.

[Figure 20 on page 114](#), [Figure 21 on page 115](#), [Figure 22 on page 115](#), [Figure 23 on page 116](#), [Figure 24 on page 117](#), and [Figure 25 on page 117](#) show the first part of the DEFAULTS report that would be produced if no changes were made to the installation options with an ICEPRMxx member of PARMLIB or with the ICEMAC macro, that is, it shows all of the IBM-supplied defaults.

z/OS DFSORT 3.2 MERGED PARMLIB/ICEMAC DEFAULTS - 1 -				
* IBM-SUPPLIED DEFAULT (ONLY SHOWN IF DIFFERENT FROM THE SPECIFIED DEFAULT)				
ITEM	JCL (ICEAM1) VALUE	INV (ICEAM2) VALUE	TSO (ICEAM3) VALUE	TSOINV (ICEAM4) VALUE
-----	-----	-----	-----	-----
ENABLE	NONE	NONE	NONE	NONE
ABCODE	MSG	MSG	MSG	MSG
ALTSEQ	SEE BELOW	SEE BELOW	SEE BELOW	SEE BELOW
ARESALL	0	0	0	0
ARESINV	NOT APPLICABLE	0	NOT APPLICABLE	0
CFW	YES	YES	YES	YES
CHALT	NO	NO	NO	NO
CHECK	YES	YES	YES	YES
CINV	YES	YES	YES	YES
COBEXIT	COB2	COB2	COB2	COB2
COLLKEY	UCA600	UCA600	UCA600	UCA600
DIAGSIM	NO	NO	NO	NO
DSA	128	128	128	128
DSPSIZE	MAX	MAX	MAX	MAX
DYNALOC	(SYSDA,4)	(SYSDA,4)	(SYSDA,4)	(SYSDA,4)
DYNAPCT	10	10	10	10
DYNAUTO	YES	YES	YES	YES
DYNAPC	256	256	256	256
EFS	NONE	NONE	NONE	NONE
EQUALS	VLBLKSET	VLBLKSET	VLBLKSET	VLBLKSET
ERET	RC16	RC16	RC16	RC16
ESTAE	YES	YES	YES	YES
EXITCK	STRONG	STRONG	STRONG	STRONG
EXPMAX	MAX	MAX	MAX	MAX
EXPOLD	50%	50%	50%	50%
EXPRES	10%	10%	10%	10%
FSZEST	NO	NO	NO	NO
GENER	NOT APPLICABLE	IEBGENR	NOT APPLICABLE	IEBGENR
GNPAD	NOT APPLICABLE	RC0	NOT APPLICABLE	RC0
GNTRUNC	NOT APPLICABLE	RC0	NOT APPLICABLE	RC0
HIPRMAX	OPTIMAL	OPTIMAL	OPTIMAL	OPTIMAL
IDRCPCT	NONE	NONE	NONE	NONE
IEXIT	NO	NO	NO	NO
IGNCKPT	YES	YES	YES	YES
IOMAXBF	35651584	35651584	35651584	35651584
LIST	YES	YES	YES	YES
LISTX	YES	YES	YES	YES
LOCALE	NONE	NONE	NONE	NONE
MAXLIM	1048576	1048576	1048576	1048576
MINLIM	450560	450560	450560	450560
MOSIZE	MAX	MAX	MAX	MAX
MOWRK	YES	YES	YES	YES
MSGCON	NONE	NONE	NONE	NONE
MSGDDN	SYSOUT	SYSOUT	SYSOUT	SYSOUT
MSGPRT	ALL	ALL	ALL	ALL
NOMSGDD	QUIT	QUIT	QUIT	QUIT
NULLOFL	RC0	RC0	RC0	RC0
NULLOUT	RC0	RC0	RC0	RC0
ODMAXBF	2097152	2097152	2097152	2097152
OUTREL	YES	YES	YES	YES
OUTSEC	YES	YES	YES	YES
OVERRGN	65536	16384	65536	16384

Figure 20. Page 1 of the output from the ICETOOL DEFAULTS job showing IBM-supplied defaults

z/OS DFSORT 3.2 MERGED PARMLIB/ICEMAC DEFAULTS - 2 -				
* IBM-SUPPLIED DEFAULT (ONLY SHOWN IF DIFFERENT FROM THE SPECIFIED DEFAULT)				
ITEM	JCL (ICEAM1) VALUE	INV (ICEAM2) VALUE	TSO (ICEAM3) VALUE	TSOINV (ICEAM4) VALUE
OVFLO	RC0	RC0	RC0	RC0
PAD	RC0	RC0	RC0	RC0
PARMDDN	DFSPARM	DFSPARM	DFSPARM	DFSPARM
RESALL	4096	4096	4096	4096
RESET	YES	YES	YES	YES
RESINV	NOT APPLICABLE	0	NOT APPLICABLE	0
SDB	INPUT	INPUT	INPUT	INPUT
SDBMSG	NO	NO	NO	NO
SIZE	MAX	MAX	MAX	MAX
SMF	NO	NO	NO	NO
SOLRF	YES	YES	YES	YES
SORTLIB	PRIVATE	PRIVATE	PRIVATE	PRIVATE
SPANINC	RC16	RC16	RC16	RC16
SVC	109	109	109	109
SZERO	YES	YES	YES	YES
TEXTIT	NO	NO	NO	NO
TMAXLIM	6291456	6291456	6291456	6291456
TRUNC	RC0	RC0	RC0	RC0
TUNE	STOR	STOR	STOR	STOR
VERIFY	NO	NO	NO	NO
VIO	NO	NO	NO	NO
VLLONG	NO	NO	NO	NO
VLSCMP	NO	NO	NO	NO
VLSHRT	NO	NO	NO	NO
VSAMBS	OPTIMAL	OPTIMAL	OPTIMAL	OPTIMAL
VSAMEMT	YES	YES	YES	YES
VSAMIO	NO	NO	NO	NO
WRKREL	YES	YES	YES	YES
WRKSEC	YES	YES	YES	YES
Y2PAST	80	80	80	80
ZDPRINT	YES	YES	YES	YES
ZSORT	NO	NO	NO	NO

Figure 21. Page 2 of the output from the ICETOOL DEFAULTS job showing IBM-supplied defaults

```

z/OS DFSORT 3.2 MERGED PARMLIB/ICEMAC DEFAULTS - 3 -

ICEAM1 ALTSEQ TABLE: SAME AS IBM-SUPPLIED ALTSEQ TABLE BELOW

ICEAM2 ALTSEQ TABLE: SAME AS IBM-SUPPLIED ALTSEQ TABLE BELOW

ICEAM3 ALTSEQ TABLE: SAME AS IBM-SUPPLIED ALTSEQ TABLE BELOW

ICEAM4 ALTSEQ TABLE: SAME AS IBM-SUPPLIED ALTSEQ TABLE BELOW

IBM-SUPPLIED ALTSEQ TABLE (IN HEXADECIMAL):
00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F
20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F 30 31 32 33 34 35 36 37 38 39 3A 3B 3C 3D 3E 3F
40 41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F 50 51 52 53 54 55 56 57 58 59 5A 5B 5C 5D 5E 5F
60 61 62 63 64 65 66 67 68 69 6A 6B 6C 6D 6E 6F 70 71 72 73 74 75 76 77 78 79 7A 7B 7C 7D 7E 7F
80 81 82 83 84 85 86 87 88 89 8A 8B 8C 8D 8E 8F 90 91 92 93 94 95 96 97 98 99 9A 9B 9C 9D 9E 9F
A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 AA AB AC AD AE AF B0 B1 B2 B3 B4 B5 B6 B7 B8 B9 BA BB BC BD BE BF
C0 C1 C2 C3 C4 C5 C6 C7 C8 C9 CA CB CC CD CE CF D0 D1 D2 D3 D4 D5 D6 D7 D8 D9 DA DB DC DD DE DF
E0 E1 E2 E3 E4 E5 E6 E7 E8 E9 EA EB EC ED EE EF F0 F1 F2 F3 F4 F5 F6 F7 F8 F9 FA FB FC FD FE FF

```

Figure 22. Page 3 of the output from the ICETOOL DEFAULTS job showing IBM-supplied defaults

z/OS DFSORT 3.2 MERGED PARMLIB/ICEMAC DEFAULTS - 4 -				
* IBM-SUPPLIED DEFAULT (ONLY SHOWN IF DIFFERENT FROM THE SPECIFIED DEFAULT)				
ITEM	TD1 (ICETD1) VALUE	TD2 (ICETD2) VALUE	TD3 (ICETD3) VALUE	TD4 (ICETD4) VALUE
SUN	NONE	NONE	NONE	NONE
MON	NONE	NONE	NONE	NONE
TUE	NONE	NONE	NONE	NONE
WED	NONE	NONE	NONE	NONE
THU	NONE	NONE	NONE	NONE
FRI	NONE	NONE	NONE	NONE
SAT	NONE	NONE	NONE	NONE
ABCODE	MSG	MSG	MSG	MSG
ALTSEQ	SEE BELOW	SEE BELOW	SEE BELOW	SEE BELOW
ARESALL	0	0	0	0
ARESDIN	0	0	0	0
CFW	YES	YES	YES	YES
CHALT	NO	NO	NO	NO
CHECK	YES	YES	YES	YES
CINV	YES	YES	YES	YES
COBEXIT	COB2	COB2	COB2	COB2
COLLKEY	UCA600	UCA600	UCA600	UCA600
DIAGSIM	NO	NO	NO	NO
DSA	128	128	128	128
DSPSIZE	MAX	MAX	MAX	MAX
DYNALOC	(SYSDA,4)	(SYSDA,4)	(SYSDA,4)	(SYSDA,4)
DYNAPCT	10	10	10	10
DYNAUTO	YES	YES	YES	YES
DYNAPC	256	256	256	256
EFS	NONE	NONE	NONE	NONE
EQUALS	VLBLKSET	VLBLKSET	VLBLKSET	VLBLKSET
ERET	RC16	RC16	RC16	RC16
ESTAE	YES	YES	YES	YES
EXITCK	STRONG	STRONG	STRONG	STRONG
EXPMAX	MAX	MAX	MAX	MAX
EXPOLD	50%	50%	50%	50%
EXPRES	10%	10%	10%	10%
FSZEST	NO	NO	NO	NO
GENER	IEBGENR	IEBGENR	IEBGENR	IEBGENR
GNPAD	RC0	RC0	RC0	RC0
GNTRUNC	RC0	RC0	RC0	RC0
HIPRMAX	OPTIMAL	OPTIMAL	OPTIMAL	OPTIMAL
IDRCPCT	NONE	NONE	NONE	NONE
IEXIT	NO	NO	NO	NO
IGNCKPT	YES	YES	YES	YES
IOMAXBF	35651584	35651584	35651584	35651584
LIST	YES	YES	YES	YES
LISTX	YES	YES	YES	YES
LOCALE	NONE	NONE	NONE	NONE
MAXLIM	1048576	1048576	1048576	1048576
MINLIM	450560	450560	450560	450560
MOSIZE	MAX	MAX	MAX	MAX
MOWRK	YES	YES	YES	YES
MSGCON	NONE	NONE	NONE	NONE
MSGDDN	SYSOUT	SYSOUT	SYSOUT	SYSOUT
MSGPRT	ALL	ALL	ALL	ALL
NOMSGDD	QUIT	QUIT	QUIT	QUIT

Figure 23. Page 4 of the output from the ICETOOL DEFAULTS job showing IBM-supplied defaults

z/OS DFSORT 3.2 MERGED PARMLIB/ICEMAC DEFAULTS - 5 -				
* IBM-SUPPLIED DEFAULT (ONLY SHOWN IF DIFFERENT FROM THE SPECIFIED DEFAULT)				
ITEM	TD1 (ICETD1) VALUE	TD2 (ICETD2) VALUE	TD3 (ICETD3) VALUE	TD4 (ICETD4) VALUE
NULLOFL	RC0	RC0	RC0	RC0
NULLOUT	RC0	RC0	RC0	RC0
ODMAXBF	2097152	2097152	2097152	2097152
OUTREL	YES	YES	YES	YES
OUTSEC	YES	YES	YES	YES
OVERRGN	65536	65536	65536	65536
OVFLO	RC0	RC0	RC0	RC0
PAD	RC0	RC0	RC0	RC0
PARMDDN	DFSPARM	DFSPARM	DFSPARM	DFSPARM
RESALL	4096	4096	4096	4096
RESET	YES	YES	YES	YES
RESINV	0	0	0	0
SDB	INPUT	INPUT	INPUT	INPUT
SDBMSG	NO	NO	NO	NO
SIZE	MAX	MAX	MAX	MAX
SMF	NO	NO	NO	NO
SOLRF	YES	YES	YES	YES
SORTLIB	PRIVATE	PRIVATE	PRIVATE	PRIVATE
SPANINC	RC16	RC16	RC16	RC16
SVC	109	109	109	109
SZERO	YES	YES	YES	YES
TEXTIT	NO	NO	NO	NO
TMAXLIM	6291456	6291456	6291456	6291456
TRUNC	RC0	RC0	RC0	RC0
TUNE	STOR	STOR	STOR	STOR
VERIFY	NO	NO	NO	NO
VIO	NO	NO	NO	NO
VLLONG	NO	NO	NO	NO
VLSCMP	NO	NO	NO	NO
VLSHRT	NO	NO	NO	NO
VSAMBS	OPTIMAL	OPTIMAL	OPTIMAL	OPTIMAL
VSAMEMT	YES	YES	YES	YES
VSAMIO	NO	NO	NO	NO
WRKREL	YES	YES	YES	YES
WRKSEC	YES	YES	YES	YES
Y2PAST	80	80	80	80
ZDPRINT	YES	YES	YES	YES
ZSORT	NO	NO	NO	NO

Figure 24. Page 5 of the output from the ICETOOL DEFAULTS job showing IP-supplied defaults

```

z/OS DFSORT 3.2 MERGED PARMLIB/ICEMAC DEFAULTS - 6 -

ICETD1 ALTSEQ TABLE: SAME AS IBM-SUPPLIED ALTSEQ TABLE BELOW

ICETD2 ALTSEQ TABLE: SAME AS IBM-SUPPLIED ALTSEQ TABLE BELOW

ICETD3 ALTSEQ TABLE: SAME AS IBM-SUPPLIED ALTSEQ TABLE BELOW

ICETD4 ALTSEQ TABLE: SAME AS IBM-SUPPLIED ALTSEQ TABLE BELOW

IBM-SUPPLIED ALTSEQ TABLE (IN HEXADECIMAL):
00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F
20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F 30 31 32 33 34 35 36 37 38 39 3A 3B 3C 3D 3E 3F
40 41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F 50 51 52 53 54 55 56 57 58 59 5A 5B 5C 5D 5E 5F
60 61 62 63 64 65 66 67 68 69 6A 6B 6C 6D 6E 6F 70 71 72 73 74 75 76 77 78 79 7A 7B 7C 7D 7E 7F
80 81 82 83 84 85 86 87 88 89 8A 8B 8C 8D 8E 8F 90 91 92 93 94 95 96 97 98 99 9A 9B 9C 9D 9E 9F
A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 AA AB AC AD AE AF B0 B1 B2 B3 B4 B5 B6 B7 B8 B9 BA BB BC BD BE BF
C0 C1 C2 C3 C4 C5 C6 C7 C8 C9 CA CB CC CD CE CF D0 D1 D2 D3 D4 D5 D6 D7 D8 D9 DA DB DC DD DE DF
E0 E1 E2 E3 E4 E5 E6 E7 E8 E9 EA EB EC ED EE EF F0 F1 F2 F3 F4 F5 F6 F7 F8 F9 FA FB FC FD FE FF

```

Figure 25. Page 6 of the output from the ICETOOL DEFAULTS job showing IBM-supplied defaults

If the merged PARMLIB/ICEMAC default value for an installation option is different from the IBM-supplied default value, the IBM-supplied default value is shown below the merged PARMLIB/ICEMAC default value. Figure 26 on page 118 and Figure 27 on page 119 show an example of how the beginning of each section of the ICETOOL DEFAULTS report might look if some of the defaults have been changed with ICEPRM03 in PARMLIB and with the ICEMAC Macro.

Z/OS DFSORT 3.2 MERGED PARMLIB/ICEMAC DEFAULTS - 1 -				
* IBM-SUPPLIED DEFAULT (ONLY SHOWN IF DIFFERENT FROM THE SPECIFIED DEFAULT)				
ITEM	JCL (ICEAM1) VALUE	INV (ICEAM2) VALUE	TSO (ICEAM3) VALUE	TSOINV (ICEAM4) VALUE
-----	-----	-----	-----	-----
ENABLE	NONE	TD1	NONE	NONE
ABCODE	MSG	99	MSG	99
		* MSG		* MSG
ALTSEQ	SEE BELOW	SEE BELOW	SEE BELOW	SEE BELOW
ARESALL	0	0	0	0
ARESINV	NOT APPLICABLE	0	NOT APPLICABLE	0
CFW	YES	YES	YES	YES
CHALT	YES	YES	NO	NO
	* NO	* NO		
CHECK	YES	YES	YES	YES
CINV	YES	YES	YES	YES
.				
.				
.				

Z/OS DFSORT 3.2 MERGED PARMLIB/ICEMAC DEFAULTS - 4 -				
* IBM-SUPPLIED DEFAULT (ONLY SHOWN IF DIFFERENT FROM THE SPECIFIED DEFAULT)				
ITEM	TD1 (ICETD1) VALUE	TD2 (ICETD2) VALUE	TD3 (ICETD3) VALUE	TD4 (ICETD4) VALUE
-----	-----	-----	-----	-----
SUN	0600-2000	NONE	NONE	NONE
	* NONE			
MON	NONE	NONE	NONE	NONE
TUE	NONE	NONE	NONE	NONE
WED	NONE	NONE	NONE	NONE
THU	NONE	NONE	NONE	NONE
FRI	NONE	NONE	NONE	NONE
SAT	0600-2000	NONE	NONE	NONE
	* NONE			
ABCODE	99	MSG	MSG	MSG
	* MSG			
ALTSEQ	SEE BELOW	SEE BELOW	SEE BELOW	SEE BELOW
ARESALL	0	0	0	0
ARESINV	0	0	0	0
CFW	YES	YES	YES	YES
CHALT	YES	NO	NO	NO
	* NO			
CHECK	YES	YES	YES	YES
CINV	YES	YES	YES	YES
.				
.				
.				

Z/OS DFSORT 3.2 PARMLIB ICEPRMXX MEMBER OPTIONS - 7 -							
ICEOPT COMMAND IN EFFECT: START ICEOPT,ICEPRM=(03)							
RELEASE:	3.2						
MODULE:	ICEPRML						
APAR LEVEL:	NONE						
COMPILED:	03/12/25						
AREA:	1						
JCL (ICEAM1) OPTIONS							
ITEM	VALUE	MEMBER					
-----	-----	-----					
RESALL	40000	ICEPRM03					
CHALT	YES	ICEPRM03					
INV (ICEAM2) OPTIONS							
ITEM	VALUE	MEMBER					
-----	-----	-----					
RESINV	40000	ICEPRM03					
CHALT	YES	ICEPRM03					
TSO (ICEAM3) OPTIONS - NONE							
TSOINV (ICEAM4) OPTIONS - NONE							

Figure 26. Output from an ICETOOL DEFAULTS job for changed defaults (Part 1 of 2)


```

TD1 (ICETD1) OPTIONS
ITEM          VALUE                                     MEMBER
-----
CHALT         YES                                     ICEPRM03

TD2 (ICETD2) OPTIONS - NONE
TD3 (ICETD3) OPTIONS - NONE TD4 (ICETD4) OPTIONS - NONE
.
.
.
Z/OS DFSORT  3.2 ICEMAC DEFAULTS                      - 8 -

ITEM          JCL (ICEAM1) VALUE                      INV (ICEAM2) VALUE                      TSO (ICEAM3) VALUE                      TSOINV (ICEAM4) VALUE
-----
MODULE        3.2
APAR LEVEL    ICEAM1
              NONE
COMPILED      03/12/25                                03/12/25                                03/12/25                                03/12/25
ENABLE        NONE                                    TD1                                       NONE                                    NONE
ABCODE        MSG                                    99                                       MSG                                    99
ALTSEQ        SEE BELOW                              SEE BELOW                              SEE BELOW                              SEE BELOW
ARESALL       0                                     0                                     0                                     0
ARESINV       0 NOT APPLICABLE                      0 NOT APPLICABLE                      0 NOT APPLICABLE                      0 NOT APPLICABLE
CFW           YES                                    YES                                    YES                                    YES
CHALT         NO                                     NO                                     NO                                     NO
CHECK         YES                                    YES                                    YES                                    YES
CINV          YES                                    YES                                    YES                                    YES
.
.
.
Z/OS DFSORT  3.2 ICEMAC DEFAULTS                      - 11 -

ITEM          TD1 (ICETD1) VALUE                      TD2 (ICETD2) VALUE                      TD3 (ICETD3) VALUE                      TD4 (ICETD4) VALUE
-----
RELEASE       3.2
MODULE        ICETD1
APAR LEVEL    NONE
COMPILED      03/12/25                                03/12/25                                03/12/25                                03/12/25
SUN           0600-2000                              NONE                                    NONE                                    NONE
MON           NONE                                    NONE                                    NONE                                    NONE
TUE           NONE                                    NONE                                    NONE                                    NONE
WED           NONE                                    NONE                                    NONE                                    NONE
THU           NONE                                    NONE                                    NONE                                    NONE
FRI           NONE                                    NONE                                    NONE                                    NONE
SAT           0600-2000                              NONE                                    NONE                                    NONE
ABCODE        99                                       MSG                                    MSG                                    MSG
ALTSEQ        SEE BELOW                              SEE BELOW                              SEE BELOW                              SEE BELOW
ARESALL       0                                     0                                     0                                     0
ARESINV       0                                     0                                     0                                     0
CFW           YES                                    YES                                    YES                                    YES
CHALT         NO                                     NO                                     NO                                     NO
CHECK         YES                                    YES                                    YES                                    YES
CINV          YES                                    YES                                    YES                                    YES
.
.
.

```

Figure 27. Output from an ICETOOL DEFAULTS job for changed defaults (Part 2 of 2)

Chapter 5. Controlling resources using ICEIEXIT

This chapter contains Programming Interface information.

You can write and install an initialization exit, ICEIEXIT, to provide more control over certain installation- and run-time options. Your ICEIEXIT can analyze the original installation- and run-time options together with information from system control blocks to determine whether to change certain options at run time.

For example, by using ICEIEXIT, your site can:

- Enforce an absolute limit on maximum storage for DFSORT jobs
- Increase or decrease the maximum storage for DFSORT jobs based on performance requirements or job name
- Cause DFSORT to reserve more storage for jobs with user exits that process a large number of data sets
- Use different options for test and production modes.

When installed and activated, ICEIEXIT receives control in the initialization phase of DFSORT after all scanning and cross-checking of options is complete. DFSORT passes a set of installation-time options and a set of run-time options to ICEIEXIT.

Note: Although an ICEIEXIT can change certain options based on the time of day, the time-of-day installation default environments provide an easier and more comprehensive method for time-of-day option controls. See [Chapter 4, “Changing the installation defaults,” on page 43](#) for details.

What run-time options can ICEIEXIT change?

Run-time options are those in effect for the application. Some options are specified when invoking a job; DFSORT chooses other run-time options based on the default settings established at installation time. Your ICEIEXIT can change certain run-time options, but keep in mind that DFSORT might override them because of conflicts between options, technique, or function restrictions, or for performance considerations.

Several run-time options can be changed:

- Maximum storage to be used
- Storage to be reserved
- Action for critical error
- Sequence checking of output records
- Use of Hiperspace for Hipersorting
- Use of data space for dataspace sorting
- Use of a memory object for memory object sorting
- Maximum OUTFIL data set buffer space

Note: Any ICEIEXIT changes to run-time options override all changes made to these options by installation options, other run-time options, or an EFS program.

See details of the parameter list interface in [“Contents of the run-time options list” on page 124](#).

What installation- and run-time information can ICEIEXIT examine?

Installation-time information

ICEIEXIT can examine these installation options:

- DSPSIZE
- ERET
- HIPRMAX
- MINLIM
- MAXLIM
- MOSIZE
- ODMAXBF
- RESALL/RESINV
- SIZE
- TMAXLIM
- VERIFY.

Run-time information

ICEIEXIT can examine:

- Whether the application is a sort, merge, or copy
- Whether DFSORT was invoked directly or through a program
- Whether the Blockset technique is being used
- Whether storage was allocated above 16 MB virtual
- The options SIZE/MAINSIZE, RESALL/RESINV, ABEND/RC16, VERIFY/NOVERIFY, HIPRMAX, DSPSIZE, ODMAXBF, and MOSIZE.

See details of the parameter list interfaces in the sections [“Contents of the installation-time options list”](#) on page 123 and [“Contents of the run-time options list”](#) on page 124 for more information.

Writing and installing ICEIEXIT

These are the steps for writing and installing ICEIEXIT:

1. Write your ICEIEXIT in assembler language using standard linkage and register conventions. The information in [“Interface points for ICEIEXIT”](#) on page 122 provides the specific values that ICEIEXIT can change or examine.
2. Include your ICEIEXIT in the library that contains the DFSORT modules or in a separate library. See [“Using initialization and termination exits”](#) on page 38 for more information on installing an ICEIEXIT.
3. Set the IEXIT=YES installation option with an ICEPRMxx member in PARMLIB or with the ICEMAC macro.

In addition, be aware of the following:

- If ICEIEXIT is installed resident, it must be reentrant.
- ICEIEXIT must reside and run below 16 MB virtual. (That is, its residency and addressing mode must be 24-bit.)
- Before returning to DFSORT, ICEIEXIT must free any storage it obtained and close any data sets it opened. Otherwise, the storage is not available to DFSORT for processing.
- ICEIEXIT might be entered more than once per DFSORT application.
- A large ICEIEXIT routine might increase the amount of storage required for DFSORT runs.

Interface points for ICEIEXIT

In order to write an ICEIEXIT, you need to know what specific values you can change or examine in:

- General register 1

- The installation-time options list
- The run-time option list

Contents of general register 1

On entry to ICEIEXIT, general register 1 contains the address of the following parameter list:

On entry to ICEIEXIT, general register 1 contains the address of the following parameter list:	
Bytes 1 through 4	
Address of run-time options list	
Address of installation-time options list	

A mapping macro, ICEDIEX, is in the DFSORT target library, ICEUSER, after you have done the SMP/E APPLY step. ICEDIEX provides an assembler DSECT for each ICEIEXIT parameter list.

When ICEIEXIT is exited, the options that can be changed are examined by DFSORT in their original locations. Therefore, changing the address of the parameter list passed to your routine by DFSORT has no effect.

Contents of the installation-time options list

The following options are set at installation time. ICEIEXIT can examine, but not change, these values.

Bytes 1 through 4	
Length of list	
Maximum storage	
Storage to be reserved	
Flags	
MINLIM	
MAXLIM	
TMAXLIM	
HIPRMAX	DSPSIZE
ODMAXBF	
Reserved	
MOSIZE	

Length of list (4 bytes)

This value defines the length of the installation-time parameter list, including this length field.

Maximum storage (4 bytes)

This value represents the maximum amount of storage DFSORT requests, taking into account the installation options SIZE, MAXLIM, TMAXLIM, and MINLIM. DFSORT also looks at whether storage can be allocated above 16 MB virtual. See the Flags2 description under [“Contents of the run-time options list”](#) on page 124 for details of flag settings.

Storage to be reserved (4 bytes)

This value represents the total storage DFSORT reserves during a run for system storage and for use by the invoking program. This value takes into account the installation options RESALL and RESINV, and whether SIZE=MAX is in effect.

Flags (4 bytes)

Bit 0 = 1:

Installation option ERET=ABEND is set

Bit 0 = 0:

Installation option ERET=RC16 is set

Bit 1 = 1:

Installation option VERIFY=YES is set

Bit 1 = 0:

Installation option VERIFY=NO is set

Bit 2 = 1:

Installation option SIZE=MAX is set

Bit 2 = 0:

Installation option SIZE=n is set

Bits 3-4:

Not used

Bit 5 = 1:

Installation option HIPRMAX=OPTIMAL is set.

Bit 5 = 0:

Installation option HIPRMAX=n is set.

Bits 6-31:

For future use

MINLIM (4 bytes)

The value for installation option MINLIM.

MAXLIM (4 bytes)

The value for installation option MAXLIM.

TMAXLIM (4 bytes)

The value for installation option TMAXLIM.

HIPRMAX (2 bytes)

When Flags bit 5 is 1, installation option HIPRMAX=OPTIMAL was specified and this field is not meaningful.

When Flags bit 5 is 0, this field contains a value between 0 and 32767 that represents the maximum amount, in MB, of Hipspace to be committed for Hipersorting. This value is determined from installation option HIPRMAX=n, or from installation option HIPRMAX=p% and the "Configured Central Storage" value shown in the Run-Time Options List.

