

DFSORT



Installation and Customization

Release 14

DFSORT



Installation and Customization

Release 14

Note!

Before using this information and the product it supports, be sure to read the general information under Appendix E, "Notices" on page 231.

Twenty-First Edition (March 2002)

This edition replaces and makes obsolete the previous edition, SC33-4034-19. The technical changes for this edition are summarized under "Summary of Changes," and are indicated by a vertical bar to the left of a change.

This edition applies to Release 14 of DFSORT (5740-SM1) and to any subsequent releases until otherwise indicated in new editions. Make sure you are using the correct edition for the level of the product.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address below.

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© **Copyright International Business Machines Corporation 1973, 2002. All rights reserved.**

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

Contents

Figures	vii
Tables	ix
Preface	xi
About This Book	xi
Required Product Knowledge	xii
Where to find more information	xii
DFSORT Publications	xii
DFSORT Library Softcopy Information	xiii
How to Send Your Comments	xiii
Notational Conventions	xiii
Summary of Changes	xvii
Twenty-First Edition, March 2002	xvii
New Programming Support for Release 14 (PTFs — March, 2002)	xvii
Twentieth Edition, July 2000	xx
New Programming Support for Release 14 (PTFs - July, 2000)	xx
Nineteenth Edition, September 1998	xxiii
New Programming Support for Release 14	xxiii

Part 1. Planning for Installation 1

Chapter 1. Getting Ready to Install DFSORT	3
DFSORT on the Web	3
DFSORT FTP Site	3
What Is on the Tape?	3
Storage Administrator Examples	3
About the DFSORT Program Directory	4
Chapter 2. System Planning Considerations	5
Where DFSORT Fits on Your System	5
Software Requirements	5
Driving System Requirements	5
Target System Requirements	6
Required Storage Devices	6
Machine Requirements	6
Choosing Where to Install DFSORT	6
Required Distribution Libraries	7
Required Target Libraries	7
Should You Run DFSORT Resident?	8
Testing DFSORT	9
Making the Target Libraries Accessible	9
Running More Than One DFSORT Release at a Time	9
Replacing a Previous DFSORT Release	10
Allocating DFSORT Libraries	12
Verifying that DFSORT Was Installed Correctly	12
ICEJCLJ and ICEINVJ Verification	13
ICETOOLJ Verification	14
ICECSRTJ Verification	15
ICECOBJ Verification	16
Deleting Private Libraries after Testing	16
Verifying the SVC Was Installed Correctly	16

DFSORT Storage Considerations	16
Required Main Storage	16
Effects of Varying Main Storage	17
Required Storage for Running DFSORT Resident	18
Required Hiperspace Storage	18
Tailoring Main Storage	18
Tailoring Storage for Hipersorting	28
Installing DFSORT in Permanent Libraries	30
Replacing IEBGENER with the ICEGENER Facility	30
Verifying ICEGENER Installation	32
Discontinuing Use of ICEGENER as a Replacement for IEBGENER	33
Chapter 3. Planning for DFSORT Panels	35
Where DFSORT Panels Fits on Your System	35
Selecting Distribution and Target Libraries	35
Required Distribution Libraries	36
Required Target Libraries	36
Allocating Storage for Installation	37
Making the DFSORT Panels Available After Installation	37
Accessing DFSORT Panels	37
Testing DFSORT Panels	40

Part 2. Customizing DFSORT 45

Chapter 4. Planning for Customization	47
Customizing the Standard DFSORT Procedures	47
Changing the DFSORT Installation Defaults	48
Cultural Environment Considerations	49
Invoking an EFS Routine Automatically	50
Using Initialization and Termination Exits	50
Initialization Exits	50
Termination Exits	51
Modifying Message Texts	51
For the Base Product	51
For DFSORT Panels	54
Chapter 5. Changing the Installation Defaults—ICEMAC Macro	55
Migrating to DFSORT from Other Sort Products	56
Tailoring Main Storage with ICEMAC	57
Tailoring Hiperspace Storage with ICEMAC	57
How to Code the ICEMAC Macro	57
ICEMAC Syntax Diagram	58
ICEMAC Parameters	62
Examples of ICEMAC Macros	114
ICEMAC JCL - Batch Direct Invocation Defaults	114
ICEMAC INV - Batch Program Invocation Defaults	115
ICEMAC TSO - TSO Direct Invocation Defaults	116
ICEMAC TSOINV - TSO Program Invocation Defaults	117
ICEMAC TD1 - First Time-of-Day Module Defaults	118
ICEMAC Defaults	118
Listing ICEMAC Defaults	119
Chapter 6. Controlling Resources Using ICEIEXIT	127
What Run-Time Options Can ICEIEXIT Change?	127
What Installation- and Run-Time Information Can ICEIEXIT Examine?	128
Installation-time Information	128

Run-time Information	128
Writing and Installing ICEIEXIT	128
Interface Points for ICEIEXIT	129
Contents of General Register 1	129
Contents of the Installation-Time Options List	129
Contents of the Run-Time Options List.	131
A Sample ICEIEXIT Routine	133
Chapter 7. Collecting Statistical Data	137
Why Use DFSORT's SMF Record?	137
Why Use an ICETEXIT?	137
Collecting Data from DFSORT's SMF Record	137
Controlling What DFSORT Passes to SMF	138
When DFSORT Cannot Pass a Record to SMF	138
SMF Record Structure.	139
Mapping the SMF Record	139
Verifying Supported Record Sections	139
Accessing Multiple Input Data Set Sections	140
Accessing Multiple OUTFIL Data Set Sections	140
Identifying What Release Produced Your SMF Record	140
Collecting Data Using an ICETEXIT.	141
Installing an ICETEXIT	144
Writing an ICETEXIT Routine	144
Controlling What DFSORT Passes to ICETEXIT	145
Mapping ICETEXIT Statistics	145
ICETEXIT Examples	146
A Nonexecutable ICETEXIT.	146
An Executable ICETEXIT	151
Chapter 8. Improving Tape Processing with DFSMSrmm or ICETPEX	159
Information Passed by DFSMSrmm or ICETPEX	159
Installing an ICETPEX Routine	159
Writing an ICETPEX Routine	159
A Sample ICETPEX Routine	161

Part 3. Appendixes 163

Appendix A. The ICEMSGs Macro	165
Appendix B. ICEMAC and Message Module Messages	177
Error Messages	177
ICEMAC Error Messages	177
Message Module Error Messages	182
ICEMAC Informational Messages.	184
Appendix C. SMF Type-16 Record.	197
Product Section	199
Data Section	199
Record-Length Distribution Section	206
Input Data Set Section	207
SORTOUT Data Set Section	208
OUTFIL Data Set Section	209
Appendix D. Termination Exit (ICETEXIT) Parameters	213
SMF Statistics.	213
General Statistics	213

Options Statistics	214
Sort/Merge Statistics	217
Virtual Storage Statistics	218
Phase-Timing Statistics	219
SORTIN Statistics	221
SORTOUT Statistics	224
Work Data Set Statistics	227
Work Data Set Entry	227
Hipersorting Statistics	229
Sorting with Data Space Statistics	229
Appendix E. Notices	231
Programming Interface Information	232
Trademarks.	232
Summary of Changes	233
New Programming Support for Release 13 (PTFs after April, 1996)	233
Additional Year 2000 Features.	233
OS/390 Registration	233
New Programming Support for Release 13 (PTFs – April, 1996)	233
Year 2000 Features.	233
Performance Improvements for FLR and VLR Blockset Sorts	233
Floating Point for SUM	234
Security Improvements	234
EXCPVR Processing Removed	234
New Device Support for Release 13 (PTFs).	234
New Programming Support for Release 13	235
DFSORT's Performance Booster for The SAS** System	235
Dynamic Hipersorting	235
Performance	235
OUTFIL Processing.	235
National Language Support.	236
ICETOOL Enhancements	236
INCLUDE/OMIT Substring Search	237
SMF Type-16 Record Enhancements	237
Other Enhancements	237
New Programming Support for Release 12 (PTFs)	238
New Device Support for Release 12 (PTFs).	239
Index	241

Figures

1.	Expected PRTAFTER SORTOUT Output from ICEJCLJ and ICEINVJ	14
2.	Expected TOOLRUN DEPTSP Output from ICETOOLJ	15
3.	Expected TOOLRUN LIST1 Output from ICETOOLJ	15
4.	Expected TOOLRUN DEPTOT Output from ICETOOLJ	15
5.	Expected CSORT SORTOUT Output from ICECSRTJ	15
6.	Expected SORT SORTOUT Output from ICECOBJ	16
7.	Storage Comparisons for SIZE and REGION	23
8.	Effect of TMAXLIM Option on Storage	24
9.	Effects of MAXLIM Option on Storage	26
10.	Effects of RESALL Option on Storage	27
11.	Sample TSO Logon Procedure.	38
12.	Sample CLIST used to ALLOCATE the DFSORT Panels Libraries	39
13.	Sample CLIST (LDEF) Using LIBDEF to Allocate Data Sets	39
14.	The DFSORT Primary Option Menu	40
15.	Selecting Data Sets on the Sort Entry Panel, Page 1	41
16.	Specifying the Output Data Set on the Sort Entry Panel, Page 2	42
17.	Organizing the Records on the Sort Statement Data Entry Panel	42
18.	Sort Entry Panel, Page 1, After a Successful Sort	43
19.	Suggested Translations for Two ICEMSGS Messages	53
20.	The ICEMSGS Macro with two Messages Changed to French	53
21.	Message HELP Panel	54
22.	Syntax of the ICEMAC Macro	59
23.	Coding the ICEMAC Macro for ICEAM1	115
24.	Coding the ICEMAC Macro for ICEAM2	116
25.	Coding the ICEMAC Macro for ICEAM3	117
26.	Coding the ICEMAC Macro for ICEAM4	117
27.	Coding the ICEMAC Macro for ICETD1	118
28.	ICETOOL DEFAULTS Job	119
29.	Page 1 of the Output from the ICETOOL DEFAULTS Job Showing IBM-supplied Defaults	120
30.	Page 1 (continued) and Page 2 of the Output from the ICETOOL DEFAULTS Job Showing IBM-supplied Defaults	121
31.	Page 3 of the Output from the ICETOOL DEFAULTS Job Showing IBM-supplied Defaults	122
32.	Page 4 of the Output from the ICETOOL DEFAULTS Job Showing IBM-supplied Defaults	123
33.	Page 4 (continued) and Page 5 of the Output from the ICETOOL DEFAULTS Job Showing IBM-supplied Defaults	124
34.	Page 6 of the Output from the ICETOOL DEFAULTS Job Showing IBM-supplied Defaults	125
35.	Output from the ICETOOL DEFAULTS Job Showing that Some of the Defaults Have Been Changed	125
36.	Sample ICEIEXIT Routine	135
37.	Information DFSORT Passes to Your ICETEXIT	142
38.	A Nonexecutable ICETEXIT	146
39.	An Executable ICETEXIT	151
40.	Sample ICETPEX Routine	161
41.	ICEMSGS Macro	166

Tables

1.	KB of Virtual Memory Required in LPA for Release 14	18
2.	How Storage-Management Installation Defaults Can Be Overridden at Run Time	18
3.	Options That Tailor Main Storage	20
4.	DD Names for DFSORT Libraries.	37
5.	Options That Can Ease Migration.	56
6.	Example of DYNMPC Primary Space	74
7.	SDB=LARGE for Tape Output Data Sets	99
8.	Controlling the Type of SMF Record DFSORT Passes to ICETEXIT	145
9.	SMF Type-16 Record.	197
10.	SMF Type-16 Record: Product Section	199
11.	SMF Type-16 Record: Data Section	199
12.	SMF Type-16 Record: Record-Length Distribution Section	206
13.	SMF Type-16 Record: Input Data Set Section.	207
14.	SMF Type-16 Record: SORTOUT Data Set Section	208
15.	SMF Type-16 Record: OUTFIL Data Set Section.	210
16.	DFSORT SMF Statistics.	213
17.	DFSORT General Statistics	213
18.	DFSORT Option Statistics	214
19.	DFSORT Sort/Merge Statistics	217
20.	DFSORT Virtual Storage Statistics	218
21.	DFSORT Phase-Timing Statistics	219
22.	DFSORT SORTIN Statistics	221
23.	SORTIN first volume or first data set information:24	221
24.	Additional SORTIN Information	223
25.	DFSORT SORTOUT Statistics	224
26.	SORTOUT first volume or first data set information:26	225
27.	SORTOUT Multivolume Information:19,28	226
28.	DFSORT Work Data Set Statistics	227
29.	DFSORT Work Data Set Entry	227
30.	DFSORT Hipersorting Statistics	229
31.	DFSORT Sorting with Data Space Statistics	229

Preface

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 book is designed to be used with the DFSORT Program Directory for planning before installation. This book is also designed to be used for customization after installation. The DFSORT Program Directory covers tasks to be done for installation. If you are installing from a ServerPac, installation tasks are covered in *ServerPac: Installing Your Order*.

The Program Directory is shipped in the same package as the DFSORT distribution tape. It 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). The publication *ServerPac: Installing Your Order* also lists these FMIDs.

This book provides planning information for installing DFSORT. The information in the DFSORT Program Directory is more current than the information in this book. If the information in these two sources differs, please use the information in the Program Directory.

The Preventive Service Planning (PSP) "bucket" identified in the Program Directory is the most current source of information and may contain updates to this book, to the Program Directory, or to both. If DFSORT was provided by ServerPac, you should also consult the applicable z/OS or OS/390 PSP upgrades and subsets identified in *ServerPac: Installing Your Order*, which is shipped with your ServerPac order.

About This Book

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 ServerPac, follow the steps in *ServerPac: Installing Your Order* to install DFSORT. If DFSORT was not provided by ServerPac, follow the steps in the DFSORT Program Directory to install DFSORT. If you then want to customize DFSORT, read the pertinent chapters in Part 2, "Customizing DFSORT" on page 45.

The information in this book 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.

- Chapter 3, “Planning for DFSORT Panels” on page 35, describes how to prepare to install the DFSORT Panels.
- Part 2, “Customizing DFSORT” on page 45, discusses how to customize DFSORT and what you need to think about before you do so.
 - Chapter 4, “Planning for Customization” on page 47, discusses items you need to think about before customizing standard DFSORT procedures, modifying DFSORT defaults, and changing the message texts.
 - Chapter 5, “Changing the Installation Defaults—ICEMAC Macro” on page 55, describes the DFSORT installation options.
 - Chapter 6, “Controlling Resources Using ICEIEXIT” on page 127, 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 7, “Collecting Statistical Data” on page 137, explains how to use the DFSORT SMF type-16 record and a termination exit (ICETEXIT) to collect statistical data.
 - Chapter 8, “Improving Tape Processing with DFSMSrmm or ICETPEX” on page 159, 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, “The ICEMSGs Macro” on page 165, depicts the ICEMSGs macro, which contains the texts for DFSORT messages.
 - Appendix B, “ICEMAC and Message Module Messages” on page 177, lists messages issued when assembling ICEMAC and message modules.
 - Appendix C, “SMF Type-16 Record” on page 197, illustrates the format of the SMF type-16 record.
 - Appendix D, “Termination Exit (ICETEXIT) Parameters” on page 213, illustrates the format of the information passed to the termination exit, ICETEXIT.
 - Appendix E, “Notices” on page 231, includes the notices, Programming Interface information, and the trademark list.

Required Product Knowledge

To use this book effectively, you should be familiar with job control language (JCL) and the IBM System Modification Program Extended (SMP/E). If you plan to make the DFSORT panels available for use, you should also be familiar with the Interactive System Productivity Facility (ISPF), the Time Sharing Option Extensions (TSO/E), and the Interactive Storage Management Facility (ISMF).

Where to find more information

For up-to-date descriptions of all of the books that support z/OS, refer to the *z/OS Information Roadmap*. You can obtain a softcopy version of this book from the z/OS Internet Library at <http://www.ibm.com/servers/eservers/zseries/zos/bkserv/>.

DFSORT Publications

DFSORT Installation and Customization Release 14 is part of a more extensive DFSORT library. The additional books in the library are listed below.

Task	Publication	Order Number
Application Programming	<i>DFSORT Application Programming Guide Release 14</i>	SC33-4035

Where to find more information

Task	Publication	Order Number
Interpreting Messages and Diagnosing Failures	<i>DFSORT Messages, Codes and Diagnosis Guide Release 14</i>	SC26-7050
Learning to use DFSORT	<i>Getting Started with DFSORT Release 14</i>	SC26-4109
Learning to use DFSORT Panels	<i>DFSORT Panels Guide</i>	GC26-7037
Quick Reference	<i>DFSORT Reference Summary Release 14</i>	SX33-8001
Tuning DFSORT	<i>DFSORT Tuning Guide Release 14</i>	SC26-3111

You can order a complete set of DFSORT books with the order number SBOF-1243, except for *DFSORT Licensed Program Specifications GC33-4032*, which must be ordered separately.

DFSORT Library Softcopy Information

A softcopy version of the DFSORT library is available on CD-ROM, as shown in the table that follows. The CD-ROM contains all of the DFSORT books for Release 13 and Release 14, with the exception of the *DFSORT Reference Summary*.

Order Number	Title
SK3T-4269	<i>z/OS Collection</i>

How to Send Your Comments

Your feedback is important in helping to provide the most accurate and high-quality information. If you have any comments about this book or any other DFSORT documentation:

- Send your comments by e-mail to:
 - IBMLink from US: starpubs@us.ibm.com
 - IBMLink from Canada: STARPUBS at TORIBM
 - IBM Mail Exchange: USIB3VVD at IBMMAIL
 - Internet: starpubs@us.ibm.com

Be sure to include the name of the book, the part number of the book, version and product name, and if applicable, the specific location of the text you are commenting on (for example, a page number or a table number).

- Fill out one of the forms at the back of this book and return it by mail or by giving it to an IBM representative. If the form has been removed, address your comments to IBM Corporation, RCF Processing Department G26/050, 5600 Cottle Road, San Jose, California 95193-0001, U.S.A.

Notational Conventions

The syntax diagrams in this book are designed to make coding the ICEMAC macro call simple and unambiguous. The lines and arrows represent a path or flowchart that connects operators, parameters, and delimiters in the order and syntax in which they must appear in your completed statement. Construct a statement by tracing a path through the appropriate diagram that includes all the parameters you need, and code them in the order that the diagram requires you to follow. Any path through the diagram gives you a correctly coded statement, if you observe these conventions:

Where to find more information

- Read the syntax diagrams from left to right and from top to bottom.
- Begin coding your statement at the spot marked with the double arrowhead.



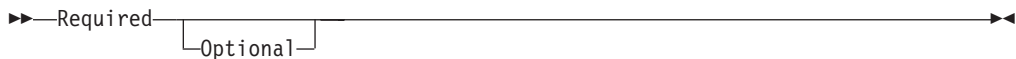
- A single arrowhead at the end of a line indicates that the diagram continues on the next line or at an indicated spot.



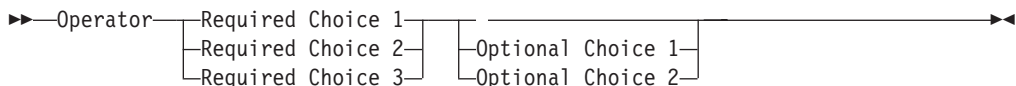
- A continuation line begins with a single arrowhead.



- Strings in upper-case letters are operators and parameters, and must be coded exactly as shown. (The conventions require that at least one blank separates the initial operator from the succeeding parameters; no blanks are allowed between parameters.)
- Punctuation (parentheses, commas, and so on), must be coded exactly as shown.
- Strings in all lowercase letters represent information that you supply.
- Required parameters appear on the same horizontal line (the main path) as the operator, while optional parameters appear in a branch below the main path.

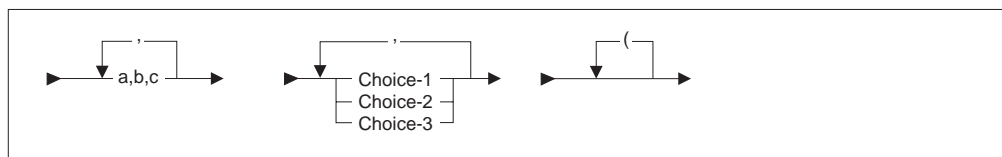


- Where you can make one choice between two or more parameters, the alternatives are stacked vertically.



If one choice within the stack lies on the main path (as in the example above, left), you *must* specify one of the alternatives. If the stack is placed below the main path (as in the example above, right), then selections are optional, and you can choose either one or none of them.

- The repeat symbol shows where you can return to an earlier position in the syntax diagram to specify a parameter more than once (see the example below, left), to specify more than one choice at a time from the same stack (see the example below, middle), or to nest parentheses (see the example below, right).



Where to find more information

Do not interpret a repeat symbol to mean that you can specify incompatible parameters. For instance, do not specify both **LIST=YES** and **LIST=NO** in the same **ICEMAC** macro call, or attempt to nest parentheses incorrectly.

Use any punctuation or delimiters that appear within the repeat symbol to separate repeated items.

A double arrowhead at the end of a line indicates the end of the syntax diagram.



Where to find more information

Summary of Changes

Twenty-First Edition, March 2002

New Programming Support for Release 14 (PTFs — March, 2002)

z900

DFSORT can now exploit the z900's 64-bit real architecture by backing storage and data spaces in real storage above 2 gigabytes, and by using central storage instead of expanded storage for Hipersorting.

Multiple Hiperspaces

DFSORT can now use multiple Hiperspaces for external storage requirements, increasing DFSORT's ability to use Hipersorting for large sort applications when sufficient system resources are available.

Managed Tapes

DFSORT can now automatically obtain accurate input file size information for tapes managed by tape management systems that supply a tape exit routine (ICETPEX). This can result in improved sort performance and more accurate dynamic work space allocation.

DFSORT can now automatically obtain input and output attributes such as RECFM, LRECL and BLKSIZE, for tapes managed by tape management systems that supply a tape exit routine (ICETPEX).

ICETOOL Enhancements

New EMPTY, NOTEMPTY, HIGHER(n), LOWER(n), EQUAL(n) and NOTEQUAL(n) options of ICETOOL's COUNT operator allow you to set RC=12 or RC=0 based on the count of records in a data set.

A new FIRSTDUP option of ICETOOL's SELECT operator allows you to keep just the first record of each set of duplicates.

A new LASTDUP option of ICETOOL's SELECT operator allows you to keep just the last record of each set of duplicates.

The DISCARD(savedd) option of ICETOOL's SELECT operator can now be specified without the TO(outdd) operand.

The maximum length for a CH field used with ICETOOL has been raised to 1500.

The maximum length for a HEX field used with ICETOOL has been raised to 1000.

New DT1, DT2 and DT3 formats of ICETOOL's DISPLAY and OCCUR operators allow you to produce meaningful representations of SMF date values.

New TM1, TM2, TM3 and TM4 formats of ICETOOL's DISPLAY and OCCUR operators allow you to produce meaningful representations of SMF time values.

A new INDENT(n) option of ICETOOL's DISPLAY and OCCUR operators allows you to specify the number of blanks for indentation.

A new BETWEEN(n) option of ICETOOL's DISPLAY and OCCUR operators allows you to specify the number of blanks between columns.

A new STATLEFT option of ICETOOL's DISPLAY operator allows you to print the headings for statistics to the left of the first column of data.

The use of formatting items has been extended to the record length, record number and break field of ICETOOL's DISPLAY operator, and to the data field, record length and value count of ICETOOL's OCCUR operator.

Additional formatting items are now provided as follows: E'pattern' (use a specified pattern for numeric digits such as phone numbers, social security numbers, dates, and so on), /D (divide by 10), /C (divide by 100), /DK (divide by 10000), /CK (divide by 100000), LZ (leading zeros for mask), NOST (no statistics), and Ndd (dd digits).