DSPSIZE (2 bytes)

The value specified by installation option DSPSIZE=n or DSPSIZE=MAX. A value of 10000 corresponds to DSPSIZE=MAX whereas a value between 0 and 9999 represents a specified or calculated value in MB.

ODMAXBF (4 bytes)

The value for installation option ODMAXBF. A value of 262144 (256 KB) to 16777216 (16 MB) corresponds to ODMAXBF=n.

MOSIZE (4 bytes)

The value specified by installation option MOSIZE=MAX, MOSIZE=n or MOSIZE=p%. A value of 2147483647 (hexadecimal 7FFFFFFF) corresponds to MOSIZE=MAX. A value between 0 and 2147483646 (hexadecimal 7FFFFFFE) represents a specified or calculated value in MB, determined from MOSIZE=n, or from MOSIZE=p% and the available central storage.

Contents of the run-time options list

ICEIEXIT can examine and change the following options:

Bytes 1 through 4	
Length of list	
Maximum storage	
Storage to be reserved	
Flags1	
Flags2	
HIPRMAX	DSPSIZE
ODMAXBF	
Reserved	
Configured Central Storage	
MOSIZE	

Length of list (4 bytes)

This value defines the length of the run-time parameter list, including this field.

Maximum storage (4 bytes)

This value represents the maximum amount of storage DFSORT requests, taking into account the run-time options SIZE and MAINSIZE and the installation options SIZE, MAXLIM, TMAXLIM, and MINLIM. DFSORT also checks whether storage can be allocated above 16 MB virtual.

If the value in this field is changed, DFSORT uses the new value as the maximum amount of storage requested. If a new value of less than 90112 (88 KB) is specified, DFSORT uses 90112 (88 KB) as the new value. If a new value greater than 2097152000 (2000 MB) is specified, DFSORT uses 2097152000 (2000 MB) as the new value. The new value appears in messages as if it were the SIZE/MAINSIZE value. A change to this field does not affect the RESALL or RESINV values.

Storage to be reserved (4 bytes)

This value represents the total storage to be reserved by DFSORT during execution. It takes into account RESALL and RESINV, and whether SIZE/MAINSIZE=MAX is in effect.

If the value in this field is changed, DFSORT uses the new value as the total storage to be reserved. The new value appears in messages as if it were the RESALL value and as if the RESINV value were 0.

Flags1 (4 bytes)**Bit 0 = 1:**

ERET=ABEND is in effect

Bit 0 = 0:

ERET=RC16 is in effect

Bit 1 = 1:

VERIFY=YES is in effect

Bit 1 = 0:

VERIFY=NO is in effect

Bits 2-4 =

Not used

Bit 5 = 1:

HIPRMAX=OPTIMAL is in effect.

Bit 5 = 0:

HIPRMAX=n is in effect.

Bits 6-31 =

For future use

Note: If a flag is changed, DFSORT uses the new corresponding option. The new option is reported in appropriate messages.

Flags2 (4 bytes)

Bit 0 = 1:

This is a sort application

Bit 0 = 0:

This is not a sort application

Bit 1 = 1:

This is a merge application

Bit 1 = 0:

This is not a merge application

Bit 2 = 1:

This is a copy application

Bit 2 = 0:

This is not a copy application

Bit 3 = 1:

DFSORT is directly invoked

Bit 3 = 0:

DFSORT is program-invoked

Bit 4 = 1:

The Blockset technique is being used

Bit 4 = 0:

The Blockset technique is not being used

Bit 5 = 1:

SIZE=MAX is in effect

Bit 5 = 0:

SIZE=MAX is not in effect

Bit 6 = 1:

Storage can be allocated above 16 MB virtual

Bit 6 = 0:

Storage must be allocated below 16 MB virtual

Bit 7 = 1:

OUTFIL processing is being used

Bit 7 = 0:

OUTFIL processing is not being used

Bits 8-31 =

For future use

Note: Flags2 is for information only. Any changes to it are ignored.

HIPRMAX (2 bytes)

When Flags1 bit 5 is 1, HIPRMAX=OPTIMAL is in effect and this field is not meaningful.

When Flags1 bit 5 is 0, this field contains a value between 0 and 32767 that represents the maximum amount, in MB, of Hiperspace to be committed for Hipersorting. The actual amount of Hiperspace used will not exceed this value but might be less if DFSORT determines that using the maximum will increase system paging significantly. This value may have been calculated from a percentage of the "Configured Central Storage" value shown. If this value is 0, Hipersorting is not used.

If Flags1 bit 5 is 1 when you return from ICEIEXIT, DFSORT uses HIPRMAX=OPTIMAL.

If Flags1 bit 5 is 0 when you return from ICEIEXIT, DFSORT uses the value from 0 to 32767 in this field as the HIPRMAX=n value. However, if the value in this field is greater than 32767, DFSORT uses HIPRMAX=OPTIMAL.

DSPSIZE (2 bytes)

If between 1 and 9999, this value specifies the maximum amount, in MB, of data space to be used for dataspace sorting. The actual amount of data space used does not exceed this value.

If this value is 0, dataspace sorting is not used.

If this value is 10000, (equivalent to DSPSIZE=MAX), DFSORT determines the maximum amount of data space it should use based on the size of the input file being sorted and the paging activity of the system.

If the value in this field is changed, DFSORT uses the new value as explained previously. If a new value greater than 10000 is specified, DFSORT uses 10000 (DSPSIZE=MAX) as the new value. The new value is reported in appropriate messages as the DSPSIZE value.

ODMAXBF (4 bytes)

If between 262144 (256 KB) and 16777216 (16 MB), this value specifies the maximum buffer space to be used for each OUTFIL data set (ODMAXBF value in effect). If OUTFIL is not being used, this value is 0 and has no meaning.

If the value in this field is changed and OUTFIL is being used, DFSORT uses the new value as the maximum buffer space for each OUTFIL data set. If a new value of less than 262144 (256 KB) is specified, DFSORT uses 262144 (256 KB) as the new value. If a new value of more than 16777216 (16 MB) is specified, DFSORT uses 16777216 (16 MB) as the new value. The new value is reported in appropriate messages as the ODMAXBF value.

Configured Central Storage (4 bytes)

This value specifies the portion of central storage DFSORT considers for the use of Hipersorting, memory object sorting, or dataspace sorting, in pages.

The value in this field cannot be changed.

MOSIZE (4 bytes)

If between 1 and 2147483646 (hexadecimal 7FFFFFFE), this value specifies the maximum size, in MB, to be used for memory object sorting. The actual size of a memory object used will not exceed this value but might be less if DFSORT determines that using the maximum will increase system paging significantly. This value may have been calculated as a percentage of the available central storage.

If this value is 0, memory object sorting is not used.

If this value is 2147483647 (hexadecimal 7FFFFFFF - equivalent to MOSIZE=MAX), DFSORT determines dynamically the maximum size of a memory object to be used for memory object sorting based on the size of the input file being sorted and the central storage usage activity.

If the value in this field is changed, DFSORT uses the new value as explained previously. If a new value greater than 2147483647 is specified, DFSORT uses 2147483647 (MOSIZE=MAX) as the new value. The new value is reported in appropriate messages as the MOSIZE value.

A sample ICEIEXIT routine

Figure 28 on page 128 and Figure 29 on page 129 are an example of a simple ICEIEXIT routine, which does the following:

- Sets the storage available to DFSORT for the weekly payroll jobs (jobname of 'PAYR*' and stepname of 'WEEKLY') to 24 MB.
- Ensures that the storage available to DFSORT for copy applications is not greater than 6 MB.

The ICEIEXIT routine in this example is provided as part of the sample job ICEIXREC in the SICESAMP library. See “Using initialization and termination exits ” on page 38 for details on how to install this sample routine.


```

ICEIEXIT CSECT
* DSECTS FOR SYSTEM CONTROL BLOCKS
  CVT DSECT=YES
  IKJTCB DSECT=YES
  IEFTIOT1 ,
* DSECT FOR ICEIEXIT PARAMETER LISTS
  ICEDIEX
* EQUATES
R0 EQU 0
R1 EQU 1
R2 EQU 2
R3 EQU 3
R4 EQU 4
R11 EQU 11
R12 EQU 12
R13 EQU 13
R14 EQU 14
R15 EQU 15
*
CURTCB EQU 4 CURRENT TCB OFFSET
ICEIEXIT CSECT
* PROLOGUE
  USING *,R12 SET UP BASE REGISTER
  SAVE (14,12) SAVE REGISTERS
  LR R12,R15 LOAD BASE REGISTER
  GETMAIN RC,LV=WORKLEN,BNDRY=DBLWD
  LTR R15,R15 WAS STORAGE OBTAINED
  BZ WORKOK IF YES, THEN CONTINUE
  RETURN (14,12) ELSE, RETURN TO DFSORT
WORKOK LR R11,R1 SAVEAREA ADDRESS IN R11
  USING WORKAREA,R11 MAKE WORKAREA ADDRESSABLE
  ST R13,SAVEAREA+4 CHAIN BACKWARD
  L R1,24(R13) LOAD ADDRESS OF PARAMETER LIST
  LR R4,R13 SAVE ADDRESS OF CALLER'S SAVEAREA
  LA R13,SAVEAREA LOAD ADDRESS OF SAVEAREA
  ST R13,8(R4) CHAIN FORWARD
* ESTABLISH ADDRESSABILITY TO PARAMETER LISTS PASSED TO ICEIEXIT
  USING ILMN,R1 ESTABLISH MAIN LIST ADDRESS
  L R2,ILMEXA GET EXECUTION LIST ADDRESS
  USING ILEXL,R2 AND ESTABLISH BASE
  L R3,ILMINA GET INSTALLATION LIST ADDRESS
  USING ILINL,R3 AND ESTABLISH BASE
  DROP R1
* DETERMINE IF THIS IS ONE OF THE WEEKLY PAYROLL JOBS.
* IF SO, IT WILL HAVE A JOBNAME OF PAYR* AND A STEPNAME OF WEEKLY.
* NOTE THAT WEEKLY IS NOT A PROCEDURE STEP.
PAYRCK DS OH
  L R1,16 GET CVT
  USING CVT,R1 CVT DSECT BASE
  L R1,CVTTCBP GET NEXT/CURRENT TCB POINTERS
  L R1,CURTCB(,R1) GET CURRENT TCB
  USING TCB,R1 TCB DSECT BASE
  L R1,TCBTIO GET TIOT
  USING TIOT1,R1 TIOT DSECT BASE
  CLC TIOCNJOB(4),PAYR IF JOBNAME IS NOT PAYR*,
  BNE CPYCK CHECK FOR A COPY APPLICATION
  CLC TIOCSTEP(8),PAYST IF STEPNAME IS NOT WEEKLY,
  BNE CPYCK CHECK FOR A COPY APPLICATION
* NOTE: IF THE STEP WAS A PROCEDURE STEP, THE STEP NAME WOULD BE
* AT TIOCSTEP+8.

```

Figure 28. Sample ICEIEXIT routine (1 of 2)


```

        DROP R1
* ONE OF THE WEEKLY PAYROLL JOBS - LET DFSORT USE UP TO
* 24 MEGABYTES OF STORAGE.
PAYRJOB DS 0H
        MVC ILEXMXS,PAYRMAX      TELL DFSORT THAT 24 MEGABYTES
*                                     IS THE LIMIT
        B GOBACK                RETURN TO DFSORT WITH CHANGED
*                                     STORAGE LIMIT
* DETERMINE IF THIS IS A COPY APPLICATION.
CPYCK DS 0H
        TM ILEXF2A,ILEXFCPY      IF IT'S NOT A COPY APPLICATION,
        BZ GOBACK                DON'T CHANGE STORAGE LIMIT
* COPY APPLICATION - DO NOT LET DFSORT USE MORE THAN
* 6 MEGABYTES OF STORAGE.
CPYAPPL DS 0H
        L R1,COPYMAX              GET MAXIMUM STORAGE FOR COPY
        CL R1,ILEXMXS             IF NOT USING MORE THAN MAXIMUM,
        BNL GOBACK                KEEP CURRENT STORAGE LIMIT
        ST R1,ILEXMXS            TELL DFSORT THAT 6 MEGABYTES
*                                     IS THE LIMIT
* EPILOGUE
GOBACK L R13,SAVEAREA+4          RESTORE SAVEAREA
        LA R0,WORKLEN            LOAD LENGTH OF WORKAREA
        LR R1,R11                LOAD ADDRESS OF WORKAREA
        FREEMAIN R,LV=(R0),A=(R1) FREE WORKAREA
        RETURN (14,12)           RETURN TO DFSORT
PAYR DC CLR'PAYR'                PAYR* JOB NAME
PAYST DC CL8'WEEKLY'            PAYROLL STEP NAME
PAYRMAX DC A(24*1024*1024)      LET WEEKLY PAYROLL JOBS USE
*                                     UP TO 24 MEGABYTES
COPYMAX DC A(6*1024*1024)      LET COPY APPLICATIONS USE
*                                     UP TO 6 MEGABYTES
WORKAREA DSECT                  WORK AREA
SAVEAREA DS 18F                 SAVE AREA
WORKLEN EQU *-WORKAREA
        DROP R2,R3,R11,R12      DROP BASE REGISTERS
        END

```

Figure 29. Sample ICEIEXIT routine (2 of 2)

Chapter 6. Collecting statistical data

This chapter contains Programming Interface information.

You can collect DFSORT statistical data in two ways:

- Collect the DFSORT system management facilities (SMF) type-16 records from the system SMF data sets
- Write a termination exit (ICETEXIT).

You can use the information DFSORT provides in its SMF record and the information passed to an ICETEXIT routine to analyze a particular aspect of DFSORT execution that you are interested in studying, such as resource usage or performance.

Why use DFSORT's SMF record?

There are several reasons why you might collect information about DFSORT applications using only the DFSORT SMF type-16 records written to the system SMF data sets:

- You already have procedures to collect this information.
- The scope of the information you are interested in collecting is limited to the information available in the SMF record.
- You do not want to write and maintain an ICETEXIT.

Why use an ICETEXIT?

There are several reasons why you might prefer to write and maintain an ICETEXIT:

- You want to collect statistics on each running application directly from DFSORT.
- You want to collect DFSORT SMF data and DFSORT SMF records without having to go through the system SMF data sets.
- You want to collect DFSORT SMF records without causing DFSORT to pass a record to SMF.
- You want to collect data that is not in the DFSORT SMF record.

Collecting data from the DFSORT SMF record

DFSORT provides several types of information in its SMF record:

Application Identification Information

The name of the job and step that executed.

Date and Time Information

The date and time that the application began and terminated.

Termination Information

Detailed information describing how DFSORT terminated and why.

Access Method Count Information

A total count of the number of GETs, PUTs, READs, WRITEs, and EXCPs issued for the SORTIN, SORTOUT and work data sets.

Timing Information

Task Control Block (TCB) and Service Request Block (SRB) time.

Data Set Information

Detailed information about SORTIN, SORTINnn, SORTOUT, and OUTFIL data sets used for the application.

To collect this information, or any information, from the SMF records written to the system SMF data sets, you must make sure that:

- The correct DFSORT SVC for this release was loaded in LPA or MLPA.
- The DFSORT installation defaults have been modified so that the SMF parameter indicates the type of SMF record you want DFSORT to pass to SMF. You can use the SMF run-time option to produce SMF records only for selected DFSORT applications.
- The SMFPRMxx active member of parmlib specifies that SMF type-16 records should be collected for the types of address spaces you are interested in. You can also specify that only specific subtypes be collected. For information about how to specify collection of SMF type-16 records, see [z/OS MVS System Management Facilities \(SMF\)](#).

Controlling what DFSORT passes to SMF

When you so request, DFSORT passes one type of record to SMF for each executing application. This record is the SMF type-16 record. By coding the installation or run-time SMF parameter properly, you can control whether DFSORT passes a short (subtype 1), a full (subtype 2), or a short (subtype 3) record to SMF.

If you code SMF=SHORT:

- A short (subtype 1) record is passed to SMF for a successful application.
- A short (subtype 3) record is passed to SMF for an unsuccessful application.

If you code SMF=FULL:

- A short (subtype 1) record is passed to SMF for a successful fixed-length record Peerage/Vale sort application.
- A full (subtype 2) record is passed to SMF for a successful Blockset application.
- A full (subtype 2) record is passed to SMF for a successful variable-length record Peerage/Vale sort application.
- A short (subtype 3) record is passed to SMF for an unsuccessful application.

If you code SMF=NO:

- No record is passed to SMF.

The installation or run-time SMF parameter can also be used to control the type of SMF record passed to your termination exit. If you intend to write a termination exit (ICETEXIT) and want to place some control over the type of record DFSORT passes to ICETEXIT, refer to [“Controlling what DFSORT passes to ICETEXIT”](#) on page 138.

Note:

1. If SMF is inactive and you specified SMF=FULL or SMF=SHORT, DFSORT issues message ICE094I and does not pass a record to SMF.
2. SMF=FULL can degrade performance for variable-length record applications.
3. For a successful or unsuccessful Conventional merge or tape work data set sort application, a short (subtype 1) record is always produced.

When DFSORT cannot pass a record to SMF

As a general rule, DFSORT does not pass a record to SMF when:

- A system or user abend occurs when NOESTAE is in effect for the Blockset or Peerage/Vale techniques.
- DFSORT abends in the Blockset technique and both ABSTP and NOESTAE are in effect.

SMF record structure

The short (subtype 1 or subtype 3) and full (subtype 2) records always contain the following:

- Header section
- Product section
- Data section

For a variable-length record application, the full (subtype 2) record also includes the following:

- Record-length distribution section

For a Blockset application, the full (subtype 2) record might also include one or more of the following:

- Input Data Set Sections; one section for each of up to 16 SORTIN (including concatenated data sets) or SORTINnn data sets
- SORTOUT Data Set Section
- OUTFIL Data Set Sections; one section for each of up to 16 OUTFIL data sets.

See [“Controlling what DFSORT passes to SMF” on page 132](#) for details of when each subtype is passed to SMF.

Note: Depending on the severity of an unsuccessful run, information might not be provided in some fields of the SMF record. For a complete description of what fields are provided when, refer to [Appendix D, “SMF Type-16 record,” on page 175](#).

Mapping the SMF record

A program that processes the DFSORT SMF record must adhere to the standard SMF convention for addressing variable portions of the record. Because the product, data, record-length distribution, and data set sections might be at different displacements from one release to the next, always use the ICESMF mapping macro to reference fields within the SMF type-16 record. ICESMF provides separate DSECTs for each SMF record section and eliminates the need to use hard-coded offsets. You can find ICESMF in the DFSORT target library ICEUSER after you have done the SMP/E APPLY step.

Note: Programs that use the DFSORT ICESMF macro DSECTs and reference fields ICERCDS, ICEBYTES, ICECPUT, and ICEWBLK show alignment warnings when assembled.

Verifying supported record sections

To verify the number of sections provided in the SMF record being processed, a program must examine the following SMF record fields:

- Number of sections descriptor
- Number of product sections (if zero, then there are no product sections)
- Number of data sections (if zero, then there are no data sections)
- Number of record-length distribution sections field (if zero, then there are no record-length distribution sections).
- Number of input (SORTIN/SORTINnn) data set sections (if zero, then there are no input data set sections; if more than one, there are multiple input data set sections)
- Number of SORTOUT data set sections (if zero, then there are no SORTOUT data set sections)
- Number of OUTFIL data set sections (if zero, then there are no OUTFIL data set sections; if more than one, there are multiple OUTFIL data set sections)

Accessing multiple input data set sections

For a successful Blockset application, with SMF=FULL in effect and concatenated SORTIN data sets (sort or copy) or more than one SORTINnn data set (merge), multiple input data set sections will be provided in the SMF record. The following fields in the SMF record header section can be used to access these sections:

- Offset to the first input data set section
- Input data set section length

- Number of input data set sections

The offset to the first input data set section can be used to locate the start of the input data set sections. Since the input data set sections will be contiguous, the second section can be located by adding the input data set section length to the start of the first section, and so on, for the number of input data set sections indicated.

Note: A maximum of 16 input data set sections is provided. The following fields in the SMF record data section can be used to determine the actual number of input data sets for the application:

- Number of SORTIN data sets, including concatenated data sets
- Number of SORTINnn data sets

Accessing multiple OUTFIL data set sections

For a successful Blockset application, with SMF=FULL in effect and more than one OUTFIL data set, multiple OUTFIL data set sections will be provided in the SMF record. The following fields in the SMF record header section can be used to access these sections:

- Offset to the first OUTFIL data set section
- OUTFIL data set section length
- Number of OUTFIL data set sections

The offset to the first OUTFIL data set section can be used to locate the start of the OUTFIL data set sections. Since the OUTFIL data set sections will be contiguous, the second section can be located by adding the OUTFIL data set section length to the start of the first section, and so on, for the number of OUTFIL data set sections indicated.

Note: A maximum of 16 OUTFIL data set sections is provided. The following field in the SMF record data section can be used to determine the actual number of OUTFIL data sets for the application:

- Number of OUTFIL data sets

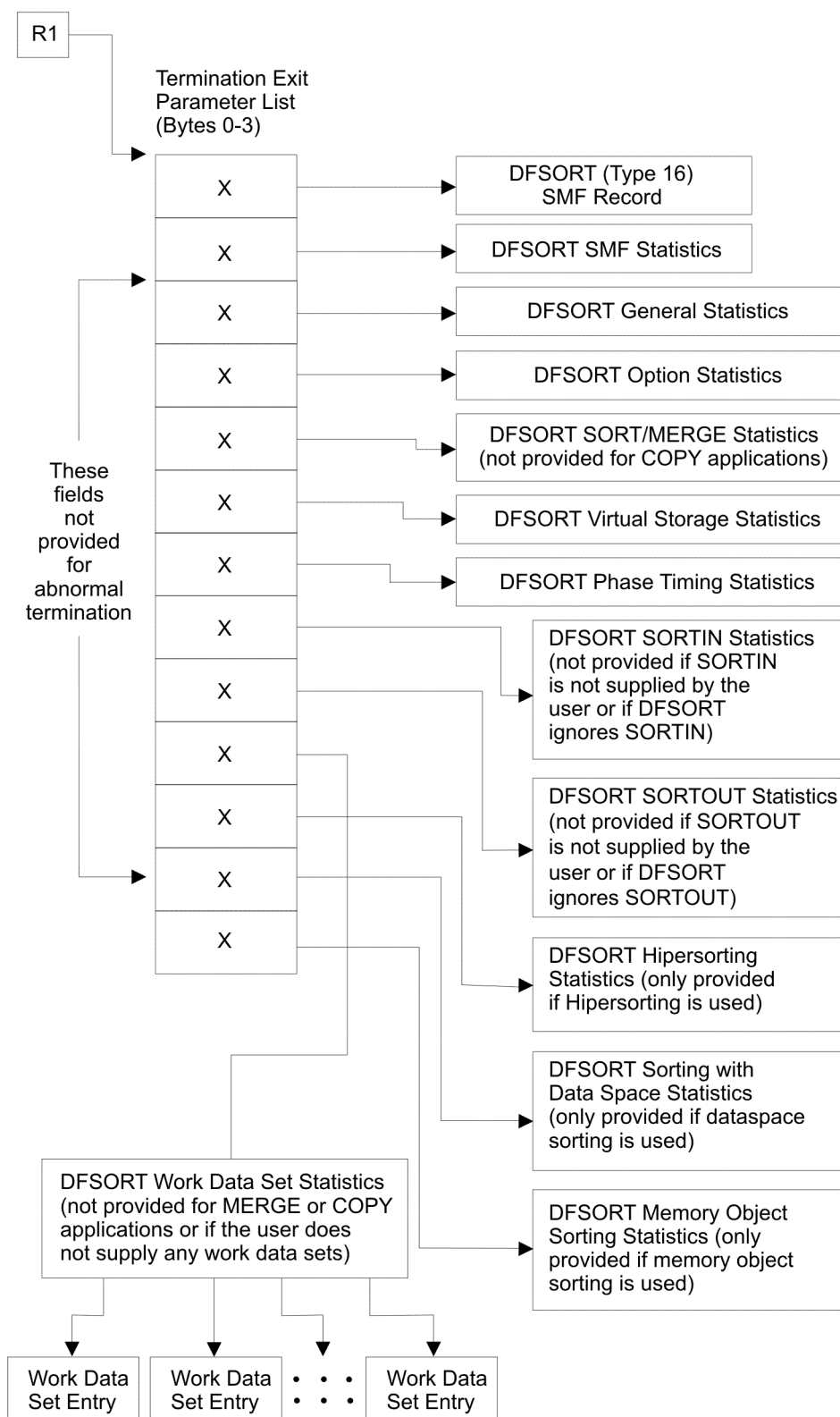
Identifying what release produced your SMF record

If your site collects SMF type-16 records created by different releases of DFSORT, you can identify the release that created a specific SMF type-16 record by examining ICERELNM, a 4-byte EBCDIC Release Level Identifier in the record.

Collecting data using an ICETEXIT

Following is a brief description of the information DFSORT provides to the ICETEXIT that you write. For a complete description of all the information provided to an ICETEXIT, refer to [Appendix E, “Termination exit \(ICETEXIT\) parameters,”](#) on page 193.

On entry to ICETEXIT, general register 1 contains the address of the parameter list shown in [Figure 30](#) on page 135.



(One Work Data Set Entry is provided for each work data set.)

Figure 30. Information DFSORT passes to your ICETEXIT

The information provided to ICETEXIT includes:

SMF Record

The DFSORT SMF type-16 record is passed to ICETEXIT whether or not DFSORT is passing the record to SMF. Refer to [“Controlling what DFSORT passes to ICETEXIT” on page 138](#) for a description of how you can use the SMF parameter to control the type of record passed to your ICETEXIT.

SMF Statistics

Indicates whether:

- SMF=FULL, SMF=SHORT, or SMF=NO is in effect
- DFSORT attempted to pass the SMF record to SMF
- SMF is active in your system
- The SMF record was successfully passed to SMF.

The return code from the SMFWTM macro to DFSORT is also provided.

General Statistics

Includes system and DFSORT release statistics:

- Identifies the calling module if DFSORT is program invoked
- Gives, when applicable, the DFSORT revert code (as it appears in message ICE800I)
- Tells whether DFSORT modules are resident for the technique in use
- Tells whether DFSORT was loaded above 16 MB virtual.

Option Statistics

Provides a list of options in effect and a list of all specified user exits.

SORT/MERGE Statistics

Provides:

- The number of sort or merge keys
- The number of intermediate merges
- The number of input files to an E32 user exit
- A listing of all field formats specified.

Virtual Storage Statistics

Provides values in effect for ARESALL, ARESINV, OVERRGN, TMAXLIM, RESALL, RESINV, MINLIM, MAXLIM, and SIZE or MAINSIZE. It also provides the region size specified by the user and the amount of virtual storage DFSORT used above and below 16 MB virtual.

Phase-Timing Statistics

SRB, TCB, and elapsed times are provided for each DFSORT phase.

SORTIN(nn) Statistics

Provides detailed information describing the SORTIN data sets.

SORTOUT Statistics

Provides detailed information describing the SORTOUT data set.

Work Data Set Statistics

Provides information on each work data set.

Hipersorting Statistics

Provides information on the amount of Hiperspace used for Hipersorting.

Statistics for Sorting with Data Space

Provides information on the amount of data space used for dataspace sorting.

Memory Object Sorting Statistics

Provides information on the size of the memory object used for memory object sorting.

Note:

1. Depending on the application running, an address to a particular set of statistics (other than the SMF record and the SMF statistics) might not be provided. In this case, the field within the parameter list contains a zero.
2. If DFSORT abends, only an address to the DFSORT SMF record and to the SMF statistics is provided. Before attempting to process the rest of the termination exit parameter list, examine the information in the SMF record related to the type of termination. This helps you decide what step to take next.
3. The ICEBTIME, ICEBDATA, and ICESID fields of the SMF record supplied to ICETEXIT contain binary zeroes. This information is normally filled in when SMF copies the record to an SMF buffer.
4. The address to the SORTIN statistics is zero if SORTIN is not present or DFSORT ignores SORTIN.
5. The address to the SORTOUT statistics is zero if SORTOUT is not present or DFSORT ignores SORTOUT.
6. The address to the work data set statistics is zero if a merge or copy application is running.
7. The address to the SORT/MERGE statistics is zero if a copy application is running.
8. ICETEXIT is not called for a Conventional merge or tape work data set sort.

Installing an ICETEXIT

If you intend to collect any of the information DFSORT passes to ICETEXIT, you must do the following:

1. Write an ICETEXIT routine.
2. Include your ICETEXIT in the library that contains the DFSORT modules or in a separate library. See [“Using initialization and termination exits ” on page 38](#) for more information about installing an ICETEXIT.
3. Set the TEXTIT=YES installation option with an ICEPRMxx member in PARMLIB or with the ICEMAC macro.

When installed and activated correctly, ICETEXIT receives control in the termination phase of DFSORT. ICETEXIT is called for each successful and unsuccessful run of DFSORT, except in the following situations:

- While a tape work data set sort application is running
- While a Conventional merge application is running
- After a critical message is issued when both NOESTAE and ABSTP are in effect
- After a system abend when NOESTAE is in effect
- After an abend by a user exit when NOESTAE is in effect
- After an abend in EFS Major Call 1.

Writing an ICETEXIT routine

When writing your own ICETEXIT, be aware that:

- ICETEXIT must be written using the interfaces described in [Appendix E, “Termination exit \(ICETEXIT\) parameters,” on page 193](#).
- If ICETEXIT is installed resident, it must be reentrant.
- ICETEXIT must reside and execute below 16 MB virtual; that is, its residency and addressing mode must be 24-bit.
- ICETEXIT is entered only once for a DFSORT run.
- Critical messages and system and user abends that occur after control is passed to ICETEXIT are not reflected in the information passed to ICETEXIT.
- Abnormal termination (including abends) that occur in EFS Major Call 4 and EFS Major Call 5 are not reflected in the information provided to ICETEXIT.
- Use of an ICETEXIT might increase the minimum storage required to execute DFSORT successfully.
- Carefully consider both the amount of working storage required by your ICETEXIT and the overall size of the ICETEXIT load module. An ICETEXIT requiring large amounts of storage might fail if the REGION is too small.

- If ICETEXIT for any DFSORT application must wait to obtain a resource, DFSORT execution will be delayed. You should carefully consider whether or not ICETEXIT should use a resource (for example, a statistics collection data set) that must be serialized.
- If an ICETEXIT is used to collect DFSORT statistics in a data set, the data set you define must be large enough to accommodate the maximum number of DFSORT applications that might run between the times that you empty this data set.
- Before returning to DFSORT, ICETEXIT must free any storage it obtained and close any data sets it opened. Otherwise, the storage used will not be available to DFSORT if it is called again in the same step.

Controlling what DFSORT passes to ICETEXIT

DFSORT always passes your ICETEXIT an SMF record. However, by selectively altering the SMF parameter, you can control whether your ICETEXIT is passed *short* or *full* records. Table 9 on page 138 describes the precise actions that DFSORT takes for different settings of the TEXTIT installation parameter and SMF installation or run-time parameter.

Table 9. Controlling the type of SMF record DFSORT passes to ICETEXIT			
ICETEXIT	SMF=FULL	SMF=SHORT	SMF=NO
TEXTIT=YES Termination exit receives control.	If the run is successful, a full (subtype 2) record is passed to SMF and to the termination exit. If the run is unsuccessful, a short (subtype 3) record is passed to SMF and to the termination exit.	If the run is successful, a short (subtype 1) record is passed to SMF and to the termination exit. If the run is unsuccessful, a short (subtype 3) record is passed to SMF and to the termination exit.	If the run is successful, a short (subtype 1) record is passed to the termination exit. No record is passed to SMF. If the run is unsuccessful, a short (subtype 3) record is passed to the termination exit. No record is passed to SMF.
TEXTIT=NO Termination exit does not receive control.	If the run is successful, a full (subtype 2) record is passed to SMF. No record is passed to the termination exit. If the run is unsuccessful, a short (subtype 3) record is passed to SMF. No record is passed to the termination exit.	If the run is successful, a short (subtype 1) record is passed to SMF. No record is passed to the termination exit. If the run is unsuccessful, a short (subtype 3) record is passed to SMF. No record is passed to the termination exit.	No record is passed to SMF or to the termination exit.

Mapping ICETEXIT statistics

DFSORT provides a set of DSECTs by means of the ICEDTEX and ICESMF macros to map out all of the statistical structures passed to ICETEXIT. These macros are in the SICEUSER library. You should use ICEDTEX and ICESMF in any program that references termination exit statistics.

ICETEXIT examples

The following examples describe two possible ways to write an ICETEXIT. Figure 31 on page 139 presents a nonexecutable overview of how you might code ICETEXIT if you intend to process all the statistics that DFSORT provides ICETEXIT.

Figure 35 on page 143 demonstrates an executable ICETEXIT you might write to collect timing information on successfully running DFSORT applications.

A nonexecutable ICETEXIT

```

        TITLE 'ICETEXIT --- TERMINATION EXIT'
ICETEXIT CSECT
*****
*
*   Description -- This program is a non-executable example of
*                  how each set of DFSORT statistics passed
*                  to ICETEXIT can be processed.
*
*   Entry Points -- ICETEXIT; called by DFSORT.
*
*   Input -- On entry, R1 points to a parameter list containing
*            pointers to DFSORT statistics.
*
*   Output -- It is the responsibility of the programmer to save
*            any information that he or she wishes to collect.
*
*****
*
*****
*   DFSORT REQUIRED MAPPING MACROS
*****
*
        ICEDTEX
        ICESMF
*
*****
*   ENTRY LINKAGE AND PREPARATION
*****
*
        . . . . .
        . . . . .
        . . . . .
*
*****
*   PROCESS THE DFSORT (TYPE 16) SMF RECORD
*****
*
        LR      R11,R1      Load addr of TEXTIT parameter list
        USING   ICETPAR,R11  Make TEXTIT parm list addressable
        L       R10,ICETSMFA Load address of DFSORT SMF record
        USING   ICESMFH,R10  Make the SMF header section addr

        . . . . . process header section of SMF record. . . . .
        LR      R9,R10      Load address of beginning of SMF rec
        A       R9,ICEPROD   Add offset to product section
        USING   ICESMFH,R9   Make product section addressable
        . . . . . process product section of SMF record . . . . .
        LR      R8,R10      Load address of beginning of SMF rec
        A       R8,ICEDATA   Add offset to data section
        USING   ICESMFD,R8   Make data section addressable
        . . . . . process data section of SMF record. . . . .
        CLC     ICERSUB,ICESUBF Is this a full SMF record?
        BNE     SMFSTATS     If not, go process SMF statistics
        LR      R7,R10      Load address of beginning of SMF rec
        A       R7,ICESTAT   Add offset to record length stats
        USING   ICESMFR,L,R7 Make record length stats addressable

```

Figure 31. A nonexecutable ICETEXIT (1 of 4)


```

. . . . process record length distributions section . .
. . DROP      R7,R8,R9,R10      Drop all SMF record sections
*
*****
*          PROCESS THE DFSORT SMF STATISTICS          *
*****
*
SMFSTATS L      R10,ICETSST      Load address of SMF statistics
          USING  ICESST,R10      Make SMF stats addressable
          . . . . .
          . . . . .
          DROP    R10            Drop SMF statistics
*
*****
*          PROCESS THE DFSORT GENERAL STATISTICS      *
*****
*
GENSTATS L      R10,ICETGEN      Load address of General Statistics
          LTR     R10,R10        Are there any general statistics?
          BZ      OPTSTATS      If not, go process option statistics
          USING   ICESST,R10     Make general stats addressable
          . . . . .
          . . . . .
          DROP    R10            Drop general statistics
*****
*          PROCESS THE DFSORT OPTION STATISTICS      *
*****
*
OPTSTATS L      R10,ICETOPTS     Load address of General Statistics
          LTR     R10,R10        Are there any option statistics?
          BZ      SMSTATS       If not, go process SORT/MERGE stats
          USING   ICEOPTS,R10    Make option stats addressable
          . . . . .
          . . . . .
          DROP    R10            Drop option statistics
*
*****
*          PROCESS THE DFSORT SORT/MERGE STATISTICS  *
*****
*
SMSTATS  L      R10,ICETSMS      Load address of SORT/MERGE stats
          LTR     R10,R10        Are there any option statistics?
          BZ      VRTSTATS       If not, go process virtual stg stats
          USING   ICESMS,R10     Make SORT/MEREG stats addressable
          . . . . .
          . . . . .
          DROP    R10            Drop SORT/MERGE statistics
*
*****
*          PROCESS THE DFSORT VIRTUAL STORAGE STATISTICS *
*****
*
VRTSTATS L      R10,ICETVIRT     Load address of virtual stg stats
          LTR     R10,R10        Are there any virt stg stats?
          BZ      PHTSTATS       If not, go process phase timings stats
          USING   ICEVSTOR,R10   Make virtual stg stats addressable
          . . . . .
          . . . . .
          DROP    R10            Drop virtual storage statistics
*

```

Figure 32. A nonexecutable ICETEXIT (2 of 4)


```

*****
*          PROCESS THE DFSORT PHASE TIMING STATISTICS          *
*****
*
PHTSTATS L      R10,ICETPTIM    Load address of phase timing stats
  LTR      R10,R10             Are there any phase timing stats?
  BZ       SINSTATS            If not, go process SORTIN statistics
  USING    ICEPHAST,R10        Make phase timing stats addressable
  . . . . .
  . . . . .
  DROP     R10                 Drop phase timing statistics
*****
*          PROCESS THE DFSORT SORTIN STATISTICS                *
*****
*
SINSTATS L      R10,ICETSIN     Load address of SORTIN statistics
  LTR      R10,R10             Are there any SORTIN statistics?
  BZ       SOTSTATS            If not, go process SORTOUT statistics
  USING    ICESRTIN,R10        Make SORTIN statistics addressable
  . . . . .
  . . . . .
  DROP     R10                 Drop SORTIN statistics
*
*****
*          PROCESS THE DFSORT SORTOUT STATISTICS                *
*****
*
SOTSTATS L      R10,ICETSOUT    Load address of SORTOUT statistics
  LTR      R10,R10             Are there any SORTOUT statistics?
  BZ       WKSTATS            If not, go process work data set stats
  USING    ICESRTOT,R10        Make SORTOUT statistics addressable
  . . . . .
  . . . . .
  DROP     R10                 Drop SORTOUT statistics
*
*****
*          PROCESS THE DFSORT WORK DATA SET STATISTICS        *
*          AND WORK DATA SET ENTRIES                          *
*****
*
WKSTATS  L      R10,ICETSWK     Load address of work data set stats
  LTR      R10,R10             Are there any work data set stats?
  BZ       HIPRSTAT            If not, go check on Hipersorting stats
  USING    ICEWRKDS,R10        Make work data set stats addressable
  . . . . .
  . . . . .
  LH       R9,ICESWKEN         Load number of work data sets
  LA       R8,ICESWK01         Load address of first entry pointer
ENTRYLOP L      R7,0(,R8)       Load address of an entry
  USING    ICEWORK,R7          Make work data set entry addressable
  . . . . .
  . . . . .
  . . . . . process work data set entry . . . . .
  . . . . .
  LA       R8,4(,R8)           Point at next entry pointer
  BCTR     R9,0                Decrement count of entries
  LTR      R9,R9               Have all entries been processed?
  BNZ     ENTRYLOP             If not, go process next entry
  DROP     R7,R10              Drop entry and work data set stats
*

```

Figure 33. A nonexecutable ICETEXIT (3 of 4)


```

*****
*          PROCESS THE DFSORT HIPERSORTING STATISTICS          *
*****
*
HIPRSTAT L          R10,ICETHIPR      Load address of Hipersorting stats
      LTR          R10,R10           Was Hipersorting used for sort?
      BZ          DATASTAT          If not, go process data space stats
      USING       ICEHIPER,R10       Make Hipersorting stats addressable
      . . . . .
      . . . . .
      . . . . .
      DROP        R10               Drop Hipersorting statistics
*
*****
*          PROCESS THE DFSORT SORTING WITH DATA SPACE STATISTICS          *
*****
*
DATASTAT L          R10,ICETDATA      Load address of data space stats
      LTR          R10, R10          Was dataspace sorting used for sort?
      BZ          MOSTAT            If not, go process memory object stats
      USING       ICEDATAS,R10      Make data space stats addressable
      . . . . .
      . . . . .
      . . . . .
      DROP        R10               Drop data space statistics
*****
*          PROCESS THE DFSORT MEMORY OBJECT STATISTICS          *
*****
*
MOSTAT  L          R10,ICETMO        Load address of memory object stats
      LTR          R10,R10          Was a memory object used for sort?
      BZ          CLOSDOWN          If not, go perform closedown function
      USING       ICEMOSEC,R10      Make memory object stats addressable
      . . . . .
      . . . . .
      . . . . .
      DROP        R10               Drop memory object sorting stats
*****
*          EXIT LINKAGE AND ANY NECESSARY CLOSE DOWN OPERATIONS          *
*****
*
CLOSDOWN DS          0H
      . . . . .
      . . . . .
      . . . . .
*
      END          ICETEXIT

```

Figure 34. A nonexecutable ICETEXIT (4 of 4)

An executable ICETEXIT

```

        TITLE 'ICETEXIT --- TERMINATION EXIT'
ICETEXIT CSECT
*****
*
*   Description -- For each successfully executing DFSORT
*                   application, this program collects information
*                   on the number of records and bytes sorted, the
*                   region size specified, and the amount of TCB,
*                   SRB, and elapsed time used by the application.
*
*   Entry Points -- ICETEXIT; called by DFSORT.
*
*   Input -- On entry, R1 points to a parameter list containing
*            pointers to DFSORT statistics.
*
*   Output -- All collected information is appended to a dynamically
*             allocated data set. No information will be saved
*             in this dynamically allocated data set if, upon entry
*             to ICETEXIT, another task already has the data set
*             allocated.
*
*   External References:
*       Macros -- GETMAIN, FREEMAIN, ENQ, DEQ, PUT,
*                ICESMF, ICEDTEX, IHADCB
*
*   Attributes: Reentrant.
*
*****
*
*****
*   EQUATE GENERAL PURPOSE REGISTERS
*
*****
R0      EQU      0
R1      EQU      1
R2      EQU      2
R3      EQU      3
R4      EQU      4
R5      EQU      5
R6      EQU      6
R7      EQU      7
R8      EQU      8
R9      EQU      9
R10     EQU     10
R11     EQU     11
R12     EQU     12
R13     EQU     13
R14     EQU     14
R15     EQU     15

```

Figure 35. An executable ICETEXIT (1 of 7)


```

*
*****
*   INCLUDE MAPPING MACROS FOR THE DFSORT (TYPE 16) SMF RECORD AND   *
*   THE DFSORT TERMINATION INSTALLATION EXIT STATISTICS             *
*****
*
      PRINT      NOGEN
      IHADCB     DSORG=PS           System DCB macro
      ICESMF     DFSORT SMF mapping macro
      ICETEX     DFSORT Termination Exit mapping macro
*
*****
*   GET WORK STORAGE AND SAVE CALLER'S REGISTERS                     *
*****
*
ICETEXIT CSECT
      STM      R14,R2,12(R13)      Save some of caller's registers
      ST       R12,20+4*12(,R13)   Save caller's register 12
      BALR     R12,0               Set up base register
      USING   *,R12               Make ICETEXIT addressable
      GETMAIN  RC,LV=STGNEED,BNDRY=DBLWD
      STM      R3,R11,32(R13)      Save rest of caller's registers
      LTR     R15,R15              Was required storage obtained?
      BZ      STGOK               If yes, then continue
      LM      R14,R12,12(R13)      Restore caller's registers
      SLR     R15,R15              Set return code to zero
      BR      R14                 Return to caller
*
STGOK   LR      R11,R1             Save storage address in R11
        USING   STORAGE,R11        Make storage area addressable
        L       R10,24(R13)        Load address of TEXT parameter list
        ST      R13,SAVEAREA+4     Save address of caller's save area
        LA      R1,SAVEAREA        Load address of ICETEXIT save area
        ST      R1,8(R13)          Save address of ICETEXIT save area
        LR      R13,R1             Load address of ICETEXIT save area
*
*****
*   INITIALIZE STORAGE WORK AREA                                     *
*****
*
      MVC      ENQLIST,ENQCMD       Save ENQ list
      MVC      DEQLIST,DEQCMD       Save DEQ list
      MVC      OPENLIST,OPENCMD     Save OPEN list
      MVC      CLOSLIST,CLOSECMD    Save CLOSE list
      MVC      DCB,COLLECT          Save collection DCB

```

Figure 36. An executable ICETEXIT (2 of 7)


```

*
*****
*          COLLECT ALL DESIRED STATISTICS          *
*****
*
      USING      ICETPAR,R10      Make TEXTIT parm list addressable
*
      L          R9,ICETSMFA      Load addr of DFSORT SMF record
      USING     ICESMFH,R9      Make SMF header section addressable
      LR        R8,R9          Load address of header section
      A         R8,ICEDATA      Determine location of data section
      USING     ICESMFD,R8      Make data section addressable
*
      TM        ICEFLBY2,ICESORT  Is application a sort?
      BNO       RETURN          If no, then return to DFSORT
      SLR       R1,R1          Clear a register
      IC        R1,ICERC        Load DFSORT return code to caller
      LTR       R1,R1          Is execution successful?
      BNZ       RETURN          If not, then return to DFSORT
*
      MVC       JOBNAME,ICEJOBNM  Save job name
      MVC       STEPNAME,ICESTPNM Save step name
      MVC       TIMECOMP,ICETIMEE Save time of completion
      MVC       DATECOMP,ICEDATEE Save date of completion
      MVC       RECORDS,ICERCDS   Save number of records sorted
      MVC       BYTES,ICEBYTES    Save number of bytes sorted
      MVC       CPUTIME,ICECPUT   Save amount of CPU time used
      L         R1,ICETIMEE      Load time of completion
      S         R1,ICETIMES      Subtract time of start
      ST        R1,ELAPTIME      Save elapsed time
      L         R1,ICESRBTE      Load ending SRB time
      S         R1,ICESRBTS      Subtract starting SRB time
      ST        R1,SRBTIME       Save SRB time
      DROP      R8,R9           Drop header and data sections
*
      L         R9,ICETVIRT      Load addr of Virtual Stg Stats
      USING     ICEVSTOR,R9      Make Virtual Stg Stats addressable
      MVC       REGION,ICEVREGN  Save region size specified
      DROP      R9,R10          Drop stats and TEXTIT parm list
*
*****
*          ENQUEUE THE COLLECTION DATA SET          *
*****
*
      LA        R3,ENQLIST      Load address of ENQ list
      ENQ       MF=(E,(R3))     ENQ data set resource
      LTR       R15,R15         Is the collection data set available?
      BNZ       RETURN          If not, then return to DFSORT

```

Figure 37. An executable ICETEXIT (3 of 7)


```

*
*****
*          BUILD A CALL TO SVC 99 AND DYNAMICALLY          *
*          ALLOCATE THE COLLECTION DATA SET                *
*****
*
      LA      R5,SVC99ALC      Load address of allocation parameters
      USING   ALLOCATE,R5      Make allocation parms addressable
      MVC     ALCPARMS(L'ALCEND),APARMS      Move in skeleton
*
      LA      R4,ALCRB        Load address of request block
      ST      R4,ALCPARMS      Store address in SVC 99 parm list
      MVI     ALCPARMS,X'80'   Indicate end of list
*
      LA      R4,ALCTXTPS      Load address of text unit pointers
      ST      R4,ALCTXTPT      Store pointer in request block
      LA      R4,ALCDDN        Load address of DDNAME text unit
      ST      R4,ALCTXTPS      Store address in list of text units
      LA      R4,ALCDSN        Load addr of data set name text unit
      ST      R4,ALCTXTPS+4    Store address in list of text units
      LA      R4,ALCSTAT       Load addr of data set stat text unit
      ST      R4,ALCTXTPS+8    Store address in list of text units
      LA      R4,ALCNDISP      Load addr of normal disp text unit
      ST      R4,ALCTXTPS+12   Store address in list of text units
      MVI     ALCTXTPS+12,X'80' Indicate end of text ptr list
*
      LR      R1,R5            Load address of SVC 99 parameters
      SVC     99               Allocate the collection data set
      LTR     R15,R15          Was data set allocated?
      BNZ     DEQUEUE          If not, then go issue DEQ
*
*****
*          OPEN THE COLLECTION DATA SET, APPEND A NEW RECORD OF      *
*          COLLECTED INFORMATION TO IT, THEN CLOSE THE DATA SET      *
*****
*
      LA      R2,DCB           Load address of DCB
      USING   IHADCB,R2        Make DCB addressable
      MVC     DCBDDNAM,ALCDDNM Save DDNAME in DCB
      DROP    R2               Drop DCB
*
      LA      R3,OPENLIST      Load address of OPEN list
      OPEN    ((R2),(OUTPUT)),MF=(E,(R3)) Open collection data set
*
      LA      R3,DCB           Load address of collection DCB
      PUT     0(R3),RECORD     Append the new record
*
      LA      R3,CLOSLIST      Load address of CLOSE list
      CLOSE   ((R2),),MF=(E,(R3)) Close the collection data set

```

Figure 38. An executable ICETEXIT (4 of 7)


```

*
*****
*          BUILD A CALL TO SVC 99 AND DYNAMICALLY          *
*          DEALLOCATE THE COLLECTION DATA SET            *
*****
*
      LA      R3,SVC99DAL      Load addr of deallocation parameters
      USING   DALLOCAT,R3      Make parameters addressable
      MVC     DALPARMS(L'DALEND),DPARMS      Move in skeleton
*
      LA      R4,DALRB         Load address of request block
      ST      R4,DALPARMS      Store address in SVC 99 parm list
      MVI     DALPARMS,X'80'   Indicate end of list
*
      LA      R4,DALTXTPS      Load address of text unit pointers
      ST      R4,DALTXTPT      Store pointer in request block
      LA      R4,DALDDN        Load address of DDNAME text unit
      ST      R4,DALTXTPS      Store address in list of text units
      MVC     DALDDNM,ALCDDNM   Place DDNAME in parameters
      LA      R4,ALCDSN        Load addr of data set name text unit
      ST      R4,DALTXTPS+4     Store address in list of text units
      MVI     DALTXTPS+4,X'80' Indicate end of list
*
      LR      R1,R3            Load address of SVC 99 parameters
      SVC     99               Deallocate the collection data set
      DROP   R3,R5             Drop allocation/deallocation parms
*
*****
*          DEQUEUE THE COLLECTION DATA                    *
*****
*
DEQUEUE  LA      R3,DEQLIST      Load address of ENQ list
      DEQ      MF=(E,(R3))      DEQ data set resource
*
*****
*          FREE WORKING STORAGE AND RETURN TO DFSORT      *
*****
*
RETURN   DS      0H
      LA      R0,STGNEED        Load length of storage work area
      LR      R1,R11            Load address of stg work area
      L       R13,SAVEAREA+4    Load addr of caller's save area
      LM      R2,R12,28(R13)     Restore caller's registers
      FREEMAIN R,LV=(R0),A=(R1)  Free work area
      L       R14,12(R13)        Load address of caller
      SLR     R15,R15            Set return code to zero
      BR      R14               Return to caller

```

Figure 39. An executable ICETEXIT (5 of 7)


```

*
*****
*          SVC 99 DYNAMIC ALLOCATION PARAMETERS          *
*      (These parameters will be moved into a GETMAINed area)      *
*****
*
APARMS  DC      A(0)
        DC      X'14010000',F'0',A(0),F'0,0'
        DC      A(0,0,0,0)
        DC      X'005500010008'
        DC      CL8' '
        DC      X'00020001002C'
DSNAME  DC      CL44'RICHDRK.DFSORT.SORT.DATA'
        DC      X'00040001000102'
        DC      X'00050001000108'
*
*****
*          SVC 99 DYNAMIC DEALLOCATION PARAMETERS        *
*      (These parameters will be moved into a GETMAINed area)      *
*****
*
DPARMS  DC      A(0)
        DC      X'14020000',F'0',A(0),F'0,0'
        DC      A(0,0)
        DC      X'000100010008'
        DC      CL8' '
*
*****
*          MAP OF DYNAMIC ALLOCATION PARAMETERS          *
*****
*
ALLOCATE DSECT
ALCPARMS DS      A          Allocation Request Block Pointer
ALCRB    DS      2F          Allocation Request Block
ALCTXPT  DS      A,2F
ALCTXTPS DS      4F          Text Unit Pointers
ALCDDN   DS      XL6         Text Unit requesting that the
ALCDDNM  DS      CL8         allocation DDNAME be returned
ALCDSN   DS      XL6         Text Unit specifying the name of
ALCDSNM  DS      CL44        the data set to dynamically allocate
ALCSTAT  DS      XL7         Text Unit indicating data set status
ALCNDISP DS      XL7         Text Unit indicating normal disposition
ALCEND   EQU     *,*-ALLOCATE
*
*****
*          MAP OF DYNAMIC DEALLOCATION PARAMETERS        *
*****
*
DALLOCAT DSECT
DALPARMS DS      A          Deallocation Request Block Pointer
DALRB    DS      2F          Deallocation Request Block

```

Figure 40. An executable ICETEXIT (6 of 7)


```

DALTXPT DS      A,2F
DALXTPTS DS      2F          Text Unit Pointers
DALDDN  DS      XL6          Text Unit indicating DDNAME
DALDDNM DS      CL8          associated with deallocation
DALEND  EQU      *,*-DALLOCAT
*
*****
*          ENQ/DEQ AND OPEN/CLOSE PARAMETERS          *
*****
*
ICETEXIT CSECT
ENQCMD  ENQ      (QUEUE,DSNAME,E,,SYSTEM),RET=USE,MF=L
ENQSIZE EQU      *-ENQCMD
DEQCMD  DEQ      (QUEUE,DSNAME,,SYSTEM),RET=HAVE,MF=L
DEQSIZE EQU      *-DEQCMD
QUEUE   DC       CL8'TEXIT'
*
OPENCMD  OPEN    (COLLECT,(OUTPUT)),MF=L
OPENSIZE EQU      *-OPENCMD
CLOSECMD CLOSE   (COLLECT,,),MF=L
CLOSSIZE EQU      *-CLOSECMD
COLLECT  DCB     DSORG=PS,MACRF=PM,BLKSIZE=L'RECORD,LRECL=L'RECORD,    X
          BUFNO=1
DCBSIZE  EQU      *-COLLECT
*
*****
*          WORKING STORAGE DSECT          *
*****
*
STORAGE DSECT
SAVEAREA DS      18F          ICETEXIT save area
*
JOBNAME  DS      CL8          Job name which ran DFSORT
STEPNAME DS      CL8          Step name which ran DFSORT
TIMECOMP DS      F           Time of completion
DATECOMP DS      PL4         Date of completion
RECORDS  DS      F           Number of records sorted
BYTES    DS      F           Number of bytes sorted
REGION   DS      F           Region size specified
CPUTIME  DS      F           CPU time
ELAPTIME DS      F           Elapsed time
SRBTIME  DS      F           SRB time
RECORD   EQU      JOBNAME,*,JOBNAME
*
ENQLIST  DS      XL(ENQSIZE)   Space for ENQ list
DEQLIST  DS      XL(DEQSIZE)   Space for DEQ list
OPENLIST DS      XL(OPENSIZE)  Space for OPEN list
CLOSLIST DS      XL(CLOSSIZE)  Space for CLOSE list
DCB       DS      XL(DCBSIZE)  Space for DCB
SVC99ALC DS      XL(L'ALCEND)  Space for allocation parms
SVC99DAL DS      XL(L'DALEND)  Space for deallocation parms
STGNEED  EQU      *-STORAGE
*
          END      ICETEXIT

```

Figure 41. An executable ICETEXIT (7 of 7)

Chapter 7. Improving tape processing with DFSMSrmm or ICETPEX

This chapter contains Programming Interface information.

When DFSORT can obtain information about tape data sets from a tape management system, it can significantly improve the way it processes tapes. DFSORT can obtain such information automatically from DFSMSrmm, but an ICETPEX routine is required to obtain the same information from other tape management systems. Check with your tape management vendor to find out if they currently have an ICETPEX routine available or have plans to provide one in the future.

It is also possible to write your own ICETPEX routine using the information provided in this chapter.

Information passed by DFSMSrmm or ICETPEX

DFSMSrmm or an ICETPEX routine can pass specific information to DFSORT for managed input and output tapes. DFSORT can use this information in the following ways, when appropriate, to significantly improve the way it processes managed tapes:

- DFSORT can obtain accurate input filesize information for managed tapes. This can result in improved sort performance and more accurate dynamic workspace allocation.

Additionally, you do not have to supply the input filesize to DFSORT when this information is available from DFSMSrmm or ICETPEX. DFSORT will automatically use the filesize it obtains from DFSMSrmm or ICETPEX to override any FILSZ=En or SIZE=En value you specify. However, you must remove any FILSZ=n, FILSZ=Un, SIZE=n or SIZE=Un value you specify in order for DFSORT to use the filesize it obtains from DFSMSrmm or ICETPEX.

- DFSORT can obtain input and output attributes such as RECFM, LRECL and BLKSIZE for managed tapes. As a result, you do not have to specify these attributes explicitly for input and output tape data sets when this information is available from DFSMSrmm or ICETPEX.

Installing an ICETPEX routine

If your vendor supplies an ICETPEX routine, or you write one yourself, just include it in the library that contains the DFSORT modules or in a separate library in the normal order of search. DFSORT will automatically call the ICETPEX routine and use the information it provides, when appropriate.