A new DATENS(abc) option of ICETOOL's DISPLAY and OCCUR operators allows you to insert the date of your ICETOOL run into your titles without a separator. Separators are included for existing options DATE and DATE(abcd).

A new TIMENS(ab) option of ICETOOL's DISPLAY and OCCUR operators allows you to insert the time of your ICETOOL run into your titles without a separator. Separators are included for existing options TIME and TIME(abc).

OUTFIL Enhancements

A new FTOV option of OUTFIL allows you to convert fixed-length input records (for example, FB) to variable-length output records (for example, VB). A new VTOF option of OUTFIL (an alias for CONVERT) allows you to convert variable-length input records (for example, VB) to fixed-length output records (for example, FB).

A new VLTRIM=byte option of OUTFIL allows you to remove trailing bytes such as blanks, binary zeros or asterisks, from variable-length output records.

A new REMOVECC option of OUTFIL allows you to remove the ANSI control characters from a report.

New DATEn, DATEn(c) and DATEnP options of OUTFIL OUTREC allow you to insert the date of your DFSORT run into your records in various forms.

New TIMEn, TIMEn(c) and TIMEnP options of OUTFIL OUTREC allow you to insert the time of your DFSORT run into your records in various forms.

New DT1, DT2 and DT3 formats of OUTFIL OUTREC allow you to produce meaningful representations of SMF date values.

New TM1, TM2, TM3 and TM4 formats of OUTFIL OUTREC allow you to produce meaningful representations of SMF time values.

A new TRAN=LTOU option of OUTFIL OUTREC allows you to change lowercase EBCDIC letters anywhere in a field to uppercase EBCDIC letters.

A new TRAN=UTOL option of OUTFIL OUTREC allows you to change uppercase EBCDIC letters anywhere in a field to lowercase EBCDIC letters.

A new TRAN=ALTSEQ option of OUTFIL OUTREC allows you to change characters anywhere in a field according to the ALTSEQ translation table in effect.

A new M26 edit mask of OUTFIL OUTREC allows you to edit numeric values according to the pattern ST...T (S is the sign and T is a significant digit).

The maximum position for the end of a field used with `OUTFIL INCLUDE` and `OMIT` has been raised to 32752.

New `DATEn`, `DATEn(c)` and `DATEnP` options of `OUTFIL INCLUDE` and `OMIT` allow you to compare fields with various formats to the date of your DFSORT run.

A positive decimal number (`n` or `+n`) can now be compared to a binary (BI) field for `OUTFIL INCLUDE` and `OMIT`.

A new `DATENS=(abc)` option of `OUTFIL HEADERx` and `TRAILERx` allows you to insert the date of your DFSORT run into your headers and trailers without a separator. Separators are included for existing options `DATE` and `DATE=(abcd)`.

A new `TIMENS=(ab)` option of `OUTFIL HEADERx` and `TRAILERx` allows you to insert the time of your DFSORT run into your headers and trailers without a separator. Separators are included for existing options `TIME` and `TIME=(abc)`.

A new `SKIP=0L` option of `OUTFIL SECTIONS` allows you to suppress blank lines between sections on the same page.

INREC and OUTREC Enhancements

New `DATEn`, `DATEn(c)` and `DATEnP` options of `INREC` and `OUTREC` allow you to insert the date of your DFSORT run into your records in various forms.

New `TIMEn`, `TIMEn(c)` and `TIMEnP` options of `INREC` and `OUTREC` allow you to insert the time of your DFSORT run into your records in various forms.

New `DT1`, `DT2` and `DT3` formats of `INREC` and `OUTREC` allow you to produce meaningful representations of SMF date values.

New `TM1`, `TM2`, `TM3` and `TM4` formats of `INREC` and `OUTREC` allow you to produce meaningful representations of SMF time values.

A new `TRAN=LTOU` option of `INREC` and `OUTREC` allows you to change lowercase EBCDIC letters anywhere in a field to uppercase EBCDIC letters.

A new `TRAN=UTOL` option of `INREC` and `OUTREC` allows you to change uppercase EBCDIC letters anywhere in a field to lowercase EBCDIC letters.

A new `TRAN=ALTSEQ` option of `INREC` and `OUTREC` allows you to change characters anywhere in a field according to the `ALTSEQ` translation table in effect.

A new `M26` edit mask of `INREC` and `OUTREC` allows you to edit numeric values according to the pattern `ST..T` (S is the sign and T is a significant digit).

SUM Enhancement

The `VLSHRT` option can now be used to process short `SORT` and `MERGE` control fields, and `INCLUDE` and `OMIT` compare fields, when a `SUM` statement is specified. Records with short `SUM` summary fields are left unsummed.

SORT and MERGE Enhancements

The maximum position for the end of a field used with `SORT` and `MERGE` has been raised to 32752.

The maximum length for an `AQ` or `AC` field used with `SORT` and `MERGE` has been raised to 4092.

INCLUDE and OMIT Enhancements

New DATEn, DATEn(c) and DATEnP options of INCLUDE and OMIT allow you to compare fields with various formats to the date of your DFSORT run.

The maximum position for the end of a field used with INCLUDE and OMIT has been raised to 32752.

A positive decimal number (n or +n) can now be compared to a binary (BI) field for INCLUDE and OMIT.

Other Enhancements

The IBM-supplied default for ICEMAC option DYNSPC has been changed from 32 megabytes to 256 megabytes. This allows DFSORT to sort more data by default with dynamically allocated work space when the input file size is unknown.

DYNSPC can now be specified as a run-time option. This allows you to adjust the dynamically allocated work space for individual sort applications for which the input file size is unknown.

DFSORT now issues message ICE118I, and sets on bit 3 of SMF type-16 record field ICEFSZFL, when the input file size is unknown for a sort application. This identifies sort applications which may benefit from a FILSZ=En or DYNSPC=n value.

DFSORT copy can now be used when ICEGENER is called by a program that uses an alternate SYSIN ddname with DUMMY. This can result in improved performance for RACF's IRRUT200 utility when ICEGENER is installed as a replacement for IEBGENER.

When all of the input for variable-length records is supplied through an E15 or E32 exit, and RECFM is not specified for the output data set, DFSORT now sets the output RECFM as blocked rather than unblocked, if that allows the use of the system-determined optimum block size for output.

A hyphen (-) can now be used as the second or subsequent character in a DFSORT symbol name.

DFSORT now issues message ICE042A and terminates if the QNAME option is specified on an output DD statement.

Twentieth Edition, July 2000

New Programming Support for Release 14 (PTFs - July, 2000)

Larger Tape Block Sizes with OS/390 R10

DFSORT can now use tape data sets with block sizes greater than 32760 bytes for input and output, providing improved performance and tape utilization.

DFSORT can now select system-determined optimum block sizes greater than 32760 bytes for tape output data sets, if allowed by the BLKSZLIM value in effect. Installation and run-time options SDB=INPUT (the new IBM-supplied default), SDB=LARGE (new), SDB=YES (or its alias SDB=SMALL) and SDB=NO allow you to control DFSORT's use of system-determined block sizes, including block sizes greater than 32760 bytes for tape output data sets.

DFSORT's ICEGENER, like IEBGENER, will use SDB=value parameters you specify.

New Device Support

DFSORT now supports FICON channels.

DFSORT now recognizes the IBM 2105 control unit and provides optimizations for it appropriate for cached control units, such as cache fast write for work data sets.

DFSORT now treats work data sets on emulated 3390 model 9 devices as desirable choices with respect to performance. DFSORT continues to treat work data sets on real 3990 model 9 devices as undesirable choices with respect to performance.

DFSMSrmm-Managed Tapes

DFSORT can now automatically obtain accurate input file size information for DFSMSrmm-managed tapes. This can result in improved sort performance and more accurate dynamic workspace allocation.

DFSORT can now automatically obtain input and output attributes such as RECFM, LRECL and BLKSIZE for DFSMSrmm-managed tapes.

VSAM Processing

DFSORT now supports the VSAM extended addressability function for extended format VSAM data sets, which provides the capability of VSAM data sets larger than four gigabytes.

DFSORT now allows an empty VSAM input data set to be accepted and processed as having zero records. New installation option VSAMEMT=YES/NO and run-time options VSAMEMT and NVSAMEMT allow you to control the processing of empty VSAM input data sets.

DFSORT now allows a VSAM output data set defined with REUSE to be processed as a NEW or MOD data set. New installation option RESET=YES/NO and run-time options RESET and NORESET allow you to control the processing of VSAM output data sets.

DFSORT now allows a VSAM data set defined with REUSE to be sorted in-place. New installation option VSAMIO=YES/NO and run-time options VSAMIO and NOVAMIO allow you to control the use of the same VSAM data set for input and output.

HFS Files

DFSORT now supports Hierarchical File System (HFS) files for input and output.

Productivity and Easier Migration from other Sort Products

By default, DFSORT now uses the L3, OUTREC or INREC record length as the SORTOUT LRECL when the SORTOUT LRECL is unavailable. New installation option SOLRF=YES/NO and run-time options SOLRF and NOSOLRF allow you to control the use of the reformatted record length for the SORTOUT LRECL.

DFSORT now provides enhanced capabilities for OUTFIL, including:

- Padding short records with blanks, by default, when converting variable-length records to fixed-length records.
- Creation of edited counts and subcounts in trailers.

DFSORT now provides enhanced capabilities for INREC and OUTREC that were previously only available with OUTFIL OUTREC, including:

- Hexadecimal display.
- Transformation of dates.
- Editing of numeric fields.
- Lookup and change.

DFSORT now provides enhanced capabilities for INREC, OUTREC and OUTFIL OUTREC, including:

- Conversion of numeric fields to other numeric formats.
- Editing of transformed dates.
- Conversion of transformed dates to numeric formats.
- Creation of sequence numbers.

DFSORT now allows INCLUDE/OMIT fields that extend beyond variable-length records (that is, short fields) to be compared as if they were padded with binary zeros. This allows both the short and non-short fields to be compared. New installation option VLSCMP=YES/NO and run-time options VLSCMP and NOVLSCMP allow you to control the use of binary zero padding for short control fields.

DFSORT now allows -0 and +0 to be treated as unsigned (that is, the same) for processing by DFSORT statements INCLUDE, INREC, MERGE, OMIT, OUTFIL, OUTREC and SORT. New installation option SZERO=YES/NO and run-time options SZERO and NOSZERO allow you to select signed or unsigned zero processing for these statements.

DFSORT now allows -0 and +0 to be treated as unsigned (that is, the same) for processing by ICETOOL operators DISPLAY, OCCUR, SELECT and UNIQUE. New option UZERO allows you to select unsigned zero processing for these operators.

DFSORT now always determines or assigns the record type (F or V) when the input is VSAM, or when an E15 or E32 exit supplies all of the input records.

DFSORT now allows an LRECL greater than 80 for SYSIN, SORTCNTL and DFSPARM; only the first 80 bytes of each control statement are used.

DFSORT now allows variable-length output records that are longer than the output data set LRECL to be truncated to the LRECL. New installation option VLLONG=YES/NO and run-time options VLLONG and NOVLLONG allow you to control the truncation of long variable-length output records.

DFSORT now accepts and processes the following as EXEC/DFSPARM PARM options: COBEXIT=COB1/COB2, NOVERIFY, NOVLSHRT, NZDPRINT, OVFLO=RC0/RC4/RC16, PAD=RC0/RC4/RC16, SPANINC=RC0/RC4/RC16, TRUNC=RC0/RC4/RC16, VERIFY, VLSHRT and ZDPRINT.

DFSORT now accepts and processes the following as OPTION statement options: EXITCK=STRONG/WEAK, WRKREL and WRKSEC.

DFSORT now accepts and processes Y2PAST=s/f as a SORT and MERGE statement option.

DFSORT now accepts, and processes or ignores as appropriate, many more aliases for DFSORT options.

DFSORT now accepts and processes the following as 24-Bit Parmlist control statement entry codes: X'10', X'16', X'18' and X'20' through X'29'.

Nineteenth Edition, September 1998

New Programming Support for Release 14

Symbols for Fields and Constants

DFSORT now provides a simple and flexible method for using symbols in DFSORT and ICETOOL statements. You can define and use a symbol for any field or constant that is recognized in a DFSORT control statement or ICETOOL operator. This makes it easy to create and reuse collections of symbols (that is, mappings) for your frequently used data.

In addition, you can obtain and use collections of DFSORT symbols created specifically for data associated with other products (for example, RACF, DFSMSrmm and DCOLLECT) or by your site.

DFSORT symbols can increase your productivity by automatically providing the positions, lengths and formats of the fields and the literals, numbers, and bit flags of the constants, associated with the particular records you are processing with DFSORT or ICETOOL.

Improvements in Performance, Capacity and Storage Usage

Blockset copy and merge applications can now use storage above 16MB virtual, providing improved performance and virtual storage constraint relief.

Blockset copy and merge modules will now reside above 16MB virtual, providing virtual storage constraint relief.

DFSORT can now handle a significantly larger number of INCLUDE and OMIT conditions.

DFSORT can now handle a significantly larger number of SUM fields.

The upper limit for the number of JCL and dynamically allocated work data sets that can be specified and used by DFSORT's Blockset technique has been raised from 100 to 255. The use of more work data sets increases the maximum amount of data DFSORT can process in a single sort application. Any valid ddname of the form SORTWKdd or SORTWKd can now be used for DASD work data sets (for example, SORTWK01, SORTWK3, SORTWK2, SORTWK#5, SORTWKA, SORTWKXY and so on).

The upper limit for the number of input data sets that can be specified and used for a Blockset merge application has been raised from 16 to 100. The use of more merge input data sets increases the maximum amount of data DFSORT can process in a single merge application.

Time-of-Day Option Controls

New time-of-day installation modules (ICETD1-4) allow different sets of installation defaults to be used, based on the day and time DFSORT applications run. Each environment installation module (ICEAM1-4) can enable one or more time-of-day installation modules. This capability allows new levels of control for installation

defaults. For example, larger storage, hiperspace and data space limits could be used only for batch program-invoked DFSORT applications that run off-shift during the week, and all weekend.

Repackaging

The product has been repackaged to simplify installation and customization:

- IBM's DFSORT, DFSMSdfp, and MVS/DFP teams have simplified the process of replacing IEBGENER with ICEGENER. You now only need to apply a DFSMS or DFP PTF that supplies an alias of "IEBGENR" for IEBGENER and place ICEGENER with an alias of "IEBGENER" ahead of IEBGENER in the system's search order for programs.
- The number of FMIDs has been reduced from 10 to 3.
- The number of libraries required to install DFSORT has been reduced from 40 to 26.
- DFSORT R14 now supports a single installation of the product for both resident and nonresident features. This allows you to decide how to use DFSORT independent of the installation method, thus reducing the number of decisions you have to make at installation time.
- All FMIDs in DFSORT R14 can be installed together, including the FMIDs for both English and Japanese messages and panels.

OUTFIL Processing Enhancements

OUTFIL now supports creation of multiple output records using the fields of the input record. This allows you to split each record into pieces, include a field in more than one record, include different fields in different records, and more.

OUTFIL now supports processing of variable-length input records which are too short to contain all specified OUTFIL OUTREC fields. OUTFIL's new VLFILL=byte operand can be used to replace missing bytes in OUTFIL OUTREC fields with the specified fill byte so the filled fields can be processed.

ICETOOL Enhancement

A new DISCARD(savedd) operand of ICETOOL's SELECT operator allows you to save the records that are not selected, in the savedd data set. Thus, in one pass, you can create an outdd data set with the records that meet your specified criteria, and a savedd data set with the records that do not meet your specified criteria. DISCARD(savedd) can be used to save the records discarded by ALLDUPS, NODUPS, HIGHER(x), LOWER(y), EQUAL(v), FIRST or LAST.

Installation and Run-Time Option Enhancements

A new p% value for the EXPRES, EXPOLD, and EXPMAX installation options and the HIPRMAX installation and run-time options is now available. p% can be used to vary the limit DFSORT calculates for the corresponding option as a percentage of the configured expanded storage on the system at run time. If the configured expanded storage on a system changes, p% will cause a corresponding change in the run-time limit calculated for the corresponding option. When sharing DFSORT installation options between systems, such as in a sysplex, p% can be used to tailor the limit DFSORT calculates for the corresponding option to the system on which the application runs.

A new SPANINC installation and run-time option allows you to specify what you want DFSORT to do if it detects incomplete spanned records. This gives you control over the action (continue by eliminating incomplete spanned records and recovering valid records, or terminate), type of message (informational or error) and return code (0, 4 or 16) for incomplete spanned records.

A new OVFL0 installation and run-time option allows you to specify what you want DFSORT to do when BI, FI, PD or ZD summary fields overflow. This gives you control over the action (continue or terminate), type of message (informational or error) and return code (0, 4 or 16) for summary overflow.

A new PAD installation and run-time option allows you to specify what you want DFSORT to do when the SORTOUT LRECL is larger than the SORTIN/SORTINnn LRECL. This gives you control over the action (continue or terminate), type of message (informational or error) and return code (0, 4 or 16) for LRECL padding.

A new TRUNC installation and run-time option allows you to specify what you want DFSORT to do when the SORTOUT LRECL is smaller than the SORTIN/SORTINnn LRECL. This gives you control over the action (continue or terminate), type of message (informational or error) and return code (0, 4 or 16) for LRECL truncation.

The IBM-supplied default for ICEMAC option DSA has been changed from 16MB to 32MB.

The IBM-supplied default for ICEMAC option GENER has been changed from IEBGENER to IEBGENR.

The maximum value for ICEMAC option OVERRGN has been changed from 64KB to 16128KB.

Other Enhancements

New messages ICE178I and ICE179A provide information about reallocation of VIO work data sets.

The option-in-effect messages (ICE127I-ICE133I) are now printed for Blockset copy and merge applications.

The user exit address constant can now be passed to E32 user exits for Blockset merge applications.

Null segments in variable spanned input records are now processed by DFSORT and no longer result in termination. A null segment means that there are no more segments in the block.

OS/390 and MVS/ESA Only

DFSORT Release 14 only supports the OS/390 and MVS/ESA environments. MVS/XA and VIRTDSP processing for MVS/XA are no longer supported.

Part 1. Planning for Installation

Chapter 1. Getting Ready to Install DFSORT	3
DFSORT on the Web	3
DFSORT FTP Site	3
What Is on the Tape?	3
Storage Administrator Examples	3
About the DFSORT Program Directory	4
Chapter 2. System Planning Considerations	5
Where DFSORT Fits on Your System	5
Software Requirements	5
Driving System Requirements	5
Target System Requirements	6
Required Storage Devices	6
Machine Requirements	6
Choosing Where to Install DFSORT	6
Required Distribution Libraries	7
Required Target Libraries	7
Should You Run DFSORT Resident?	8
Testing DFSORT	9
Making the Target Libraries Accessible	9
Running More Than One DFSORT Release at a Time	9
Replacing a Previous DFSORT Release	10
Required Migration Actions	10
Additional Migration Actions for CBPDO Installation	12
Allocating DFSORT Libraries	12
Verifying that DFSORT Was Installed Correctly	12
ICEJCLJ and ICEINVJ Verification	13
ICETOOLJ Verification	14
ICECSRTJ Verification	15
ICECOBJ Verification	16
Deleting Private Libraries after Testing	16
Verifying the SVC Was Installed Correctly	16
DFSORT Storage Considerations	16
Required Main Storage	16
Effects of Varying Main Storage	17
Required Storage for Running DFSORT Resident	18
Required Hiperspace Storage	18
Tailoring Main Storage	18
Options That Allocate Main Storage	21
Options That Limit Storage Used by DFSORT	21
Options That Reserve Storage for System Use	22
Storage Considerations	22
Tailoring Storage for Hipersorting	28
Installing DFSORT in Permanent Libraries	30
Replacing IEBGENER with the ICEGENER Facility	30
Verifying ICEGENER Installation	32
Discontinuing Use of ICEGENER as a Replacement for IEBGENER	33
Chapter 3. Planning for DFSORT Panels	35
Where DFSORT Panels Fits on Your System	35
Selecting Distribution and Target Libraries	35
Required Distribution Libraries	36
Required Target Libraries	36

Allocating Storage for Installation	37
Making the DFSORT Panels Available After Installation	37
Accessing DFSORT Panels	37
Example: TSO Logon Procedure	38
Example: Allocation CLIST	38
Example: Allocating DFSORT ISPF Libraries Using LIBDEF	39
Testing DFSORT Panels	40

Chapter 1. Getting Ready to Install DFSORT

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 DFSORT Program Directory in conjunction with this book 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 book will help you customize DFSORT for your site.

DFSORT on the Web

For articles, online books, news, tips, techniques, examples, and more, visit the DFSORT/MVS home page at:

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

DFSORT FTP Site

You can obtain DFSORT articles and examples via anonymous FTP to:

<ftp.software.ibm.com/dfsort/mvs/>

What Is on the Tape?

The DFSORT product tape contains:

- SMP/E control statements
- DFSORT modules
- DFSORT macros
- Source code for the modules containing installation defaults and message texts
- Sample jobs for installing DFSORT
- DFSORT Panels CLISTS
- DFSORT Panels skeletons
- DFSORT Panels tables
- English and Japanese versions of the DFSORT Panels.

DFSORT is distributed on a tape that you can install using IBM® System Modification Program Extended (SMP/E). This tape contains functional modification identifiers (FMIDs) that identify DFSORT to SMP/E. See the DFSORT Program Directory for details on the installation of DFSORT. See *SMP/E Reference* and *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 DFSMSHsm™, DFSMSrmm™, DCOLLECT, and SMF. These sample jobs can be found in the ICESTGEX member of the SICESAMP library after DFSORT is installed. You can

Getting Ready to Install DFSORT

also download these sample jobs from the DFSORT FTP site listed in “DFSORT FTP Site” on page 3. These sample jobs show some of the many ways that 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

DFHSMEX2

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

RMMEX1

DFSMSrmm Example 1: SMF audit report

RMMEX2

DFSMSrmm Example 2: Create ADDVOLUME commands

About the DFSORT Program Directory

The DFSORT Program Directory provides step-by-step instructions for installing DFSORT.

A printed copy of the DFSORT Program Directory is shipped when you order DFSORT as a separate product. If you are installing DFSORT as part of a ServerPac or SystemPac, the Program Directory is available for reference, and can be found in softcopy in the CPAC.PGMDIR data set. See the \$INDEXPD member of the data set to find the member name for the DFSORT Program Directory.

Chapter 2. System Planning Considerations

This chapter supplements the step-by-step installation procedures in the Program Directory 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 chapter also reviews:

- The questions you should answer before deciding whether to run DFSORT resident or nonresident
- The steps you must follow if you are keeping previous releases of DFSORT
- The storage requirements for main, expanded 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 following operating systems:

- z/OS[®]
- OS/390[®]

Additionally, DFSORT runs under z/OS or OS/390 when it is running as a guest under z/VM[®] or VM/ESA[®].

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 or OS/390 available.

System Planning Considerations

Target System Requirements

To use DFSORT, you must have a supported release of z/OS or OS/390 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.
- Direct-access storage for the distribution and target libraries (storage estimates are given under "DFSORT Storage Considerations" on page 16). You can choose any direct access storage device supported by your operating system.

To *use* DFSORT, you need:

- Devices supported by QSAM or VSAM for input/output.
- Direct-access or tape devices for work storage. You can choose any direct access storage device or tape device supported by your operating system.

For work data set storage during DFSORT processing, high-speed direct-access devices such as the 3390 offer the best performance. For more information on how different devices affect performance, see *Application Programming Guide*.