Writing an ICETPEX routine

If you write your own ICETPEX routine, you must be aware of the following:

- Your ICETPEX routine must be written in assembler language using standard linkage register conventions, and must operate in 31-bit addressing mode. If you install your ICETPEX routine resident, it must be reentrant.
- If DFSORT needs information about an input or output tape, it will call your ICETPEX routine with the address of the following parameter list in general register 1:

Bytes 1 through 4
Address of JFCB
Address of return area

The format of the return area is as follows:

Bytes 1 through 4	
LRECL	
Reserved	
BLKSIZE	
Reserved	
BLKCOUNT	
RECFM	Reserved
Reserved	
Reserved	
Reserved	
Reserved	

The return area will initially be set to binary zeros each time your ICETPEX routine is called. If you want to pass a particular value to DFSORT (for example, BLKCOUNT), set it in the return area. If a particular value (for example, BLKCOUNT) in the return area is not updated, zero will be used for that particular value.

LRECL is the logical record length for the tape data set. If the LRECL value in the JFCB is zero, DFSORT will use the LRECL value in the return area.

BLKSIZE is the block size for the tape data set. If the BLKSIZE value in the JFCB is zero, DFSORT will use the BLKSIZE value in the return area.

BLKCOUNT is the total block count for the tape data set. If the tape data set is multivolume, BLKCOUNT represents the total block count for all volumes. If the BLKCOUNT is not updated, DFSORT will use a zero block count as the number of blocks to be sorted. If DFSORT uses a zero block count and the actual block count is significantly higher, the following may result:

- Degraded performance
- Inaccurate dynamically allocated workspace
- DFSORT out of space error message

RECFM is the record format for the tape data set. RECFM is one byte with the same format as JFCRECFM. For example, RECFM=FB has a value of X'90'. If the RECFM value in the JFCB is zero, DFSORT will use the RECFM in the return area.

- Your ICETPEX routine could be called multiple times for a single DFSORT run.
- Before returning to DFSORT, your ICETPEX routine should free any storage it obtained and close any data sets it opened. DFSORT will not be able to free the storage you obtain, or close the data sets you open.
- A large ICETPEX routine might increase the amount of storage required for DFSORT applications.

A sample ICETPEX routine

Figure 42 on page 153 is a simple example that gives the structure of an ICETPEX routine that sets the RECFM and BLKCOUNT, with the actual data collection represented by comments:

Figure 42. Sample ICETPEX routine

Appendix A. Sample jobs

This appendix describes the DFSORT sample jobs.

Customizing DFSORT

The following sample jobs can be used as appropriate, to customize DFSORT to the needs of your site.

Procedures

The target and distribution libraries for the following procedures are SICEPROC and AICESAMP. If any modifications are required to the following procs, update a copy and place it in the customer managed proc library (for example, SYS1.PROCLIB). Do not modify these procs in the SMP/E managed proc library.

- ICEPROC - Adds the standard DFSORT SORT procedure
- ICEPROC - Adds the standard DFSORT SORTD procedure

Usermods

The target and distribution libraries for the following sample usermods are SICESAMP and AICESAMP. If any modifications are required to the following usermods, update a copy and place it in a customer managed library. Do not modify these usermods in the SMP/E managed sample library.

- ICESVREC - Receives the SVC user modifications
- ICESVAPP - Applies the SVC user modifications
- ICEOPREC - Receives the options user modifications
- ICEOPAPP - Applies the options user modifications
- ICEMSREC - Receives message text user modifications
- ICEMSAPP - Applies message text user modifications
- ICEIXREC - Receives installation exit user modifications
- ICEIXAPP - Applies installation exit user modifications

Note:

1. If you customize DFSORT after you apply the product, the target library SICESAMP contains the source information needed for DFSORT customization sample jobs ICEOPREC and ICEMSREC.
2. If you customize DFSORT after you accept the product, the AICESRCE distribution library contains the source information needed for DFSORT customization sample jobs ICEOPREC and ICEMSREC.

Using DFSORT and ICETOOL

ICESTGEX has DFSORT/ICETOOL sample jobs for Storage Administrators.

Appendix B. ICEOPT messages and return codes

When you issue a START ICEOPT command from the console, or include it in an appropriate COMMNDxx member in PARMLIB, to activate specified ICEPRMxx members in PARMLIB, ICEOPT issues messages to the console to indicate the status of your installation default changes as well as any errors it detects.

ICEOPT messages

**ICEP500I ICEOPT COMPLETED - OPTIONS
CLEARED/SAVED IN AREA n**

Explanation

ICEOPT completed processing of the specified ICEPRMxx members. Any previously saved installation option values were cleared. Any valid installation option values from the specified ICEPRMxx members were saved in DFSORT storage area 1 or storage area 2. The new installation option values saved in the storage area will be merged with the ICEMAC defaults at run-time.

System action

DFSORT's ICEPRMxx installation area was changed.

Programmer response

None required. You can run an ICETOOL DEFAULTS report to list the merged PARMLIB/ICEMAC installation defaults. See [“Listing installation defaults” on page 113](#) for more information.

**ICEP501I ICEOPT COMPLETED - OPTIONS
NOT CLEARED/SAVED**

Explanation

ICEOPT completed processing of the specified ICEPRMxx members. ICEOPT detected errors (indicated by other ICEP5xxI messages) that prevented it from clearing or saving any installation option values. Any new option values will not be used; any previously saved installation option values will continue to be used.

System action

DFSORT's ICEPRMxx installation area was not changed.

Programmer response

Correct the indicated errors and reissue the START ICEOPT command.

**ICEP502I ICEOPT: PARAMETER IS
NOT ICEPRM=XX OR
ICEPRM=(XX,...,XX) - MEMBERS
IGNORED**

Explanation

An invalid form of the iceprm parameter was specified in the START ICEOPT command or the iceprm parameter was missing. iceprm=xx and iceprm=(xx,...,xx) are the only valid forms of this parameter. ICEOPT did not process any ICEPRMxx members.

System action

DFSORT's ICEPRMxx installation area was not changed.

Programmer response

Issue the START ICEOPT command again with a valid form of the iceprm parameter.

**ICEP503I ICEOPT: MORE THAN 10
MEMBERS SPECIFIED - EXCESS
MEMBERS IGNORED**

Explanation

An iceprm=(xx,...,xx) parameter was specified in the START ICEOPT command that specified more than 10 member suffixes. ICEOPT processed the first 10 members and ignored the other members.

System action

DFSORT's ICEPRMxx installation area was changed.

Programmer response

None required. However, if the ignored members contain needed installation option values, consolidate the installation option values into only 10 members and issue the START ICEOPT command again with those 10 members.

ICEP504I ICEOPT: PARMLIB ALLOCATION FAILED - MEMBERS IGNORED

Explanation

The IEFPRMLB service used by ICEOPT to allocate the logical parmlib data sets failed. The specified ICEPRMxx members were not processed.

System action

DFSORT's ICEPRMxx installation area was not changed.

Programmer response

Refer to the IEFxxxI message issued by the IEFPRMLB service.

ICEP505I ICEOPT: MEMBER 'ICEPRMxx' NOT FOUND OR IN ERROR - MEMBER IGNORED

Explanation

xx was specified in the iceprm parameter for the START ICEOPT command, but the IEFPRMLB service used by ICEOPT could not find member ICEPRMxx in any of the data sets that make up the PARMLIB concatenation, or an error occurred during PARMLIB service processing. The ICEPRMxx member was not processed. Any other specified ICEPRMxx members that were found were checked for errors but no installation option values were saved.

System action

DFSORT's ICEPRMxx installation area was not changed.

Programmer response

If the ICEPRMxx member already exists in PARMLIB, contact the system programmer. Otherwise, create the indicated ICEPRMxx member in PARMLIB and issue the START ICEOPT command again. Alternatively, issue a START ICEOPT command specifying suffixes for ICEPRMxx members that exist in PARMLIB.

Additionally, refer to the IEF764I, IEF105I, or IEFxxxI message issued by the IEFPRMLB service.

ICEP506I ICEOPT: TYPE (JCL, INV, TSO, TSOINV, TD1, TD2, TD3 OR TD4) MISSING BEFORE OPTIONS IN MEMBER 'ICEPRMxx'

Explanation

Member ICEPRMxx did not contain a type parameter (JCL, INV, TSO, TSOINV, TD1, TD2, TD3 or TD4) before the first option. For example, it contained:

```
RESINV=20480
```

Instead of:

```
JCL
RESINV=20480
```

The options in the ICEPRMxx member were checked for errors but no installation option values were saved.

System action

DFSORT's ICEPRMxx installation area was not changed.

Programmer response

Specify a type parameter before the first option in the indicated ICEPRMxx member and issue the START ICEOPT command again.

ICEP507I ICEOPT: KEYWORD 'keyword' IS INVALID OR UNRECOGNIZED FOR TYPE 'type' IN MEMBER 'ICEPRMxx'

Explanation

The 'keyword' after 'type' (for example, JCL) in member ICEPRMxx is not one of the valid installation option keywords for this release of DFSORT. If a keyword longer than 7 characters is specified, only the first 8 characters of the keyword are shown. The 'keyword' may be invalid for all levels of DFSORT (for example, 'EQUAL' instead of 'EQUALS') or only allowed for a later level of DFSORT than the one being used. The valid keywords for this release of DFSORT are listed in ["Installation options syntax diagram" on page 52](#). The option with the indicated keyword is ignored. Other valid options are saved.

System action

The option was ignored, but DFSORT's ICEPRMxx installation area was changed.

Programmer response

If the 'keyword' is only allowed for a later level of DFSORT, it will be ignored and no other action is required. If the 'keyword' is invalid, correct it and issue the START ICEOPT command again.

**ICEP508I ICEOPT: OPTION 'keyword' HAS
AN INVALID VALUE FOR TYPE
'type' IN MEMBER 'ICEPRMxx' -
REASON=rsn**

Explanation

The 'keyword' (for example, PAD) after 'type' (for example, JCL) in member ICEPRMxx is followed by an invalid value for that option or is followed by a comma or blank delimiter instead of an '=' or '(' delimiter. The valid values for each option are listed in "Installation options syntax diagram" on page 52. The option with the indicated keyword is ignored. Other valid options are saved. rsn is the reason code associated with the error as follows:

1. The value is not one of the valid choices (shown in uppercase letters in the syntax diagram) for the option or for the level of DFSORT being used or does not end with a blank or a comma. Examples: 'CHECK=ALL' instead of 'CHECK=YES' or 'CHECK=NO', or 'PAD=RC4,' instead of 'PAD=RC4 ' or 'PAD=RC4,'.
2. For the ENABLE option, the value TD1, TD2, TD3 or TD4 was specified more than once. Example: 'ENABLE=(TD1,TD2,TD1)' instead of 'ENABLE=(TD1,TD2,TD3)'.
3. The value is not followed by a valid delimiter. Examples: 'CHALT(YES,' instead of 'CHALT(YES)', 'SAT=(0800)' instead of 'SAT=(0800,1000)' or 'DYNALOC=SYSDA)' instead of 'DYNALOC=(SYSDA)'.
4. For a day option, a starting time value is equal to or greater than the corresponding ending time value. Examples: 'SUN=(0600,0600)' instead of 'SUN=(0600,0659)', or 'FRI=(2200,1159)' instead of 'FRI=(2200,2359)' and 'SAT=(0,1159)'.
5. The number of characters for the value is longer than the maximum characters allowed or shorter than the minimum characters allowed. Examples: 'DYNALOC=LARGEVOLS' with LARGEVOLS longer than 8 characters instead of 'DYNALOC=LARGEVOL', or 'ALTSEQ=(C1)' with C1 shorter than 4 characters instead of 'ALTSEQ=(C1F1)'.
6. The value is null. Example: 'RESALL=' instead of 'RESALL=20480', or 'CHALT=()' instead of 'CHALT=(YES)' or 'CHALT=(NO)'.
7. A numeric value was required, but the value contains a character other than '0' through '9' or does not end with a blank or a comma. Examples: 'DSA=MAX' instead of 'DSA=128', or 'SVC(ALT)' instead of 'SVC(109,ALT)' or 'SVC(,ALT)', or 'ABCODE=10;' instead of 'ABCODE=10 ' or 'ABCODE=10,'.

8. A numeric value is less than the minimum allowed. Example: 'IDRCPT=10' instead of 'IDRCPT=100'.
9. A numeric value is greater than the maximum allowed. Examples: 'IDRCPT=1200' instead of 'IDRCPT=1000', or 'Y2PAST=101' instead of 'Y2PAST=100', or 'Y2PAST=3001' instead of 'Y2PAST=3000'.
10. For the ALTSEQ option, the value contained an invalid hexadecimal digit, that is, a character other than '0' through '9' or 'A' through 'F'. Example: 'ALTSEQ=(C1G1)' instead of 'ALTSEQ=(C1F1)'.

System action

The option was ignored, but DFSORT's ICEPRMxx installation area was changed.

Programmer response

Correct the invalid value and issue the START ICEOPT command again.

**ICEP509I ICEOPT: OPTION 'ALTSEQ' HAS A
CONTINUATION ERROR FOR TYPE
'type' IN MEMBER 'ICEPRMxx' -
PROCESSING TERMINATED**

Explanation

The ALTSEQ option after 'type' (for example, JCL) in member ICEPRMxx has a continuation error in its values. A line ends with a comma, but there is no next line in the member, or the next line in the member does not contain valid hexadecimal pairs or has an invalid delimiter. Example:

```
JCL
  ALTSEQ=(C1F1,C2F2,
INV
  RESALL=20480
```

Instead of:

```
JCL
  ALTSEQ=(C1F1,C2F2,
           C3F3)
INV
  RESALL=20480
```

Processing was terminated and no installation options were saved.

System action

DFSORT's ICEPRMxx installation area was not changed.

Programmer response

Correct the continuation error and issue the START ICEOPT command again.

ICEP510I	ICEOPT: COULD NOT OBTAIN nnnnnnnnnnn BYTES OF NEEDED STORAGE IN PRIVATE AREA
-----------------	---

Explanation

The indicated number of bytes of storage was needed in the private area to run the START ICEOPT command, but could not be obtained. Processing was terminated and no installation options were saved.

System action

DFSORT's ICEPRMxx installation area was not changed.

Programmer response

Refer to the documentation for system abend 878 reason code 10.

ICEP511I

**ICEOPT: COULD NOT OBTAIN
nnnnnnnnnnn BYTES OF NEEDED
STORAGE IN CSA**

Explanation

The indicated number of bytes of storage was needed in the common storage area to run the START ICEOPT command, but could not be obtained. Processing was terminated and no installation options were saved.

System action

DFSORT's ICEPRMxx installation area was not changed.

Programmer response

Refer to the documentation for system abend 878 reason code 08.

ICEOPT return codes

When you issue a START ICEOPT command, ICEOPT returns the highest return code it sets. The meanings of the return codes set by ICEOPT are:

0

Successful completion. No errors were detected. Any previously specified installation options were cleared. Any specified installation options were saved.

4

Successful completion. Any previously specified installation options were cleared. One or more invalid or unrecognized installation options were detected and not saved. Any valid installation options were saved.

8

Unsuccessful completion. One or more errors were detected. Previously specified installation options were not cleared. Any specified installation options were not saved.

If you execute the ICEPRML program directly instead of as a started task with a START ICEOPT command, ICEPRML sets return code 12.

Appendix C. ICEMAC and message module messages

Messages are produced by the assembler program during the expansion of your ICEMAC macro instruction and during the expansion of your message modules. These messages are printed as part of the assembler listing in the SYSPRINT data set. Two types of messages are produced: error messages and informative messages.

Messages produced by the program while running are listed and explained in [z/OS DFSORT Messages, Codes and Diagnosis Guide](#).

Error messages

The general format of DFSORT installation error messages is:

```
s,***ICESMnnnn text
```

where *s* is the severity code (8 or 12), ICESM1 is the prefix, *nnn* is the message number, and *text* is the message.

ICEMAC error messages

The messages in the following pages are produced during expansion of ICEMAC. If JCL, INV, TSO, TSOINV, TD1, TD2, TD3 or TD4 is specified, analysis of the ICEMAC operands continues after an error message is issued. See [“Using the ICEMAC macro” on page 51](#) for information on what to do if you receive an error message.

12,*ICESM1 JCL, INV, TSO, TSOINV, TD1, TD2,
000 TD3 OR TD4 NOT SPECIFIED -
GENERATION TERMINATED**

Explanation

JCL, INV, TSO, TSOINV, TD1, TD2, TD3 or TD4 must be specified as the first ICEMAC parameter. The generation procedure was unsuccessful. Specify JCL, INV, TSO, TSOINV, TD1, TD2, TD3 or TD4 and restart generation.

8,*ICESM10 SIZE VALUE x INVALID
01**

Explanation

The value *x* specified for SIZE is not a valid decimal number or MAX, or is not in the valid range for SIZE.

8,*ICESM10 MAXLIM VALUE x INVALID
02**

Explanation

The value *x* specified for MAXLIM is not a valid decimal number or is not in the valid range for MAXLIM.

8,*ICESM10 MINLIM VALUE x INVALID
03**

Explanation

The value *x* specified for MINLIM is not a valid decimal number or is not in the valid range for MINLIM.

8,*ICESM10 MSGPRT VALUE x INVALID
04**

Explanation

The value *x* specified for MSGPRT is not a valid keyword.

8,*ICESM10 EQUALS VALUE x INVALID
05**

Explanation

The value x specified for EQUALS is not a valid keyword.

8,*ICESM10 LIST VALUE x INVALID**
06

Explanation

The value x specified for LIST is not a valid keyword.

8,*ICESM10 ERET VALUE x INVALID**
07

Explanation

The value x specified for ERET is not a valid keyword.

8,*ICESM10 MSGDDN VALUE x GREATER THAN**
09 8 CHARACTERS

Explanation

The value x specified for MSGDDN is too long.

8,*ICESM10 VIO VALUE x INVALID**
11

Explanation

The value x specified for VIO is not a valid keyword.

8,*ICESM10 PARMDDN VALUE x GREATER**
12 THAN 8 CHARACTERS

Explanation

The value x specified for PARMDDN is too long.

8,*ICESM10 ESTAE VALUE x INVALID**
13

Explanation

The value x specified for ESTAE is not a valid keyword.

8,*ICESM10 IGNCKPT VALUE x INVALID**
14

Explanation

The value x specified for IGNCKPT is not a valid keyword.

8,*ICESM10 WRKSEC VALUE x INVALID**
15

Explanation

The value x specified for WRKSEC is not a valid decimal number or keyword, or is not in the valid range for WRKSEC.

8,*ICESM10 VERIFY VALUE x INVALID**
16

Explanation

The value x specified for VERIFY is not a valid keyword.

8,*ICESM10 SVC VALUE x INVALID**
17

Explanation

The value x specified for the user SVC number is not decimal 109 or decimal 200-255.

8,*ICESM10 DYNALOC VALUE x GREATER THAN**
18 8 CHARACTERS

Explanation

The value x specified for DYNALOC device type is too long.

8,*ICESM10 DYNALOC VALUE x INVALID**
19

Explanation

The value x specified for DYNALOC number of devices is not a valid decimal number.

8,*ICESM10 ALTSEQ VALUE x INVALID**
20

Explanation

The value x specified for ALTSEQ is not four digits, a valid pair of hexadecimal digits (xxxx), or the keyword TABLE. The message appears once for each invalid value.

8,*ICESM10 RESALL VALUE x INVALID**
21

Explanation

The value x specified for RESALL is not a valid decimal number or is not in the valid range for RESALL.

8,*ICESM10 RESINV VALUE x INVALID**
22

Explanation

The value x specified for RESINV is not a valid decimal number.

8,*ICESM10 SMF VALUE x INVALID**
23

Explanation

The value x specified for SMF is not a valid keyword.

8,*ICESM10 CHALT VALUE x INVALID**
24

Explanation

The value x specified for CHALT is not a valid keyword.

8,*ICESM10 CHECK VALUE x INVALID**
25

Explanation

The value x specified for CHECK is not a valid keyword.

8,*ICESM10 WRKREL VALUE x INVALID**
26

Explanation

The value x specified for WRKREL is not a valid keyword.

8,*ICESM10 OUTREL VALUE x INVALID**
27

Explanation

The value x specified for OUTREL is not a valid keyword.

8,*ICESM10 MINLIM VALUE x IS GREATER**
28 **THAN SIZE VALUE**

Explanation

The value x specified for MINLIM is larger than the value specified for SIZE.

8,*ICESM10 MINLIM VALUE x IS GREATER**
29 **THAN MAXLIM VALUE**

Explanation

The value x specified for MINLIM is larger than the value specified for MAXLIM.

8,*ICESM10 OUTSEC VALUE x INVALID**
30

Explanation

The value x specified for OUTSEC is not a valid decimal number or keyword, or is not in the valid range for OUTSEC.

8,*ICESM10 OVERRGN VALUE x INVALID**
32

Explanation

The value x specified for OVERRGN is not a valid decimal number or is not in the valid range for OVERRGN.

8,*ICESM10 TMAXLIM VALUE x INVALID**
33

Explanation

The value x specified for TMAXLIM is not a valid decimal number or is not in the valid range for TMAXLIM.

8,*ICESM10 ARESALL VALUE x INVALID**
34

Explanation

The value x specified for ARESALL is not a valid decimal number or is not in the valid range for ARESALL.

8,*ICESM10 ARESINV VALUE x INVALID**
35

Explanation

The value x specified for ARESINV is not a valid decimal number or is not in the valid range for ARESINV.

8,*ICESM10 COBEXIT VALUE x INVALID**
36

Explanation

The value x specified for COBEXIT is not valid.

8,*ICESM10 TMAXLIM VALUE IS LESS THAN**
37 **MAXLIM VALUE**

Explanation

The value specified for TMAXLIM is less than the MAXLIM value.

8,*ICESM10 MSGCON VALUE x INVALID**
38

Explanation

The value x specified for MSGCON is not a valid keyword.

8,*ICESM10 VLSHRT VALUE x INVALID**
39

Explanation

The value x specified for VLSHRT is not a valid keyword.

8,*ICESM10 CINV VALUE x INVALID**
40

Explanation

The value x specified for CINV is not a valid keyword.

8,*ICESM10 EFS VALUE x GREATER THAN 8**
41 CHARACTERS

Explanation

The value x specified for the EFS program name is too long.

8,*ICESM10 LISTX VALUE x INVALID**
42

Explanation

The value x specified for LISTX is not a valid keyword.

8,*ICESM10 VSAMBSP VALUE x INVALID**
43

Explanation

The value x specified for VSAMBSP is not a valid keyword.

8,*ICESM10 DIAGSIM VALUE x INVALID**
44

Explanation

The value x specified for DIAGSIM is not a valid keyword.

8,*ICESM10 SORTLIB VALUE x INVALID**
46

Explanation

The value x specified for SORTLIB is not a valid keyword.

8,*ICESM10 DYNSPC VALUE x INVALID**
58

Explanation

The value x specified for DYNSPC is not a valid decimal number or is not in the valid range for DYNSPC.

8,*ICESM10 SVC VALUE x IS NOT ALT**
59

Explanation

The value x is specified instead of the keyword ALT.

8,*ICESM10 ALT IS INVALID WITH SVC x**
60

Explanation

The value x specified for the user SVC number is not decimal 109 and is therefore invalid with ALT.

8,*ICESM10 ZDPRINT VALUE x INVALID**
61

Explanation

The value x specified for ZDPRINT is not a valid keyword.

8,*ICESM10 IEXIT VALUE x INVALID**
62

Explanation

The value x specified for IEXIT is not a valid keyword.

8,*ICESM10 TEXTIT VALUE x INVALID**
63

Explanation

The value x specified for TEXTIT is not a valid keyword.

8,*ICESM10 GENER VALUE x GREATER THAN 8**
73 CHAR

Explanation

The value x specified for GENER is too long.

8,*ICESM10 DYNAUTO VALUE x INVALID**
77

Explanation

The value x specified for DYNAUTO is not a valid keyword.

8,*ICESM10 ABCODE VALUE x INVALID**
78

Explanation

The value x specified for ABCODE is not a valid decimal number or MSG, or is not in the valid range for ABCODE.

8,*ICESM10 EXITCK VALUE x INVALID
81**

Explanation

The value x specified for EXITCK is not a valid keyword.

8,*ICESM10 NOMSGDD VALUE x INVALID
82**

Explanation

The value x specified for NOMSGDD is not a valid keyword.

8,*ICESM10 TABLE IS INVALID WITH OTHER
83 ALTSEQ VALUES**

Explanation

The value TABLE is specified for ALTSEQ along with other parameters. There should not be any additional parameters when TABLE is specified.

8,*ICESM10 HIPRMAX VALUE x INVALID
85**

Explanation

The value x specified for HIPRMAX is not a valid decimal number or OPTIMAL, or is not in the valid range for HIPRMAX.

8,*ICESM10 FSZEST VALUE x INVALID
86**

Explanation

The value x specified for FSZEST is not a valid keyword.

8,*ICESM10 SDB VALUE x INVALID
87**

Explanation

The value x specified for SDB is not a valid keyword.

8,*ICESM10 IDRCPT VALUE x INVALID
89**

Explanation

The value x specified for IDRCPT is not a valid decimal number or NONE, or is not in the valid range for IDRCPT.

8,*ICESM10 ODMAXBF VALUE x INVALID
90**

Explanation

The value x specified for ODMAXBF is not a valid decimal number or is not in the valid range for ODMAXBF.

8,*ICESM10 CFW VALUE x INVALID
91**

Explanation

The value x specified for CFW is not a valid keyword.

8,*ICESM10 GNPAD VALUE x INVALID
92**

Explanation

The value x specified for GNPAD is not a valid keyword.

8,*ICESM10 GNTRUNC VALUE x INVALID
93**

Explanation

The value x specified for GNTRUNC is not a valid keyword.

8,*ICESM10 DSPSIZE VALUE x INVALID
94**

Explanation

The value x specified for DSPSIZE is not a valid decimal number or MAX, or is not in the valid range for DSPSIZE.

8,*ICESM10 SDBMSG VALUE x INVALID
96**

Explanation

The value x specified for SDBMSG is not a valid keyword.

8,*ICESM10 LOCALE VALUE x GREATER THAN
97 32 CHARACTERS**

Explanation

The value x specified for the LOCALE name is too long.

8,*ICESM10 DSA VALUE x INVALID**
98

Explanation

The value x specified for DSA is not a valid decimal number or is not in the valid range for DSA.

8,*ICESM10 Y2PAST VALUE x INVALID**
99

Explanation

The value x specified for Y2PAST is not a valid decimal number or is not in the valid range for Y2PAST.

8,*ICESM11 EXPMAX VALUE x INVALID**
00

Explanation

The value x specified for EXPMAX is not a valid decimal number or MAX, or is not in the valid range for EXPMAX.

8,*ICESM11 EXPOLD VALUE x INVALID**
01

Explanation

The value x specified for EXPOLD is not a valid decimal number or MAX, or is not in the valid range for EXPOLD.

8,*ICESM11 EXPRES VALUE x INVALID**
02

Explanation

The value x specified for EXPRES is not a valid decimal number or MAX, or is not in the valid range for EXPRES.

8,*ICESM11 IOMAXBF VALUE x INVALID**
03

Explanation

The value x specified for IOMAXBF is not a valid decimal number or is not in the valid range for IOMAXBF.

8,*ICESM11 ENABLE VALUE x INVALID**
04

Explanation

The value x specified for ENABLE is not a valid keyword.

8,*ICESM11 ENABLE VALUE IS NULL**
05

Explanation

A null value was specified for ENABLE instead of a valid keyword.

8,*ICESM11 ENABLE VALUE x DUPLICATED**
06

Explanation

The value x was specified for ENABLE more than once.

8,*ICESM11 day VALUE x INVALID**
07

Explanation

The value x specified for the indicated *day* parameter is not a valid hhmm value, is specified as both the start and end time, or is not a valid keyword.

8,*ICESM11 day VALUE IS NULL**
08

Explanation

A null value was specified for the indicated *day* parameter instead of a valid hhmm value or keyword.

8,*ICESM11 day CONFLICTS WITH ANOTHER**
09 **PARAMETER**

Explanation

The indicated *day* parameter was specified with another day parameter with which it is not allowed.

8,*ICESM11 day VALUES ARE UNPAIRED**
10

Explanation

A starting hhmm value was specified without an ending hhmm value for the indicated *day* parameter.

8,*ICESM11 PAD VALUE x INVALID**
11

Explanation

The value x specified for PAD is not a valid keyword.

8,*ICESM11 TRUNC VALUE x INVALID**
12

Explanation

The value x specified for TRUNC is not a valid keyword.

8,*ICESM11 OVFO VALUE x INVALID**
13

Explanation

The value x specified for OVFL0 is not a valid keyword.

8,*ICESM11 SPANINC VALUE x INVALID**
14

Explanation

The value x specified for SPANINC is not a valid keyword.