Machine Requirements

DFSORT can run in any hardware environment that supports the software described in "Software Requirements" on page 5. If Japanese National Language Support (JSM1G11) is selected, workstations or terminals used to display panels or panel messages must support the double byte character set (DBCS).

Choosing Where to Install DFSORT

You can install DFSORT in either the same target and DLIB zones as your operating system, or in separate zones. We recommend that you install it in the same zones 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.

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 DASD, 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.
-----------------	------------------

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
---------------	--------------------

SORTLPA	The LPA-eligible DFSORT Blockset technique modules.
----------------	---

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

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.
-----------------	------------------

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.
-----------------	---

System Planning Considerations

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?”.

If you plan to use standard DFSORT cataloged 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 47 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 DASD 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 18.

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. You can also test DFSORT if you installed it using separate libraries and SMP/E zones.

You should 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.

To run DFSORT nonresident, you can either:

- Add SICELINK and SORTLPA (in that order) to the link list, or
- 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 DASD 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.

Note: 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 SA22-7591* and *z/OS MVS Initialization and Tuning Reference SA22-7592*.

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

Running More Than One DFSORT Release at a Time

You can run more than one release of DFSORT at the same time on the same system. However, if you do this, note that:

- Each release must be installed using different data set names (for example, using a different high-level qualifier).
- Only one release can be run resident at a time.
- Only one release can be accessed at a time, using the system search order for programs.

System Planning Considerations

- Only one release can use the default SVC number.

For information about using a different SVC number for DFSORT, see the "Change the DFSORT SVC Name" section of the Program Directory.

The following recommendations can help you install the SVC for this DFSORT release so that it does not replace the SVC of a prior release:

- If you installed the DFSORT SVC for the previous release using the default SVC 109 module IGX00017, install the DFSORT SVC for the current release using the alternate SVC 109 module IGX00038.

In this case, you must specify ICEMAC option SVC=(109,ALT) and rename the SVC module to IGX00038.

- You should install the DFSORT SVC for the current release using default SVC 109 module IGX00017 under the following circumstances:
 - If you did not install the DFSORT SVC for your previous release
 - If you installed the DFSORT SVC for your previous release using an SVC number other than 109 (that is, you used a number between 200-255)
 - If you installed the DFSORT SVC for your previous release using alternate SVC 109 module IGX00038.

If any of these circumstances are true, *no changes* to the default installation procedures are required.

- Alternatively, you can install the DFSORT SVC for the current release using an SVC number other than 109 (a number between 200-255) as long as the number you choose was *not used* to install the DFSORT SVC for the previous release. In this case, you must specify ICEMAC option SVC=n and rename the SVC module appropriately.

Replacing a Previous DFSORT Release

You can install DFSORT R14 to replace a previous release of DFSORT in two ways:

- By installing it from a CBPDO or product tape
- By installing it as part of a ServerPac order.

DFSORT's packaging has changed significantly in R14. The first part of this section details migration actions you must take, regardless of which method above you choose. The second part of this section details some additional actions you must take if you choose to install DFSORT R14 using CBPDO.

Required Migration Actions

- Update the link list to remove obsolete libraries.

DFSORT's load libraries were restructured to simplify installation and allow a single installation path for running DFSORT resident and nonresident. The following libraries were consolidated into the SICELINK library. Remove them from the link list if they are present:

 - ICEDGTL
 - ICEISPL
 - SICELNKN (introduced by DFSORT R13 PTF UQ00530)
 - SORTPV
- Update the link list if you are running DFSORT R14 nonresident.

Place the SORTLPA library after the SICELINK library in the link list.

System Planning Considerations

Note: This action is not required if you are migrating from DFSORT R13 and have PTF UQ00530 installed.

- Update ISPF concatenations if you are using ISPF or ISMF Panels for DFSORT R14.

The packaging of these libraries was changed to allow both English and Japanese panels to be installed together. Remove the obsolete data sets for the associated DD names, as shown below, from ISPF concatenations in logon procedures, allocation CLISTs, and EXECs:

- SYSPROC
 - ICEDGTC (moved to SICECLIB)
 - ICEISPC (moved to SICECLIB)
- ISPLLIB
 - ICEISPL (moved to SICELINK)
 - ICEDGTL (moved to SICELINK)
- ISPSLIB
 - ICEDGTS (moved to SICESLIB)
 - ICEISPS (moved to SICESLIB)
- ISPPLIB
 - ICEDGTP (moved to SICEPENU or SICEPJPN)
 - ICEISPP (moved to SICEPENU or SICEPJPN)
- ISPMLIB
 - ICEDGTM (moved to SICEMENU or SICEMJPN)
 - ICEISPM (moved to SICEMENU or SICEMJPN)
- ISPTLIB
 - ICEDGTT (moved to SICETLIB)
 - ICEISPT (moved to SICETLIB)

Concatenate each new DFSORT ISPF data set to each associated ISPF DD name as shown below:

- SYSPROC
 - SICECLIB
- ISPSLIB
 - SICESLIB
- ISPTLIB
 - SICETLIB
- ISPMLIB
 - SICEMENU (English) or SICEMJPN (Japanese)
- ISPPLIB
 - SICEPENU (English) or SICEPJPN (Japanese)
- Review and change your DFSORT installation options.

Review your installation defaults for the previous release using the ICETOOL DEFAULTS operator, noting any changes you made to the IBM-supplied defaults. Review the installation defaults for DFSORT R14 and determine if you want to keep any of your previous changes and make additional changes. If so, change the defaults as directed in “Changing the DFSORT Installation Defaults” on page 48.
- Update cataloged procedures.

System Planning Considerations

The cataloged procedures for DFSORT have changed because of the restructuring of the libraries. You should take one of the following actions:

- If you use the DFSORT procedures as shipped, you can copy the new cataloged procedures to a procedure library in the JES concatenation used for batch jobs, replacing the old procedures.
- If you previously modified the DFSORT procedures, you can examine the changes made to the R14 procedures and make equivalent changes in your modified procedures as appropriate.
- You can allocate empty libraries for SORTPV, and SICELNKN if applicable, and continue to use your existing procedures. If your site makes extensive use of DFSORT JCL that does not use procedures, you should seriously consider this option. If you choose this option, you can allocate each library using RECFM=U, LRECL=0, BLKSIZE=(any block size), and SPACE=(TRK,(1,0,1)).

Additional Migration Actions for CBPDO Installation

Prior releases of DFSORT were installed into a greater number of libraries, and some libraries were renamed when DFSORT R14 was repackaged to allow all of its features to be installed at the same time. The old libraries must be available for SMP/E processing even though they will not be used by DFSORT R14. When you finish installing DFSORT R14, you can choose to delete these libraries and their SMP/E DDDEFs.

After you finish installing DFSORT R14, and are no longer using the SMP/E global zone for the prior release of DFSORT, you can clean up the global zone to reduce DASD space used for the prior release. The prior release's PTFs can be REJECTed to save DASD space, and the global zone FMID entries for the prior release's FMIDs can be deleted so that new PTFs for the prior release are not RECEIVED.

For more information about replacing a prior release of DFSORT with DFSORT R14, and about global zone cleanup, see the Program Directory.

Allocating DFSORT Libraries

The ICEALLOC sample job allocates space for all DFSORT target and distribution libraries. The libraries for the DFSORT panels and their messages are only required if you will install panels. All other libraries are required. For more information about the libraries that are required, see the Program Directory 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 below 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

System Planning Considerations

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.
- ICECOBJ, if you have VS COBOL II, to verify that DFSORT is installed correctly for use with COBOL E15 and E35 exits. To run this job, the VS COBOL II run-time library must be available.

Modify this job appropriately if you have COBOL for MVS & VM, COBOL for OS/390 & VM, or COBOL for z/OS and OS/390.

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 14, which shows what the first 20 records and last 10 records of the 360 output records look like.

System Planning Considerations

```
*****000001*****ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789*****
*****000037*****ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789*****
*****000073*****ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789*****
*****000109*****ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789*****
*****000145*****ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789*****
*****000181*****ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789*****
*****000217*****ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789*****
*****000253*****ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789*****
*****000289*****ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789*****
*****000325*****ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789*****
*****000002*****BCDEFGHIJKLMNOPQRSTUVWXYZ0123456789A*****
*****000038*****BCDEFGHIJKLMNOPQRSTUVWXYZ0123456789A*****
*****000074*****BCDEFGHIJKLMNOPQRSTUVWXYZ0123456789A*****
*****000110*****BCDEFGHIJKLMNOPQRSTUVWXYZ0123456789A*****
*****000146*****BCDEFGHIJKLMNOPQRSTUVWXYZ0123456789A*****
*****000182*****BCDEFGHIJKLMNOPQRSTUVWXYZ0123456789A*****
*****000218*****BCDEFGHIJKLMNOPQRSTUVWXYZ0123456789A*****
*****000254*****BCDEFGHIJKLMNOPQRSTUVWXYZ0123456789A*****
*****000290*****BCDEFGHIJKLMNOPQRSTUVWXYZ0123456789A*****
*****000326*****BCDEFGHIJKLMNOPQRSTUVWXYZ0123456789A*****
.
.
.
*****000036*****9ABCDEFGHIJKLMNPQRSTUVWXYZ012345678*****
*****000072*****9ABCDEFGHIJKLMNPQRSTUVWXYZ012345678*****
*****000108*****9ABCDEFGHIJKLMNPQRSTUVWXYZ012345678*****
*****000144*****9ABCDEFGHIJKLMNPQRSTUVWXYZ012345678*****
*****000180*****9ABCDEFGHIJKLMNPQRSTUVWXYZ012345678*****
*****000216*****9ABCDEFGHIJKLMNPQRSTUVWXYZ012345678*****
*****000252*****9ABCDEFGHIJKLMNPQRSTUVWXYZ012345678*****
*****000288*****9ABCDEFGHIJKLMNPQRSTUVWXYZ012345678*****
*****000324*****9ABCDEFGHIJKLMNPQRSTUVWXYZ012345678*****
*****000360*****9ABCDEFGHIJKLMNPQRSTUVWXYZ012345678*****
```

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 the RELEASE is shown as “14.0” for ICEAM1-4 and ICETD1-4 in the DFLTS output of step TOOLRUN. “Listing ICEMAC Defaults” on page 119 shows an example of how the complete DFLTS output will look if you have not changed any DFSORT installation options. If you have changed DFSORT installation options, the output should reflect the options 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 15.
4. Comparing the LIST1 output of step TOOLRUN to Figure 3 on page 15. (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 15. (Carriage control characters are shown in position 1; do not be concerned if you do not see them.)

System Planning Considerations

```

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
DXYZ123ABC           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                  000000000000001
J69                  000000000000006
J82                  000000000000003
L92                  000000000000005
  
```

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.

```

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

Figure 5. Expected CSORT SORTOUT Output from ICECSRTJ

System Planning Considerations

ICECOBJ Verification

Verify that the ICECOBJ job ran correctly by:

1. Checking that the condition code for each step is 0.
2. Comparing the SORTOUT output of step SORT to Figure 6.

```
AAA99AAAAAAAAA AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAA88AAAAAAAAA AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAA77AAAAAAAAA AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAA66AAAAAAAAA AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAA55AAAAAAAAA AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAA44AAAAAAAAA AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAA33AAAAAAAAA AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAA22AAAAAAAAA AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAA11AAAAAAAAA AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAA00AAAAAAAAA AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

Figure 6. Expected SORT SORTOUT Output from ICECOBJ

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 DASD devices. 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 DASD devices

In either case, message ICE187I is issued if the SVC is installed at the wrong level. If you are using a cached DASD 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 *DFSORT Tuning Guide*.

Required Main Storage

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

System Planning Considerations

For best performance, it is strongly recommended that you 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.

You can limit the amount of main storage DFSORT allocates by using the SIZE option at installation time on ICEMAC or at run time on an EXEC statement, at run time with the OPTION control statement option MAINSIZE, 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 6, “Controlling Resources Using ICEIEXIT” on page 127 for more information. This limit applies to the total storage allocated both above and below 16MB virtual. Storage above 16MB 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 16MB virtual for sort, copy and merge applications. You can see how much storage is used exclusively for OUTFIL processing, both above and below 16MB 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.

System Planning Considerations

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 1 shows approximate kilobytes (KB) of virtual memory required in the LPA for DFSORT Release 14.

Table 1. KB of Virtual Memory Required in LPA for Release 14

	Below	Above
Running DFSORT resident	110KB	758KB

Note: These approximations are based on the initial shipment of the distribution tape for Release 14. The size required in the LPA might increase with the installation of a Program Temporary Fix applied to DFSORT Release 14.

Required Hiperspace Storage

DFSORT requires expanded storage to use Hipersorting[®] unless you are running in 64-bit real mode where expanded storage no longer exists; in this environment, hiperspaces are backed by central storage. DFSORT will use Hipersorting when a sufficient amount of expanded or 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 and expanded storage allocations.

See the discussion of the HIPRMAX parameter in “ICEMAC Parameters” on page 62 for a more detailed discussion of Hipersorting.

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 the ICEMAC installation macro (see Chapter 5, “Changing the Installation Defaults—ICEMAC Macro” on page 55).

At run time, 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 6, “Controlling Resources Using ICEIEXIT” on page 127. The EXEC statement, the OPTION statement, and using an EFS program are described in *Application Programming Guide*.

Table 2 summarizes how storage-related installation defaults can be overridden at run time.

System Planning Considerations

Table 2. How Storage-Management Installation Defaults Can Be Overridden at Run Time

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

Note: ¹ The MAINSIZE parameter of the OPTION statement is equivalent to the SIZE parameter of ICEMAC.

The primary *system* storage limits for an application are set by your site's IEFUSI exit, or by default to 1MB below and 32MB above 16MB 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 16MB virtual, but will minimize its use of storage below 16MB virtual.

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

Table 3 on page 20 summarizes the DFSORT default parameters that you can set at installation time.

System Planning Considerations

Table 3. 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 16MB 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 16MB virtual storage above the REGION value that DFSORT can use.	Limited by IEFUSI exit, or 64KB by default.
	MAXLIM	Sets upper limit to amount of storage available below 16MB virtual.	Always used for Blockset sort applications. Used for copy and Blockset merge applications unless the use of more storage below 16MB 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 16MB 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 16MB virtual.	Only used when SIZE=MAX is in effect.
	ARESALL	Specifies amount of storage reserved for system use above 16MB virtual.	None.
	RESINV	Specifies amount of storage reserved for the invoking program below 16MB virtual.	Only used when SIZE=MAX is in effect.
	ARESINV	Specifies amount of storage reserved for the invoking program above 16MB 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 16MB 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 books 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.

System Planning Considerations

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. OUTFIL 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. OUTFIL data set buffers can be placed above 16MB 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 16MB virtual, and what the total amount of storage was. Use these values to calculate how much storage was used above and below 16MB virtual, and compare that amount to the amount you requested.

Options That Allocate Main Storage

Use the SIZE parameter in the ICEMAC installation macro 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 parameter).

Use the OVERRGN parameter during installation 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 (64KB 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 set in ICEMAC. MAINSIZE, which is the equivalent of SIZE, can be specified on the OPTION program control statement, and an equivalent parameter 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 16MB virtual is used for Blockset sort, copy, and merge applications. TMAXLIM is the limit for the total storage above and below 16MB virtual, when SIZE=MAX is in effect. MAXLIM is the limit for the storage below 16MB 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 16MB 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 parameter 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 (440KB), depending on the requirements of your site. Lowering MINLIM below

System Planning Considerations

440KB can degrade performance for some applications. This performance degradation can become more pronounced and affect more applications the more MINLIM is lowered below 440KB.

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 parameters RESALL, ARESALL, RESINV, and ARESINV work together to reserve storage for system use.

The RESALL parameter 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 16MB 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 16MB virtual.

The RESINV parameter 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 parameter 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 16MB virtual for user exits controlled by the invoking program. Normally the site limit to total storage is sufficient. There is storage above 16MB 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 16MB virtual. However, many of the concepts illustrated also apply when Blockset is not selected and only storage below 16MB virtual is used.

SIZE

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

System Planning Considerations

Lowering SIZE below 4MB can degrade performance of some applications. This performance degradation can become more pronounced and affect more applications the more SIZE is lowered below 4MB.

The MAXLIM and REGION (plus OVERRGN) values limit the amount of storage used below 16MB virtual. Any storage that cannot be obtained below 16MB virtual will be obtained above 16MB virtual by DFSORT. See Figure 7 for examples of how SIZE affects DFSORT storage.

DFSORT Storage Considerations
Total Storage System Will Allow

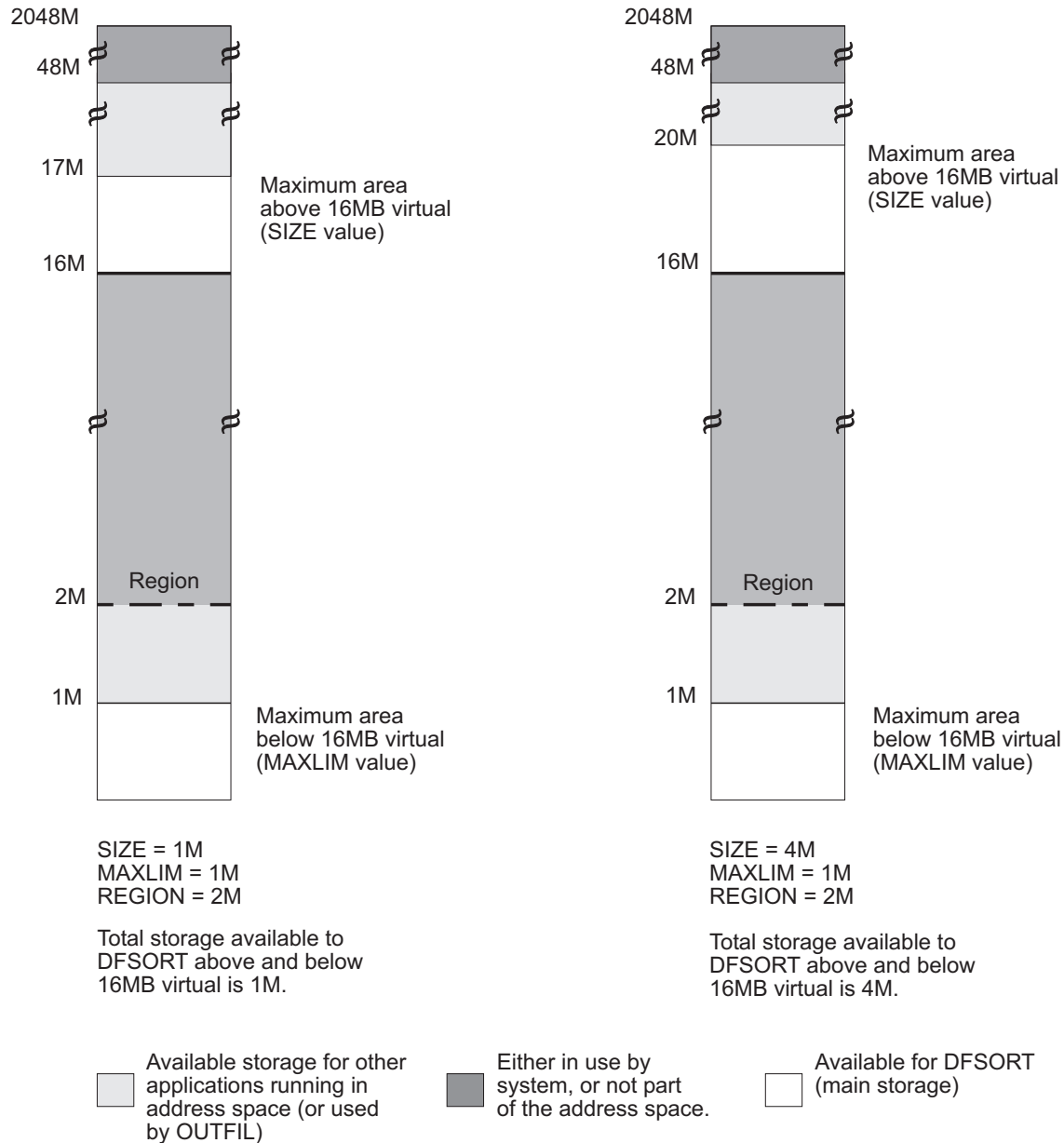


Figure 7. Storage Comparisons for SIZE and REGION

TMAXLIM

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

System Planning Considerations

You can increase or decrease the setting of this parameter from the default of 4194304 bytes (4MB), depending on the requirements of your site. Lowering TMAXLIM below 4MB can degrade performance for some applications. This performance degradation can become more pronounced and affect more applications the more TMAXLIM is lowered below 4MB.

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 8 for examples of how TMAXLIM affects DFSORT storage.

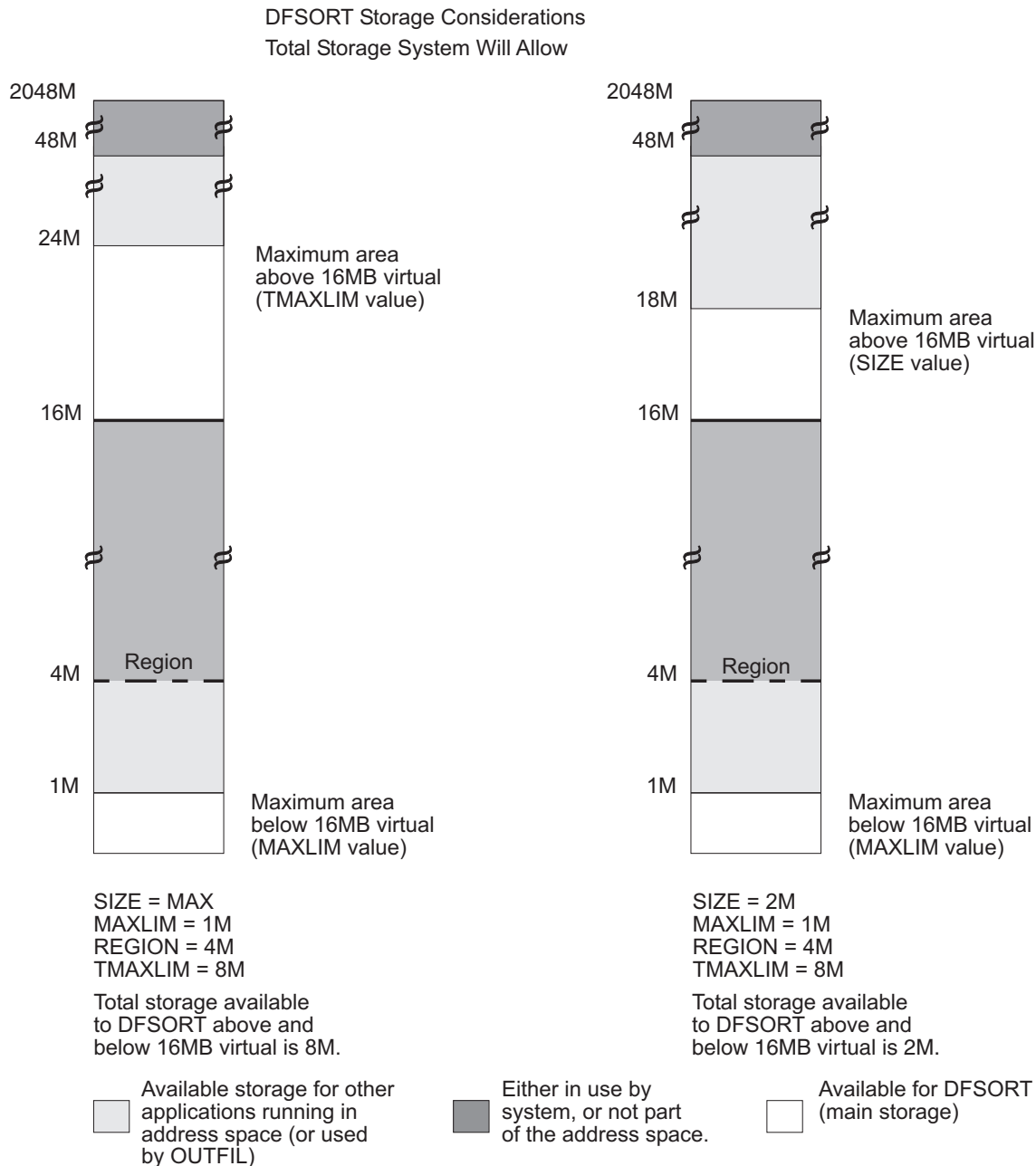


Figure 8. Effect of TMAXLIM Option on Storage

TMAXLIM is ignored when SIZE is not equal to MAX.

MAXLIM

If MAXLIM is greater than the REGION size, DFSORT uses storage below 16MB virtual only up to the size of REGION plus OVERRGN. You can increase or decrease the setting of this parameter from the default of 1048576 bytes (1MB), depending on the requirements of your site. Lowering MAXLIM below 1MB can degrade performance for some applications. This performance degradation can become more pronounced and affect more applications the more MAXLIM is lowered below 1MB. See Figure 9 on page 26 for examples of how MAXLIM affects DFSORT storage.

Normally, DFSORT uses less than 1MB of storage below 16MB virtual, so this parameter 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 16MB virtual.

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

System Planning Considerations

DFSORT Storage Considerations Total Storage System Will Allow

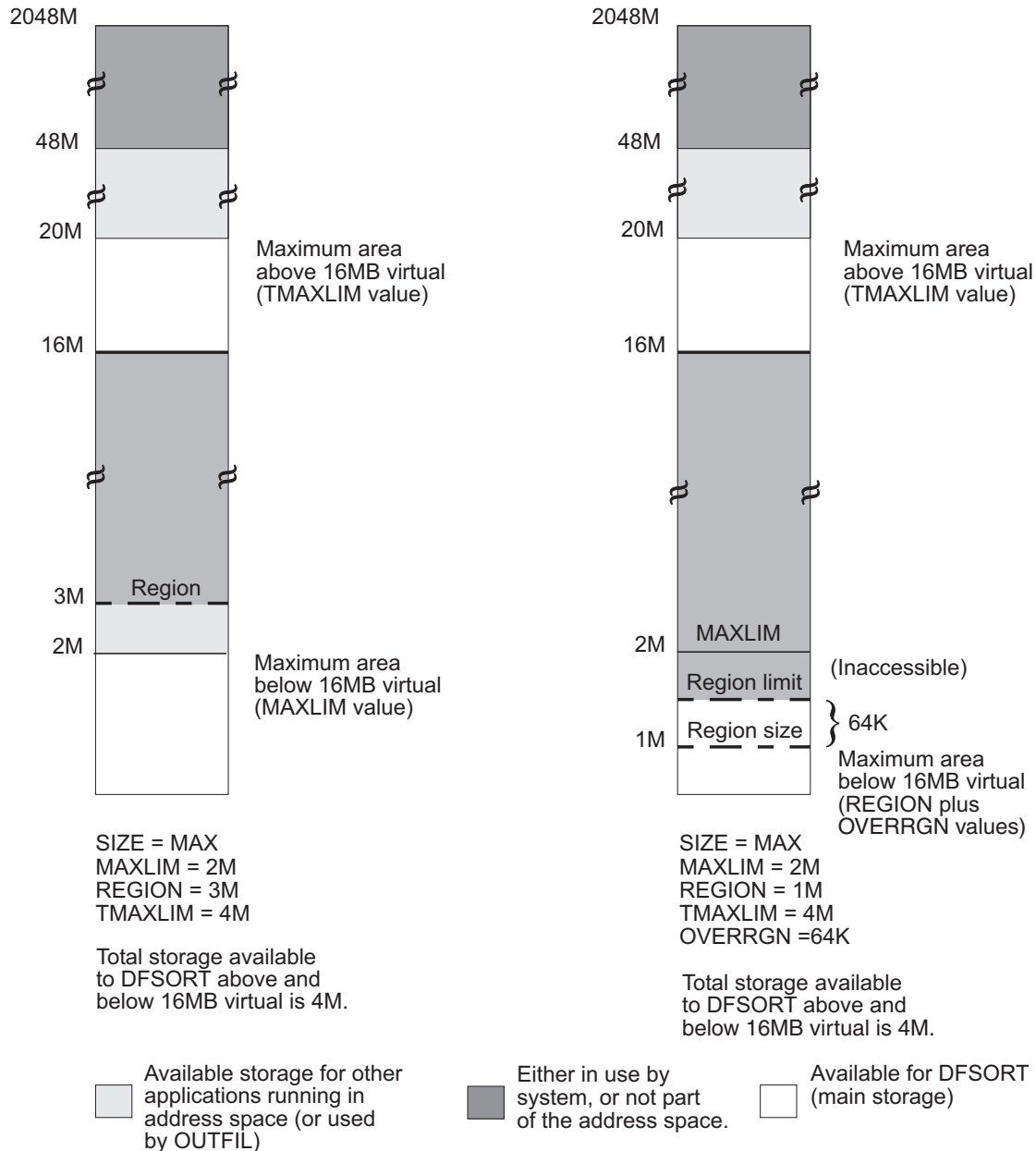


Figure 9. Effects of MAXLIM Option on Storage

When REGION is greater than MAXLIM, DFSORT uses storage below 16MB 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

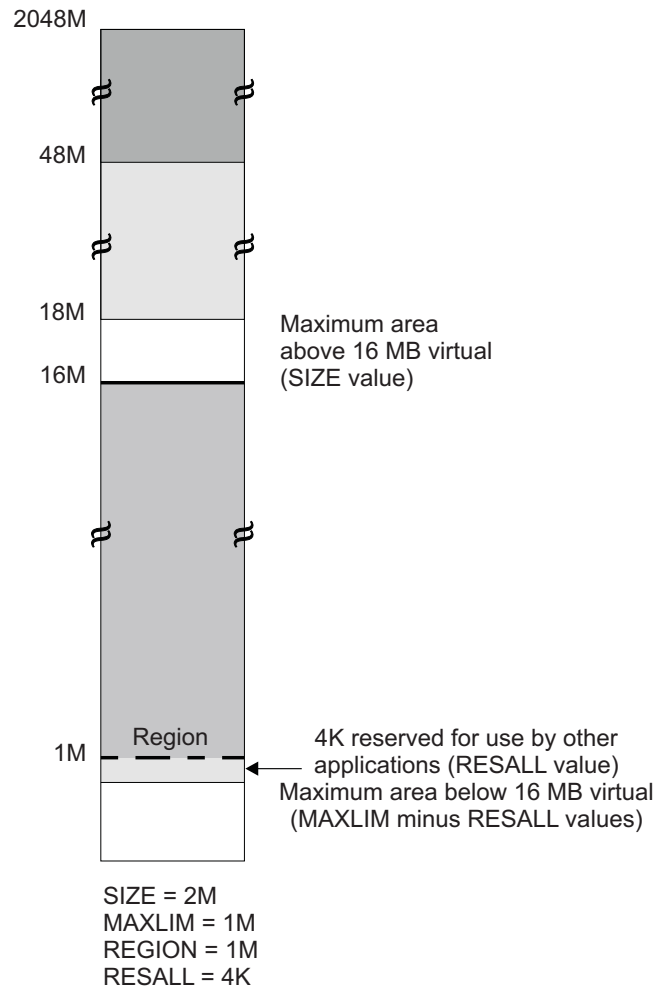
RESALL reserves storage below 16MB virtual for system use, when SIZE=MAX is in effect. The amount for RESALL is released below 16MB virtual, and an equivalent amount can then be taken above 16MB virtual.

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

System Planning Considerations

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 10 for an example of how RESALL affects DFSORT storage.

DFSORT Storage Considerations
Total Storage System Will Allow



Total storage available to DFSORT above and below 16MB virtual is 2M

- Available storage for other applications running in address spaces (or used by OUTFIL)
- Either in use by system, or not part of the address space
- Available for DFSORT (main storage)

Figure 10. Effects of RESALL Option on Storage

RESINV

RESINV reserves storage below 16MB 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.

System Planning Considerations

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 (16KB) 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 10 on page 27 for RESALL also apply to RESINV.

ARESALL

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

The default for ARESALL is 0 bytes.

ARESINV

ARESINV is similar to RESINV, except that it applies only to storage above 16MB virtual. The ARESINV parameter is used only when DFSORT is program invoked. It specifies the amount of storage to be reserved for the invoking program above 16MB virtual. This parameter is normally not needed because of the large amount of storage available above 16MB 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 64KB of additional storage to be obtained; this default can be changed in an IEFUSI exit.

The default and recommended setting for this parameter for directly invoked jobs is 65536 bytes (64KB). To avoid problems caused by incorrect use of the RESALL and RESINV parameters, the default setting for program invoked jobs is 16384 bytes (16KB). If your site uses RESALL and RESINV correctly or is installing DFSORT for the first time, it is recommended that you increase the OVERRGN value for program invoked applications to 65536 bytes. This is especially important if you are running benchmarks.

Tailoring Storage for Hipersorting

You can tailor the way DFSORT uses storage for Hipersorting applications to meet the requirements of your site. Hipersorting is a DFSORT feature that uses Hiperspaces™ as a way to store intermediate data in expanded storage, or in central storage for 64-bit real mode, instead of DASD work data sets. DFSORT provides several parameters to control the use of Hipersorting. This section describes the default settings you can establish for these parameters by using the ICEMAC installation macro (see Chapter 5, "Changing the Installation Defaults—ICEMAC Macro" on page 55).

Expanded storage, or central storage for 64-bit real mode, 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. "Available" storage is the storage used to back new Hiperspace data, and consists of two types:

1. Free storage. This is storage that is not being used by any application.

System Planning Considerations

2. Old storage. This is storage that is being used by other applications, but whose data has been unreferenced for a sufficiently long period such that the system migrates it to auxiliary storage to make room for new Hiperspace data.

The following DFSORT installation parameters are available to control the use of Hiperspace storage:

EXPMAX

Specifies the maximum total amount of available storage to be used at any one time by all Hipersorting applications

EXPOLD

Specifies the maximum total amount of old storage to be used at any one time by all Hipersorting applications

EXPRES

Specifies the minimum amount of available storage to be reserved for use by non-Hipersorting applications

HIPRMAX

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

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

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

The ICEIEXIT routine is described in Chapter 6, "Controlling Resources Using ICEIEXIT" on page 127. The EXEC statement, the OPTION statement, and using an EFS program are described in *Application Programming Guide*.

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

Hiperspace usage by individual DFSORT applications can be limited through 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.

Since expanded storage and central storage are system resources, 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 DASD work data sets if there is insufficient available storage to back new Hiperspace data.

System-wide storage usage by one or more Hipersorting applications is limited by the EXPMAX, EXPOLD, and EXPRES installation options. EXPMAX limits the total amount of available storage used by all Hipersorting applications in a system. EXPOLD limits the total amount of old storage used by all Hipersorting applications in a system. EXPRES reserves a specified amount of available storage for use by non-Hipersorting applications.

System Planning Considerations

The following are general guidelines for setting the DFSORT installation options that limit Hiperspace usage:

1. Avoid setting HIPRMAX to anything but OPTIMAL at installation time. DFSORT performs better when it has access to more Hiperspaces. It is more efficient to limit total storage usage by all Hipersorting applications (through EXPMAX, EXPOLD, and EXPRES) than it is to impose a limit on each individual Hipersorting application.
2. Setting EXPRES to a small value is the most efficient way to control the total storage used by all Hipersorting applications. When there is a sudden increased need for storage by non-Hipersorting applications, a small value for EXPRES can help to ensure there is sufficient available storage to meet the demand.
3. Avoid setting EXPOLD to a value other than MAX unless large migration of old storage data to auxiliary storage, as a result of Hipersorting activity, creates a problem at your site. If such migration does cause a problem, set EXPOLD to the maximum amount of migration that your site can sustain at any one time.
4. If you want to restrict DFSORT to using no more than a specified amount of storage for all Hipersorting activity, then EXPMAX is the only way to accomplish it with DFSORT parameters. This is a much better solution than trying to use the system to do storage isolation for DFSORT applications, since EXPMAX still allows full use of storage by non-DFSORT applications.
5. Each of the installation options EXPMAX, EXPOLD, EXPRES, and HIPRMAX fulfills a different role, and each works somewhat independently of the others. Setting EXPMAX=MAX, EXPOLD=MAX, EXPRES=0, and HIPRMAX=OPTIMAL (the defaults) permits the maximum amount of Hipersorting, subject to available storage and other concurrent Hipersorting activity. Setting either EXPMAX=0, EXPRES=MAX, or HIPRMAX=0 disables Hipersorting (though HIPRMAX can be overridden at run time). In general, setting EXPMAX, EXPOLD, and HIPRMAX to large values (and EXPRES to a small value) allows a generous amount of Hipersorting, while setting either EXPMAX, EXPOLD, or HIPRMAX to a small value (or EXPRES to a large value) allows only a limited amount of Hipersorting.

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 Program Directory, 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 environment installation module (ICEAM2 or ICEAM4) or time-of-day installation module (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, when appropriate. If you do not supply an SDB=value parameter, ICEGENER will use your site's DFSORT installation default for SDB, when appropriate (the

System Planning Considerations

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 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 SA22-7592*.

Notes:

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 module 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

System Planning Considerations

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 ICEMAC option 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 ICEMAC option 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 ICEMAC option 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.

System Planning Considerations

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 above):

- 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.

Chapter 3. Planning for DFSORT Panels

This chapter discusses what you should consider before using DFSORT Panels. You can read this chapter either before or after installing DFSORT Panels.

DFSORT Panels supports interactive panels for a **subset** of the functions available with DFSORT. Although interactive panels for functions such as DFSORT's Year 2000 features and the OUTFIL statement are **not** provided, you can use Free Form Entry panels to specify complete DFSORT control statements for these functions.

Where DFSORT Panels Fits on Your System

You can add an option for DFSORT to an ISPF panel in any of these ways:

- Use the sample ISPF panels (ISR@390, ISR@390S, and ISR@390U) to add options for all ISPF panels, including those for DFSORT. For information about how to do this, see the *z/OS Program Directory* or *OS/390 Program Directory*.
- If you are installing DFSORT as part of a ServerPac, use the ServerPac-supplied CustomPac Master Applications Menu. For more information, see *ServerPac: Installing Your Order*.
- You can modify an ISPF or ISMF panel to add an option for DFSORT.

You can also use the DFSORT Panels without adding them as an option to another panel, by issuing the ICEPANEL TSO/E command. If you choose to modify an ISPF or ISMF panel, you can choose among:

- The ISPF/PDF Primary Option Menu (ISR@PRIM)
- The ISPF Master Application Menu (ISP@MSTR)
- Another menu panel, like the Utility Selection Panel (ISRUTIL) or the ISMF Primary Options Menu (DGTSMMD1).

Notes:

1. The ISMF SORTREC line operator will be available on the ISMF Data Set List panel (DGTLP11) after you install DFSORT.
2. The ISPF panels are in the SISPPENU library. The ISMF panels are in the DGTPLIB library.

To install DFSORT/ISPF Panels and DFSORT/ISMF Panels, you must install either z/OS or OS/390.

If you want to install an option on an ISPF or ISMF panel, refer to *Interactive System Productivity Facility (ISPF) Dialog Tag Language Guide and Reference* and *Interactive System Productivity Facility (ISPF) Planning and Customizing for OS/390 Version 2 Release 5*.

Selecting Distribution and Target Libraries

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

Planning for DFSORT Panels

Required Distribution Libraries

The distribution libraries should all be private libraries.

To install DFSORT Panels, you need the following distribution libraries:

DDname	Description
AICECLIB	CLIST library.
AICEMENU	U. S. English message library.
AICEPENU	U. S. English panel library.
AICEMJPN	Japanese message library.
AICEPJPN	Japanese panel library.
AICESLIB	Skeleton library.
AICETLIB	Table library.

Notes:

- You install AICECLIB, AICESLIB, and AICETLIB using FMID HSM1G00.
- You install AICEMENU and AICEPENU using FMID JSM1G01.
- You install AICEMJPN and AICEPJPN using FMID JSM1G11.
- You can use just JSM1G01, just JSM1G11, or both.

Required Target Libraries

DFSORT Panels target libraries should be private libraries. See the "DASD Storage Requirements" section of the Program Directory for further information about the amount of space needed for these private libraries.

The following target libraries must be installed for the DFSORT Panels:

DDname	Description
SICECLIB	CLIST library.
SICEMENU	U. S. English message library.
SICEPENU	U. S. English panel library.
SICEMJPN	Japanese message library.
SICEPJPN	Japanese panel library.

SICESLIB

Skeleton library.

SICETLIB

Table library.

Notes:

- You install the panel load modules into SICELINK using FMID HSM1G00.
- You install SICECLIB, SICESLIB, and SICETLIB using FMID HSM1G00.
- You install SICEMENU and SICEPENU using FMID JSM1G01.
- You install SICEMJPN and SICEPJPN using FMID JSM1G11.
- You can use just JSM1G01, just JSM1G11, or both.

Allocating Storage for Installation

The Program Directory contains tables detailing the amounts of storage needed for DFSORT Panels installation.

Making the DFSORT Panels Available After Installation

To use DFSORT Panels, DFSORT's ISPF data sets (including those supplied by one of the English or Japanese Panels) must be allocated. You can use the DFSORT Panels by adding them to an ISPF or ISMF panel as a selectable option, or by issuing the ICEPANEL command from TSO.

Accessing DFSORT Panels

After DFSORT and DFSORT Panels (just English panels, just Japanese panels, or both) have been installed, you must make the DFSORT libraries accessible. Concatenate the DFSORT ISPF libraries to the ISPF DD names as shown in Figure 15, by adding these libraries to your logon procedures or allocation CLISTs. Alternatively, you can use ISPF's LIBDEF service to make the DFSORT libraries accessible.

Even though you may have installed both English and Japanese panels, you can use only one or the other at any given time; that is, you can access English panels or Japanese panels in a logon procedure, but not both.

Note: If you allocated the SORTLIB library with a name other than SYS1.SORTLIB, you must modify the ICEQFG04 CLIST in the SICECLIB library to run DFSORT Panels applications in the foreground. Change the name of the SORTLIB library to the name you used.

Table 4. DD Names for DFSORT Libraries

DD Name	DFSORT Library	Comments
ISPLLIB	SICEPENU or SICEPJPN	For U.S. English ISPF panels only For Japanese ISPF panels only
ISPMLIB	SICEMENU or SICEMJPN	For U.S. English ISPF messages only For Japanese ISPF messages only
ISPSLIB	SICESLIB	ISPF Skeletons (Required)
ISPTLIB	SICETLIB	ISPF Tables (Required)
SYSPROC	SICECLIB	CLISTs (Required)

Planning for DFSORT Panels

Example: TSO Logon Procedure

If you choose to add DFSORT's ISPF libraries to your logon procedures, add them as shown below:

```
/** ISPF LOGON PROCEDURE INCLUDING DFSORT LIBRARIES
/**
//SORTISPF EXEC PGM=IKJEFT01,REGION=0M,DYNAMNBR=200,PARM=' ISPF'
//SYSHLP DD DSN=SYS1.HELP,DISP=SHR /* TSO/E */
// DD DSN=ISP.SISPHLP,DISP=SHR /* ISPF */
//SYSPROC DD DSN=SYS1.SICECLIB,DISP=SHR /* DFSORT */
// DD DSN=ISP.SISPCLIB,DISP=SHR /* ISPF */
// DD DSN=SYS1.DGTCLIB,DISP=SHR /* ISMF */
//ISPEXEC DD DSN=ISP.SISPEXEC,DISP=SHR /* ISPF */
//ISPMLIB DD DSN=SYS1.SICEMENU,DISP=SHR /* DFSORT ENU */
/** DD DSN=SYS1.SICEMJPN,DISP=SHR /* OR DFSORT JPN,NOT BOTH*/
// DD DSN=ISP.SISPMENU,DISP=SHR /* ISPF */
// DD DSN=SYS1.DGTMLIB,DISP=SHR /* ISMF */
//ISPPLIB DD DSN=SYS1.SICEPENU,DISP=SHR /* DFSORT ENU */
/** DD DSN=SYS1.SICEPJPN,DISP=SHR /* OR DFSORT JPN,NOT BOTH*/
// DD DSN=ISP.SISPPENU,DISP=SHR /* ISPF */
// DD DSN=SYS1.DGTPLIB,DISP=SHR /* ISMF */
//ISPPLIB DD DSN=SYS1.SICESLIB,DISP=SHR /* DFSORT */
// DD DSN=ISP.SISPLIB,DISP=SHR /* ISPF */
// DD DSN=SYS1.DGTSLIB,DISP=SHR /* ISMF */
//ISPPLIB DD DSN=SYS1.SICETLIB,DISP=SHR /* DFSORT */
// DD DSN=ISP.SISPTENU,DISP=SHR /* ISPF */
// DD DSN=SYS1.DGTTLIB,DISP=SHR /* ISMF */
//ISPCLT1 DD DISP=NEW,UNIT=SYSALLDA,SPACE=(CYL,(1,1)),
// LRECL=80,BLKSIZE=800,RECFM=FB
//ISPCLT2 DD DISP=NEW,UNIT=SYSALLDA,SPACE=(CYL,(1,1)),
// LRECL=80,BLKSIZE=800,RECFM=FB
//ISPCLT3 DD DISP=NEW,UNIT=SYSALLDA,SPACE=(CYL,(1,1)),
// LRECL=80,BLKSIZE=800,RECFM=FB
//SYSRINT DD TERM=TS
//SYSIN DD TERM=TS
```

Figure 11. Sample TSO Logon Procedure

Example: Allocation CLIST

As an alternative or supplement to modifying JCL logon procedures, you can also choose to allocate DFSORT Panels libraries, ISPF libraries, and ISMF libraries in a CLIST, as shown in the example CLIST below.

```

ALLOC F(ISPMLIB) DS(+
    'SYS1.SICEMENU' + /* SYS1.SICEMJPN for Japanese messages */
    'ISP.SISPMENU' + /* ISP.SISPMJPN for Japanese messages */
    'SYS1.DGTMLIB' +
) REUSE
ALLOC F(ISPPLIB) DS(+
    'SYS1.SICEPENU' + /* SYS1.SICEPJPN for Japanese panels */
    'ISP.SISPPENU' + /* ISP.SISPPJPN for Japanese panels */
    'SYS1.DGTPLIB' +
) REUSE
ALLOC F(ISPSLIB) DS(+
    'SYS1.SICESLIB' +
    'ISP.SISPSENU' +
    'SYS1.DGTSLIB' +
) REUSE
ALLOC F(ISPTLIB) DS(+
    'SYS1.SICETLIB' +
    'ISP.SISPTENU' +
    'SYS1.DGTTLIB' +
) REUSE
ALLOC F(SYSPROC) DS(+
    'SYS1.SICECLIB' +
    'ISP.SISPCLIB' +
    'SYS1.DGTCLIB' +
) SHR REU

ALLOC F(SYSHELP) DS(+
    'SYS1.HELP' +
    'ISP.SISPHELP' +
) SHR REU

```

Figure 12. Sample CLIST used to ALLOCATE the DFSORT Panels Libraries

Example: Allocating DFSORT ISPF Libraries Using LIBDEF

Some installations prefer to defer the allocation of some ISPF libraries to speed logon processing, and perform the necessary allocations when an application is chosen using ISPF's LIBDEF service. For more information about LIBDEF, see *z/OS ISPF Services Guide SC34-4819*. Figure 13 shows an example of a CLIST that uses LIBDEF to allocate DFSORT's ISPF libraries. Because LIBDEF cannot be used to allocate data sets for DD name SYSPROC, DFSORT's CLISTs must be allocated in the logon procedure or by using the TSO/E ALLOCATE command in a CLIST.