8,*ICESM11 SOLRF VALUE x INVALID**
15

Explanation

The value x specified for SOLRF is not a valid keyword.

8,*ICESM11 VLSCMP VALUE x INVALID**
17

Explanation

The value x specified for VLSCMP is not a valid keyword.

8,*ICESM11 SZERO VALUE x INVALID**
18

Explanation

The value x specified for SZERO is not a valid keyword.

8,*ICESM11 VLLONG VALUE x INVALID**
19

Explanation

The value x specified for VLLONG is not a valid keyword.

8,*ICESM11 VSAMEMT VALUE x INVALID**
20

Explanation

The value x specified for VSAMEMT is not a valid keyword.

8,*ICESM11 RESET VALUE x INVALID**
21

Explanation

The value x specified for RESET is not a valid keyword.

8,*ICESM11 VSAMIO VALUE x INVALID**
22

Explanation

The value x specified for VSAMIO is not a valid keyword.

8,*ICESM11 MOSIZE VALUE x INVALID**
23

Explanation

The value x specified for MOSIZE is not a valid keyword.

8,*ICESM11 NULLOUT VALUE x INVALID**
24

Explanation

The value x specified for NULLOUT is not a valid keyword.

8,*ICESM11 NULLOFL VALUE x INVALID**
25

Explanation

The value x specified for NULLOFL is not a valid keyword.

8,*ICESM11 MOWRK VALUE x INVALID**
26

Explanation

The value x specified for MOWRK is not a valid keyword.

8,*ICESM11 TUNE VALUE x INVALID**
27

Explanation

The value x specified for TUNE is not a valid keyword.

8,*ICESM11 DYNAPCT VALUE x INVALID**
28

Explanation

The value x specified for DYNACT is not a valid keyword or not in the valid range for DYNAPCT.

8,*ICESM11 COLLKEY VALUE x GREATER THAN**
29 64 CHARACTERS

Explanation

The value x specified for the COLLKEY name is too long.

8,*ICESM11 ZSORT VALUE *x* Invalid**
30

Explanation

The value *x* specified for ZSORT is not a valid keyword.

8,***
GENERATION

TERMINATED

Explanation

The generation procedure was unsuccessful. Correct errors and restart generation.

Message module error messages

The messages in the following pages are produced during expansion of the message text modules after you have changed the ICESMSG macro. See [“Modifying message texts” on page 39](#) for information on what to do if you receive an error message.

12,*ICESM1 TEXT FOR MESSAGE *n* IS NOT**
050 PRESENT IN ICESMSG

Explanation

The message text requested was not found in the ICESMSG macro; that is, a statement of the form:

```
ICESSET n, 'textn'
```

was not found in ICESMSG.

Programmer response

Check to make sure ICESMSG was updated correctly. RESTORE the SMP/E usermod to see the original ICESMSG macro in SICEUSER.

Locate the statement:

```
ICESSET n, 'textn'
```

in the original version of ICESMSG. Add this statement to the SMP/E usermod version of this statement, and modify it if appropriate.

If the ICESMSG macro in the SMP/E target zone has the same RMID as the ICESMSG macro in the distribution zone, you can also use the ICESMSG macro in the AICEUSER library for reference.

12,*ICESM1 MESSAGE *n* TEXT LENGTH IS *m***
051 BYTES TOO LONG

Explanation

The message text requested is *m* bytes too long in the ICESMSG macro.

Programmer response

Check to make sure ICESMSG was updated correctly. RESTORE the SMP/E usermod to see the original ICESMSG macro in SICEUSER.

Locate the statement:

```
ICESSET n, 'textn'
```

in the SMP/E usermod and shorten *textn* to 110 characters or fewer.

If the ICESMSG macro in the SMP/E target zone has the same RMID as the ICESMSG macro in the distribution zone, you can also use the ICESMSG macro in the AICEUSER library for reference.

12,*ICESM1 MESSAGE TEXTS LENGTH IS *n***
052 BYTES TOO LONG

Explanation

The total length of the message texts requested is too long as they appear in the ICESMSG macro. The message texts requested are indicated by the statements:

```
ICEDMSG a, ...
ICEDMSG b, ...
.
.
.
```

in this CSECT.

Programmer response

Check to make sure ICESMSG was updated correctly. RESTORE the SMP/E usermod to see the original ICESMSG macro in SICEUSER.

The ICEDMSG statements in the CSECT identify the messages requested in ICESMSG. Locate the corresponding statements in ICESMSG:

```
ICESSET a, 'texta'
ICESSET b, 'textb'
.
.
.
```

and shorten the total length of the text in the SMP/E usermod version of these statements by *n* bytes.

If the ICEMSGSG macro in the SMP/E target zone has the same RMID as the ICEMSGSG macro in the distribution zone, you can also use the ICEMSGSG macro in the AICEUSER library for reference.

12,*ICESM1 TEXT FOR MESSAGE *n* HAS
053 INCORRECT INSERT FIELD(S)**

Explanation

The message text requested has incorrect insert field(s) as it appears in the ICEMSGSG macro; that is, an insert field character (? ,%, or !) was deleted, added, or changed, or the relative position of the insert fields was changed.

Programmer response

Check to make sure ICEMSGSG was updated correctly. RESTORE the SMP/E usermod to see the original ICEMSGSG macro in SICEUSER.

Locate the statement:

```
ICEMSET n, 'textn'
```

in the original version of ICEMSGSG. Change the insert field(s) in the SMP/E usermod version of this statement to match the insert field(s) in the original version.

If the ICEMSGSG macro in the SMP/E target zone has the same RMID as the ICEMSGSG macro in the distribution zone, you can also use the ICEMSGSG macro in the AICEUSER library for reference.

Informational messages

ICEMAC informational messages

The following messages document the defaults generated for your DFSORT program, as specified in the ICEMAC statement. All options included in your generated program are described, whether the specification was explicit or implicit. During the ICEMAC installation, one message will be printed from each group of messages in this section to describe the default you have selected (or the IBM-supplied default).

Message 1	*, 5655-ZOS Z/OS DFSORT 3.2	Message 10	*, ONE OR MORE ICETDX MODULES ARE ENABLED
Message 2	*, ICEAM1 DEFAULTS: BATCH JCL DIRECT ENVIRONMENT MODULE	Message 11	*, ICETDX MODULES ARE NOT ENABLED
Message 3	*, ICEAM2 DEFAULTS: BATCH PROGRAM ENVIRONMENT MODULE	Message 12	*, ENABLE IGNORED FOR ICETDX MODULE
Message 4	*, ICEAM3 DEFAULTS: TSO DIRECT ENVIRONMENT MODULE	Message 13	*, <i>day</i> START AND END TIMES: <i>hhmm</i> TO <i>hhmm</i>
Message 5	*, ICEAM4 DEFAULTS: TSO PROGRAM ENVIRONMENT MODULE	Message 14	*, <i>day</i> START AND END TIMES: ALL
Message 6	*, ICETD1 DEFAULTS: FIRST TIME- OF-DAY MODULE	Message 15	*, <i>day</i> START AND END TIMES: NONE
Message 7	*, ICETD2 DEFAULTS: SECOND TIME-OF-DAY MODULE	Message 16	*, ONE OR MORE DAY OPERANDS IGNORED FOR ICEAMX MODULE
Message 8	*, ICETD3 DEFAULTS: THIRD TIME-OF-DAY MODULE	Message 17	*, EXCPVR IGNORED
Message 9	*, ICETD4 DEFAULTS: FOURTH TIME-OF-DAY MODULE	Message 18	*, VIRTDSP IGNORED
		Message 19	*, STIMER IGNORED
		Message 20	*, SIZE IS <i>x</i> BYTES

Message 21	*, SIZE IS LIMITED BY TMAXLIM/ MAXLIM	Message 43	*, NO AUTOMATIC SECONDARY ALLOCATION FOR OUTPUT DATA SETS
Message 22	*, x BYTES IS LIMIT FOR MAXLIM	Message 44	*, AUTOMATIC SECONDARY ALLOCATION FOR OUTPUT DATA SETS
Message 23	*, 1024K BYTES IS LIMIT FOR MAXLIM	Message 45	*, DO NOT PRINT MESSAGES
Message 24	*, MIN MAIN STORAGE DFSORT *,WILL TRY TO USE: x BYTES	Message 46	*, PRINT ALL MESSAGES
Message 25	*, MIN MAIN STORAGE DFSORT *,WILL TRY TO USE: 440K BYTES	Message 47	*, PRINT CRITICAL MESSAGES
Message 26	*, x BYTES RESERVED FOR SYSTEM USE	Message 48	*, ORDER OF EQUALS WILL BE PRESERVED
Message 27	*, 4096 BYTES RESERVED FOR SYSTEM USE	Message 49	*, ORDER OF EQUALS WILL NOT BE PRESERVED
Message 28	*, x BYTES RESERVED FOR PROGRAM INVOKED DFSORT	Message 50	*, ORDER OF EQUALS WILL BE PRESERVED FOR VLR BLOCKSET
Message 29	*, NO BYTES RESERVED FOR PROGRAM INVOKED DFSORT	Message 51	*, CONTROL STATEMENTS WILL NOT BE LISTED
Message 30	*, RESINV IGNORED FOR DIRECTLY INVOKED DFSORT	Message 52	*, CONTROL STATEMENTS WILL BE LISTED
Message 31	*, DYNAMIC ALLOCATION NAME IS x	Message 53	*, ABEND IF DFSORT UNSUCCESSFUL
Message 32	*, DYNAMIC ALLOCATION NAME IS SYSDA	Message 54	*, RC=16 IF DFSORT UNSUCCESSFUL
Message 33	*, DYNAMIC ALLOCATION NUMBER IS x	Message 55	*, x IS ALTERNATE MESSAGE DDNAME
Message 34	*, DYNAMIC ALLOCATION DEFAULT NUMBER IS 4	Message 56	*, SYSOUT IS ALTERNATE MESSAGE DDNAME
Message 35	*, DYNAMIC ALLOCATION NUMBER IS x, DEFAULTED TO 255 (MAXIMUM)	Message 57	*, ALLOW VIRTUAL I/O FOR WORK DATA SETS
Message 36	*, AUTOMATIC DYNAMIC ALLOCATION WILL NOT BE USED	Message 58	*, AVOID VIRTUAL I/O FOR WORK DATA SETS
Message 37	*, AUTOMATIC DYNAMIC ALLOCATION WILL BE USED	Message 59	*, UNUSED SORTWKDD SPACE WILL BE RELEASED
Message 38	*, SORTWKDD DATA SETS WILL BE DEALLOCATED	Message 60	*, UNUSED SORTWKDD SPACE WILL NOT BE RELEASED
Message 39	*, ABEND CODE WILL BE MESSAGE NUMBER	Message 61	*, UNUSED OUTPUT DATA SET SPACE WILL BE RELEASED
Message 40	*, ABEND CODE WILL BE x	Message 62	*, UNUSED OUTPUT DATA SET SPACE WILL NOT BE RELEASED
Message 41	*, NO AUTOMATIC SECONDARY ALLOCATION FOR WORK DATA SETS	Message 63	*, MEMORY OBJECTS CAN BE USED AS WORK STORAGE
Message 42	*, AUTOMATIC SECONDARY ALLOCATION FOR WORK DATA SETS	Message 64	*, MEMORY OBJECTS CANNOT BE USED AS WORK STORAGE
		Message 65	*, OUTPUT RECORDS WILL BE VERIFIED

Message 66	*, OUTPUT RECORDS WILL NOT BE VERIFIED	Message 88	*, ICEIEXIT WILL NOT BE USED
Message 67	*, ALTSEQ TABLE USED ON FORMAT AQ CONTROL FIELDS	Message 89	*, ICEIEXIT WILL BE USED
Message 68	*, ALTSEQ TABLE USED ON FORMAT AQ AND CH CONTROL FIELDS	Message 90	*, ICETEXIT WILL NOT BE USED
Message 69	*, SVC NUMBER IS x	Message 91	*, ICETEXIT WILL BE USED
Message 70	*, THE DEFAULT SVC 109 MODULE IGX00017 WILL BE USED	Message 92	*, x BYTES OVER REGION FOR BLOCKSET
Message 71	*, THE ALTERNATE SVC 109 MODULE IGX00038 WILL BE USED	Message 93	*, 64K BYTES OVER REGION FOR BLOCKSET
Message 72	*, RECORD COUNTERS NOT CHECKED WHEN E35 IS PRESENT	Message 94	*, 16K BYTES OVER REGION FOR BLOCKSET
Message 73	*, RECORD COUNTERS CHECKED WHEN E35 IS PRESENT	Message 95	*, x BYTES IS LIMIT FOR TMAXLIM
Message 74	*, NO SMF RECORDS WILL BE WRITTEN TO SMF DATA SET	Message 96	*, 6144K BYTES IS LIMIT FOR TMAXLIM
Message 75	*, FULL SMF RECORDS WILL BE WRITTEN TO SMF DATA SET	Message 97	*, x BYTES RESERVED ABOVE 16 MB VIRTUAL FOR SYSTEM USE
Message 76	*, SHORT SMF RECORDS WILL BE WRITTEN TO SMF DATA SET	Message 98	*, NO BYTES RESERVED ABOVE 16 MB VIRTUAL FOR SYSTEM USE
Message 77	*, BLOCKSET TECHNIQUE MAY BE USED IF CKPT IS SPECIFIED	Message 99	*, x BYTES RESERVED ABOVE 16 MB VIRTUAL *, FOR PROGRAM INVOKED DFSORT
Message 78	*, BLOCKSET TECHNIQUE MAY NOT BE USED IF CKPT IS SPECIFIED	Message 100	*, NO BYTES RESERVED ABOVE 16 MB VIRTUAL *, FOR PROGRAM INVOKED DFSORT
Message 79	*, COBOL EXITS USE LANGUAGE ENVIRONMENT	Message 101	*, ARESINV IGNORED FOR DIRECTLY INVOKED DFSORT
Message 80	*, COBOL EXITS DO NOT USE LANGUAGE ENVIRONMENT	Message 102	*, POSITIVE ZD SUMS WILL NOT BE PRINTABLE AS NUMBERS
Message 81	*, UNICODE PROCESSING WILL USE x AS THE ACTIVE STANDARD CHARACTER SUITE	Message 103	*, POSITIVE ZD SUMS WILL BE PRINTABLE AS NUMBERS
Message 82	*, UNICODE PROCESSING WILL USE THE CURRENT ACTIVE STANDARD CHARACTER SUITE x	Message 104	*, E15/E35 RETURN CODE CHECKING WILL BE STRONG
Message 83	*, DO NOT WRITE MESSAGES TO MASTER CONSOLE	Message 105	*, E15/E35 RETURN CODE CHECKING WILL BE WEAK
Message 84	*, WRITE ALL MESSAGES TO MASTER CONSOLE	Message 106	*, TERMINATE IF MESSAGE DATA SET MISSING
Message 85	*, WRITE CRITICAL MESSAGES TO MASTER CONSOLE	Message 107	*, CONTINUE IF MESSAGE DATA SET MISSING (MSGCON=ALL)
Message 86	*, SHORT VARIABLE RECORDS WILL BE ACCEPTED	Message 108	*, CONTINUE IF MESSAGE DATA SET MISSING (MSGCON=CRITICAL)
Message 87	*, SHORT VARIABLE RECORDS WILL NOT BE ACCEPTED	Message 109	*, CONTINUE IF MESSAGE DATA SET MISSING (MSGCON=NONE)

Message 110	*, EFS MODIFIED CONTROL STATEMENTS WILL NOT BE LISTED	Message 131	*, EXACT FILESIZE VALUES WILL BE ALLOWED
Message 111	*, EFS MODIFIED CONTROL STATEMENTS WILL BE LISTED	Message 132	*, SYSTEM-DETERMINED BLOCK SIZE FOR OUTPUT DATA SETS *, CAN BE GREATER THAN 32760 BYTES *, IF INPUT BLOCK SIZE IS GREATER THAN 32760 BYTES
Message 112	*, EFS PROGRAM x WILL BE CALLED	Message 133	*, SYSTEM-DETERMINED BLOCK SIZE FOR OUTPUT DATA SETS *, CAN BE GREATER THAN 32760 BYTES
Message 113	*, NO EFS PROGRAM WILL BE CALLED	Message 134	*, SYSTEM-DETERMINED BLOCK SIZE FOR OUTPUT DATA SETS *, CANNOT BE GREATER THAN 32760 BYTES
Message 114	*, x WILL BE USED FOR IEBGENER MODULE NAME	Message 135	*, SYSTEM-DETERMINED BLOCK SIZE WILL NOT BE USED FOR *, OUTPUT DATA SETS
Message 115	*, IEBGENR WILL BE USED FOR IEBGENER MODULE NAME	Message 136	*, ICEGENER WILL RETURN RC=0 FOR LRECL PADDING
Message 116	*, GENER VALUE IGNORED FOR DIRECTLY INVOKED DFSORT	Message 137	*, ICEGENER WILL RETURN RC=4 FOR LRECL PADDING
Message 117	*, HIPRMAX IS LIMITED TO x MEGABYTES	Message 138	*, ICEGENER WILL TRANSFER TO IEBGENER FOR LRECL PADDING
Message 118	*, HIPRMAX IS LIMITED BY OPTIMAL	Message 139	*, GNPAD VALUE IGNORED FOR DIRECTLY INVOKED DFSORT
Message 119	*, HIPRMAX IS LIMITED TO x PERCENT	Message 140	*, ICEGENER WILL RETURN RC=0 FOR LRECL TRUNCATION
Message 120	*, REFORMATTED LENGTH WILL BE USED FOR SORTOUT LRECL	Message 141	*, ICEGENER WILL RETURN RC=4 FOR LRECL TRUNCATION
Message 121	*, REFORMATTED LENGTH WILL NOT BE USED FOR SORTOUT LRECL	Message 142	*, ICEGENER WILL TRANSFER TO IEBGENER FOR LRECL TRUNCATION
Message 122	*, ESTAE RECOVERY ROUTINE WILL NOT BE DELETED EARLY	Message 143	*, GNTRUNC VALUE IGNORED FOR DIRECTLY INVOKED DFSORT
Message 123	*, ESTAE RECOVERY ROUTINE WILL BE DELETED EARLY	Message 144	*, CACHE FAST WRITE CAN BE USED
Message 124	*, x IS THE ALTERNATE PARAMETER DATA SET DDNAME	Message 145	*, CACHE FAST WRITE CANNOT BE USED
Message 125	*, DFSPARM IS THE ALTERNATE PARAMETER DATA SET DDNAME	Message 146	*, THE COMPACTION PERCENTAGE FOR IDRC TAPES IS UNKNOWN
Message 126	*, VSAM CONTROL INTERVAL ACCESS CAN BE USED	Message 147	*, A COMPACTION FACTOR OF x PERCENT WILL BE USED FOR IDRC TAPES
Message 127	*, VSAM CONTROL INTERVAL ACCESS CANNOT BE USED	Message 148	*, MAXIMUM VSAM BUFFER SPACE CAN BE USED
Message 128	*, THE CONVENTIONAL TECHNIQUE MODULES RESIDE IN A SYSTEM LIBRARY	Message 149	*, MINIMUM VSAM BUFFER SPACE WILL BE USED
Message 129	*, THE CONVENTIONAL TECHNIQUE MODULES RESIDE IN A PRIVATE LIBRARY		
Message 130	*, ALL FILESIZE VALUES WILL BE TREATED AS ESTIMATES		

Message 150	*, OPTIMAL VSAM BUFFER SPACE CAN BE USED		*, INCOMPLETE SPANNED RECORDS
Message 151	*, SORTDIAG WILL BE SIMULATED	Message 170	*, DFSORT WILL RETURN RC=0 FOR LRECL PADDING
Message 152	*, SORTDIAG WILL NOT BE SIMULATED	Message 171	*, DFSORT WILL RETURN RC=4 FOR LRECL PADDING
Message 153	*, MAXIMUM DATA SPACE SIZE WILL BE x MEGABYTES	Message 172	*, DFSORT WILL RETURN RC=16 AND TERMINATE FOR LRECL PADDING
Message 154	*, MAXIMUM DATA SPACE SIZE OBTAINABLE WILL BE USED	Message 173	*, DFSORT WILL RETURN RC=0 FOR LRECL TRUNCATION
Message 155	*, CENTRAL STORAGE WILL BE ALLOCATED AS NEEDED	Message 174	*, DFSORT WILL RETURN RC=4 FOR LRECL TRUNCATION
Message 156	*, CENTRAL STORAGE WILL BE ALLOCATED AT INITIALIZATION	Message 175	*, DFSORT WILL RETURN RC=16 AND TERMINATE FOR LRECL TRUNCATION
Message 157	*, CENTRAL STORAGE WILL BE ALLOCATED AT INITIALIZATION WHEN DYNALLOC USED	Message 176	*, DFSORT WILL RETURN RC=0 FOR SUM OVERFLOW
Message 158	*, CENTRAL STORAGE WILL BE ALLOCATED IN FIXED SIZE INCREMENTS	Message 177	*, DFSORT WILL RETURN RC=4 FOR SUM OVERFLOW
Message 159	.*SHORT VARIABLE COMPARE FIELDS WILL NOT BE PADDED	Message 178	*, DFSORT WILL RETURN RC=16 AND TERMINATE FOR SUM OVERFLOW
Message 160	.*SHORT VARIABLE COMPARE FIELDS WILL BE PADDED	Message 179	*, THE CENTURY WINDOW STARTS 80 YEARS BEFORE THE CURRENT YEAR
Message 161	*, DYNAMIC ALLOCATION DEFAULT SPACE IS x MEGABYTES	Message 180	*, THE CENTURY WINDOW STARTS x YEARS BEFORE THE CURRENT YEAR
Message 162	*, DYNAMIC ALLOCATION DEFAULT SPACE IS 256 MEGABYTES	Message 181	*, THE CENTURY WINDOW STARTS AT x
Message 163	*, SYSTEM-DETERMINED BLOCK SIZE WILL BE USED *, FOR MESSAGE AND LIST DATA SETS	Message 182	*, x BYTES IS LIMIT FOR SORTIN/SORTOUT BUFFER SPACE
Message 164	*, SYSTEM-DETERMINED BLOCK SIZE WILL NOT BE USED *, FOR MESSAGE AND LIST DATA SETS	Message 183	*, 32768K BYTES IS LIMIT FOR SORTIN/SORTOUT BUFFER SPACE
Message 165	*, ZERO WILL BE TREATED AS SIGNED	Message 184	*, x MEGABYTES IS THE LIMIT FOR DSA
Message 166	*, ZERO WILL BE TREATED AS UNSIGNED	Message 185	*, 0 MEGABYTES IS THE LIMIT FOR DSA BECAUSE *, DSA IS LESS THAN OR EQUAL TO TMAXLIM
Message 167	*, DFSORT WILL RETURN RC=0 FOR INCOMPLETE SPANNED RECORDS	Message 186	*, 64 MEGABYTES IS THE LIMIT FOR DSA
Message 168	*, DFSORT WILL RETURN RC=4 FOR INCOMPLETE SPANNED RECORDS	Message 187	*, x BYTES IS LIMIT FOR BUFFER SPACE PER OUTFIL DATA SET
Message 169	*, DFSORT WILL RETURN RC=16 AND TERMINATE FOR	Message 188	*, 2048K BYTES IS LIMIT FOR BUFFER SPACE PER OUTFIL DATA SET

Messages

Message 189	*, EXPMAX IS LIMITED TO x MEGABYTES	Message 208	*, EMPTY VSAM INPUT DATA SET WILL NOT BE ACCEPTED
Message 190	*, EXPMAX IS LIMITED BY MAX	Message 209	*, MOSIZE IS LIMITED TO x MEGABYTES
Message 191	*, EXPMAX IS LIMITED TO x PERCENT	Message 210	*, MOSIZE IS LIMITED BY MAX
Message 192	*, EXPOLD IS LIMITED TO x MEGABYTES	Message 211	*, MOSIZE IS LIMITED TO x PERCENT
Message 193	*, EXPOLD IS LIMITED BY MAX	Message 212	*, DFSORT WILL RETURN RC=0 FOR NULL SORTOUT
Message 194	*, EXPOLD IS LIMITED TO x PERCENT	Message 213	*, DFSORT WILL RETURN RC=4 FOR NULL SORTOUT
Message 195	*, EXPRES IS LIMITED TO x MEGABYTES	Message 214	*, DFSORT WILL RETURN RC=16 AND TERMINATE FOR NULL SORTOUT
Message 196	*, EXPRES IS LIMITED BY MAX	Message 215	*, DFSORT WILL RETURN RC=0 FOR NULL OUTFIL
Message 197	*, EXPRES IS LIMITED TO x PERCENT	Message 216	*, DFSORT WILL RETURN RC=4 FOR NULL OUTFIL
Message 198	*, LOCALE PROCESSING WILL USE x AS THE ACTIVE LOCALE	Message 217	*, DFSORT WILL RETURN RC=16 AND TERMINATE FOR NULL OUTFIL
Message 199	*, LOCALE PROCESSING WILL USE THE CURRENT ACTIVE LOCALE	Message 218	*, 10 PERCENT ADDITIONAL WORK DATA SETS WILL BE USED
Message 200	*, LOCALE PROCESSING WILL NOT BE USED	Message 219	*, x PERCENT ADDITIONAL WORK DATA SETS WILL BE USED
Message 201	*, LONG VARIABLE OUTPUT RECORDS WILL NOT BE ACCEPTED	Message 220	*, 1 ADDITIONAL WORK DATA SETS WILL BE USED
Message 202	*, LONG VARIABLE OUTPUT RECORDS WILL BE TRUNCATED	Message 221	*, CHANGED ALTSEQ CODE GENERATED:
Message 203	*, SAME VSAM REUSE DATA SET CANNOT BE USED FOR INPUT/OUTPUT	Message 222	*, ALTSEQ CODE GENERATED:
Message 204	*, SAME VSAM REUSE DATA SET CAN BE USED FOR INPUT/OUTPUT	Message 223	*, SUPPLIED 256-BYTE ALTSEQ TABLE MUST FOLLOW:
Message 205	*, VSAM OUTPUT REUSE DATA SET WILL BE PROCESSED AS NEW	Message 224	*, ** GENERATION SUCCESSFUL **
Message 206	*, VSAM OUTPUT REUSE DATA SET WILL BE PROCESSED AS MOD		
Message 207	*, EMPTY VSAM INPUT DATA SET WILL BE ACCEPTED		

Appendix D. SMF Type-16 record

This appendix contains Programming Interface information.

The following table shows the format of the DFSORT SMF type-16 record.

DFSORT creates an SMF record when SMF=FULL, SMF=SHORT or TEXT=YES is in effect (see [Chapter 6, “Collecting statistical data,”](#) on [page 131](#) for more details on these options). However, values for some fields are not provided due to certain circumstances such as technique used, abnormal termination, and so on. The notation column of [Table 10 on page 175](#) indicates such circumstances:

- a** This value is always provided.
- b** This value is only provided for a Blockset application.
- f** This value is only provided when a Blockset sort, copy, or merge application is successful and SMF=FULL is specified.
- n** This value is not provided when DFSORT terminates abnormally (that is, with record subtype X'03').
- s** This value is not provided in the SMF record passed to the termination exit.
- t** If DFSORT terminates abnormally, the value provided might not be identical to the value provided when termination is successful.
- v** This value is only provided when a variable-length record application is successful and SMF=FULL is specified.

Table 10. SMF Type-16 record						
Offsets		Name	Length	Format	Description	Notation
Dec.	Hex					
+0	+0	SMF16LEN	2	binary	Record length.	a
+2	+2	SMF16SEG	2	binary	Segment descriptor.	s
+4	+4	ICESIND	1	binary	System indicator	a
+5	+5	ICERTYP	1	binary	Record type.	a
+6	+6	ICEBTIME	4	binary	Time, in hundredths of a second, record was moved to SMF buffer.	s
+10	+A	ICEBDATE	4	packed	Date record was moved to SMF buffer, in the form 0CYDDDF where F is the sign.	s
+14	+E	ICESID	4	EBCDIC	System identification (taken from SID parameter).	s
+18	+12	ICEJOBNM	8	EBCDIC	Job name.	a
+26	+1A	ICERST	4	binary	Time reader recognized job card.	s
+30	+1E	ICERDS	4	packed	Date reader recognized job card.	s
+34	+22	ICEUIF	8	EBCDIC	Installation dependent information.	s
+42	+2A	ICESTN	1	binary	Step number.	s
+43	+2B	ICERES1	1		Reserved.	