Once you have written a CLIST, it must be placed in a library concatenated to DD name SYSPROC and called by the DFSORT option you add to an ISPF or ISMF panel, as discussed in "Where DFSORT Panels Fits on Your System" on page 35.

```

PROC 1 ZCMD
ISPEXEC LIBDEF ISPPLIB DATASET ID('SYS1.SICEPENU') /*U. S. English panels*/
/*'SYS1.SICEPJPN' for Japanese panels */
ISPEXEC LIBDEF ISPMLIB DATASET ID('SYS1.SICEMENU') /*U. S. English messages*/
/*'SYS1.SICEMJPN' for Japanese messages */
ISPEXEC LIBDEF ISPSLIB DATASET ID('SYS1.SICESLIB') /*skeletons*/
ISPEXEC LIBDEF ISPTLIB DATASET ID('SYS1.SICETLIB') /*tables*/
/* ALLOC F(SYSPROC) DA('SYS1.SICECLIB') SHR */
ISPEXEC SELECT PGM(ICEPMD01) NOCHECK PARM(&ZCMD)

```

Figure 13. Sample CLIST (LDEF) Using LIBDEF to Allocate Data Sets


```

                                SORT ENTRY PANEL                                Page 1 of 2
COMMAND ==>>

SPECIFY ONE OR MORE INPUT DATA SET(S):
  INPUT DSN  1 ==>> SORT.SAMPIN

  (USE DSN 2 TO 12 FOR CONCATENATED DATA SETS)
  INPUT DSN  2 ==>>
  INPUT DSN  3 ==>>
  INPUT DSN  4 ==>>
  INPUT DSN  5 ==>>
  INPUT DSN  6 ==>>
  INPUT DSN  7 ==>>
  INPUT DSN  8 ==>>
  INPUT DSN  9 ==>>
  INPUT DSN 10 ==>>
  INPUT DSN 11 ==>>
  INPUT DSN 12 ==>>

  VERIFY INPUT DSN ==>> Y   (Y or N)

USE ENTER TO CONTINUE;
USE HELP COMMAND FOR HELP; USE END COMMAND TO EXIT.

```

Figure 15. Selecting Data Sets on the Sort Entry Panel, Page 1

6. Specify the input data set (INPUT DSN) SORT.SAMPIN (see Figure 15) and press **ENTER** to continue to Page 2.
7. On Page 2 of the Sort Entry panel, specify the name of the output data set, SORT.SAMPOUT. Use the default values displayed in the next 2 fields. Choose **1** for foreground execution, use the default values displayed in the work data set space fields, and set the three options on the bottom of the panel to **N** (see Figure 16 on page 42).

Planning for DFSORT Panels

```

                                SORT ENTRY PANEL                                Page 2 of 2
COMMAND ==>

SPECIFY EXISTING OR NEW OUTPUT DATA SET:
  OUTPUT DSN ==> SORT.SAMPOUT
  ALLOCATE NEW OUTPUT DATA SET ==> N (Y or N)
  VERIFY OUTPUT DSN      ==> Y (Y or N)

SPECIFY EXECUTION ENVIRONMENT ==> 1
  1 FOREGROUND (Execute as TSO job)
  2 BACKGROUND (Submit or save job)

SPECIFY WORK DATA SET SPACE:
  PRIMARY  QUANTITY ==> 5 (in Cylinders)
  SECONDARY QUANTITY ==> 5 (in Cylinders)

VIEW CONTROL STATEMENTS SELECTION PANEL ==> N (Y or N)

REVIEW CONTROL STATEMENTS BEFORE EXECUTION ==> N (Y or N)

ENTER SORT CONTROL STATEMENT IN FREE FORM ==> N (Y or N)

USE ENTER TO CONTINUE;
USE HELP COMMAND FOR HELP; USE END COMMAND TO RETURN TO PAGE 1.

```

Figure 16. Specifying the Output Data Set on the Sort Entry Panel, Page 2

8. Press **ENTER**.

The next panel is the Sort Statement Data Entry panel, where you specify the values to sort.

```

                                SORT STATEMENT DATA ENTRY                                ROW 1 OF 4
COMMAND ==>                                SCROLL ==> PAGE

SPECIFY VALUES FOR ONE OR MORE SORT FIELDS.

OPTIONALLY SPECIFY LINE COMMANDS TO EDIT SORT FIELDS
(I Insert, D Delete, R Repeat, M Move, C Copy, A After, B Before).

(USE ENTER TO CONTINUE, UP/DOWN COMMANDS TO SCROLL,
HELP COMMAND FOR HELP, END COMMAND TO SAVE AND EXIT.)

  LINE      SORT
  COMMAND  FIELD NO.  POSITION  LENGTH  FORMAT  SORT
  .....   ..  ..  ..  ..  ..  ..
  .....   1    ==> 1    ==> 75    ==> CH    ==> A
  .....   2    ==>    ==>    ==> CH    ==> A
  .....   3    ==>    ==>    ==> CH    ==> A
  .....   4    ==>    ==>    ==> CH    ==> A
          ***** END OF SORT FIELDS *****

```

Figure 17. Organizing the Records on the Sort Statement Data Entry Panel

9. Enter a **1** in the Position column, a **75** in the Length column, and make sure that Format is set to **CH**, and Sort Order is set to **A** (see Figure 17).
10. Press **ENTER** twice to submit your job.
A screen of TSO messages and DFSORT program messages appears.

Planning for DFSORT Panels

11. Check to see that your sort job completed.
 - Make sure that the SORTLIB reflects the correct library, as shown in “Example: Allocation CLIST” on page 38.
 - Make sure the correct SICELINK appears on the execution CLIST.
 - Make sure the correct input and output data sets were accessed.
 - Make sure the last line of the TSO output is the “ICE052I END OF DFSORT” message, followed by a line of three asterisks (***)).

If there are problems with any of these items, your sort will be unsuccessful.

12. Press **ENTER** to return to DFSORT.

Depending on the success of your sort, the sort Entry Panel appears on your screen with the message “DFSORT Successful” or “DFSORT Unsuccessful” in the upper right-hand corner of the panel (see Figure 18).

```
COMMAND ==>>                                SORT ENTRY PANEL                                DFSORT SUCCESSFUL

SPECIFY ONE OR MORE INPUT DATA SET(S):
INPUT DSN 1 ==>> SORT.SAMPIN

(USE DSN 2 TO 12 FOR CONCATENATED DATA SETS)
INPUT DSN 2 ==>>
INPUT DSN 3 ==>>
INPUT DSN 4 ==>>
INPUT DSN 5 ==>>
INPUT DSN 6 ==>>
INPUT DSN 7 ==>>
INPUT DSN 8 ==>>
INPUT DSN 9 ==>>
INPUT DSN 10 ==>>
INPUT DSN 11 ==>>
INPUT DSN 12 ==>>

VERIFY INPUT DSN ==>> Y (Y or N)

USE ENTER TO CONTINUE;
USE HELP COMMAND FOR HELP; USE END COMMAND TO EXIT.
```

Figure 18. Sort Entry Panel, Page 1, After a Successful Sort

If your sort job did not process correctly (“DFSORT Unsuccessful”) and there was no problem with the input/output data sets or libraries, then the DFSORT Panels are not correctly installed. Contact your IBM representative for assistance.

Part 2. Customizing DFSORT

Chapter 4. Planning for Customization	47
Customizing the Standard DFSORT Procedures	47
Changing the DFSORT Installation Defaults	48
Cultural Environment Considerations	49
Invoking an EFS Routine Automatically	50
Using Initialization and Termination Exits	50
Initialization Exits	50
Termination Exits	51
Modifying Message Texts	51
For the Base Product	51
Coding the Changes to ICEMSGS	52
Receiving and Applying SMP/E Usermods	53
For DFSORT Panels	54
Finding the Message You Want to Change	54
Making the Change	54
Chapter 5. Changing the Installation Defaults—ICEMAC Macro	55
Migrating to DFSORT from Other Sort Products	56
Tailoring Main Storage with ICEMAC	57
Tailoring Hiperspace Storage with ICEMAC	57
How to Code the ICEMAC Macro	57
ICEMAC Syntax Diagram	58
ICEMAC Parameters	62
Examples of ICEMAC Macros	114
ICEMAC JCL - Batch Direct Invocation Defaults	114
ICEMAC INV - Batch Program Invocation Defaults	115
ICEMAC TSO - TSO Direct Invocation Defaults	116
ICEMAC TSOINV - TSO Program Invocation Defaults	117
ICEMAC TD1 - First Time-of-Day Module Defaults	118
ICEMAC Defaults	118
Listing ICEMAC Defaults	119
Chapter 6. Controlling Resources Using ICEIEXIT	127
What Run-Time Options Can ICEIEXIT Change?	127
What Installation- and Run-Time Information Can ICEIEXIT Examine?	128
Installation-time Information	128
Run-time Information	128
Writing and Installing ICEIEXIT	128
Interface Points for ICEIEXIT	129
Contents of General Register 1	129
Contents of the Installation-Time Options List	129
Contents of the Run-Time Options List	131
A Sample ICEIEXIT Routine	133
Chapter 7. Collecting Statistical Data	137
Why Use DFSORT's SMF Record?	137
Why Use an ICETEXIT?	137
Collecting Data from DFSORT's SMF Record	137
Controlling What DFSORT Passes to SMF	138
When DFSORT Cannot Pass a Record to SMF	138
SMF Record Structure	139
Mapping the SMF Record	139
Verifying Supported Record Sections	139

Accessing Multiple Input Data Set Sections	140
Accessing Multiple OUTFIL Data Set Sections	140
Identifying What Release Produced Your SMF Record	140
Collecting Data Using an ICETEXIT	141
Installing an ICETEXIT	144
Writing an ICETEXIT Routine	144
Controlling What DFSORT Passes to ICETEXIT	145
Mapping ICETEXIT Statistics	145
ICETEXIT Examples	146
A Nonexecutable ICETEXIT	146
An Executable ICETEXIT	151
Chapter 8. Improving Tape Processing with DFSMSrmm or ICETPEX	159
Information Passed by DFSMSrmm or ICETPEX	159
Installing an ICETPEX Routine	159
Writing an ICETPEX Routine	159
A Sample ICETPEX Routine	161

Chapter 4. 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 book discuss these topics in more detail. It describes planning considerations for:

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

For planning considerations for customizing the DFSORT Panels user interface, see *Panels Guide*.

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 ICEPROC.D 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 using ICEMAC.
- 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. See "Required Target Libraries" on page 36 for more information on using private or system libraries for SORTLIB.
- If your installation uses COBOL E15 or COBOL E35 exit routines frequently, you could concatenate the appropriate COBOL 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.

Planning for Customization

- If you are allocating temporary work data sets in the procedures, see the explanation of WRKREL and WRKSEC in “ICEMAC Parameters” on page 62.
- 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 5, “Changing the Installation Defaults—ICEMAC Macro” on page 55. These defaults are in effect if you install DFSORT without specifying your own defaults.

Many of the defaults can be changed at any time after installing DFSORT. 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 28 on page 119.

You can change the IBM-supplied defaults in several ways. You can use the ICEMAC macro to change the defaults for four different environment installation modules (ICEAM1–4) and four different time-of-day installation modules (ICETD1–4) (see Chapter 5, “Changing the Installation Defaults—ICEMAC Macro” on page 55 for details). Some of the defaults can also be overridden by using an EFS program or an ICEIEXIT routine.

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 *Application Programming Guide*.

For additional information on the ICEMAC macro and instructions on how to code it, read Chapter 5, “Changing the Installation Defaults—ICEMAC Macro” on page 55.

To change the IBM-supplied defaults:

1. Create an SMP/E usermod to modify the ICEMAC macro call in the source statement for any installation module (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. For a description of the IEBUPDTE syntax, see *Utilities*. Examples and rules for coding the ICEMAC macro are given in Chapter 5, “Changing the Installation Defaults—ICEMAC Macro” on page 55.

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 B, "ICEMAC and Message Module Messages" on page 177.

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 *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.

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:

- in ICEMAC
- 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, use Run-Time Library Services (RTLS), place heavily-used modules in

Planning for Customization

LPA, or use a JOBLIB or STEPLIB. For more information about making the SCEERUN library available for use, see *z/OS Language Environment Customization SA22-7564*.

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.

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 *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:

- In ICEMAC
- 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 book; 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 6, "Controlling Resources Using ICEIEXIT" on page 127 and Chapter 7, "Collecting Statistical Data" on page 137.

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 36 on page 135.

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 5, “Changing the Installation Defaults—ICEMAC Macro” on page 55.

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 146.

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 TEXTIT=YES installation option to activate your ICETEXIT, as described in Chapter 5, “Changing the Installation Defaults—ICEMAC Macro” on page 55.

Modifying Message Texts

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

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. You can also change DFSORT panels using one or more SMP/E usermods modeled on ICEMSREC.

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.

Planning for Customization

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 ICEMSG macro appears in Appendix A, "The ICEMSG Macro" on page 165.

The format of the ICEMSG macro is:

```
MACRO
ICEMSG
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 ICEMSG

To make changes to DFSORT messages, you must create an SMP/E usermod to modify the ICEMSG 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 ICEMSG in the AICEUSER library.

Remember that 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 19 shows how the text of messages ICE134I and ICE142I can be translated into French.

ICE134I:

NUMBER OF BYTES SORTED: n

NOMBRE D'OCTETS TRIES: n

ICE142I:

ddname NOT FOUND - SYSOUT USED

ddname NON TROUVE - SYSOUT UTILISE

Figure 19. Suggested Translations for Two ICEMSGS Messages

Figure 20 shows the changes made in the ICEMSGS macro. Appendix A, “The ICEMSGS Macro” on page 165 shows the original text of the complete ICEMSGS macro.

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

Figure 20. The ICEMSGS Macro with two 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 182.) You can *recover* any message that you render invalid by using:

- The ICEMSGS macro in Appendix A, “The ICEMSGS Macro” on page 165

or

- The copy of ICEMSGS in the DFSORT 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.

Planning for Customization

For DFSORT Panels

You must follow certain rules when modifying messages:

- Do not change the names of any of the variables contained in the messages.
- Do not change the message number.
- Short messages cannot exceed 24 characters.
- Long messages cannot exceed 78 characters.
- Message text can be entered in uppercase and lowercase, but the other fields in the message — the message number, variables, keywords, and the help panel ID — must be in uppercase.
- When you change the text of a message, you should change the corresponding message help panel.
- For more information about ISPF messages and how to code or modify them, see *z/OS ISPF Dialog Tag Language SC34-4824*.

Finding the Message You Want to Change

To find the message you want to change you need the message number listed at the top of each message help panel (see Figure 21).

```
HELP-----DFSORT PANELS MESSAGE-----HELP
COMMAND ==>>
MESSAGE NUMBER:  ICEDV020
SHORT MESSAGE:   DATA SET NOT CATALOGED
LONG MESSAGE:   A123456.DFS.SORTOUT
                NOT FOUND IN CATALOG
```

Figure 21. Message HELP Panel

Related short and long messages are stored together in members of the SICEMENU library (English messages) and members of the SICEMJPN library (Japanese messages). To determine where the message you want to change is stored, truncate the last digit of the message number. This will give you the member name. Thus, the message ICEDV020 is stored in ICEDV02 with other messages that begin with ICEDV02.

The help panel associated with each message is identified on the first line of that message on the HELP parameter. For example, the first line of message ICEDV020 contains HELP=ICEMDV20, and ICEMDV20 is the help panel for that message. The help panels can be found in the panel library for the same language as the messages they are associated with. The panel libraries are SICEPENU (for English) and SICEPJPN (for Japanese).

Making the Change

Once you have identified the member the message is stored in, you are ready to modify the message and save your change. For example, to change message ICEDV020, you would edit the message itself in member ICEDV02 and the related text in message help panel ICEMDV20.

Chapter 5. Changing the Installation Defaults—ICEMAC Macro

DFSORT allows you to maintain separate sets of installation defaults for four different environment installation modules (ICEAM1-4) and four different time-of-day installation modules (ICETD1-4). The set of defaults used for a particular run is determined by the environment for the run, the time-of-day of the run (that is, day and time) and the defaults in effect for each installation module.

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

For example, you could set up the ICETD1 module to use larger storage values, such as 48MB for DSA, 6MB 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.

The eight installation modules are as follows:

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

You can change the IBM-supplied DFSORT defaults during installation by using the ICEMAC macro (see “Changing the DFSORT Installation Defaults” on page 48 for more information). You can choose to establish separate defaults for any of the eight installation modules. The defaults for each installation module are completely independent of the defaults for any other installation module. Thus if you enable an

ICEMAC Macro

ICETDx module for one or more ICEAMx modules, be sure the ICETDx module includes the same default changes as the ICEAMx module, 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 *Application Programming Guide* for details of these override methods), or with ICEIEXIT (see Chapter 6, “Controlling Resources Using ICEIEXIT” on page 127), as appropriate.

You can list the installation defaults selected for all of the ICEAMx and ICETDx modules at any time using the ICETOOL DEFAULTS operator, as shown in Figure 28 on page 119.

This chapter explains how to code the ICEMAC macro to change parameters for the eight installation modules and shows examples of coding the macro for several of them. This chapter also lists the IBM-supplied defaults and explains all of the ICEMAC parameters in detail.

The ICEMAC macro can be found in the SICEUSER library. The source statement for any installation module can be found in the AICESRCE library. See “Changing the DFSORT Installation Defaults” on page 48 for the steps to follow to change the IBM-supplied defaults.

Migrating to DFSORT from Other Sort Products

If you are migrating to DFSORT, you should review the IBM-supplied ICEMAC 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 ICEMAC options, described later in this Chapter, can be used to change the way DFSORT works globally by default. The run-time options, described in *DFSORT Application Programming Guide*, can be used to override the ICEMAC options for specific jobs.

Table 5. Options That Can Ease Migration

ICEMAC Options	Run-Time Options
ABCODE=MSG/n	
DYNALOC=(d,n)	DYNALOC=(d,n)
DYNAUTO=YES/IGNWKDD/NO	DYNALOC=(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

Table 5. Options That Can Ease Migration (continued)

ICEMAC Options	Run-Time Options
VSAMEMT=YES/NO	VSAMEMT/NVSAMEMT
VSAMIO=YES/NO	VSAMIO/NOVSAMIO
ZDPRINT=YES/NO	ZDPRINT/NZDPRINT

Tailoring Main Storage with ICEMAC

You can use the following ICEMAC 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 parameters carefully before you install DFSORT. These parameters are described in this chapter. To see how the parameters relate to planning decisions, read "Tailoring Main Storage" on page 18.

Tailoring Hipspace Storage with ICEMAC

You can use ICEMAC to tailor the way DFSORT uses storage for Hipersorting to fit the requirements of your site. These parameters control Hipersorting storage decisions:

EXPMAX
EXPOLD
EXPRES
HIPRMAX

Study the values for these parameters carefully before you install DFSORT. These parameters are described in this chapter. To see how these parameters relate to planning decisions, read "Tailoring Storage for Hipersorting" on page 28.

How to Code 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 space, and parameters must be separated by commas. See "Notational Conventions" on page xiii for information about notational conventions used in this book. For more complete rules, refer to your assembler language manual.

ICEMAC Syntax Diagram

Figure 22 on page 59 is the syntax diagram for ICEMAC. You must specify one of the following keywords to indicate the installation module 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 -- one or more of the parameters can be used, but the individual parameters (such as ABCODE) cannot be repeated. The ENABLE parameter can be used with JCL, INV, TSO or TSOINV to indicate one or more time-of-day modules to be associated with an ICEAMx module. One or more *day* parameters can be used with TD1, TD2, TD3 and TD4 to indicate the days and times when an ICETDx module can be activated.

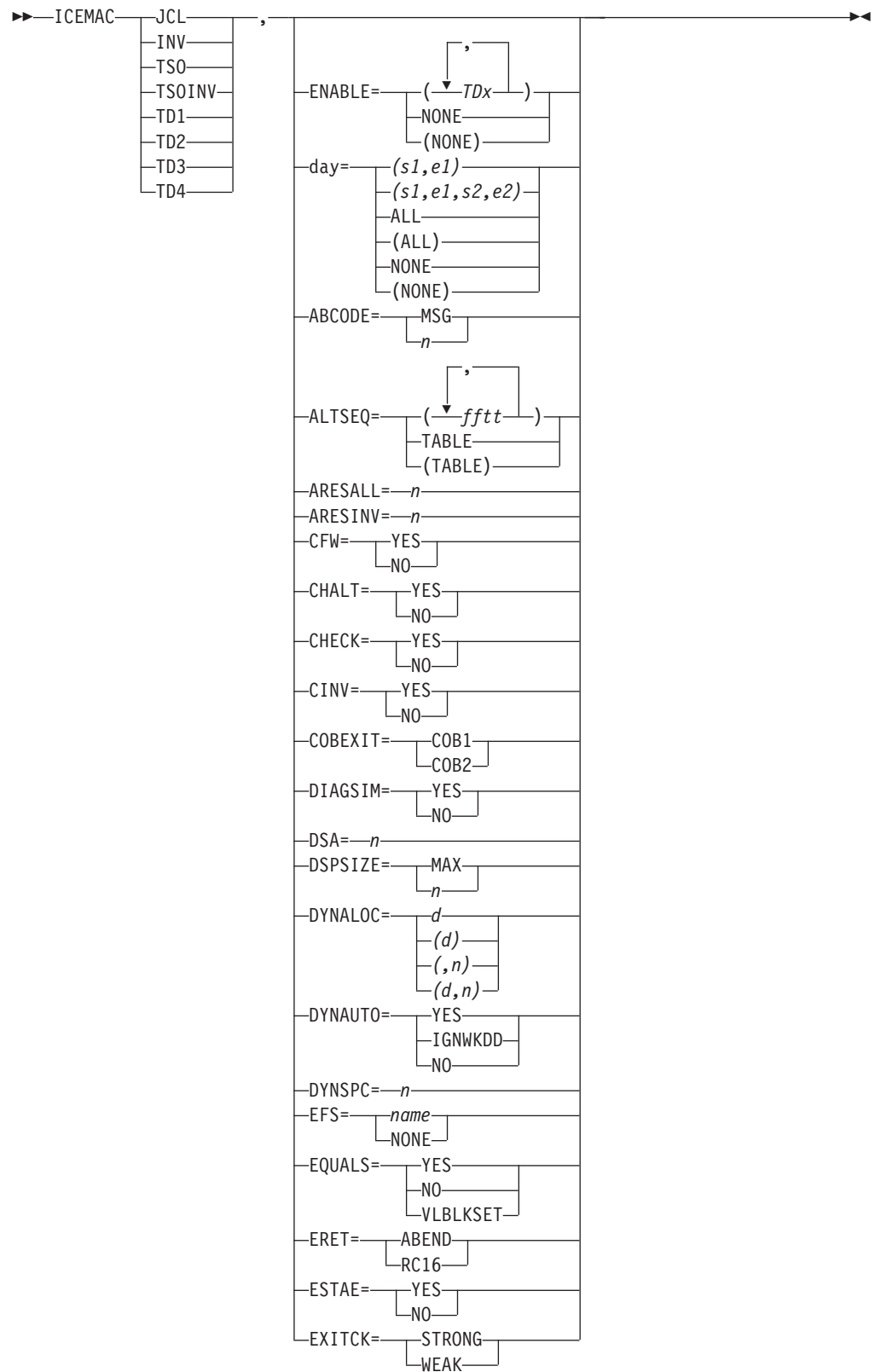


Figure 22. Syntax of the ICEMAC Macro (Part 1 of 4)

ICEMAC Macro

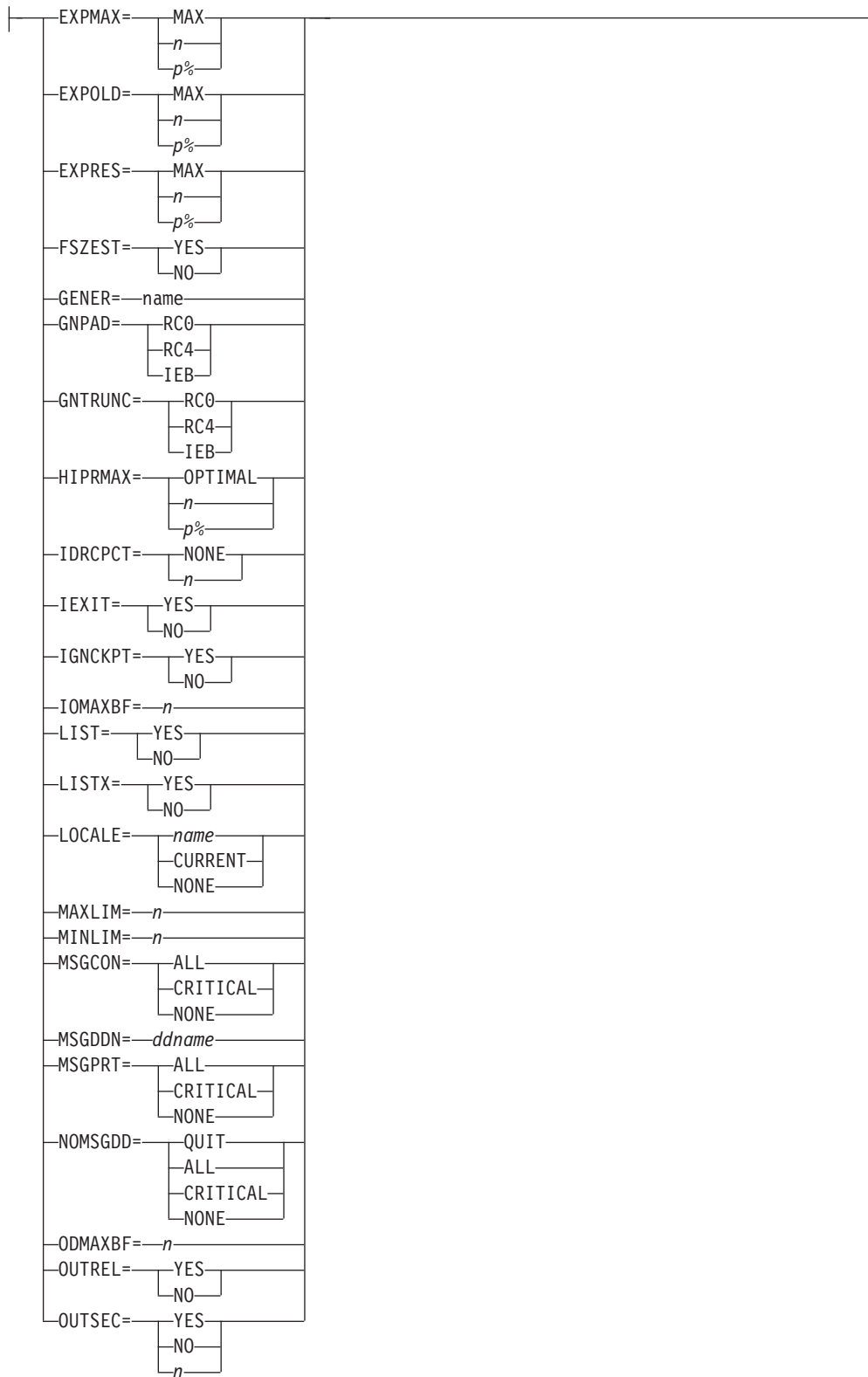


Figure 22. Syntax of the ICEMAC Macro (Part 2 of 4)

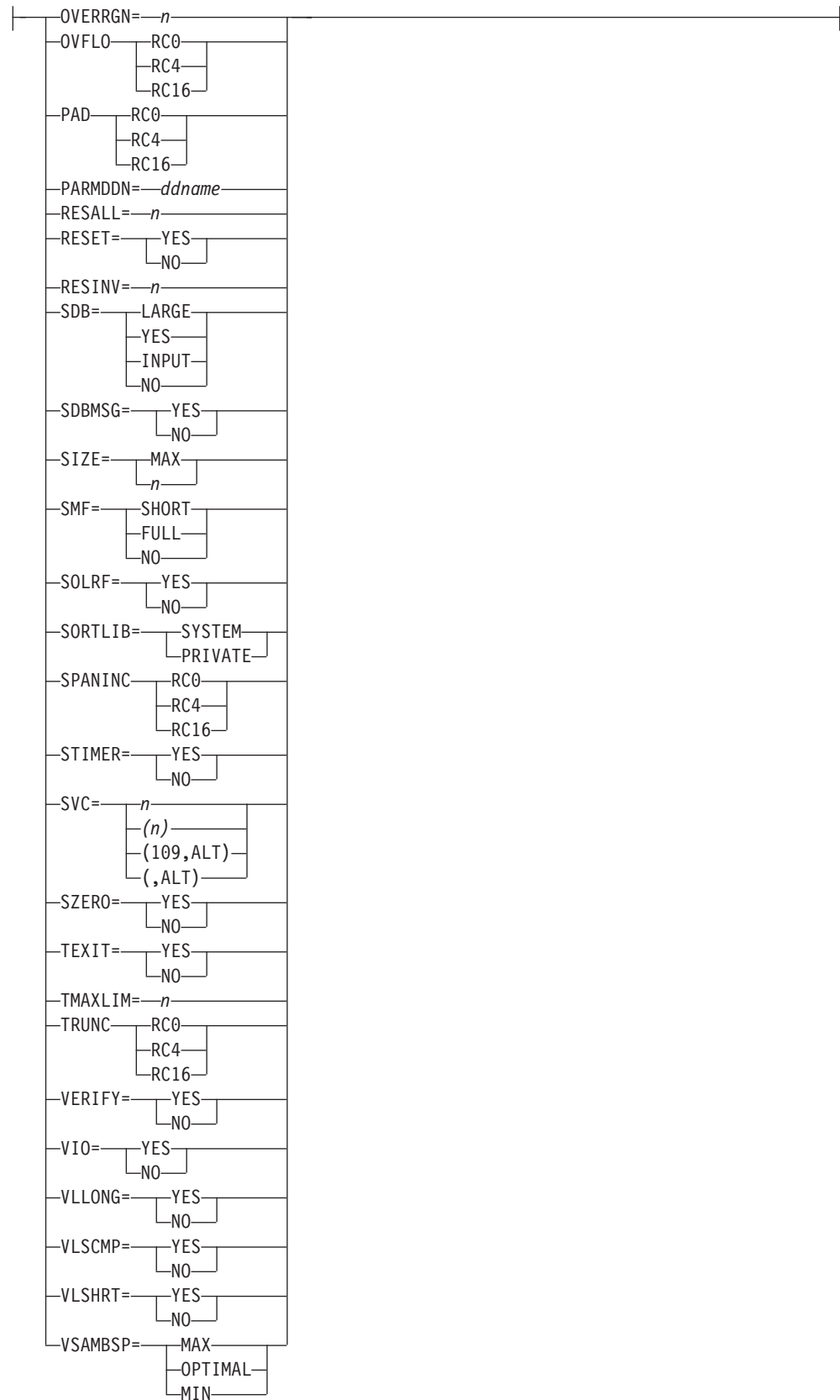


Figure 22. Syntax of the ICEMAC Macro (Part 3 of 4)

ICEMAC Macro

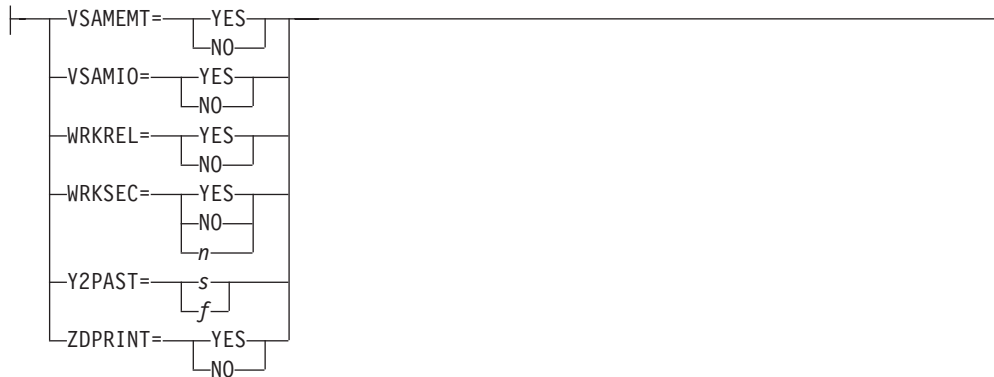
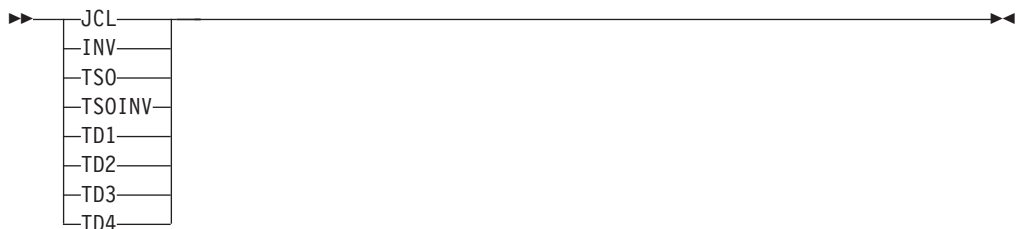


Figure 22. Syntax of the ICEMAC Macro (Part 4 of 4)

ICEMAC Parameters

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



specifies the environment installation module (ICEAMx where x is 1, 2, 3 or 4) or time-of-day installation module (ICETDx where x is 1, 2, 3 or 4) for which this set of ICEMAC 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 modules the other parameters apply.

The ENABLE parameter can be specified with JCL, INV, TSO or TSOINV to enable one or more time-of-day installation modules (ICETDx) for the selected environment installation module (ICEAMx). One or more of the ICETDx modules can be enabled for any ICEAMx module. 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 module (ICETDx) is to be activated.

JCL

specifies that this set of ICEMAC defaults is for ICEAM1, the batch direct invocation environment installation module. 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 installation module (ICETDx) is not activated. All of the ICEMAC parameters can be specified with ICEMAC JCL. However, parameters ARESINV, RESINV, GENER, GNPAD and GNTRUNC are not meaningful for direct invocation and are ignored.

INV

specifies that this set of ICEMAC defaults is for ICEAM2, the batch program

invocation environment installation module. ICEAM2 is used at run time when DFSORT is invoked through batch programs, provided that an enabled time-of-day installation module (ICETDx) is not activated. All of the ICEMAC parameters can be specified with ICEMAC INV.

TSO

specifies that this set of ICEMAC defaults is for ICEAM3, the TSO direct invocation environment installation module. 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 module (ICETDx) is not activated. All of the ICEMAC parameters can be specified with ICEMAC TSO. However, parameters ARESINV, RESINV, GENER, GNPAD and GNTRUNC are not meaningful for direct invocation and are ignored.

TSOINV

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

TD1

specifies that this set of ICEMAC defaults is for ICETD1, the first time-of-day installation module. ICETD1 is used at run time when it is activated for the time-of-day of the run, provided it is enabled by the environment installation module (ICEAMx) in effect. All of the ICEMAC parameters can be specified with ICEMAC TD1. However, parameters ARESINV, RESINV, GENER, GNPAD and GNTRUNC are not meaningful for direct invocation and are ignored if the enabling installation module is ICEAM1 or ICEAM3.

TD2

specifies that this set of ICEMAC defaults is for ICETD2, the second time-of-day installation module. ICETD2 is used at run time when it is activated for the time-of-day of the run, provided it is enabled by the environment installation module (ICEAMx) in effect. All of the ICEMAC parameters can be specified with ICEMAC TD2. However, parameters ARESINV, RESINV, GENER, GNPAD and GNTRUNC are not meaningful for direct invocation and are ignored if the enabling installation module is ICEAM1 or ICEAM3.

TD3

specifies that this set of ICEMAC defaults is for ICETD3, the third time-of-day installation module. ICETD3 is used at run time when it is activated for the time-of-day of the run, provided it is enabled by the environment installation module (ICEAMx) in effect. All of the ICEMAC parameters can be specified with ICEMAC TD3. However, parameters ARESINV, RESINV, GENER, GNPAD and GNTRUNC are not meaningful for direct invocation and are ignored if the enabling installation module is ICEAM1 or ICEAM3.

TD4

specifies that this set of ICEMAC defaults is for ICETD4, the fourth time-of-day installation module. ICETD4 is used at run time when it is activated for the time-of-day of the run, provided it is enabled by the environment installation module (ICEAMx) in effect. All of the ICEMAC parameters can be specified with ICEMAC TD4. However, parameters

ICEMAC Macro

ARESINV, RESINV, GENER, GNPAD and GNTRUNC are not meaningful for direct invocation and are ignored if the enabling installation module is ICEAM1 or ICEAM3.

ENABLE



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

(TDx...,TDx)

specifies the ICETDx installation modules in effect for this ICEAMx environment installation module, 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 modules are not to be used for this ICEAMx installation module.

Default: NONE

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

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 will be used if ICETD3 is activated for the time and day of the run. If ICETD3 is not activated, the ICETD2 defaults will be used if ICETD2 is activated. If ICETD3 is not activated and ICETD2 is not activated, the ICEAM2 defaults will be used.

day



specifies the time ranges for each day of the week when this time-of-day installation module (ICETDx) is to be activated. The time ranges may be specified for all days, the weekend days, the weekdays or each day of the week; any combination may be used, but 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 module must:

- be specified by the ENABLE parameter of the environment installation module (ICEAMx) in effect for the run-time 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

this 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 module 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 11:59pm (2359). 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's 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 DFSORT's ICE000I message) is between 6:30am and 7:59pm (630,1959) on Sunday, between 10:00pm and 11:59pm (2200,2359) on Friday or between midnight and 11:59am (0,1159) on Saturday.

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

ICEMAC Macro

- 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 module 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 module can be activated for the specified day. See s1,e1 above for details.

ALL | (ALL)

specifies that this time-of-day module is always to be activated on the specified day. This is equivalent to a time range of midnight to 11:59pm (0,2359). ALL overrides any s1,e1 or s2,e2 times that cross midnight of the previous day.

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 11:59pm (1200,2359) on Monday through Friday, between midnight and 7:29am (0,729) on Tuesday through Friday or between midnight and 11:59pm (0,2359) on Saturday (that is, any time on Saturday).

NONE | (NONE)

specifies that this time-of-day module is not to be activated on the specified day. s1,e1 or s2,e2 times that cross midnight of the previous day override NONE (either specified or defaulted).

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 8:00pm and 11:59pm (2000,2359) on Friday or between midnight and 12:40pm (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)

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

Example 1

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

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 parameters, will be used on Monday through Friday from midnight to 7:59am (0,759) and from 6:00pm to 11:59pm (1800,2359).

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

Example 2

```
ICEMAC JCL,ENABLE=(TD2),SVC=252
END
ICEMAC TD2,WKDAY=(2130,615),SAT=(1205,1559,1800,2359), *
      SUN=ALL,SVC=252,DSA=48,TMAXLIM=33554432
END
```

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 parameters, will be used as follows:

- Sunday: midnight-11:59pm (0,2359)
- Monday: 9:30pm-11:59pm (2130,2359)
- Tuesday: midnight-6:15am (0,615) and 9:30pm-11:59pm (2130,2359)
- Wednesday: midnight-6:15am (0,615) and 9:30pm-11:59pm (2130,2359)
- Thursday: midnight-6:15am (0,615) and 9:30pm-11:59pm (2130,2359)
- Friday: midnight-6:15am (0,615) and 9:30pm-11:59pm (2130,2359)
- Saturday: midnight-6:15am (0,615), 12:05pm-3:59pm (1205,1559) and 6:00pm-11:59pm (1800,2359)

The ICEAM1 set of defaults consisting of SVC=252 and the IBM-supplied defaults for all other parameters, 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 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 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.

ICEMAC Macro

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 *DFSORT Application Programming Guide R14* for additional details and examples.

TABLE | (TABLE)

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. When using ALTSEQ=(TABLE), TABLE must be the only parameter. If there is more than one parameter, 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'FFFEFDFCFBFAF9F8F7F6F5F4F3F2F1F0'
DC X'EFEEDECEBEAE9E8E7E6E5E4E3E2E1E0'
DC X'DFDEDDDCDBDAD9D8D7D6D5D4D3D2D1D0'
DC X'CFCECDCCBCAC9C8C7C6C5C4C3C2C1C0'
DC X'BFBEBCBBAB9B8B7B6B5B4B3B2B1B0'
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.

Note: Use of ALTSEQ can degrade performance.

ARESALL

▶▶—ARESALL=*n*————▶▶

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

Note: RESALL specifies the amount of storage reserved below 16MB virtual. For more information, see “Tailoring Main Storage” on page 18.

n specifies the number of bytes as a decimal value within the range of 0 to 16515072 (16128KB). 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 16MB virtual, while DFSORT is running, for an invoking program’s user exit routines that reside above or use space above 16MB 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 18. Normally, this parameter is not needed.

Note: RESINV specifies the amount of storage reserved below 16MB virtual. For more information, see “Tailoring Main Storage” on page 18.

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

Default: 0 bytes

Note: 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 3990-3 control units. (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

ICEMAC Macro

CHALT

▶▶—CHALT=—YES—
 └─NO—┘

specifies whether ALTSEQ is to apply to character format fields (CH).

YES

specifies that ALTSEQ is to apply to character format fields.

NO

specifies that ALTSEQ is not to apply to character format fields.

Default: NO

CHECK

▶▶—CHECK=—YES—
 └─NO—┘

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

▶▶—CINV=—YES—
 └─NO—┘

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

▶▶—COBEXIT=—COB1—
 └─COB2—┘

specifies the library for COBOL E15 and E35 routines. See *Application Programming Guide* for details on the use of COBEXIT and run-time libraries with COBOL E15 and E35 routines.

COB1

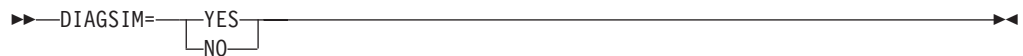
specifies that COBOL E15 and E35 routines are run with the OS/VS COBOL run-time library or, in some cases, with no COBOL run-time library.

COB2

specifies that COBOL E15 and E35 routines are run with either the VS COBOL II run-time library or the Language Environment run-time library.

Default: COB1

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 *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 later in this chapter for complete details.

NO

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

Default: NO

DSA



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

ICEMAC Macro

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.

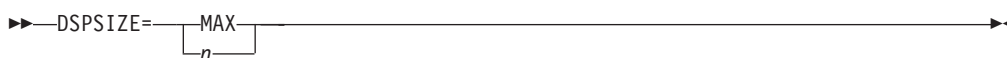
The performance improvement from dynamic storage adjustment usually provides a good tradeoff against the increased storage used by DFSORT. On storage constrained systems, however, the DSA value should be set low enough to prevent unacceptable paging.

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

If n is less than or equal to the value specified for TMAXLIM, n is set to 0 to indicate that storage will not be dynamically adjusted.

Default: 32

DSPSIZE



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, expanded, 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.

The amount of data space used by DFSORT is limited to the installation or user-specified DSPSIZE value and by the IEFUSI exit of your system. DSPSIZE=MAX (the IBM-supplied default) means that DFSORT selects the maximum amount of data space to be used based on the size of the input file and the paging activity of the system. You can further limit the amount of data space that DFSORT uses by specifying a maximum value, in megabytes.

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 DASD as temporary work space. When dataspace sorting is enabled, DFSORT changes DYNAUTO=NO to DYNAUTO=YES in some cases. Hiperspace is 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 ICEMAC DYNAUTO parameter or the DYNALOC run-time 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 direct access storage 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.

For best performance, specify an emulated 3390-9 device (such as RAMAC) or another high-speed IBM DASD device as the default, and avoid specifying a tape, virtual (VIO), or real 3390-9 devices as the default.

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.

Note: 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.

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 DYNALOC/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

ICEMAC Macro

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

►►—DYNSPC=—*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 run-time 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 DASD space available for DFSORT work data sets. As a guideline, Table 6 on page 75 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 6. Example of DYNMPC Primary Space

DYNMPC value (megabytes)	Primary space (cylinders)
32	48
64	93
128	183
256	366
512	732

The larger your DYNMPC 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 or data space) with the primary space shown in Table 6, and all of the corresponding secondary space, Blockset is able to sort approximately 150 megabytes with DYNMPC=32 and approximately 1200 megabytes with DYNMPC=256. If Hiperspace or data space can be used along with dynamically allocated work space, the amount of data DFSORT can sort will increase according to the amount of Hiperspace or data space 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 DASD space, because this causes dynamic allocation to fail for sort applications that use this default.

Default: 256

EFS

►► 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 *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

►► 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 *Application Programming Guide*.

ICEMAC Macro

YES

means the order should be preserved.

NO

means the order need not be preserved.

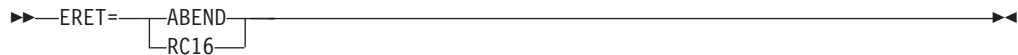
VLBLKSET

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

Note: 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 run-time 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, or with a specific number between 1 and 99, as set during installation with the ICEMAC parameter ABCODE.

To avoid problems with DFSORT Panels running in the foreground, ERET=ABEND should not be specified with the TSO parameter, or with any ICETDx modules enabled for the ICEAM3 module.

RC16

specifies that DFSORT terminates with a return code of 16.

Default: RC16

Note: If DFSORT determines at run-time that a SmartBatch pipe data set is being used, it automatically forces the ABEND run-time option on to ensure that an abend will be generated if an error is detected. This allows for appropriate error propagation by the system to other applications that might be accessing the same SmartBatch pipe data set.

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 above.

YES

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

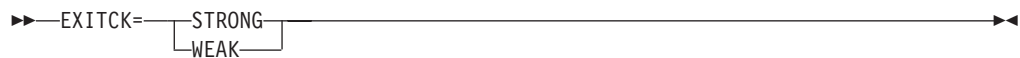
NO

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

Note: See *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 *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.

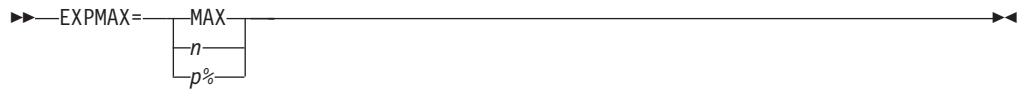
Default: STRONG

Note: 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

ICEMAC Macro



specifies the maximum total amount of available expanded storage, or central storage for 64-bit real mode, to be used at any one time by all Hipersorting applications. See the discussion of the HIPRMAX parameter for more information about Hipersorting.

“Available” storage is the storage used to back new Hiperspace data. EXPMAX can be used to restrict Hipersorting activity to a specified portion of available storage. In order for total Hipersorting 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 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 as a percentage of the configured expanded storage on the system at run-time. In 64-bit real mode, EXPMAX=p% specifies a percentage of an appropriate portion of central storage. 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.

For more information, see “Tailoring Storage for Hipersorting” on page 28.

- MAX** specifies that DFSORT determines dynamically the maximum total amount of available storage to be used at any one time by all Hipersorting 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 applications on the system, subject to a maximum of nMB. n must be a value between 0 and 8388606. If n is 0, Hipersorting is 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 applications on the system, subject to a maximum of p percent of the configured expanded storage. In 64-bit real mode, EXPMAX=p% specifies a percentage of an appropriate portion of 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 8388606MB.

Default: MAX

EXPOLD



specifies the maximum total amount of old expanded storage, or central storage for 64-bit real mode, to be used at any one time by all Hipersorting applications. See the discussion of the HIPRMAX parameter for more information about Hipersorting.

“Old” storage is 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 migrates it to auxiliary storage to make room for new hiperspace data. EXPOLD limits the total amount of such migration at any one time by all Hipersorting applications on the system. It should be set to a value other than MAX if this migration causes an unacceptable impact on the system.

EXPOLD=n specifies a fixed value for EXPOLD. EXPOLD=p% specifies a value for EXPOLD that varies as a percentage of the configured expanded storage on the system at run-time. In 64-bit real mode, EXPOLD=p% specifies a percentage of an appropriate portion of central storage. 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.

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

For more information, see “Tailoring Storage for Hipersorting” on page 28.

- MAX** specifies that DFSORT can use up to all of the old storage for Hipersorting.
- n** specifies that DFSORT limits the total amount of old storage to be used at any one time by all Hipersorting applications on the system to a maximum of nMB. n must be a value between 0 and 8388606. If n is 0, old storage is not used for Hipersorting.
- p%** specifies that DFSORT limits the total amount of old storage to be used at any one time by all Hipersorting applications on the system to a maximum of p percent of the configured expanded storage. In 64-bit real mode, EXPOLD=p% specifies a percentage of an appropriate portion of central storage. p must be a value between 0 and 100. If p is 0, old storage is not used for Hipersorting. The value calculated for p% is limited to 8388606MB.

Default: MAX

EXPRES



ICEMAC Macro

specifies the minimum amount of available expanded storage, or central storage for 64-bit real mode, to be reserved for use by non-Hipersorting applications. This is storage that would otherwise be available for Hipersorting applications. See the discussion of the HIPRMAX parameter for more information about Hipersorting.

EXPRES defines a threshold amount of available storage below which all currently running Hipersorting applications switch from using Hiperspace to using DASD work data sets. In addition, no new DFSORT applications use Hipersorting 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-Hipersorting applications. Regardless of the setting of EXPRES, DFSORT is always aware of the future needs of other concurrent Hipersorting applications, and never interferes with storage that is being actively used by non-Hipersorting applications. Sudden increases in the use of storage by non-Hipersorting 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 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 Hipersorting 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 as a percentage of the configured expanded storage on the system at run-time. In 64-bit real mode, EXPRES=*p*% specifies a percentage of an appropriate portion of central storage. 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*.

For more information, see "Tailoring Storage for Hipersorting" on page 28.

- MAX** specifies that DFSORT reserves all available storage for use by non-Hipersorting applications. This disables Hipersorting.
- n** specifies that DFSORT reserves *n*MB of available storage for use by non-Hipersorting applications. *n* must be a value between 0 and 8388606. If *n* is 0, DFSORT considers all available storage eligible for Hipersorting applications.
- p%** specifies that DFSORT reserves *p* percent of the configured storage for use by non-Hipersorting applications. In 64-bit real mode, EXPRES=*p*% specifies a percentage of an appropriate portion of central storage. *p* must be a value between 0 and 100. If *p* is 0, DFSORT considers all available storage eligible for Hipersorting applications. The value calculated for *p*% is limited to 8388606MB.

Default: 0

FSZEST

▶▶ FSZEST=

YES
NO

 ▶▶

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 run-time, 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 run-time, n is treated as an exact value. This causes DFSORT applications to terminate when the number of records is not exactly equal to n.

Notes:

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

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

GNPAD

▶▶ GNPAD=

RC0
RC4
IEB

 ▶▶

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.

ICEMAC Macro

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

Notes:

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

Notes:

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 expanded storage, or in central storage for 64-bit real mode, and is backed by auxiliary storage (if necessary). Because I/O processing is reduced for Hipersorting, elapsed time, EXCP counts, and channel usage are also reduced.

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

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

Available storage is a constantly changing amount, 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 applications further limit the amount of available storage. A Hipersorting application is aware of the storage needs of every other Hipersorting application on the system, and does not attempt to back its Hiperspace data with storage needed by another Hipersorting application. This prevents overcommitment of storage resources in the event of multiple large concurrent Hipersorting 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. EXPOLD limits the total amount of old storage that can be used at any one time to back DFSORT hiperspaces. EXPRES sets aside a specified amount of available storage for use by non-Hipersorting applications.

Some of these limits depend on system and other Hipersorting activity throughout 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 expanded storage on the system at run-time. In 64-bit real mode, HIPRMAX=*p*% specifies a percentage of an appropriate portion of central storage. If the storage on a

ICEMAC Macro

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” on page 28.

If the amount of Hiperspace available for Hipersorting is insufficient for temporary storage of the records, intermediate DASD 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 nMB. 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 expanded storage. In 64-bit real mode, HIPRMAX=p% specifies a percentage of an appropriate portion of 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 32767MB, 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 IDRCPCT value to estimate the file size for compacted input tape data sets that are not managed by DFSMSrmm.

The specified IDRCPCT 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 200MB, 800MB, and 4000MB, 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 IDRCPCT percentage.

The IDRCPCT 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 *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. Note that this may 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