Table 10. SMF Type-16 record (continued)

Offsets		Name	Length	Format	Description	Notation
Dec.	Hex					
+44	+2C	ICETRN	2	binary	Number of triplets supported by DFSORT. ⁹ A triplet is a set of values that define an SMF record section. These descriptors begin at ICEPROD.	a
+46	+2E	ICESUBID	4	EBCDIC	Subsystem identification.	a
+50	+32	ICERSUB	2	binary	Record subtype: <ul style="list-style-type: none"> • 1: Short record, successful execution. • 2: Full record, successful execution. • 3: Short record, unsuccessful execution. 	a a a
+52	+34	ICEPROD	4	binary	Offset to product section.	a
+56	+38	ICEPRODL	2	binary	Product section length.	a
+58	+3A	ICEPRODN	2	binary	Number of product sections. This field will be zero if no product section is provided.	a
+60	+3C	ICEDATA	4	binary	Offset to common data section.	a
+64	+40	ICEDATAL	2	binary	Common data section length.	a
+66	+42	ICEDATAN	2	binary	Number of data sections. This field will be zero if no data section is provided.	a
+68	+44	ICESTAT	4	binary	Offset to record-length distribution section.	a
+72	+48	ICESTATL	2	binary	Record-length distribution section length.	a
+74	+4A	ICESTATN	2	binary	Number of record-length distribution sections. This field will be zero if no record-length distribution section is provided.	a
+76	+4C	ICEINDS	4	binary	Offset to first input data set section.	a
+80	+50	ICEINDSL	2	binary	Input data set section length.	a
+82	+52	ICEINDSN	2	binary	Number of input data set sections. This field will be zero if no input data set sections are provided.	a
+84	+54	ICEOTDS	4	binary	Offset to SORTOUT data set section.	a
+88	+58	ICEOTDSL	2	binary	SORTOUT data set section length.	a
+90	+5A	ICEOTDSN	2	binary	Number of SORTOUT data set sections. This field will be zero if no SORTOUT data set section is provided.	a
+92	+5C	ICEOFDS	4	binary	Offset to first OUTFIL data set section.	a
+96	+60	ICEOFDSL	2	binary	OUTFIL data set section length.	a
+98	+62	ICEOFDSN	2	binary	Number of OUTFIL data set sections. This field will be zero if no OUTFIL data set sections are provided.	a
+100	+64	ICEHRLN	2	binary	Length of this section up to this field.	a
+102	+66	ICESPGN	2	binary	Performance group. This information is collected from OUCB field OUCBSPG.	
+104	+68	ICEUSER	8	EBCDIC	User ID for which the job or session is being executed. This field is provided only when RACF® is active.	a
+112	+70	ICEGROUP	8	EBCDIC	Group ID for which the job or session is being executed. This field is provided only when RACF is active.	a

Table 10. SMF Type-16 record (continued)						
Offsets		Name	Length	Format	Description	Notation
Dec.	Hex					
+120	+78	ICERES1D	8		Reserved.	
+128	+80	ICEZSRT	4	binary	Offset to ZSORT section.	a
+132	+84	ICEZSRTL	2	binary	Length of ZSORT statistics section.	a
+134	+86	ICEZSRTN	2	binary	Number of ZSORT statistics sections. This field will be zero if no ZSORT statistics section is provided.	a

Product section

Table 11. SMF Type-16 Record: Product section						
Offsets		Name	Length	Format	Description	Notation
Dec.	Hex					
+0	+0	ICERECV	2	EBCDIC	Record version: "01"	a
+2	+2	ICEPRDCT	8	EBCDIC	Product name: "5655ZOS "	a
+10	+A	ICERELNM	4	EBCDIC	DFSORT release level such as 3.2	DFSORT release level.
+14	+E	ICERES1C	2		Reserved.	

Data section

Table 12. SMF Type-16 Record: Data Section						
Offsets		Name	Length	Format	Description	Notation
Dec.	Hex					
+0	+0	ICERES2	2		Reserved.	
+2	+2	ICESTPNM	8	EBCDIC	Step name; blank if no step name.	a
+10	+A	ICERCDS ⁵	4	binary	Number of records sorted. ¹⁰ A 64-bit integer version of this field is defined as ICEEXRCS.	t
+14	+E	ICEBYTES ⁵	4	binary	Number of bytes sorted. ¹⁰ A 64-bit integer version of this field is defined as ICEEXBYS.	t
+18	+12	ICECPUT ⁵	4	binary	Processor time (TCB time), in hundredths of a second. ¹	a
+22	+16	ICELEN	2	binary	Specified record length.	n
+24	+18	ICEIBLK	2	binary	Maximum input block size or control interval size (15-bit). ² A 31-bit version of this field is defined as ICEIBLKF. ¹⁷	n
+26	+1A	ICEOBLK	2	binary	Maximum output block size or control interval size (15-bit). ² A 31-bit version of this field is defined as ICEOBLKF. ¹⁷	n
+28	+1C	ICEKEYLN	2	binary	Total control field length (number of bytes actually compared by DFSORT).	n
+30	+1E	ICEWBLK ⁵	4	binary	Number of work data set tracks used.	t

Table 12. SMF Type-16 Record: Data Section (continued)

Offsets		Name	Length	Format	Description	Notation
Dec.	Hex					
+34	+22	ICEFLBYT	1	binary	Bit Meaning when set 0 Reserved 1–2³ 00=Fixed-length records 01=Variable-length records 10=Variable-length spanned records 3–4 00=Blockset technique 01=Peerage technique 10=Vale technique 11=Conventional technique 5 DFSORT was invoked through a program. 6 Sorting was completed in memory (work space was not needed) 7 Reserved	n n n n t t t t a n
+35	+23	ICENDYNA	1	binary	Number of allocated work data sets.	t
+36	+24	ICEFLBY2	1	binary	Bit Meaning when set 0 A sort was specified. 1 A merge was specified. 2 A copy was specified. 3 DFSORT was called by ICETOOL. 4 DFSORT running APF authorized. 5 Memory objects used as work storage. 6 Invoked by JOINKEYS Subtask 1. 7 Invoked by JOINKEYS Subtask 2.	t t t t t t, b t t

Table 12. SMF Type-16 Record: Data Section (continued)

Offsets		Name	Length	Format	Description	Notation
Dec.	Hex					
+37	+25	ICEIOTYP	1	binary	Type of input source/output destination: Bit Meaning when set 0 E15 exit was used. 1 E32 exit was used. 2 E35 exit was used. 3 SORTIN data sets were used. 4 SORTINnn data sets were used. 5 SORTOUT data set was used. 6 OUTFIL data sets were used. 7 Reserved.	t, b t, b t, b t, b t, b t, b t, b
+38	+26	ICECSFLG	1	binary	Control statement flags byte: Bit Meaning when set 0 ALTSEQ was specified. 1 INREC was specified. 2 INCLUDE was specified. 3 OMIT was specified. 4 OUTREC was specified. 5 SUM was specified. 6 OUTFIL was specified. 7 JOINKEYS Main task was specified.	t, b t, b t, b t, b t, b t, b t, b
+39	+27	ICERES3	1		Reserved.	
+40	+28	ICETIMES	4	binary	Time DFSORT started processing (local time, in hundredths of a second).	a
+44	+2C	ICEDATES	4	packed	Date DFSORT started processing, in the form 0CYDDDF where F is the sign (local date).	a
+48	+30	ICETIMEE	4	binary	Time DFSORT terminated processing (local time, in hundredths of a second).	a
+52	+34	ICEDATEE	4	packed	Date DFSORT terminated processing, in the form 0CYDDDF where F is the sign (local date).	a

Data Section

Table 12. SMF Type-16 Record: Data Section (continued)

Offsets		Name	Length	Format	Description	Notation
Dec.	Hex					
+56	+38	ICERCBYT	1	binary	Return code status: ⁸ Bit Meaning when set 0 System abend detected. 1 User abend detected. 2–3 Reserved. 4 User requested ABEND issued by DFSORT. 5 RC=16 returned to caller. 6 RC=20 returned to caller. 7 Reserved.	a
+57	+39	ICERC	1	binary	Return code of DFSORT to its invoker. This field will contain X'FF' if a user-requested or system ABEND is issued.	a
+58	+3A	ICERESN	2	binary	Reason code. If ICERC is not 0 or 4, then this field will contain the system or user ABEND code, or the number of the DFSORT message describing the reason for an unsuccessful execution.	a
+60	+3C	ICEAVLR	4	binary	Average record length of sorted variable length records.	n
+64	+40	ICEDSA	2	binary	DSA value in effect (in megabytes).	t, b
+66	+42	ICEFLBY3	1	binary	Bit Meaning when set 0 Hipersorting was used. 1 Data space was used. 2 Work data sets were used. 3 Locale processing was used for a SORT or MERGE control field. 4 Locale processing was used for an INCLUDE or OMIT compare field. 5 EQUALS was used for a sort or merge application. 6 31-bit block size fields are available. 7 A memory object was used.	t t t t t t, b t
+67	+43	ICEWKFLG	1	binary	Work data set flag byte: Bit Meaning when set 0 Work data set tracks were allocated dynamically. 1 Cache fast write was used for one or more work data sets. 2–7 Reserved.	t

Table 12. SMF Type-16 Record: Data Section (continued)

Offsets		Name	Length	Format	Description	Notation
Dec.	Hex					
+68	+44	ICEWEXS	2	binary	Number of extents initially allocated for all work data sets.	t
+70	+46	ICEWEEXE	2	binary	Number of extents allocated for all work data sets when sorting terminated.	t
+72	+48	ICEWALLS	4	binary	Number of tracks initially allocated for work data sets.	t
+76	+4C	ICEWALLE	4	binary	Number of tracks allocated to work data sets when sorting terminated.	t
+80	+50	ICERES5	2		Reserved.	
+82	+52	ICEFLBY4	1	binary	SORTIN access method byte: ⁷ Bit Meaning when set 0 SORT or MERGE control field was specified with a Unicode data format. 1 INCLUDE or OMIT control field was specified with a Unicode data format. 2–7 Reserved.	t
+83	+53	ICEIAMB	1	binary	SORTIN access method byte: ⁷ Bit Meaning when set 0 EXCP was used for SORTIN. 1 VSAM was used for SORTIN. 2 BSAM was used for SORTIN. 3–7 Reserved.	t t t
+84	+54	ICEINIO	4	binary	Number of calls to the access method used for SORTIN. ⁶ A 64-bit integer version of this field is defined as ICEEXINN.	t
+88	+58	ICERES6	3		Reserved.	
+91	+5B	ICEOAMB	1	binary	SORTOUT access method byte: ⁷ Bit Meaning when set 0 EXCP was used for SORTOUT. 1 VSAM was used for SORTOUT. 2 BSAM was used for SOROUT. 3–7 Reserved.	t t t
+92	+5C	ICEOUTIO	4	binary	Number of calls to the access method used for SORTOUT. ⁶ A 64-bit version of this field is defined as ICEEXOUT.	t
+96	+60	ICEIBLKF	4	binary	Maximum input block size or control interval size (31–bit) ² . A 15–bit version of this field is defined as ICEIBLK. ¹⁷	t,b
+100	+64	ICEWKIO	4	binary	Number of EXCPs for all work data sets.	t
+104	+68	ICESRBTS	4	binary	Cumulative value of the SRB time for the region when DFSORT started processing, in hundredths of a second.	a

Table 12. SMF Type-16 Record: Data Section (continued)

Offsets		Name	Length	Format	Description	Notation
Dec.	Hex					
+108	+6C	ICESRBTE	4	binary	Cumulative value of the SRB time charged to the address space when DFSORT terminated processing. This field is obtained from the ASCBSRBT field in the ASCB control block. The difference between this value and the ICESRBTS value will be the SRB time charged to this DFSORT application, provided there is no other activity in the address space.	a
+112	+70	ICETCBS	2	binary	Number of TCBs defined in the region while DFSORT is processing. If the number is different at the end of processing from the number at the start of processing, this field will contain the larger number.	a
+114	+72	ICEKEYNM	2	binary	Number of Sort or Merge control fields.	a
+116	+74	ICEHSPMX	2	binary	HIPRMAX value in effect.	t, b
+118	+76	ICEDSPSZ	2	binary	DSPSIZE value in effect.	t, b
+120	+78	ICEEXRCS	8	binary	Number of records sorted (64-bit integer). ¹⁰ A 32-bit integer version of this field is defined as ICERCDS. ¹⁶	t
+128	+80	ICEEXBYS	8	binary	Number of bytes sorted (64-bit integer). ¹⁰ A 32-bit integer version of this field is defined as ICEBYTES. ¹⁶	t
+136	+88	ICEEXINN	8	binary	Number of calls to the access method used for SORTIN (64-bit integer). ⁶ A 32-bit integer version of this field is defined as ICEINIO. ¹⁶	t
+144	+90	ICEEXOUT	8	binary	Number of calls to the access method used for SORTOUT (64-bit integer). ⁶ A 32-bit integer version of this field is defined as ICEOUTIO. ¹⁶	t
+152	+98	ICEHSPN	2	binary	Number of Hiperspaces created.	t
+154	+9A	ICEHSPU	4	binary	Number of Hiperspace pages used.	t
+158	+9E	ICEDSPN	2	binary	Number of data spaces created.	t
+160	+A0	ICEDSPU	4	binary	Number of data space pages used.	t
+164	+A4	ICEPRCNM	8	EBCDIC	Proc step name; blank if no proc step name.	a
+172	+AC	ICEIDSNM	44	EBCDIC	SORTIN data set name. ¹⁴	t
+216	+D8	ICEIVOLS	6	EBCDIC	SORTIN volume serial. ¹⁴	t
+222	+DE	ICEODSNM	44	EBCDIC	SORTOUT data set name.	t
+266	+10A	ICEOVOLS	6	EBCDIC	SORTOUT volume serial. ¹⁵	t
+272	+110	ICEINPDS	2	binary	Number of SORTIN data sets, including concatenated data sets.	t, b
+274	+112	ICEINNDS	2	binary	Number of SORTINnn data sets.	t, b
+276	+114	ICEOUTDS	2	binary	Number of SORTOUT data sets.	t, b
+278	+116	ICEOFLDS	2	binary	Number of OUTFIL data sets.	t, b
+280	+118	ICERCINP	8	binary	Number of input records (64-bit integer). ¹¹	t, b
+288	+120	ICERCOUT	8	binary	Number of output records (64-bit integer). ¹¹	t, b
+296	+128	ICERCINS	8	binary	Number of inserted records (64-bit integer). ¹²	t, b
+304	+130	ICERCDEL	8	binary	Number of deleted records (64-bit integer). ¹²	t, b
+312	+138	ICEMD15N	8	EBCDIC	Routine name specified for MODS E15 exit.	t, b
+320	+140	ICEMD15M	4	binary	Storage specified for MODS E15 exit.	t, b
+324	+144	ICEMD15S	8	EBCDIC	Library ddname specified for MODS E15 exit.	t, b

Table 12. SMF Type-16 Record: Data Section (continued)

Offsets		Name	Length	Format	Description	Notation
Dec.	Hex					
+332	+14C	ICEMD15E	1	EBCDIC	Requirements specified for MODS E15 exit.	t, b
+333	+14D	ICERES10	3		Reserved.	
+336	+150	ICEMD35N	8	EBCDIC	Routine name specified for MODS E35 exit.	t, b
+344	+158	ICEMD35M	4	binary	Storage specified for MODS E35 exit.	t, b
+348	+15C	ICEMD35S	8	EBCDIC	Library ddname specified for MODS E35 exit.	t, b
+356	+164	ICEMD35E	1	EBCDIC	Requirements specified for MODS E35 exit.	t, b
+357	+165	ICERES11	3		Reserved.	
+360	+168	ICELCALE	32	EBCDIC	Active locale name or NONE.	t, b
+392	+188	ICEESMAX	4	binary	EXPMAX value in effect.	t, b
+396	+18C	ICEESOLD	4	binary	EXPOLD value in effect.	t, b
+400	+190	ICEESRES	4	binary	EXPRES value in effect.	t, b
+404	+194	ICEOBLKF	4	binary	Maximum output block size of control interval size (31–bit). ² A 15–bit version of this field is defined as ICEOBLK. ¹⁷	t, b
+408	+198	ICEFILSZ	8	binary	Value specified for FILSZ/SIZE (64-bit integer).	t, b
+416	+1A0	ICEAVGRL	4	binary	Value specified for AVGRLEN.	t, b
+420	+1A4	ICEFSZFL	1	binary	FILSZ/SIZE flags byte: Bit Meaning when set 0 FILSZ/SIZE value specified as n. 1 FILSZ/SIZE value specified as En. 2 FILSZ/SIZE value specified as Un. 3 FILSZ/SIZE value not specified and file size could not be determined. 4–7 Reserved.	t, b t, b t, b t, b
+421	+1A5	ICERES2A	2		Reserved.	
+423	+1A7	ICERES2B	1		Reserved.	
+424	+1A8	ICERES2C	8		Reserved.	
+432	+1B0	ICEMOSIZ	4	binary	MOSIZE value in effect.	t, b
+436	+1B4	ICERES12	2		Reserved.	
+438	+1B6	ICEMON	2	binary	Number of memory objects allocated.	t, b
+440	+1B8	ICEMOUSE	8	binary	Number of megabytes used for memory objects (64-bit integer).	t, b
+448	+1C0	ICEMNVLX	4	binary	The main storage value specified in kilobytes (subject to the SIZE/MAINSIZE limit), or supplied by default	a
+452	+1C4	ICEMNVLY	4	binary	The main storage theoretically available to DFSORT in kilobytes, considering the MINLIM value specified when the program was installed.	a

Table 12. SMF Type-16 Record: Data Section (continued)

Offsets		Name	Length	Format	Description	Notation
Dec.	Hex					
+456	+1C8	ICEMNV LZ	4	binary	The main storage actually available to DFSORT in kilobytes, after any other program took what it needed from the region (invoking program or exit routines).	a
+460	+1CC	ICEMN FLG	1	binary	Bit Meaning when set 0 SIZE=MAX in effect 1-7 Reserved	a
+461	+1CD	ICERES2D	1		Reserved.	
+462	+1CE	ICEINMRG	2	binary	Number of intermediate merges performed for a Sort application. This field is zero if no intermediate merges were required to complete the Sort application. This field is always zero for a Copy or Merge application.	t
+464	+1D0	ICECPUZE	4	binary	Subset of the processor time reported in field ICECPUT that was eligible to be dispatched on a zIIP, in hundredths of a second.	t, b
+468	+1D4	ICECPUZP	4	binary	Subset of the processor time reported in field ICECPUZE that was actually dispatched on a zIIP, in hundredths of a second.	t, b
+472	+1D8	ICETUNE	1	EBCDIC	TUNE value in effect CHAR Meaning O OLD S STOR D DISK J DDYN	b
+473	+1D9	ICERES3D	3	binary	Reserved.	
+476	+1DC	ICEDYINC	4	binary	Initial increment dynamically calculated for Hiperspace or Memory Object work files	t, b
+480	+1E0	ICEDYMAX	4	binary	Final EXPMAX value dynamically calculated	t, b
+484	+1E4	ICEDYOLD	4	binary	Final EXPOLD value dynamically calculated	t, b
+488	+1E8	ICEDYRES	4	binary	Final EXPRES value dynamically calculated	t, b
+492	+1EC	ICERES4D	4	binary	Reserved.	
+496	+1F0	ICECOLLK	64	EBCDIC	Active COLLKEY Value.	t, b
+560	+230	ICERES5D	8	EBCDIC	Reserved.	
+568	+238	ICETCBT	4	binary	Total TCB time, in hundredths of a second.	f
+572	+23C	ICEFLBY5	1	binary	Flag byte 5 Bit Meaning when set 0 ZSORT is used 1-7 Reserved.	f
+573	+23D	ICEZSRNU	1	binary	ZSORT Non-usage code. ¹⁸	f

Table 12. SMF Type-16 Record: Data Section (continued)						
Offsets		Name	Length	Format	Description	Notation
Dec.	Hex					
+574	+23E	ICERES6D	2		Reserved.	

ZSORT statistics section

Table 13. SMF Type-16 record: ZSORT statistics section						
Offsets		Name	Length	Format	Description	Notation
Dec.	Hex					
+0	+0	ICEZSFLG	1	binary	ZSORT flags Bit Meaning when set 0 ZSORT statistics available and valid. 1–7 Reserved.	f
+1	+1	ICEZSRVS	3		Reserved	
+4	+4	ICEZSPH1	4	binary	ZSORT Phase 1 elapsed time, in hundredths of a second.	f
+8	+8	ICEZSPH3	4	binary	ZSORT Phase 3 elapsed time, in hundredths of a second.	f
+12	+C	ICEZSTC1	4	binary	ZSORT Phase 1 TCB time, in hundredths of a second.	f
+16	+10	ICEZSTC3	4	binary	ZSORT Phase 3 TCB time, in hundredths of a second.	f
+20	+14	ICEZSRS1	24		Reserved.	
+44	+2C	ICEZSDIA	400	binary	ZSORT diagnostic information area.	f
+444	+1BC	ICEZSDIV	1	EBCDIC	ZSORT diagnostic version.	f
+445	+1BD	ICEZSRS2	3		Reserved.	f

Record-length distribution section

Table 14. SMF Type-16 record: Record-length distribution section						
Offsets		Name	Length	Format	Description	Notation
Dec.	Hex					
+0	+0	ICECTR01	4	binary	Records with length 5-15.	v
+4	+4	ICECTR02	4	binary	Records with length 16-31.	v
+8	+8	ICECTR03	4	binary	Records with length 32-63.	v
+12	+C	ICECTR04	4	binary	Records with length 64-127.	v
+16	+10	ICECTR05	4	binary	Records with lengths 128-191.	v
+20	+14	ICECTR06	4	binary	Records with lengths 192-255.	v
+24	+18	ICECTR07	4	binary	Records with lengths 256-511.	v
+28	+1C	ICECTR08	4	binary	Records with lengths 512-1023.	v
+32	+20	ICECTR09	4	binary	Records with lengths 1024-2047.	v
+36	+24	ICECTR10	4	binary	Records with lengths 2048-4095.	v
+40	+28	ICECTR11	4	binary	Records with lengths 4096-7167.	v
+44	+2C	ICECTR12	4	binary	Records with lengths 7168-10751.	v
+48	+30	ICECTR13	4	binary	Records with lengths 10752-15359.	v

Table 14. SMF Type-16 record: Record-length distribution section (continued)						
Offsets		Name	Length	Format	Description	Notation
Dec.	Hex					
+52	+34	ICECTR14	4	binary	Records with lengths 15360-20991.	v
+56	+38	ICECTR15	4	binary	Records with lengths 20992-26623.	v
+60	+3C	ICECTR16	4	binary	Records with lengths 26624-32756.	v

Input data set section

One section is provided for each of up to 16 SORTIN data sets (including concatenations) or for each of up to 16 SORTINnn data sets.

Table 15. SMF Type-16 record: input data set section						
Offsets		Name	Length	Format	Description	Notation
Dec.	Hex					
+0	+0	ICEINFL1	1	binary	Bit Meaning when set 0 SORTIN data set. 1 SORTINnn data set. 2-7 Reserved.	f
+1	+1	ICEINFL2	1	binary	Bit Meaning when set 0 Pipe data set. 1 Striped data set. 2 Compressed data set. 3 VSAM extended addressability data set. 4 HFS file. 5-7 Reserved.	f
+2	+2	ICEINAMB	1	binary	Access method flags byte: Bit Meaning when set 0 EXCP was used for data set. 1 VSAM was used for data set. 2 BSAM was used for data set. 3-7 Reserved.	f

Table 15. SMF Type-16 record: input data set section (continued)

Offsets		Name	Length	Format	Description	Notation
Dec.	Hex					
+3	+3	ICEINTYP	1	binary	Data set type flags byte: Bit Meaning when set 0 Tape data set. 1 Disk data set. 2 Spool, dummy or pipe data set, or HFS file. 3-7 Reserved.	f
+4	+4	ICEINRCF	1	binary	Record format (flags are identical to those in JFCRECFM).	f
+5	+5	ICEINRS1	3		Reserved.	f
+8	+8	ICEINBYT	8	binary	Number of bytes read from data set (64-bit integer).	f
+16	+10	ICEINAMC	8	binary	Number of calls to access method for data set (64-bit integer). ⁶	f
+24	+18	ICEINRS2	2		Reserved.	f
+26	+1A	ICEINLRL	2	binary	Logical record length (LRECL).	f
+28	+1C	ICEINRS3	2		Reserved.	f
+30	+1E	ICEINBKZ	2	binary	Block size or control interval size (15-bit). A 31-bit version of this field is defined as ICEINBKF. ¹⁷	f
+32	+20	ICEINDDN	8	EBCDIC	Data set ddname.	f
+40	+28	ICEINNAM	44	EBCDIC	Data set name.	f
+84	+54	ICEINVOL	6	EBCDIC	First volume serial.	f
+90	+5A	ICEINRS4	2		Reserved.	f
+92	+5C	ICEINBKF	4	binary	Block size or control interval size (31-bit). A 15-bit version of this field is defined as ICEINBKZ. ¹⁷	f

SORTOUT data set section

Table 16. SMF Type-16 record: SORTOUT data set section

Offsets		Name	Length	Format	Description	Notation
Dec.	Hex					
+0	+0	ICEOTFL1	1	binary	Bit Meaning when set 0 Pipe data set. 1 Striped data set. 2 Compressed data set. 3 VSAM extended addressability data set. 4 HFS file. 5-7 Reserved.	f
+1	+1	ICEOTAMB	1	binary	Access method flags byte: Bit Meaning when set 0 EXCP was used for data set. 1 VSAM was used for data set. 2 BSAM was used for data set. 3-7 Reserved.	f
+2	+2	ICEOTYP	1	binary	Data set type flags byte: Bit Meaning when set 0 Tape data set. 1 Disk data set. 2 Spool, dummy or pipe data set, or HFS file. 3-7 Reserved.	f
+3	+3	ICEOTRCF	1	binary	Record format (flags are identical to those in JFCRECFM).	f
+4	+4	ICEOTRS1	4		Reserved.	f
+8	+8	ICEOTBYT	8	binary	Number of bytes written to data set (64-bit integer).	f
+16	+10	ICEOTREC	8	binary	Number of records written to data set (64-bit integer).	f
+24	+18	ICEOTAMC	8	binary	Number of calls to access method for data set (64-bit integer). ⁶	f
+32	+20	ICEOTRS2	2		Reserved.	f
+34	+22	ICEOTLRL	2	binary	Logical record length (LRECL).	f

Table 16. SMF Type-16 record: SORTOUT data set section (continued)

Offsets		Name	Length	Format	Description	Notation
Dec.	Hex					
+36	+24	ICEOTRS3	2		Reserved.	f
+38	+26	ICEOTBKZ	2	binary	Block size or control interval size (15-bit). A 31-bit version of this field is defined as ICEOTBKF. ¹⁷	f
+40	+28	ICEOTDDN	8	EBCDIC	Data set ddname.	f
+48	+30	ICEOTNAM	44	EBCDIC	Data set name.	f
+92	+5C	ICEOTVOL	6	EBCDIC	First volume serial.	f
+98	+62	ICEOTRS4	2		Reserved.	f
+100	+64	ICEOTBKF	4	binary	Block size or control interval size (31-bit). A 15-bit version of this field is defined as ICEOTBKZ. ¹⁷	f

OUTFIL data set section

One section is provided for each of up to 16 OUTFIL data sets.