```
IDRCPCT=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 200MB (3480), 800MB (3490), or 4000MB (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

```
IDRCPCT=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 IDRCPCT 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 IDRCPCT=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 IDRCPCT 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.

ICEMAC Macro

Notes:

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 4MB 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 800MB. This type of over-allocation increases with the use of IDRCPCT and compacted tape data sets. For example, if the 4MB 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 2800MB (3.5*800MB) of data.

IEXIT

►► IEXIT= YES NO ◀◀

specifies whether DFSORT is to pass control to an initialization exit (ICEIEXIT) as described in Chapter 6, "Controlling Resources Using ICEIEXIT" on page 127.

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

►► IGNCKPT= YES NO ◀◀

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 the SORTIN, SORTINnn, and SORTOUT data sets. 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 16MB 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 (8MB), then at run time DFSORT uses:

- Up to 8MB for the SORTIN buffers for sort and copy applications
- Up to 8MB for all of the SORTINnn buffers for merge applications
- Up to 8MB 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 3MB per second, the IBM shipped default of 33554432 (32MB) for IOMAXBF engages the channel for a maximum of about 11 seconds at any one time. This is well below the IBM supplied missing interrupt handler (MIH) value of 15 seconds for most direct access devices. Exceeding this MIH value can result in I/O terminations of DFSORT applications. IBM recommends that this MIH value be raised for newer devices, such as 3390 and RAMAC DASD. 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 3MB per second, and you wish to limit DFSORT to a maximum channel connect time of 8 seconds, set IOMAXBF to 25165824 (24MB). Likewise, with a channel transfer rate of 4MB per second and an MIH value of 30 seconds, limiting DFSORT to no more than 25 seconds of channel connect time by setting IOMAXBF to 104857600 (100MB), will prevent MIH terminations.

Lowering IOMAXBF below the default value of 33554432 (32MB) can cause performance degradation.

n specifies the maximum bytes of storage to be used for the SORTIN,

ICEMAC Macro

SORTINnn, and SORTOUT data set buffers as a decimal value within the range of 8388608 (8MB) to 268435456 (256MB).

Default: 33554432 (32MB)

Example 1

```
IOMAXBF=37748736
```

At run time, DFSORT uses up to 36MB for the SORTIN buffers for sort and copy applications, up to 36MB for all of the SORTINnn buffers for merge applications, and up to 36MB for the SORTOUT buffers for sort, copy, and merge applications.

At a transfer rate of 3MB per second, this IOMAXBF value limits DFSORT to a maximum connect time of 12 seconds.

Example 2

```
IOMAXBF=20971520
```

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

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

LIST

```
▶▶—LIST=—YES—————▶▶
           |
           | NO
```

specifies whether DFSORT program control statements are printed. For full details on use of the message data set, see *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

```
▶▶—LISTX=—YES—————▶▶
           |
           | NO
```

specifies whether to print control statements returned by an EFS program. For full details on use of the message data set, see *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.

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 *Using Locales* 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

Notes:

1. 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.
2. 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.
3. To use an IBM supplied locale, DFSORT must have access to the Language Environment run-time 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.
4. If locale processing is used for SORT, MERGE, INCLUDE, or OMIT fields:
 - VLSHRT is not used for SORT or MERGE
 - CHALT, INREC, an EFS program, or an E61 exit must not be used
5. 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 16MB virtual, available to DFSORT. When Blockset is selected, MAXLIM is used to limit the storage below 16MB virtual, and SIZE/MAINSIZE is used to limit the total storage above and below 16MB 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 18.

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

is lowered below 1MB. If the value specified is not a multiple of 8, it is rounded down to the next multiple of 8.

Default: 1048576 (1MB)

Notes:

1. The MAXLIM limit is not observed for a copy or Blockset merge application when the use of more storage below 16MB virtual would improve performance.
2. The MAXLIM limit is not observed for OUTFIL 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 88KB of storage is available below 16MB virtual. For more information, see “Tailoring Main Storage” on page 18.

n specifies the number of bytes as a decimal value. The minimum value is 90112 (88KB). In addition, the value must be less than or equal to both the SIZE and MAXLIM values specified for the ICEMAC macro. Lowering MINLIM below 440KB can degrade performance for some applications. This performance degradation can become more pronounced and affect more applications the more MINLIM is lowered below 440KB. If the value specified is not a multiple of 8, it is rounded down to the next multiple of 8.

Default: 450560 (440KB)

MSGCON

►►—MSGCON=

ALL
CRITICAL
NONE

—————►►

specifies the class of messages to be written to the master console. For full details about having DFSORT write to the master console, see *Messages, Codes and Diagnosis Guide*.

ALL

specifies that all messages, except parameter-in-effect messages (ICE128I to ICE133I) 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

►►—MSGDDN=*ddname*—————►►

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 *Messages, Codes and Diagnosis Guide*.

Default: SYSOUT

MSGPRT

►►—MSGPRT=

ALL
CRITICAL
NONE

—————►►

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 *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 *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

ODMAXBF



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 16MB virtual for OUTFIL processing whenever possible.

The default value of 2097152 (2MB) 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 (256KB) to 16777216 (16MB).

ICEMAC Macro

Default: 2097152 (2MB)

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

▶▶—OUTSEC=—

YES
NO
n

————▶▶

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

OVERRGN

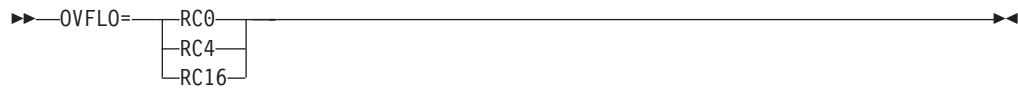
▶▶—OVERRGN=*n*————▶▶

specifies the amount of main storage, below 16MB 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 64KB). For more information, see "Tailoring Main Storage" on page 18.

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

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

ICEMAC Macro

Default: RC0

Notes:

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: If PARMDDN=DFSPARM is specified or defaulted, and a DFSPARM data set is not available at run time, DFSORT will use a \$ORTPARM data set if available.

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 4KB (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 16MB virtual. ARESALL applies to the amount of main storage above 16MB virtual.

For more information, see “Tailoring Main Storage” on page 18.

n specifies the number of bytes as a decimal value within the range of 4096 (4KB) to 16515072 (16128KB). If the value specified is not a multiple of 8, it is rounded up to the next multiple of 8.

Default: 4096 (4KB)

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 16MB virtual. ARESINV applies to the amount of main storage above 16MB virtual. For more information, see “Tailoring Main Storage” on page 18.

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 (16KB) is sufficient. The reserved space is not meant to be used for the invoking program’s executable code.

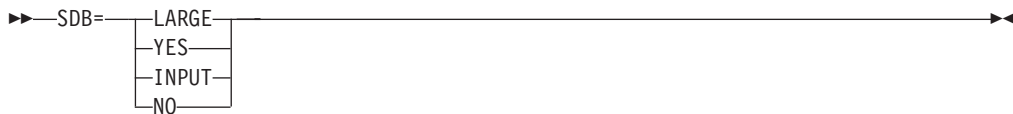
n specifies the number of bytes as a decimal value within the range of 0 to 16515072 (16128KB). If the value specified is not a multiple of 8, it is rounded up to the next multiple of 8.

Default: 0 bytes

Notes:

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



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.

With OS/390 Release 10 and above, 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 DASD 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. With OS/390 Release 10 and above, 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 DASD 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 DASD 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 7. 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, non-zero) in the JFCB (DASD or tape) or format 1 DSCB (DASD) or tape label (only for DISP=MOD with AL, SL, or NSL label, when appropriate)
- Not MVS/DFP 3.1.0 or above
- Output is a spool, dummy, VSAM, or unmovable data set, or an HFS file
- The output data set is on tape with a label type of AL
- DFSORT's Blockset technique is **not** selected.

In the above 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

ICEMAC Macro

set based on the obtained or derived output or input attributes. SDB=NO limits the selected block sizes to a maximum of 32760 bytes.

SDB=NO works like SDB=YES if the input block size is greater than 32760 bytes (only possible for tape input data sets). See the discussion of SDB=YES for more information.

Notes:

1. With OS/390 R9 and below, 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 changed to specify that block size **explicitly**.
4. SDB=SMALL can be used instead of SDB=YES.

Default: INPUT

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 DASD 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, non-zero)
- Not MVS/DFP 3.1.0 or above
- 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.

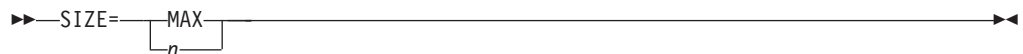
Note: SDBMSG=NO does not prevent the use of system-determined block size for message or list data sets at allocation or in other cases where the block size is set before DFSORT or ICETOOL gets control.

Default: NO

Notes:

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 18.

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 (88KB) to 2097152000 (2000MB). Lowering SIZE below 4MB 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 4MB. If the value specified is not a multiple of 8, it is rounded down to the next multiple of 8.

Default: MAX

ICEMAC Macro

Notes:

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 16MB 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 main 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 7, "Collecting Statistical Data" on page 137.

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.

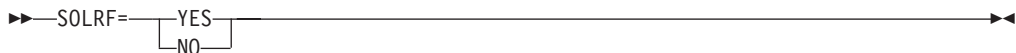
Note: SMF=FULL can degrade performance for variable-length record applications.

NO

DFSORT does not produce any SMF type-16 records.

Default: NO

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

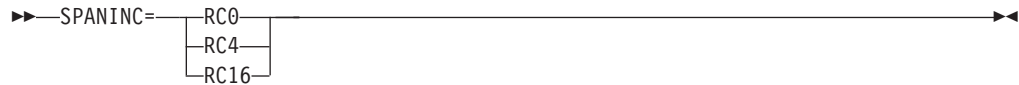
specifies that the Conventional technique modules reside in a private library

ICEMAC Macro

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 it detects. Valid records will be recovered.

RC16

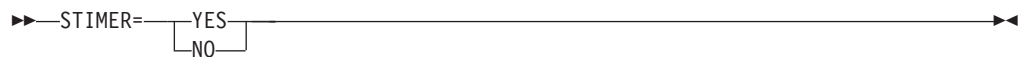
specifies that DFSORT should issue message ICE204A, terminate, and give a return code of 16 when an incomplete spanned record is detected.

Default: RC16

Notes:

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.

STIMER



specifies whether DFSORT issues an STIMER to monitor the processor time.

YES

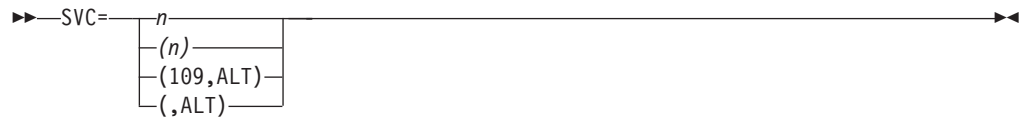
An STIMER is issued and processor time data appears in SMF records and ICETEXIT statistics.

NO

No STIMER is issued and processor time does not appear in SMF records or ICETEXIT statistics.

Default: YES

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*.

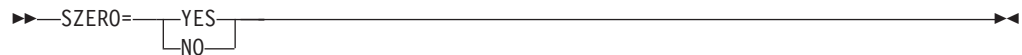
ALT

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).

For more information, see “Running More Than One DFSORT Release at a Time” on page 9. You can also find more information on installing SVCs in *Initialization and Tuning*.

Default: 109

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 INCLUDE and OMIT fields and constants, -0 compares as less than +0.
- For editing and conversions of INREC, OUTREC and OUTFIL OUTREC fields, -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.

ICEMAC Macro

- For comparisons of INCLUDE, OMIT and OUTFIL INCLUDE and OMIT fields and constants, -0 compares as equal to +0.
- For editing and conversions of INREC, OUTREC and OUTFIL OUTREC fields, -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

►► TEXTIT=

YES
NO

 ◀◀

specifies whether DFSORT is to pass control to a termination exit (ICETEXIT) as described in Chapter 7, “Collecting Statistical Data” on page 137.

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

►► TMAXLIM=*n* ◀◀

specifies an upper limit to the total amount of main storage, both above and below 16MB virtual, available to DFSORT when SIZE/MAINSIZE=MAX is in effect and Blockset is selected. For more information, see “Tailoring Main Storage” on page 18.

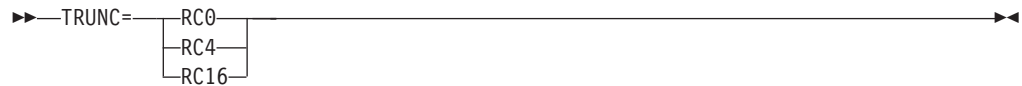
n specifies the number of bytes as a decimal value. The maximum value is 2097152000 (2000MB). In addition, the value must not be less than the MAXLIM value. Lowering TMAXLIM below 4MB can degrade performance for some applications. This performance degradation can become more pronounced and affect more applications the more TMAXLIM is lowered below 4MB. If the value specified is not a multiple of 8, it is rounded down to the next multiple of 8.

Default: 4194304 (4MB)

Notes:

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.

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

Notes:

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

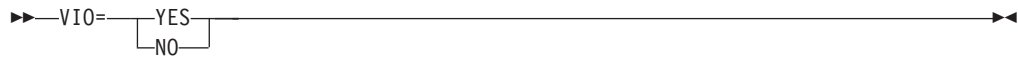
specifies that the sequence is not to be verified.

Default: NO

Note: VERIFY=YES can degrade performance, especially when sorting short records.

VIO

ICEMAC Macro



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

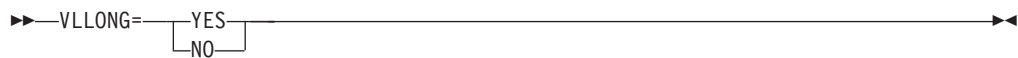
DFSORT dynamically reallocates any virtual allocation of sort work data sets to get real disk locations.

Default: NO

Notes:

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.

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

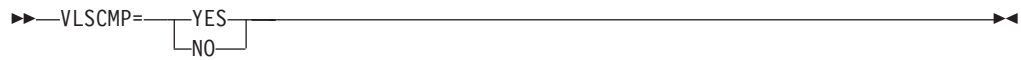
specifies that DFSORT terminates if a long variable-length output record is found.

Default: NO

Notes:

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 run-time 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 below). 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'hh00' (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'))
```

ICEMAC Macro

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 *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

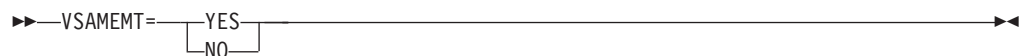
specifies that DFSORT can use the optimal number of VSAM buffers when sorting. OPTIMAL can provide substantial performance benefits with minimal impact on system paging activity.

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

ICEMAC Macro

►►—VSAMIO=—YES
 └─NO—┘

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 above conditions are not met.

NO

specifies that DFSORT terminates if the same VSAM data set is specified for input and output

Default: NO

WRKREL

►►—WRKREL=—YES
 └─NO—┘

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

Notes:

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

►►—WRKSEC=—YES
 └─NO—┘
 └─n—┘

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

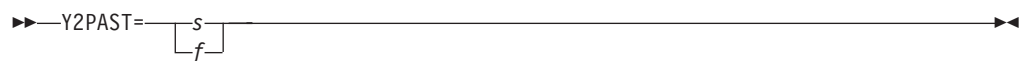
specifies that automatic secondary allocation for temporary JCL SORTWKdd data sets not be used.

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: WRKSEC has no effect on dynamically allocated work data sets.

Y2PAST



specifies the sliding (s) or fixed (f) century window. The century window is used with DFSORT's Y2C, Y2Z, Y2P, Y2D, Y2S, and Y2B 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. s must be a value between 0 and 100.

f specifies the beginning of the fixed century window. f must be a value between 1000 and 3000.

Default: 80

Example 1

Y2PAST=81

If the current year is 1996, the century window will be 1915 to 2014. This century window results in the following interpretation of Y2 format data by DFSORT:

Two-digit year data	Interpretation
-----	-----
00	2000
14	2014
15	1915
61	1961
62	1962
99	1999

ICEMAC Macro

Note that when the current year changes to 1997, the century window for Y2PAST=81 will be 1916 to 2015; the century window slides as the current year changes.

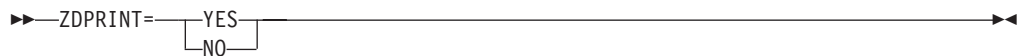
Example 2

Y2PAST=1962

The century window will be 1962 to 2061. This century window results in the following interpretation of Y2 format data by DFSORT:

Two-digit year data	Interpretation
00	2000
14	2014
15	2015
61	2061
62	1962
99	1999

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: NO

Examples of ICEMAC Macros

This section shows you examples of the ICEMAC macro for several of the eight installation modules.

ICEMAC JCL - Batch Direct Invocation Defaults

Figure 23 is a sample of the ICEMAC macro update statements to produce module ICEAM1, the JCL defaults module.

column			
1	16	72 73	80
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 23. Coding the ICEMAC Macro for ICEAM1

Line Explanation

- 01** This ICEMAC call sets up your installation parameter 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 (5MB) both above and below 16MB virtual.
- 03** When Blockset is selected, DFSORT can use up to 2097152 bytes (2MB) below 16MB virtual. When Blockset is not selected, DFSORT makes use of all available space (the default is SIZE=MAX) up to 2097152 bytes (2MB) below 16MB virtual.
- 04** DFSORT always tries to use at least 524288 bytes (512KB).
- 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 29 on page 120, Figure 30 on page 121, Figure 31 on page 122, Figure 32 on page 123, Figure 33 on page 124, and Figure 34 on page 125 for the IBM-supplied defaults for all of the ICEMAC parameters.

ICEMAC INV - Batch Program Invocation Defaults

Figure 24 on page 116 is a sample of the ICEMAC macro update statements to produce module ICEAM2, the INV defaults module.

ICEMAC Macro

column			
1	16	72 73	80
	ICEMAC INV,	* 00550000	01
	DYNALOC=3380,	* 00550100	02
	IGNCKPT=NO,	* 00550200	03
	MSGDDN=MSGOUT,	* 00550300	04
	MSPRT=CRITICAL,	* 00550400	05
	OUTREL=NO,	* 00550500	06
	WRKREL=NO,	* 00550600	07
	LIST=YES	00550700	08

Figure 24. Coding the ICEMAC Macro for ICEAM2

Line Explanation

- 01** This ICEMAC call sets up your installation parameter defaults for batch program invoked DFSORT applications.
- 02** This sets 3380 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 29 on page 120, Figure 30 on page 121, Figure 31 on page 122, Figure 32 on page 123, Figure 33 on page 124, and Figure 34 on page 125 for the IBM-supplied defaults for all of the ICEMAC parameters.