Table 17. SMF Type-16 record: OUTFIL data set section

Offsets		Name	Length	Format	Description	Notation
Dec.	Hex					
+0	+0	ICEOFFL1	1	binary	Bit Meaning when set 0 Pipe data set. 1 Striped data set. 2 Compressed data set. 3 VSAM extended addressability data set. 4 HFS file. 5-7 Reserved.	f
+1	+1	ICEOFAMB	1	binary	Access method flags byte: Bit Meaning when set 0 EXCP was used for data set. 1 VSAM was used for data set. 2 BSAM was used for data set. 3-7 Reserved.	f

Table 17. SMF Type-16 record: OUTFIL data set section (continued)

Offsets		Name	Length	Format	Description	Notation
Dec.	Hex					
+2	+2	ICEOFTYP	1	binary	Data set type flags byte: Bit Meaning when set 0 Tape data set. 1 Disk data set. 2 Spool, dummy or pipe data set, or HFS file. 3-7 Reserved.	f
+3	+3	ICEOFRCF	1	binary	Record format (flags are identical to those in JFCRECFM).	f
+4	+4	ICEOFPRM	1	binary	OUTFIL parameter flags byte: Bit Meaning when set 0 OUTFIL STARTREC or ENDREC parameter specified. 1 OUTFIL INCLUDE, OMIT, or SAVE parameter specified. 2 OUTFIL SPLIT parameter specified. 3 OUTFIL OUTREC parameter specified. 4 OUTFIL, VTOF, CONVERT, or FTOV parameter specified 5 OUTFIL report parameter specified. 6 OUTFIL VLFILL parameter specified. 7 OUTFIL VLTRIM parameter specified.	f
+5	+5	ICEOFPR2	1	binary	OUTFIL parameter flags byte: Bit Meaning when set 0 OUTFIL REMOVECC parameter specified. 1-7 Reserved.	f
+6	+6	ICEOFRS1	2		Reserved.	f
+8	+8	ICEOFBYT	8	binary	Number of bytes written to data set (64-bit integer).	f
+16	+10	ICEOFREC	8	binary	Number of records written to data set (64-bit integer).	f
+24	+18	ICEOFAMC	8	binary	Number of calls to access method for data set (64-bit integer). ⁶	f
+32	+20	ICEOFRS2	2		Reserved.	f

Table 17. SMF Type-16 record: OUTFIL data set section (continued)

Offsets		Name	Length	Format	Description	Notation
Dec.	Hex					
+34	+22	ICEOFLRL	2	binary	Logical record length (LRECL).	f
+36	+24	ICEOFRS3	2		Reserved.	f
+38	+26	ICEOFBKZ	2	binary	Block size or control interval size (15-bit). A 31-bit version of this field is defined as ICEOFBKF. ¹⁷	f
+40	+28	ICEOFDDN	8	EBCDIC	Data set ddname.	f
+48	+30	ICEOFNAM	44	EBCDIC	Data set name.	f
+92	+5C	ICEOFVOL	6	EBCDIC	First volume serial.	f
+98	+62	ICEOFRS4	2		Reserved.	f
+100	+64	ICEOFBKF	4	binary	Block size or control interval size (31-bit). A 15-bit version of this field is defined as ICEOFBKZ. ¹⁷	f

Footnotes

Footnotes:

- ¹ ICECPUT is set to zero if the TIMEUSED service used to obtain CPU time fails.
- ² ICEIBLK and ICEIBLKf are set to zero if the input data set is not present. ICEOBLK and ICEOBLKF are set to zero if the output data set is not present.
- ³ If ICEFLBYT bits 1-2 are binary zeros (indicating fixed length record), the short form of the SMF record is produced, even if the user has specified the full SMF record. In addition, ICERSUB is initialized to X'0001' to indicate that a short record will be written.
- ⁴ If all bits of ICEFLBY2 are zero, then the function to be performed could not be determined.
- ⁵ Since ICERCDS, ICEBYTES, ICECPUT, and ICEWBLK are not each on full word boundaries within the DSECT for the data section, alignment errors occur when assembling programs referencing these fields as full word fields. See "Mapping the SMF record" on page 133 for additional information.
- ⁶ The number of calls to the access method used for a particular data set will be the total count of EXCPs, or READ/WRITES (BSAM), or GET/PUTs (VSAM).
- ⁷ If all the bits of ICEIAMB or ICEOAMB are zero and DFSORT terminated abnormally, then the type of access method used for SORTIN and SORTOUT could not be determined.
- ⁸ This field is zero if termination is normal.
- ⁹ Depending on the application, this number may not be representative of the number of SMF record sections that DFSORT actually provides in its SMF record.
- ¹⁰ Refer to message ICE134I for a description of the meaning of the number of records/bytes sorted. The Blockset technique provides a 64-bit value; the Peerage and Vale techniques provide a 31-bit value. You should reference the 64-bit fields containing these values. For Blockset, the smaller fields contain only the least significant 32 bits.
- ¹¹ Refer to message ICE054I for a description of the meaning of the number of input/output records.
- ¹² Refer to message ICE055I for a description of the meaning of the number of inserted/deleted records.
- ¹³ The following fields are provided for the Conventional technique SMF record:
 - All of the fields in the header section except ICETRN, ICEUSER and ICEGROUP.
 - All of the fields in the Product section.
 - ICESTPNM and the technique and program invoked flags of ICEFLBYT in the Data section.
- ¹⁴ This information is provided for the first SORTIN or SORTINnn data set only. This field will contain binary zeroes if no SORTIN or SORTINnn data set is present.
- ¹⁵ This information is provided for the first SORTOUT volume serial only. This field will contain binary zeroes if no SORTOUT data set is present.
- ¹⁶ The Blockset technique provides a 64-bit value; the Peerage and Vale techniques provide a value in the lower 31 bits only.

¹⁷ The Blockset technique provides a 31-bit value; the Peerage and Vale techniques provide a 15-bit value only. For Blockset, the smaller fields contain only the least significant 16 bits.

¹⁸ See message ICE267I (in *z/OS DFSORT Messages, Codes and Diagnosis Guide*) which indicates the reason codes that explain why the IBM Integrated Accelerator for Z Sort was not used.

Appendix E. Termination exit (ICETEXIT) parameters

This appendix contains Programming Interface information.

The following diagrams illustrate the format of the information passed to the termination exit (ICETEXIT). See Chapter 6, “Collecting statistical data,” on page 131 for a description of the parameters.

Notes to all the tables are listed after “Sorting with data space statistics” on page 216.

SMF statistics

Table 18. DFSORT SMF statistics				
Offsets		Name	Length	Description
Dec.	Hex			
+0	+0	ICESMFST	4	Length of the DFSORT SMF statistics, including this field.
+4	+4	ICESMFRC	2	SMF return code. This value is the return code from the SMFWTM macro. ¹⁹
+6	+6	ICESMFS1	1	SMF status information: Bit Meaning 0 An attempt was made to write an SMF record. 1 The DFSORT SMF record was successfully transferred to an SMF buffer. 2 SMF is not active in the system. 3 SMF=SHORT is in effect. 4 SMF=FULL is in effect. 5 SMF=NO is in effect. 6 The TIMEUSED service was used 7 Reserved.
+7	+7	ICESMFR1	1	Reserved.

General statistics

Table 19. DFSORT general statistics				
Offsets		Name	Length	Description
Dec.	Hex			
+0	+0	ICEGSTAT	4	Length of the DFSORT general statistics, including this field.
+4	+4	ICEGRES1	4	Reserved.

Table 19. DFSORT general statistics (continued)

Offsets		Name	Length	Description
Dec.	Hex			
+8	+8	ICEGPDNM	8	Product name of control program. This information is copied from CVT field CVTPRODN (EBCDIC). This field contains a release indicator (not the actual version and release numbers).
+16	+10	ICEGPDID	8	Product FMID identifier for the control program. This information is collected from CVT field CVTPRODI (EBCDIC—such as "JBB2125").
+24	+18	ICEGOPSV	16	Optional user personalization of Software System Version. This information is collected from CVT field CVTVERID.
+40	+28	ICEGSYRL	4	System release. This information is copied from CVT field CVTRELNO (EBCDIC). Its value should always be "038".
+44	+2C	ICEGDFRL	4	DFSORT release/level (EBCDIC) (such as "2.04").
+48	+30	ICEGCPUM	2	CPU Model (signless packed decimal) (such as 3081).
+50	+32	ICEGRES2	2	Reserved.
+52	+34	ICEGMOD	8	Identification of the module that invoked DFSORT (when available). ³⁸ This field will contain binary zeroes when DFSORT is not invoked through a program.
+60	+3C	ICEGRVTR	2	Reason that DFSORT reverted to the Peerage/Vale technique. This code is the same as that which appears with the ICE800I message. This field is zero when the Blockset technique is used.
+62	+3E	ICEGBLKC	2	Blockset critical message number which caused a revert to the Peerage/Vale technique. This information is provided only when DFSORT reverts to Peerage/Vale with a reason code of 1.
+64	+40	ICEGGIB1	1	DFSORT general information, byte 1: Bit Meaning 0 DFSORT modules for the technique in use are resident. 1 DFSORT was loaded above 16 MB virtual. ²⁸ 2 The DFSORT SVC is installed. 3-7 Reserved.
+65	+41	ICEGSVCU	1	DFSORT SVC number (binary). The number is one of the following decimal values: 17 = SVC 109 MODULE IGX00017 38 = SVC 109 MODULE IGX00038 N = SVC N (N = 200 - 255)
+66	+42	ICEGRES3	6	Reserved.

Options statistics

Table 20. DFSORT option statistics				
Offsets		Name	Length	Description
Dec.	Hex			
+0	+0	ICEOSTAT	4	Length of the DFSORT options statistics, including this field.
+4	+4	ICEORES1	4	Reserved.
+8	+8	ICEOCSE	1	Control statements specified: Bit Meaning 0 ALTSEQ was specified. 1 INREC was specified. 2 INCLUDE was specified. 3 OMIT was specified. 4 OUTREC was specified. 5 SUM was specified. 6 UTFIL was specified. 7 Reserved.
+9	+9	ICEORES2	1	Reserved.
+10	+A	ICEOUE1	1	User exits specified through MODS statement, byte 1: Bit Meaning 0 E11 specified. 1 E15 specified. 2 E17 specified. 3 E18 specified. 4-7 Reserved.

Table 20. DFSORT option statistics (continued)

Offsets		Name	Length	Description
Dec.	Hex			
+11	+B	ICEOUE2	1	User exits specified through MODS statement, byte 2: Bit Meaning 0 E31 specified. 1 E35 specified. 2 E37 specified. 3 E38 specified. 4 E39 specified. 5 E61 specified. 6-7 Reserved.
+12	+C	ICEORES3	5	Reserved.
+17	+11	ICEOUEP1	1	User exits specified in 24- or 31-bit parameter list: Bit Meaning 0 E15 specified. 1 E18 specified. 2 E32 specified. 3 E35 specified. 4 E39 specified. 5-7 Reserved.
+18	+12	ICEORES4	6	Reserved.
+24	+18	ICEOE1	1	Options in effect, byte 1: Bit Meaning 0 EQUALS is in effect. 1 VERIFY is in effect. 2 CHALT is in effect. 3-7 Reserved.

Table 20. DFSORT option statistics (continued)

Offsets		Name	Length	Description
Dec.	Hex			
+25	+19	ICEOE2	1	Options in effect, byte 2: Bit Meaning 0 CHECK is in effect. 1 LIST is in effect. 2-4 Not used. 5-7 Reserved.
+26	+1A	ICEOE3	1	Options in effect, byte 3: Bit Meaning 0 NOBLKSET is in effect. 1 SKIPREC is in effect. 2 DYNALLOC is in effect. 3 NOOUTREL is in effect. 4 NOWRKSEC is in effect. 5 NOWRKREL is in effect. 6 NOOUTSEC is in effect. 7 VLSHRT is in effect.
+27	+1B	ICEOE4	1	Options in effect, byte 4: Bit Meaning 0 STOPAFT is in effect. 1-7 Reserved.

Table 20. DFSORT option statistics (continued)

Offsets		Name	Length	Description
Dec.	Hex			
+28	+1C	ICEOE5	1	Options in effect, byte 5: Bit Meaning 0 COBEXIT=COB2 is in effect. 1 SORTDD is in effect. 2 SORTIN is in effect. 3 SORTOUT is in effect. 4 MSGDDN is in effect. 5-7 Reserved.
+29	+1D	ICEOE6	1	Options in effect, byte 6: Bit Meaning 0 MSGPRT=ALL or equivalent is in effect. 1 MSGPRT=NONE or equivalent is in effect. 2 MSGPRT=CRITICAL or equivalent is in effect. 3 MSGCON=ALL or equivalent is in effect. 4 MSGCON=NONE or equivalent is in effect. 5 MSGCON=CRITICAL or equivalent is in effect. 6 CHKPT is in effect. 7 VIO is in effect.
+30	+1E	ICEORES5	1	Reserved.
+31	+1F	ICEOFLSB	1	FILSZ/SIZE specified: ³² Bit Meaning 0 FILSZ/SIZE is an estimate. 1 FILSZ/SIZE is an exact amount. 2-7 Reserved.
+32	+20	ICEOFLSZ	8	FILSZ/SIZE specified. ³²

Table 20. DFSORT option statistics (continued)

Offsets		Name	Length	Description
Dec.	Hex			
+40	+28	ICEODEB	1	Debug statement parameters specified: Bit Meaning 0 Reserved. 1 NOASSIST was specified. 2-7 Reserved.
+41	+29	ICEORES7	1	Reserved.
+42	+2A	ICEORES8	4	Reserved.
+46	+2E	ICEOEFS	1	EFS options in effect: Bit Meaning 0 EFS is in effect. 1 LISTX is in effect. 2-7 Reserved.
+47	+2F	ICEORES9	1	Reserved.

Sort/merge statistics

Table 21. DFSORT sort/merge statistics

Offsets		Name	Length	Description
Dec.	Hex			
+0	+0	ICEFSTAT	4	Length of the DFSORT sort or merge field statistics, including this field.
+4	+4	ICEFRES1	4	Reserved.
+8	+8	ICESKEYS	2	Number of SORT or MERGE keys.
+10	+A	ICESINMG	1	Number of intermediate merges. This value is supplied for a Blockset sort only.
+11	+B	ICEFRES2	1	Reserved.
+12	+C	ICEMFILS	4	Specified number of input files to merge when input is specified through the E32 user exit (FILES). This value is provided only when an E32 user exit is supplied.
+16	+10	ICEFRES3	2	Reserved.

Table 21. DFSORT sort/merge statistics (continued)

Offsets		Name	Length	Description
Dec.	Hex			
+18	+12	ICEFSRIF	1	SORT information: Bit Meaning when set 0 Alternate Blockset path was used. 1 Phase 3 sorting instructions were used. 2-7 Reserved.
+19	+13	ICEFFLDI	1	SORT/MERGE field information: Bit Meaning when set 0 One or more descending fields were specified. 1-7 Reserved.
+20	+14	ICEFLFM1	1	Field formats used, byte 1: Bit Meaning when set 0 CH format was used. 1 BI format was used. 2 FI format was used. 3 FL format was used. 4 PD or PDO format was used. 5-7 Reserved.

Table 21. DFSORT sort/merge statistics (continued)

Offsets		Name	Length	Description
Dec.	Hex			
+21	+15	ICEFLFM2	1	Field formats used, byte 2: Bit Meaning when set 0 ZD format was used. 1 AC format was used. 2 AQ format was used. 3 D1 format was used. 4 Y2x format was used. 5 AUF format was used. 6 ASF format was used. 7 Reserved.
+22	+16	ICEFLFM3	1	Field formats used, byte 3: Bit Meaning when set 0 CSL or LS format was used. 1 CLO or OL format was used. 2 CST or TS format was used. 3 CTO or OT format was used. 4 ASL format was used. 5 AST format was used. 6 CSF or FS format was used. 7 Reserved.
+23	+17	ICEFRES4	1	Reserved.

Virtual storage statistics

Table 22. DFSORT virtual storage statistics

Offsets		Name	Length	Description
Dec.	Hex			
+0	+0	ICEVSTAT	4	Length of the virtual storage statistics, including this field.

Table 22. DFSORT virtual storage statistics (continued)				
Offsets		Name	Length	Description
Dec.	Hex			
+4	+4	ICEVARSA	4	ARESALL value in effect. ²¹
+8	+8	ICEVARIV	4	ARESINV value in effect. ²¹
+12	+C	ICEVOVER	4	OVERRGN value in effect. ²¹
+16	+10	ICEVTMAX	4	TMAXLIM value in effect. ²¹
+20	+14	ICEVRSAL	4	RESALL value in effect.
+24	+18	ICEVRSIV	4	RESINV value in effect.
+28	+1C	ICEVMINL	4	MINLIM value in effect.
+32	+20	ICEVMAXL	4	MAXLIM value in effect.
+36	+24	ICEVRES1	4	Reserved.
+40	+28	ICEVREGN	4	REGION size specified. This value is rounded up to the next page boundary. It is taken from the LDAREGRQ field of the LDA.
+44	+2C	ICEVRES2	4	Reserved.
+48	+30	ICEVSIZE	4	SIZE/MAINSIZE specified or supplied by default. This field will contain X'FFFFFFFF' when SIZE=MAX was specified or defaulted. ²²
+52	+34	ICEVTHRY	4	Total amount of virtual storage theoretically available to DFSORT considering MINLIM. ²²
+56	+38	ICEVVIRT	4	Total amount of virtual storage actually available to DFSORT after any other program or exit has taken what it needed from the region. ^{22, 23}
+60	+3C	ICEVTABV	4	Amount of virtual storage theoretically available to DFSORT above 16 MB virtual. ²⁴
+64	+40	ICEVAABV	4	Amount of virtual storage available above 16 MB virtual after DFSORT has released the ARESALL and ARESINV space. ²⁴
+68	+44	ICEVRES3	4	Reserved.

Phase-timing statistics

Note: Because DFSORT makes phase-timing measurements internally, consider the values provided in Table 23 on page 202 to be approximate timings only.

Table 23. DFSORT phase-timing statistics				
Offsets		Name	Length	Description
Dec.	Hex			
+0	+0	ICEPTIME	4	Length of the DFSORT phase-timing statistics, including this field.
+4	+4	ICEPRES1	4	Reserved.

Table 23. DFSORT phase-timing statistics (continued)

Offsets		Name	Length	Description
Dec.	Hex			
+8	+8	ICEPTST	1	Phase-timing status flags: Bit Meaning when set 0 Timing information exists for phase 0. 1 Timing information exists for phase 1. 2 Timing information exists for phase 2. 3 Sort timing information exists for phase 3. 4 Merge or copy timing information exists for phase 3. 5 Timing information exists for phase 4. 6 TCB phase timing values are not valid. 7 Reserved.
+9	+9	ICEPRES2	3	Reserved.

Table 23. DFSORT phase-timing statistics

Phase 0 Timings:				
Offsets		Name	Length	Description
Dec.	Hex			
+12	+C	ICEPOTCB	4	TCB time for this phase, in hundredths of seconds. ³³
+16	+10	ICEPOELT	4	Elapsed time, in hundredths of seconds.
+20	+14	ICEPOSRB	4	SRB time for this phase, in hundredths of seconds.

Table 23. DFSORT phase-timing statistics

Phase 1 Timings:				
Offsets		Name	Length	Description
Dec.	Hex			
+24	+18	ICEP1TCB	4	TCB time for this phase, in hundredths of seconds. ³³
+28	+1C	ICEP1ELT	4	Elapsed time, in hundredths of seconds.
+32	+20	ICEP1SRB	4	SRB time for this phase, in hundredths of seconds.

Table 23. DFSORT phase-timing statistics

Phase 2 Timings:				
Offsets		Name	Length	Description
Dec.	Hex			
+36	+24	ICEP2TCB	4	TCB time for this phase, in hundredths of seconds. ³³
+40	+28	ICEP2ELT	4	Elapsed time, in hundredths of seconds.

Table 23. DFSORT phase-timing statistics (continued)				
Phase 2 Timings:				
Offsets		Name	Length	Description
Dec.	Hex			
+44	+2C	ICEP2SRB	4	SRB time for this phase, in hundredths of seconds.

Table 23. DFSORT phase-timing statistics				
Phase 3 SORT Timings:				
Offsets		Name	Length	Description
Dec.	Hex			
+48	+30	ICEP3TCB	4	TCB time for this phase, in hundredths of seconds. ³³
+52	+34	ICEP3ELT	4	Elapsed time, in hundredths of seconds.
+56	+38	ICEP3SRB	4	SRB time for this phase, in hundredths of seconds.

Table 23. DFSORT phase-timing statistics				
Phase 3 MERGE or COPY Timings:				
Offsets		Name	Length	Description
Dec.	Hex			
+60	+3C	ICEP3TBM	4	TCB time for this phase, in hundredths of seconds. ³³
+64	+40	ICEP3ETM	4	Elapsed time, in hundredths of seconds.
+68	+44	ICEP3SBM	4	SRB time for this phase, in hundredths of seconds.

Table 23. DFSORT phase-timing statistics				
Phase 4 Timings:				
Offsets		Name	Length	Description
Dec.	Hex			
+72	+48	ICEP4TCB	4	TCB time for this phase, in hundredths of seconds. ³³
+76	+4C	ICEP4ELT	4	Elapsed time, in hundredths of seconds.
+80	+50	ICEP4SRB	4	SRB time for this phase, in hundredths of seconds.
+84	+54	ICEPRES3	4	Reserved.

SORTIN statistics

Table 24. DFSORT SORTIN statistics				
Offsets		Name	Length	Description
Dec.	Hex			
+0	+0	ICESORTI	4	Length of the DFSORT SORTIN statistics including this field.
+4	+4	ICESINR1	4	Reserved.
+8	+8	ICESINNM	44	SORTIN data set name (EBCDIC) or HFS file name (EBCDIC). ³⁷

Table 24. DFSORT SORTIN statistics (continued)

Offsets		Name	Length	Description
Dec.	Hex			
+52	+34	ICESINCM	1	SORTIN information flags: Bit Meaning when set 0 SORTIN is concatenated. ³⁰ 1 SORTIN is SORTINnn. 2-5 Reserved. 6 Read Track CCWs were used. 7 DFSORT read the SORTIN data set.
+53	+35	ICESINR2	1	Reserved.
+54	+36	ICESINVS	1	SORTIN VSAM flags: ³⁷ Bit Meaning when set 0 SORTIN is a VSAM data set. 1 SORTIN is a KSDS. 2 SORTIN is an ESDS. 3 SORTIN is a RRDS. 4 SORTIN is a VRRDS. 5-6 Reserved. 7 SORTIN is a spanned VSAM data set.
+55	+37	ICESINR3	1	Reserved.

Table 25. SORTIN first volume or first data set information:24

Offsets		Name	Length	Description
Dec.	Hex			
+56	+38	ICESINDV	1	SORTIN device flags: Bit Meaning when set 0 SORTIN is a spool, dummy or pipe data set, or an HFS file. 1 SORTIN is tape. 2 SORTIN is disk. 3-6 Reserved. 7 SORTIN is multivolume.
+57	+39	ICESINR4	3	Reserved.
+60	+3C	ICESINDT	4	SORTIN device type. This information is taken from the UCBTYP field of the UCB.
+64	+40	ICESINR5	4	Reserved.
+68	+44	ICESINDA	1	SORTIN disk flags, byte 1: Bit Meaning when set 0 SORTIN resides on a shared disk. 1 SORTIN resides on a non-synchronous disk. 2 Reserved. 3 Reserved. 4 Reserved. 5 Reserved. 6 Reserved. 7 Reserved.

Table 25. SORTIN first volume or first data set information:24 (continued)

Offsets		Name	Length	Description
Dec.	Hex			
+69	+45	ICESIND2	1	SORTIN disk flags, byte 2: Bit Meaning when set 0 Reserved. 1 Reserved. 2 Reserved. 3 SORTIN resides on a disk connected to a 2105 control unit. 4 SORTIN resides on a disk connected to a 2107 control unit. 5-7 Reserved.
+70	+46	ICESINR6	2	Reserved.
+72	+48	ICESINLR	2	SORTIN LRECL.
+74	+4A	ICESINBK	2	SORTIN block size or control interval size (15-bit). A 31-bit version of this field is defined as ICESIBKF. ³⁹
+76	+4C	ICESINR7	3	Reserved.
+79	+4F	ICESINSP	1	SORTIN space allocation request: Bit Meaning when set 0 Cylinder request. 1 Track request. 2 Block request. ³⁴ 3-7 Reserved. This value is provided when available. This value is provided only for a disk data set.
+80	+50	ICESINBS	4	Average block size specified for block allocation requests. This value is provided only for a disk data set.
+84	+54	ICESIBKF	4	SORTIN block size or control interval size (31-bit). A 15-bit version of this field is defined as ICESINBK. ³⁹
+88	+58	ICESINSA	4	SORTIN secondary allocation (cylinders, tracks, or number of blocks). This value is provided when available. This value is provided only for a disk data set.
+92	+5C	ICESINEX	2	Total number of extents allocated. This value is provided only for a disk data set.

Table 26. Additional SORTIN information				
Offsets		Name	Length	Description
Dec.	Hex			
+94	+5E	ICESNCDV	1	Additional SORTIN information, byte 1: ²⁶ , ³¹ Bit Meaning when set 0 A tape was contained in the SORTIN concatenation list. 1 A disk was contained in the SORTIN concatenation list. 2 A multivolume data set was contained in the SORTIN concatenation list. 3-7 Reserved.
+95	+5F	ICESNCDA	1	Additional SORTIN information, byte 2: ²⁰ Bit Meaning when set 0 A concatenated SORTIN data set, or one or more volumes of a multivolume SORTIN data set (other than the first volume), resides on a shared disk. 1 A concatenated SORTIN data set, or one or more volumes of a multivolume SORTIN data set (other than the first volume), resides on a non-synchronous disk. 2 Reserved. 3 Reserved. 4 Reserved. 5 Reserved. 6 Reserved. 7 Reserved.

Table 26. Additional SORTIN information (continued)

Offsets		Name	Length	Description
Dec.	Hex			
+96	+60	ICESNCD2	1	Additional SORTIN information, byte 3: ²⁰ Bit Meaning when set 0 Reserved. 1 Reserved. 2 Reserved. 3 A concatenated SORTIN data set, or one or more volumes of a multivolume SORTIN data set (other than the first volume) resides on a disk connected to a 2105 control unit. 4 A concatenated SORTIN data set, or one or more volumes of a multivolume SORTIN data set (other than the first volume) resides on a disk connected to a 2107 control unit. 5-7 Reserved.
+97	+61	ICESINR9	7	Reserved.

SORTOUT statistics

Table 27. DFSORT SORTOUT statistics

Offsets		Name	Length	Description
Dec.	Hex			
+0	+0	ICESORT0	4	Length of the SORTOUT statistics, including this field.
+4	+4	ICESOTR1	4	Reserved.
+8	+8	ICESOTNM	44	SORTOUT data set name (EBCDIC) or HFS file name (EBCDIC). ³⁷
+52	+34	ICESOTCM	1	SORTOUT information flags: Bit Meaning when set 0 SORTOUT is a multivolume data set. 1-6 Reserved. 7 DFSORT wrote to the SORTOUT data set.
+53	+35	ICESOTR2	1	Reserved.

Table 27. DFSORT SORTOUT statistics (continued)

Offsets		Name	Length	Description
Dec.	Hex			
+54	+36	ICESOTVS	1	SORTOUT VSAM flags: Bit Meaning when set 0 SORTOUT is a VSAM data set. 1 SORTOUT is a KSDS. 2 SORTOUT is an ESDS. 3 SORTOUT is a RRDS. 4 SORTOUT is a VRRDS. 5-6 Reserved. 7 SORTOUT is a spanned VSAM data set.
+55	+37	ICESOTR3	1	Reserved.

Table 28. SORTOUT first volume or first data set information:26

Offsets		Name	Length	Description
Dec.	Hex			
+56	+38	ICESOTDV	1	SORTOUT device flags: Bit Meaning when set 0 SORTOUT is a spool, dummy or pipe data set, or an HFS file. 1 SORTOUT is tape. 2 SORTOUT is disk. 3-7 Reserved.
+57	+39	ICESOTR4	3	Reserved.
+60	+3C	ICESOTDT	4	SORTOUT device type: This information is taken from the UCBTYP field of the UCB.
+64	+40	ICESOTR5	4	Reserved.

Table 28. SORTOUT first volume or first data set information:26 (continued)

Offsets		Name	Length	Description
Dec.	Hex			
+68	+44	ICESOTDA	1	SORTOUT disk flags, byte 1: Bit Meaning when set 0 SORTOUT resides on a shared disk. 1 SORTOUT resides on a non-synchronous disk. 2 Reserved. 3 Reserved. 4 Reserved. 5 Reserved. 6 Reserved. 7 Reserved.
+69	+45	ICESOTD2	1	SORTOUT disk flags, byte 2: Bit Meaning when set 0 Reserved 1 Reserved. 2 Reserved. 3 SORTOUT resides on a disk connected to a 2105 control unit. 4 SORTOUT resides on a disk connected to a 2107 control unit. 5-7 Reserved.
+70	+46	ICESOTR6	2	Reserved.
+72	+48	ICESOTLR	2	SORTOUT LRECL.
+74	+4A	ICESOTBK	2	SORTOUT block size or control interval size (15-bit). A 31-bit version of this field is defined as ICESOBKF. ³⁹
+76	+4C	ICESOTR7	3	Reserved.

Table 28. SORTOUT first volume or first data set information:26 (continued)

Offsets		Name	Length	Description
Dec.	Hex			
+79	+4F	ICESOTSP	1	<p>SORTOUT space allocation request:</p> <p>Bit</p> <p>Meaning when set</p> <p>0 Cylinder request.</p> <p>1 Track request.</p> <p>2 Block request.³⁵</p> <p>3-7 Reserved.</p> <p>This value is provided when available. This value is provided only for a disk data set.</p>
+80	+50	ICESOTBS	4	Average block size specified for block allocation requests. This value is provided only for a disk data set.
+84	+54	ICESOBKF	4	SORTOUT block size or control interval size (31-bit). A 15-bit version of this field is defined as ICESOTBK. ³⁹
+88	+58	ICESOTSA	4	SORTOUT secondary allocation (cylinders, tracks, or number of blocks). This value is provided when available. This value is provided only for a disk data set.
+92	+5C	ICESOTEX	2	Total number of extents allocated. This value is provided only for a disk data set.

Table 29. SORTOUT multivolume Information:19,28

Offsets		Name	Length	Description
Dec.	Hex			
+94	+5E	ICESTCDA	1	SORTOUT multivolume flags, byte 1: Bit Meaning when set 0 One or more volumes (but not the first volume) of the multivolume SORTOUT data set resides on a shared disk. 1 One or more volumes (but not the first volume) of the multivolume SORTOUT data set resides on a non-synchronous disk. 2 Reserved. 3 Reserved. 4 Reserved. 5 Reserved. 6 Reserved. 7 Reserved.
+95	+5F	ICESTCD2	1	SORTOUT multivolume flags, byte 2: Bit Meaning when set 0 Reserved. 1 Reserved. 2 Reserved. 3 One or more volumes (but not the first volume) of the multivolume SORTOUT data set resides on a disk connected to a 2105 control unit. 4 One or more volumes (but not the first volume) of the multivolume SORTOUT data set resides on a disk connected to a 2107 control unit. 5-7 Reserved.

Work data set statistics

Table 30. DFSORT work data set statistics

Offsets		Name	Length	Description
Dec.	Hex			
+0	+0	ICESORTW	4	Length of the DFSORT work data set statistics, including this field. The length of this table depends on the number of work data sets. However, if more than 32 work data sets are allocated, statistics will only be shown for 32.
+4	+4	ICESWKR1	4	Reserved.
+8	+8	ICESWBLK	2	Work data set block size (signless).
+10	+A	ICESWKR2	10	Reserved.

Table 30. DFSORT work data set statistics

EXCP Counts per DFSORT phase:				
Offsets		Name	Length	Description
Dec.	Hex			
+20	+14	ICESWXP1	4	Number of EXCPs issued in phase 1.
+24	+18	ICESWXP2	4	Number of EXCPs issued in phase 2.
+28	+1C	ICESWXP3	4	Number of EXCPs issued in phase 3.
+32	+20	ICESWKR3	6	Reserved.

Table 30. DFSORT work data set statistics

Work Data Set Entries: ³⁶				
Offsets		Name	Length	Description
Dec.	Hex			
+38	+26	ICESWKEN	2	Number of work data set entries. If more than 32 work data sets are allocated, work data set entries will only be shown for 32.
+40	+28	ICESWK01	4	Pointer to the first work data set entry.
+44	+2C	ICESWK02	4	Pointer to the second work data set entry.
⋮	⋮	⋮	⋮	⋮
+164	+A4	ICESWK32	4	Pointer to the 32nd work data set entry.

Work data set entry

One work data set entry is provided for each work data set, up to a maximum of 32.

Table 31. DFSORT work data set entry

Offsets		Name	Length	Description
Dec.	Hex			
+0	+0	ICESWORK	4	Length of this entry including this field.
+4	+4	ICESER1	4	Reserved.
+8	+8	ICESEDDN	8	DDname of this work data set.

Table 31. DFSORT work data set entry (continued)				
Offsets		Name	Length	Description
Dec.	Hex			
+16	+10	ICESEFLG	1	Work data set flags: Bit Meaning when set 0 This data set resides on a shared disk. 1 This data set resides on a non-synchronous disk. 2 Reserved. 3 Reserved. 4 Reserved. 5 Reserved. 6 Reserved. 7 Reserved.
+17	+11	ICESEUNT	3	3-digit unit address for this work data set (EBCDIC). 'UCB' indicates the unit address has 4-digits. ICESEADR has the 4-digit unit address.
+20	+14	ICESEDEV	4	Device type of this work data set. This information is taken from the UCBTYP field in the UCB.
+24	+18	ICESEFL2	1	Additional work data set flags: Bit Meaning when set 0 READ-TRACK CCWs were used. 1 This data set resides on a disk connected to a cached 9343 control unit. 2 This data set resides on a disk connected to a 3990 Model 6 control unit. 3 This data set resides on a disk connected to a 2105 control unit. 4 This data set resides on a disk connected to a 2107 control unit. 5-7 Reserved.
+25	+19	ICESER2	3	Reserved.
+28	+1C	ICESETKI	4	Number of tracks allocated initially.
+32	+20	ICESETKF	4	Number of tracks allocated when sorting terminated.
+36	+24	ICESER3	4	Reserved.

Table 31. DFSORT work data set entry (continued)

Offsets		Name	Length	Description
Dec.	Hex			
+40	+28	ICESEEXI	2	Number of extents allocated initially.
+42	+2A	ICESEEXF	2	Number of extents allocated when sorting terminated.
+44	+2C	ICESEXPT	4	Total number of EXCPs for this work data set.
+48	+30	ICESEXP1	4	Number of EXCPs issued in phase 1.
+52	+34	ICESEXP2	4	Number of EXCPs issued in phase 2.
+56	+38	ICESEXP3	4	Number of EXCPs issued in phase 3.
+60	+3C	ICESEADR	4	4-digit unit address for this work data set (EBCDIC).

Hipersorting statistics

Table 32. DFSORT Hipersorting statistics

Offsets		Name	Length	Description
Dec.	Hex			
+0	+0	ICEHSORT	4	Length of the Hipersorting statistics, including this field.
+4	+4	ICEHRMAX	2	HIPRMAX value in effect.
+6	+6	ICEHSNBR	2	Number of Hiperspaces.
+8	+8	ICEHPGTA	4	Pages of central storage theoretically available to DFSORT for this run (for Hiperspace).
+12	+C	ICEHPGSP	4	Pages of central storage available at run time that will cause some paging (for Hiperspace).
+16	+10	ICEHPGNI	4	Pages of central storage available at run time that will not impact paging (for Hiperspace).
+20	+14	ICEHRES1	4	Reserved.
+24	+18	ICEHPGCR	4	Pages of Hiperspace created.
+28	+1C	ICEHPGUS	4	Pages of Hiperspace used.

Sorting with data space statistics

Table 33. DFSORT Sorting with data space statistics

Offsets		Name	Length	Description
Dec.	Hex			
+0	+0	ICEDSORT	4	Length of the sorting with data space statistics, including this field.
+4	+4	ICEDATMX	2	DSPSIZE value in effect.
+6	+6	ICEDSNBR	2	Number of data spaces created.
+8	+8	ICEDSBTA	4	Pages of central storage available that will have little impact on performance.
+12	+C	ICEDSBTB	4	Pages of central storage available that will have some impact on performance.

Table 33. DFSORT Sorting with data space statistics (continued)

Offsets		Name	Length	Description
Dec.	Hex			
+16	+10	ICEDSBTC	4	Pages of central storage available that will have high impact on performance.
+20	+14	ICERS1	4	Reserved.
+24	+18	ICERS2	4	Reserved.
+28	+1C	ICERS3	4	Reserved.
+32	+20	ICEDSCR	4	Pages of data space created.
+36	+24	ICEDSUS	4	Pages of data space used.

Memory object sorting statistics

Table 34. DFSORT memory object sorting statistics

Offsets		Name	Length	Description
Dec.	Hex			
+0	+0	ICEMOLNG	4	Length of the memory object sorting statistics, including this field.
+4	+4	ICEMOSZ	4	MOSIZE value in effect.
+8	+8	ICEMONBR	2	Number of memory objects allocated.
+10	+A	ICEMEMLS	1	MEMLIMIT source identifier (8-bit integer): 1 MEMLIMIT set by SMF 2 MEMLIMIT set by JCL 3 MEMLIMIT unlimited (REGION=0K/0M) 4 MEMLIMIT set by IEFUSI 5–8 MEMLIMIT set by UNIX functions 9 MEMLIMIT set by system authorized functions
+11	+B	ICEMOSLF	1	Memory object limit source identifier (8-bit integer): 1 Limit set by MOSIZE 2 Limit set by MEMLIMIT 3 Limit set by SYSEVENT STGTEST 4 Limit set by IARV64 5 Limit set by enqueueing process

Table 34. DFSORT memory object sorting statistics (continued)

Offsets		Name	Length	Description
Dec.	Hex			
+12	+C	ICEMOOPT	1	Memory objects usage flags: Bit Meaning when set 0 Memory objects were used as work storage. 1 Memory objects were used as main storage. 2-7 Reserved.
+13	+D	ICEMORS1	3	Reserved.
+16	+10	ICEMEML	8	MEMLIMIT value in megabytes (64-bit integer).
+24	+18	ICEMOCR	8	Megabytes for allocated memory objects (64-bit integer).
+32	+20	ICEMOUS	8	Megabytes for used memory objects (64-bit integer).
+40	+28	ICEMOSTG	8	Megabytes for central storage not in use (64-bit integer).
+48	+30	ICEMOSYS	8	Megabytes for memory object limit in effect (64-bit integer).
+56	+38	ICEMORS2	8	Reserved.

Footnotes

Footnotes:

- ¹⁹ The SMF return code is supplied only when SMF is active and either “SMF=FULL” or “SMF=SHORT” is in effect.
- ²⁰ The information in this field is provided for data sets and volumes containing active data only.
- ²¹ This value is provided only when Blockset is selected.
- ²² This value is the same as the corresponding value in message ICE092I or ICE093I.
- ²³ This value is not meaningful when the storage taken by an invoking program or exit is greater than REGION or SIZE/MAINSIZE.
- ²⁴ This value is the same as the corresponding value in message ICE156I. It is provided only when Blockset is selected.
- ²⁵ The information in this area applies to the first volume of a multivolume data set, or the first data set of a concatenated set of data sets. The former case takes precedence when SORTIN is both concatenated and the first data set is multivolume. No information, with the exception of LRECL and CI size, is provided in this area when SORTIN is a VSAM data set. No information, with the exception of the spool/dummy flag, LRECL and BLKSIZE, is provided in this area when SORTIN is a spool or dummy data set.
- ²⁶ This information is provided when SORTIN is concatenated, multivolume, or both. This information does not apply to the first data set of a concatenated list or to the first volume of a multivolume data set (which is the first or only SORTIN data set). No information is provided in this area when SORTIN is a VSAM data set or when the application is a merge.
- ²⁷ If SORTOUT is a multivolume data set, the information in this area refers to the first volume of the data set. With the exception of LRECL control interval size, no information is supplied in this area when SORTOUT is a VSAM data set. No information, with the exception of the spool/dummy flag, LRECL and BLKSIZE, is provided in this area when SORTOUT is a spool or dummy data set.
- ²⁸ This bit will not be set if DFSORT modules for the technique in use are resident.
- ²⁹ This information is provided only when SORTOUT is a multivolume data set. If it is, then this information will apply to all the volumes of the data set except the first. No information is provided in this area when execution is not in the Blockset technique.
- ³⁰ If this bit is set, then the additional SORTIN information bytes below contains valid information.

-
- ³¹ No information is provided in this area when execution is not in the Blockset technique.
 - ³² This field is zero if FILSZ/SIZE is not specified.
 - ³³ This value is not provided if the TIMEUSED service used to obtain CPU time fails.
 - ³⁴ When this bit is set, ICESINBS contains the average block length space request.
 - ³⁵ When this bit is set, ICESOTBS contains the average block length space request.
 - ³⁶ All work data set entries, if any, are ordered by most heavily used to least heavily used.
 - ³⁷ This information is provided for the first SORTIN or SORTOUT data set or HFS file. For an HFS file, only the first 44 bytes of the name is provided.
 - ³⁸ This field contains binary zero when DFSORT is JCL invoked, DFSORT is running resident, or the information is not generally available to DFSORT. If the invoker of DFSORT is running resident, this field is meaningless.
 - ³⁹ The Blockset technique provides a 31-bit value; the Peerage and Vale techniques provide a 15-bit value only. For Blockset, the smaller fields contain only the least significant 16 bits.

Appendix F. Accessibility

Accessible publications for this product are offered through [IBM Documentation for z/OS \(www.ibm.com/docs/en/zos\)](http://www.ibm.com/docs/en/zos).

If you experience difficulty with the accessibility of any z/OS documentation see [How to Send Feedback to IBM](#) to leave documentation feedback.

Notices

This information was developed for products and services that are offered in the USA or elsewhere.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

*IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
United States of America*

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

*Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan*

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

This information could include missing, incorrect, or broken hyperlinks. Hyperlinks are maintained in only the HTML plug-in output for IBM Documentation. Use of hyperlinks in other output formats of this information is at your own risk.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

*IBM Corporation
Site Counsel
2455 South Road*

Poughkeepsie, NY 12601-5400
USA

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions.

Applicability

These terms and conditions are in addition to any terms of use for the IBM website.

Personal use

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or

reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

IBM Online Privacy Statement

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user, or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

Depending upon the configurations deployed, this Software Offering may use session cookies that collect each user's name, email address, phone number, or other personally identifiable information for purposes of enhanced user usability and single sign-on configuration. These cookies can be disabled, but disabling them will also eliminate the functionality they enable.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM's Privacy Policy at ibm.com/privacy and IBM's Online Privacy Statement at ibm.com/privacy/details in the section entitled "Cookies, Web Beacons and Other Technologies," and the "IBM Software Products and Software-as-a-Service Privacy Statement" at ibm.com/software/info/product-privacy.

Policy for unsupported hardware

Various z/OS elements, such as DFSMSdfp, JES2, and MVS, contain code that supports specific hardware servers or devices. In some cases, this device-related element support remains in the product even after the hardware devices pass their announced End of Service date. z/OS may continue to service element code; however, it will not provide service related to unsupported hardware devices. Software problems related to these devices will not be accepted for service, and current service activity will cease if a problem is determined to be associated with out-of-support devices. In such cases, fixes will not be issued.

Minimum supported hardware

The minimum supported hardware for z/OS releases identified in z/OS announcements can subsequently change when service for particular servers or devices is withdrawn. Likewise, the levels of other software products supported on a particular release of z/OS are subject to the service support lifecycle of those

products. Therefore, z/OS and its product publications (for example, panels, samples, messages, and product documentation) can include references to hardware and software that is no longer supported.

- For information about software support lifecycle, see: [IBM Lifecycle Support for z/OS \(www.ibm.com/software/support/systemsz/lifecycle\)](http://www.ibm.com/software/support/systemsz/lifecycle)
- For information about currently-supported IBM hardware, contact your IBM representative.

Programming interface information

This publication primarily documents information that is NOT intended to be used as Programming Interfaces of DFSORT.

This publication also documents intended Programming Interfaces that allow the customer to write programs to obtain the services of DFSORT. This information is identified where it occurs, either by an introductory statement to a chapter or section or by the following marking:

Trademarks

IBM, the IBM logo, and [ibm.com](http://www.ibm.com) are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at [Copyright and Trademark information \(www.ibm.com/legal/copytrade.shtml\)](http://www.ibm.com/legal/copytrade.shtml).

Index

Special Characters

\$ORTPARM [91](#)

A

ABCODE parameter [61](#)
ABEND parameter [71](#)
accessibility
 contact IBM [221](#)
allocating storage
 temporary libraries [9](#)
alternate collating sequence [61](#), [63](#)
alternate message data set [86](#)
ALTSEQ parameter [61](#), [63](#)
applying SMP/E usermods [41](#)
ARESALL parameter [16](#), [25](#), [62](#)
ARESINV parameter [16](#), [25](#), [63](#)
assistive technologies [221](#)
Attention
 message ICE046A termination when DYNALLOC is in effect [79](#)
 use of ALTSEQ [62](#)

B

BCOPY [31](#)
block size, system determined [95](#)
buffer space, OUTFIL data set [89](#)

C

cache fast write [63](#)
century window [107](#)
CFW parameter [63](#)
CHALT parameter [63](#)
CHECK parameter [63](#)
checking record count [63](#)
checkpoint/restart [80](#)
CINV parameter [64](#)
COBEXIT parameter [64](#)
COBOL
 invoking DFSORT [93](#)
 user exits [64](#)
collating sequence
 EBCDIC [61](#)
 equal records [70](#)
 modifying [61](#), [63](#)
contact
 z/OS [221](#)
control statements, printing modified [82](#)
converting zoned-decimal (ZD) fields [107](#)
cultural environment
 considerations [36](#)
 storage considerations [83](#)
 using locales [82](#)

D

data compaction percentage [78](#)
Data Facility Sort (DFSORT)
 verifying installation of [9](#)
data set
 block size, system determined [95](#)
 maximum buffer space, OUTFIL [89](#)
dataspace sorting
 changing use with ICEIEXIT [121](#)
 ICETEXIT statistics [216](#)
 SMF record statistics [136](#)
day parameter [58](#)
ddname for message data set [86](#)
default primary space allocation [69](#)
defaults [35](#), [56](#), [113](#)
DEFAULTS operator
 listing installation defaults with [113](#)
deleting private libraries [11](#)
description of installation defaults [113](#)
devices required for storage [6](#)
DFSMSrmm [151](#)
DFSORT home page [3](#)
DFSPARM [91](#)
DIAGSIM parameter [65](#)
distribution tape
 contents [3](#)
 examples for storage administrators [3](#)
DSA parameter [66](#)
DSPSIZE parameter [66](#)
DYNALLOC parameter [67](#), [109](#)
DYNALOC parameter [67](#)
dynamic allocation
 DYNALOC parameter [67](#)
 DYNAPCT parameter [68](#)
 DYNAUTO parameter [69](#)
 example, ICEMAC macro [109](#)
Dynamic Storage Adjustment (DSA) [66](#)
DYNAPCT default [68](#)
DYNAPCT parameter [68](#)
DYNAUTO parameter [69](#)
DYNAPCT parameter [69](#)

E

EBCDIC collating sequence, modifying [61](#), [63](#)
EFS parameter [70](#)
EFS program [38](#)
ENABLE parameter [57](#)
environment installation modules [43](#)
equally collating records [70](#)
EQUALS parameter [70](#)
ERET parameter [71](#)
error messages [161](#), [169](#)
ESTAE parameter [71](#)
examples for storage administrators [3](#)
EXITCK parameter [72](#)

- exits, user
 - COBOL [64](#)
 - E15 [64](#)
 - E35 [63](#), [64](#)
- expanded storage
 - required [13](#)
 - tailoring [25](#), [46](#)
- EXPMAX parameter [72](#)
- EXPOLD parameter [73](#)
- EXPRES parameter [74](#)
- Extended Function Support (EFS) program
 - invoking [38](#)
 - parameter [70](#)

F

- format
 - error messages [161](#)
 - ICEMAC macro [51](#)
- FSZEST parameter [75](#)

G

- GCOPY1 [31](#)
- GCOPY2 [31](#)
- GEN [31](#)
- GENER parameter [76](#)
- GNPAD parameter [76](#)
- GNTRUNC parameter [77](#)

H

- Hipersorting
 - changing use with ICEIEXIT [121](#)
 - ICEMAC parameters [46](#)
 - ICETEXIT statistics [216](#)
 - maximum expanded storage [72](#)
 - maximum old expanded storage [73](#)
 - minimum expanded storage [74](#)
 - SMF record statistics [136](#)
- HIPRMAX parameter [77](#)
- home page (web) [3](#)

I

- ICEGENER
 - discontinuing use [31](#)
 - GENER parameter [76](#)
 - verifying correct installation [30](#)
- ICEIEXIT
 - general register 1 [123](#)
 - IEXIT parameter [80](#)
 - installing [122](#)
 - interface points [122](#)
 - overview of installation control [121](#)
 - parameter list for installation-time options [123](#)
 - parameter list for run-time options [124](#)
 - requirements [122](#)
 - sample routine [127](#)
 - using [38](#)
 - writing [122](#)
- ICEMAC macro
 - coding [51](#)

- ICEMAC macro (*continued*)
 - examples
 - batch JCL direct invocation defaults [108](#)
 - batch program invocation defaults [109](#)
 - time-of-day defaults [111](#)
 - TSO direct invocation defaults [110](#)
 - TSO program invocation defaults [110](#)
 - tailoring expanded storage [46](#)
 - tailoring main storage [14](#)
 - tailoring storage [45](#)
- ICEOPT message and return code [157](#)
- ICETEXIT
 - designing [137](#)
 - example, executable [143](#)
 - example, nonexecutable [139](#)
 - installing [137](#)
 - mapping statistics [138](#)
 - parameters [193](#)
 - statistical data collection [131](#)
 - TEXT parameter [100](#)
 - using [38](#)
- ICETOOL
 - examples for storage administrators [3](#)
 - verifying installation
 - in temporary libraries [10](#)
 - with sample job, ICETOOLJ [9](#)
- ICETPEX
 - Information passed [151](#)
 - Installing [151](#)
 - Sample [152](#)
 - Writing [151](#)
- identically collating records [70](#)
- IDRCPCT parameter [78](#)
- IEBGENER, using ICEGENER as substitute for [28](#), [76](#)
- IEXIT parameter [80](#)
- IGNCKPT parameter [80](#)
- informative messages [169](#)
- initialization exit (ICEIEXIT) [38](#)
- installation
 - defaults
 - IBM-supplied [113](#)
 - listing [113](#)
 - modifying [35](#)
 - temporary libraries [8](#)
 - verifying
 - ICEJCLJ and ICEINVJ [9](#)
 - ICETOOL [10](#)
 - Locale Processing [11](#)
- installation-time options
 - parameters changed by ICEIEXIT [123](#)
- installation-written exits
 - ICEIEXIT [38](#)
 - ICETEXIT [38](#)
- Internet [3](#)
- invoking DFSORT [56](#)
- IOMAXBF parameter [80](#)

K

- keyboard
 - navigation [221](#)
 - PF keys [221](#)
 - shortcut keys [221](#)

L

- libraries
 - deleting [11](#)
 - required [6](#)
- link-pack area (LPA) [13](#)
- LIST parameter [82](#)
- listing installation defaults [113](#)
- LISTX parameter [82](#)
- LOCALE parameter [82](#)
- locale processing
 - considerations for installation-wide default settings [83](#)
 - environment considerations [36](#)
 - LOCALE parameter [82](#)
 - storage considerations [83](#)
- Locale Processing
 - verifying installation in temporary libraries [11](#)

M

- machine requirements [6](#)
- main storage
 - allocating [16](#)
 - considerations [17](#)
 - factors affecting performance [12](#)
 - limiting [16](#)
 - reserving [16](#)
 - tailoring [14](#)
 - varying [13](#)
- maximum buffer space, OUTFIL data sets [89](#)
- MAXLIM parameter [16](#), [20](#), [83](#)
- memory object sorting
 - changing use with ICEIEXIT [121](#)
 - ICETEXIT statistics [217](#)
 - maximum central storage [72](#)
 - maximum old central storage [73](#)
 - minimum central storage [74](#)
 - SMF record statistics [136](#)
- memory object storage
 - required [13](#)
- message data set
 - LIST parameter [82](#)
 - LISTX parameter [82](#)
 - MSGDDN parameter [86](#)
 - MSGPRT parameter [87](#)
 - NOMSGDD parameter [87](#)
- message module error messages [168](#)
- message texts
 - modifying [39](#)
 - translating [39](#)
- messages
 - error [161](#)
 - explanation [161](#)
 - format [161](#)
 - informational [169](#)
 - informative [169](#)
 - message module error [168](#)
- migrating from other sort products [45](#)
- MINLIM parameter [16](#), [84](#)
- MNOTE messages [169](#)
- modified control statements, printing [82](#)
- modifying
 - installation defaults [35](#)

- modifying (*continued*)
 - message texts [39](#)
- MOSIZE parameter [84](#)
- MOWRK parameter [86](#)
- MSGCON parameter [86](#)
- MSGDDN parameter [86](#)
- MSGPRT parameter [87](#)

N

- navigation
 - keyboard [221](#)
- NOMSGDD parameter [87](#)
- NULLOFL parameter [88](#)
- NULLOUT parameter [88](#)

O

- ODMAXBF parameter [89](#)
- operating system
 - conventions [5](#)
 - requirements [5](#)
 - standards [5](#)
- OS/VS COBOL
 - invoking DFSORT [93](#)
 - user exits [64](#)
- OUTFIL processing
 - changing maximum buffer space with ICEIEXIT [121](#)
 - examples for storage administrators [3](#)
 - maximum buffer space [89](#)
- OUTREL parameter [89](#)
- OUTSEC parameter [90](#)
- OVERRGN parameter [16](#), [25](#), [26](#), [90](#)
- OVFLO parameter [90](#)

P

- PAD parameter [91](#)
- performance
 - factors affecting, main storage [12](#)
- performance considerations [12](#)
- PL/I [93](#)
- prerequisite products [5](#)
- primary space allocation, default [69](#)
- printing all control statements and messages [65](#)
- printing modified control statements [82](#)
- procedures
 - modifying [35](#)
 - SORT standard procedure [35](#)
 - SORTD standard procedure [35](#)

R

- receiving SMP/E usermods [41](#)
- Recommendation
 - performance [12](#)
- record count [63](#)
- reinstalling DFSORT [28](#)
- require
 - machine [6](#)
- required
 - expanded storage [13](#)
 - libraries [6](#)

- required (*continued*)
 - main storage [12](#)
 - memory object storage [13](#)
- RESALL parameter [16](#), [23](#), [92](#)
- RESET parameter [92](#)
- RESINV parameter [16](#), [25](#), [92](#)
- return codes, ERET parameter [71](#)
- run time
 - changing options [121](#)
 - examining information [121](#)
 - options, changed by ICEIEXIT [124](#)
- running DFSORT nonresident [8](#)
- running DFSORT resident [8](#)
- running resident
 - required storage [13](#)

S

- sample jobs
 - ICECOBJ [9](#)
 - ICECSRTJ [9](#)
 - ICEINVJ [9](#)
 - ICEJCLJ [9](#)
 - ICETOOLJ [9](#)
- SDB parameter [93](#)
- SDBMSG parameter [95](#)
- shortcut keys [221](#)
- simulating SORTDIAG DD statement [65](#)
- SIZE parameter [14](#), [16](#), [17](#), [96](#)
- SMF parameter [97](#)
- SMF record
 - collecting data from [131](#)
 - format [175](#)
 - mapping [133](#)
 - type [16](#) [175](#)
 - type-16 [132](#)
- SMF statistics for ICETEXIT [193](#)
- SMF type-16 record [132](#), [175](#)
- SMF=FULL [132](#)
- SMF=NO [132](#)
- SMF=SHORT [132](#)
- SMP/E usermod
 - applying [41](#)
 - receiving [41](#)
- software requirements [5](#)
- SOLRF parameter [97](#)
- sorting instructions [12](#)
- SORTLIB parameter [98](#)
- SPANINC parameter [98](#)
- specifying maximum storage [66](#)
- statistical data collection
 - overview [131](#)
 - using ICETEXIT [131](#)
 - using SMF records [131](#)
- storage
 - amount required for running resident [13](#)
 - decreasing main [13](#)
 - devices [6](#)
 - expanded [13](#), [25](#), [46](#)
 - for temporary private libraries [9](#)
 - memory object [13](#)
 - modifying with ICEIEXIT [121](#)
 - required [12](#), [13](#)
 - required, main [12](#)

- storage (*continued*)
 - tailoring expanded
 - with ICEMAC [46](#)
 - tailoring main
 - with ICEMAC [45](#)
 - varying main [13](#)
- storage administrator examples [3](#)
- summary of changes [xvii](#)
- SVC
 - parameter [99](#)
- SVC installation
 - verifying [11](#)
- SYSDA default [68](#)
- system determined block size [95](#)
- system management facilities (SMF) record [131](#)
- SZERO parameter [99](#)

T

- tailoring main storage [14](#)
- Tape processing [151](#)
- target libraries [8](#)
- termination exit (ICETEXIT) [38](#)
- testing DFSORT [8](#)
- TEXT parameter [100](#)
- time-of-day installation modules [43](#)
- TMAXLIM parameter [16](#), [18](#), [100](#)
- TRUNC parameter [101](#)
- TUNE parameter [101](#)

U

- unicode [37](#)
- unicode Environment Considerations [37](#)
- user exits
 - COBOL [64](#)
 - E15 [64](#)
 - E35 [63](#), [64](#)
- user interface
 - ISPF [221](#)
 - TSO/E [221](#)

V

- VERIFY parameter [102](#)
- verifying installation
 - ICEINVJ [9](#)
 - ICEJCLJ [9](#)
 - ICETOOL [10](#)
 - Locale Processing [11](#)
- verifying SVC installation [11](#)
- verifying that DFSORT was installed correctly [9](#)
- VIO parameter [102](#)
- virtual allocation [102](#)
- VLLONG parameter [103](#)
- VLSCMP parameter [103](#)
- VLSHRT parameter [104](#)
- VS COBOL II
 - invoking DFSORT [93](#)
 - user exits [64](#)
- VSAM buffers, specifying the number of [105](#)
- VSAMBSP parameter [105](#)
- VSAMEMT parameter [105](#)

VSAMIO parameter [106](#)

W

Web [3](#)

web site [3](#)

work data sets

- choice of [6](#)

- dedicated [35](#)

- dynamic allocation [67–69](#)

- releasing unused space [106](#)

- secondary allocation [106](#)

- virtual allocation [102](#)

WRKREL parameter [106](#)

WRKSEC parameter [106](#)

Y

Y2PAST parameter [107](#)

Year 2000 [107](#)

Z

ZDPRINT parameter [107](#)

ZSORT parameter [107](#)



Product Number: 5655-ZOS

SC23-6881-70