ICEMAC TSO - TSO Direct Invocation Defaults

Figure 25 is a sample of the ICEMAC macro update statements to produce module ICEAM3, the TSO defaults module.


```

column
1          16          72 73   80

ICEMAC TSO,
           ENABLE=(TD1),          * 00550000          01
           NOMSGDD=CRITICAL,      * 00550100          02
           TMAXLIM=3145728,       * 00550200          03
           DYNALOC=(3390,3)       * 00550300          04
                                   00550400          05

```

Figure 25. Coding the ICEMAC Macro for ICEAM3

Line Explanation

- 01** This ICEMAC call sets up your installation parameter defaults for TSO directly invoked DFSORT applications.
- 02** Enables the ICETD1 time-of-day module 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 (3MB) both above and below 16MB 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 29 on page 120, Figure 30 on page 121, Figure 31 on page 122, Figure 32 on page 123, Figure 33 on page 124, and Figure 34 on page 125 for the IBM-supplied defaults for all of the ICEMAC parameters.

ICEMAC TSOINV - TSO Program Invocation Defaults

Figure 26 is a sample of the ICEMAC macro update statements to produce module ICEAM4, the TSOINV defaults module.

```

column
1          16          72 73   80

ICEMAC TSOINV,
           RESINV=20480,          * 00550000          01
           COBEXIT=COB2          * 00550100          02
                                   00550200          03

```

Figure 26. Coding the ICEMAC Macro for ICEAM4

Line Explanation

- 01** This ICEMAC call sets up your installation parameter defaults for TSO program invoked DFSORT applications.
- 02** DFSORT reserves 20480 bytes (20KB) for use by the invoking program while DFSORT is running.
- 03** COBOL E15 and E35 user exit routines are executed with either the VS COBOL II run-time library or the Language Environment run-time library.

ICEMAC Macro

No other standard defaults for TSO program invoked applications are overridden in this example. See Figure 29 on page 120, Figure 30 on page 121, Figure 31 on page 122, Figure 32 on page 123, Figure 33 on page 124, and Figure 34 on page 125 for the IBM-supplied defaults for all of the ICEMAC parameters.

ICEMAC TD1 - First Time-of-Day Module Defaults

Figure 27 is a sample of the ICEMAC macro update statements to produce module ICETD1, the first time-of-day defaults module.

```
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 27. Coding the ICEMAC Macro for ICETD1

Line	Explanation
01	This ICEMAC call sets up your installation parameter defaults for the first time-of-day module.
02	DFSORT uses these ICETD1 defaults on Saturday or Sunday whenever the ICEAMx module 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: <ul style="list-style-type: none">• 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 (6MB) both above and below 16MB 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 48MB of space for Blockset sort applications when doing so should improve performance.

No other standard defaults for the first time-of-day module are overridden in this example. See Figure 29 on page 120, Figure 30 on page 121, Figure 31 on page 122, Figure 32 on page 123, Figure 33 on page 124, and Figure 34 on page 125 for the IBM-supplied defaults for all of the ICEMAC parameters.

ICEMAC Defaults

Figure 29 on page 120, Figure 30 on page 121, Figure 31 on page 122, Figure 32 on page 123, Figure 33 on page 124, and Figure 34 on page 125, which were produced using the job shown in Figure 28, list the IBM-supplied defaults for each parameter for the eight installation modules. The IBM-supplied defaults are used for each

installation module unless you change them. The defaults for each installation module can be changed independently of the others.

A detailed description of each parameter follows Figure 35 on page 125.

Listing ICEMAC Defaults

You can use a job similar to the one in Figure 28 to list the ICEMAC installation defaults currently 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 28. ICETOOL DEFAULTS Job

The DEFAULTS report shows the values for each parameter as they are set in the ICEAM1-4 and ICETD1-4 modules loaded for the step.

Note: Be sure that the ICEAMx/ICETDx modules are loaded for the DEFAULTS run in the same way they will be loaded 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.

The ICEAM1-4 report is shown first, followed by the ICETD1-4 report.

ICEMAC Macro

DFSORT REL 14.0 INSTALLATION (ICEMAC) DEFAULTS

- 1 -

* IBM-SUPPLIED DEFAULT (ONLY SHOWN IF DIFFERENT FROM THE SPECIFIED DEFAULT)

ITEM	JCL (ICEAM1)	INV (ICEAM2)	TSO (ICEAM3)	TSOINV (ICEAM4)
RELEASE	14.0	14.0	14.0	14.0
MODULE	ICEAM1	ICEAM2	ICEAM3	ICEAM4
APAR LEVEL	PQ56402	PQ56402	PQ56402	PQ56402
COMPILED	03/06/02	03/06/02	03/06/02	03/06/02
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	COB1	COB1	COB1	COB1
DIAGSIM	NO	NO	NO	NO
DSA	32	32	32	32
DSPSIZE	MAX	MAX	MAX	MAX
DYNALOC	(SYSDA,4)	(SYSDA,4)	(SYSDA,4)	(SYSDA,4)
DYNAUTO	YES	YES	YES	YES
DYNSPC	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	MAX	MAX	MAX	MAX
EXPRES	0	0	0	0
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	33554432	33554432	33554432	33554432
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
MSGCON	NONE	NONE	NONE	NONE
MSGDDN	SYSOUT	SYSOUT	SYSOUT	SYSOUT
MSGPRT	ALL	ALL	ALL	ALL
NOMSGDD	QUIT	QUIT	QUIT	QUIT
ODMAXBF	2097152	2097152	2097152	2097152

Figure 29. Page 1 of the Output from the ICETOOL DEFAULTS Job Showing IBM-supplied Defaults

DFSORT REL 14.0 INSTALLATION (ICEMAC) DEFAULTS

- 1 -

* IBM-SUPPLIED DEFAULT (ONLY SHOWN IF DIFFERENT FROM THE SPECIFIED DEFAULT)

ITEM	JCL (ICEAM1)	INV (ICEAM2)	TSO (ICEAM3)	TSOINV (ICEAM4)
OUTREL	YES	YES	YES	YES
OUTSEC	YES	YES	YES	YES
OVERRGN	65536	16384	65536	16384

DFSORT REL 14.0 INSTALLATION (ICEMAC) DEFAULTS

- 2 -

* IBM-SUPPLIED DEFAULT (ONLY SHOWN IF DIFFERENT FROM THE SPECIFIED DEFAULT)

ITEM	JCL (ICEAM1)	INV (ICEAM2)	TSO (ICEAM3)	TSOINV (ICEAM4)
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	R16	R16	R16	R16
STIMER	YES	YES	YES	YES
SVC	109	109	109	109
SZERO	YES	YES	YES	YES
TEXTIT	NO	NO	NO	NO
TMAXLIM	4194304	4194304	4194304	4194304
TRUNC	RC0	RC0	RC0	RC0
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
VSAMBSP	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	NO	NO	NO	NO

Figure 30. Page 1 (continued) and Page 2 of the Output from the ICETOOL DEFAULTS Job Showing IBM-supplied Defaults

ICEMAC Macro

DFSORT REL 14.0 INSTALLATION (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 31. Page 3 of the Output from the ICETOOL DEFAULTS Job Showing IBM-supplied Defaults

DFSORT REL 14.0 INSTALLATION (ICEMAC) DEFAULTS

- 4 -

* IBM-SUPPLIED DEFAULT (ONLY SHOWN IF DIFFERENT FROM THE SPECIFIED DEFAULT)

ITEM	TD1 (ICETD1)	TD2 (ICETD2)	TD3 (ICETD3)	TD4 (ICETD4)
RELEASE	14.0	14.0	14.0	14.0
MODULE	ICETD1	ICETD2	ICETD3	ICETD4
APAR LEVEL	PQ56402	PQ56402	PQ56402	PQ56402
COMPILED	03/06/02	03/06/02	03/06/02	03/06/02
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
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
COBEXIT	COB1	COB1	COB1	COB1
DIAGSIM	NO	NO	NO	NO
DSA	32	32	32	32
DSPSIZE	MAX	MAX	MAX	MAX
DYNALOC	(SYSDA,4)	(SYSDA,4)	(SYSDA,4)	(SYSDA,4)
DYNAUTO	YES	YES	YES	YES
DYNSPC	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	MAX	MAX	MAX	MAX
EXPRES	0	0	0	0
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	33554432	33554432	33554432	33554432
LIST	YES	YES	YES	YES
LISTX	YES	YES	YES	YES
LOCALE	NONE	NONE	NONE	NONE
MAXLIM	1048576	1048576	1048576	1048576

Figure 32. Page 4 of the Output from the ICETOOL DEFAULTS Job Showing IBM-supplied Defaults

ICEMAC Macro

DFSORT REL 14.0 INSTALLATION (ICEMAC) DEFAULTS

- 4 -

* IBM-SUPPLIED DEFAULT (ONLY SHOWN IF DIFFERENT FROM THE SPECIFIED DEFAULT)

ITEM	TD1 (ICETD1)	TD2 (ICETD2)	TD3 (ICETD3)	TD4 (ICETD4)
MINLIM	450560	450560	450560	450560
MSGCON	NONE	NONE	NONE	NONE
MSGDDN	SYSOUT	SYSOUT	SYSOUT	SYSOUT

DFSORT REL 14.0 INSTALLATION (ICEMAC) DEFAULTS

- 5 -

* IBM-SUPPLIED DEFAULT (ONLY SHOWN IF DIFFERENT FROM THE SPECIFIED DEFAULT)

ITEM	TD1 (ICETD1)	TD2 (ICETD2)	TD3 (ICETD3)	TD4 (ICETD4)
MSGPRT	ALL	ALL	ALL	ALL
NOMSGDD	QUIT	QUIT	QUIT	QUIT
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	R16	R16	R16	R16
STIMER	YES	YES	YES	YES
SVC	109	109	109	109
SZERO	YES	YES	YES	YES
TEXTIT	NO	NO	NO	NO
TMAXLIM	4194304	4194304	4194304	4194304
TRUNC	RC0	RC0	RC0	RC0
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
VSAMBSP	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	NO	NO	NO	NO

Figure 33. Page 4 (continued) and Page 5 of the Output from the ICETOOL DEFAULTS Job Showing IBM-supplied Defaults

DFSORT REL 14.0 INSTALLATION (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 34. Page 6 of the Output from the ICETOOL DEFAULTS Job Showing IBM-supplied Defaults

If the installation default value for a parameter is different from the IBM-supplied default value, the IBM-supplied default value is shown below the installation default value. Figure 35 is an example of what the output from the ICETOOL DEFAULTS job looks like if some of the defaults have been changed from the IBM-supplied defaults.

DFSORT REL 14.0 INSTALLATION (ICEMAC) DEFAULTS - 1 -

* IBM-SUPPLIED DEFAULT (ONLY SHOWN IF DIFFERENT FROM THE SPECIFIED DEFAULT)

ITEM	JCL (ICEAM1)	INV (ICEAM2)	TSO (ICEAM3)	TSOINV (ICEAM4)
RELEASE	14.0	14.0	14.0	14.0
MODULE	ICEAM1	ICEAM2	ICEAM3	ICEAM4
APAR LEVEL	PQ56402	PQ56402	PQ56402	PQ56402
COMPILED	03/06/02	03/06/02	03/06/02	03/06/02
ENABLE	NONE	NONE	NONE	NONE
ABCODE	MSG	99 * MSG	MSG	99 * 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		
.				
.				
.				

Figure 35. Output from the ICETOOL DEFAULTS Job Showing that Some of the Defaults Have Been Changed

Chapter 6. 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 modules provide an easier and more comprehensive method for time-of-day option controls. See Chapter 5, “Changing the Installation Defaults—ICEMAC Macro” on page 55 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.
- 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 131.

What Installation- and Run-Time Information Can ICEIEXIT Examine?

Installation-time Information

ICEIEXIT can examine these ICEMAC options:

DSPSIZE
ERET
HIPRMAX
MINLIM
MAXLIM
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 16MB virtual
- The options SIZE/MAINSIZE, RESALL/RESINV, ABEND/RC16, VERIFY/NOVERIFY, HIPRMAX, DSPSIZE, and ODMAXBF.

See details of the parameter list interfaces in the sections “Contents of the Installation-Time Options List” on page 129 and “Contents of the Run-Time Options List” on page 131 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 129 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 50 for more information on installing an ICEIEXIT.
3. Change installation defaults with the ICEMAC macro to specify IEXIT=YES.

In addition, be aware of the following:

- If ICEIEXIT is installed resident, it must be reentrant.
- ICEIEXIT must reside and run below 16MB 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:

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	

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 ICEMAC options SIZE, MAXLIM, TMAXLIM, and MINLIM. DFSORT also looks at whether storage can be allocated above 16MB virtual. See the Flags2 description under “Contents of the Run-Time Options List” on page 131 for details of flag settings.

Controlling Resources Using ICEEXIT

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 ICEMAC options RESALL and RESINV, and whether SIZE=MAX is in effect.

Flags (4 bytes)

Bit 0 = 1:

ICEMAC option ERET=ABEND is set

Bit 0 = 0:

ICEMAC option ERET=RC16 is set

Bit 1 = 1:

ICEMAC option VERIFY=YES is set

Bit 1 = 0:

ICEMAC option VERIFY=NO is set

Bit 2 = 1:

ICEMAC option SIZE=MAX is set

Bit 2 = 0:

ICEMAC option SIZE=n is set

Bits 3-4 =

Not used

Bit 5 = 1:

ICEMAC option HIPRMAX=OPTIMAL is set.

Bit 5 = 0:

ICEMAC option HIPRMAX=n is set.

Bits 6-31 =

For future use

MINLIM (4 bytes)

The value for ICEMAC option MINLIM.

MAXLIM (4 bytes)

The value for ICEMAC option MAXLIM.

TMAXLIM (4 bytes)

The value for ICEMAC option TMAXLIM.

HIPRMAX (2 bytes)

When Flags bit 5 is 1, ICEMAC 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 Hiperspace to be committed for Hipersorting. This value is determined from ICEMAC option HIPRMAX=n, or from ICEMAC option HIPRMAX=p% and the "Total Hiperspace storage" value shown in the Run-Time Options List.

DSPSIZE (2 bytes)

The value specified by ICEMAC 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 ICEMAC option ODMAXBF. A value of 262144 (256KB) to 16777216 (16MB) corresponds to ODMAXBF=n.

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	
Total Hiperspace Storage	

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 ICEMAC options SIZE, MAXLIM, TMAXLIM, and MINLIM. DFSORT also checks whether storage can be allocated above 16MB 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 (88KB) is specified, DFSORT uses 90112 (88KB) as the new value. If a new value greater than 2097152000 (2000MB) is specified, DFSORT uses 2097152000 (2000MB) 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

Controlling Resources Using ICEEXIT

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 16MB virtual

Bit 6 = 0:
Storage must be allocated below 16MB virtual

Bit 7 = 1:
OUTFIL processing is being used

Bit 7 = 0:
OUTFIL processing is not being used

Bits 8–31 =
For future use

Controlling Resources Using ICEIEXIT

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 "Total Hiperspace 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 above. 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 (256KB) and 16777216 (16MB), 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 (256KB) is specified, DFSORT uses 262144 (256KB) as the new value. If a new value of more than 16777216 (16MB) is specified, DFSORT uses 16777216 (16MB) as the new value. The new value is reported in appropriate messages as the ODMAXBF value.

Total Hiperspace storage (4 bytes)

This value specifies the amount of configured expanded storage on the system, in pages. For 64-bit real mode, this value specifies the portion of central storage DFSORT considers for the use of Hipersorting, in pages.

The value in this field cannot be changed.

A Sample ICEIEXIT Routine

Figure 36 on page 135 is an example of a simple ICEIEXIT routine, which does the following:

Controlling Resources Using ICEIEXIT

- Sets the storage available to DFSORT for the weekly payroll jobs (jobname of 'PAYR*' and stepname of 'WEEKLY') to 24MB.
- Ensures that the storage available to DFSORT for copy applications is not greater than 6MB.

The ICEIEXIT routine in this example is provided as part of the sample job ICEXREC in the SICESAMP library. See "Using Initialization and Termination Exits" on page 50 for details on how to install this sample routine.

Controlling Resources Using ICEIEXIT

```

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 WORKKOK IF YES, THEN CONTINUE
  RETURN (14,12) ELSE, RETURN TO DFSORT
WORKKOK 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 TIOCSSTEP(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 TIOCSSTEP+8.

```

Figure 36. Sample ICEIEXIT Routine (Part 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 36. Sample ICEIEXIT Routine (Part 2 of 2)

Chapter 7. 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 DFSORT's 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 ended.

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.

Collecting Statistical Data

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 ICEMAC 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) SA22-7630*

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 ICEMAC or OPTION 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 ICEMAC or OPTION 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 145.

Notes:

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 138 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 C, “SMF Type-16 Record” on page 197.

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).

Collecting Statistical Data

- 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

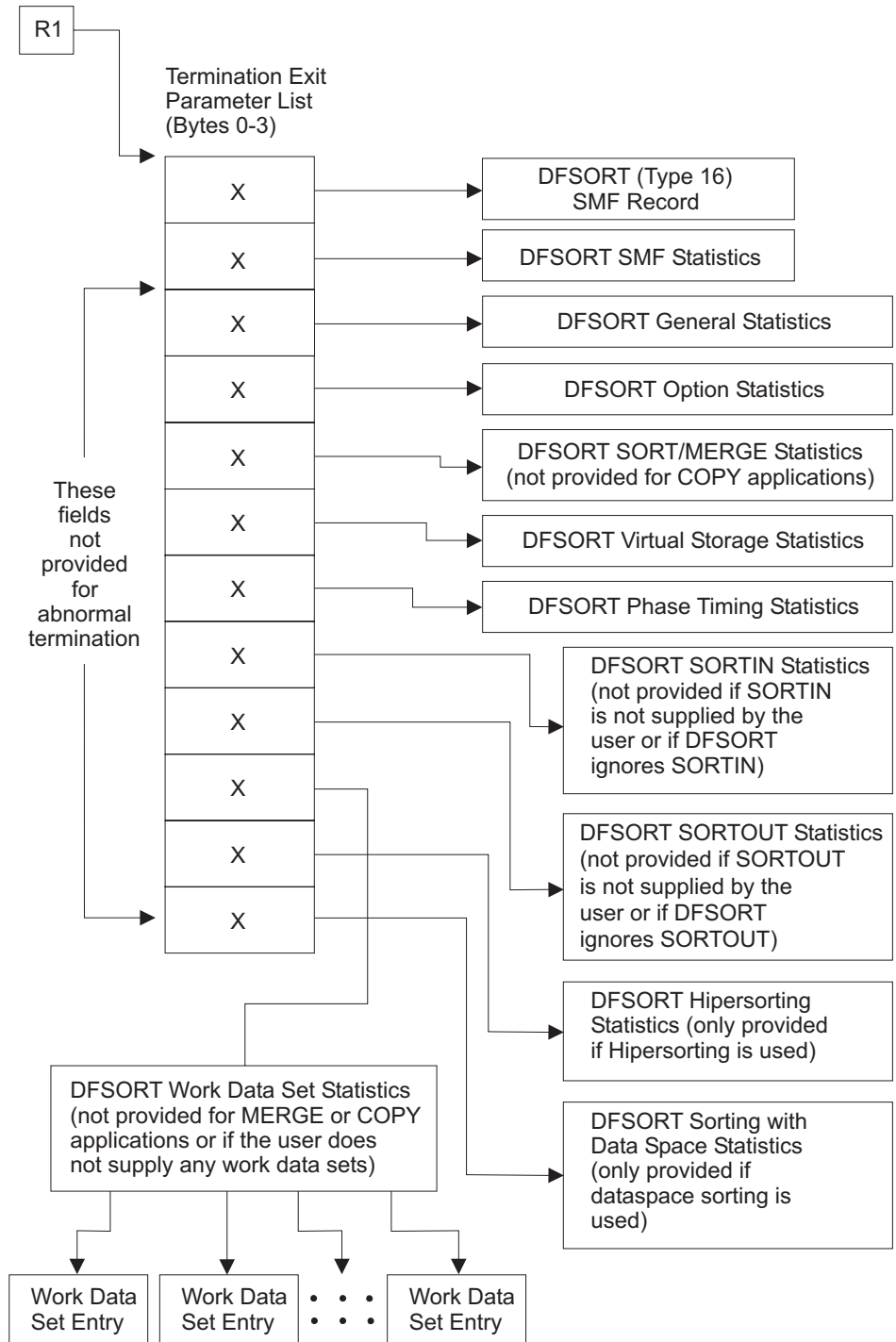
If your site collects SMF type-16 records created by different releases of DFSORT, you can identify the release that created a specific SMF type-16 record by examining ICERELNM, a 4-byte EBCDIC Release Level Identifier in the record.

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 D, "Termination Exit (ICETEXIT) Parameters" on page 213.

On entry to ICETEXIT, general register 1 contains the address of the parameter list shown in Figure 37 on page 142.

Collecting Statistical Data



(One Work Data Set Entry is provided for each work data set.)

Figure 37. 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 145 for a description of how you can use the ICEMAC 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
- STIMER=YES is in effect
- 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 16MB 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 16MB virtual.

Phase-Timing Statistics

SRB, TCB, and elapsed times are provided for each DFSORT phase. If STIMER=NO, TCB time is not provided.

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.

Notes:

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.

Collecting Statistical Data

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 50 for more information about installing an ICETEXIT.
3. Reinstall the ICEMAC installation options with TEXTIT=YES.

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 D, “Termination Exit (ICETEXIT) Parameters” on page 213.
- If ICETEXIT is installed resident, it must be reentrant.
- ICETEXIT must reside and execute below 16MB 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 ICEMAC SMF parameter, you can control whether your ICETEXIT is passed *short* or *full* records. Table 8 describes the precise actions that DFSORT takes for different settings of the ICEMAC TEXTIT and SMF parameters.

Table 8. Controlling the Type of SMF Record DFSORT Passes to ICETEXIT

ICETEXIT	SMF=FULL	SMF=SHORT	SMF=NO
<p>TEXTIT=YES</p> <p>Termination exit receives control.</p>	<p>If the run is successful, a full (subtype 2) record is passed to SMF and to the termination exit.</p> <p>If the run is unsuccessful, a short (subtype 3) record is passed to SMF and to the termination exit.</p>	<p>If the run is successful, a short (subtype 1) record is passed to SMF and to the termination exit.</p> <p>If the run is unsuccessful, a short (subtype 3) record is passed to SMF and to the termination exit.</p>	<p>If the run is successful, a short (subtype 1) record is passed to the termination exit. No record is passed to SMF.</p> <p>If the run is unsuccessful, a short (subtype 3) record is passed to the termination exit. No record is passed to SMF.</p>
<p>TEXTIT=NO</p> <p>Termination exit does not receive control.</p>	<p>If the run is successful, a full (subtype 2) record is passed to SMF. No record is passed to the termination exit.</p> <p>If the run is unsuccessful, a short (subtype 3) record is passed to SMF. No record is passed to the termination exit.</p>	<p>If the run is successful, a short (subtype 1) record is passed to SMF. No record is passed to the termination exit.</p> <p>If the run is unsuccessful, a short (subtype 3) record is passed to SMF. No record is passed to the termination exit.</p>	<p>No record is passed to SMF or to the termination exit.</p>

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

Collecting Statistical Data

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 38 presents a nonexecutable overview of how you might code ICETEXIT if you intend to process all the statistics that DFSORT provides ICETEXIT.

Figure 39 on page 151 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 TEXT parameter list
                USING  ICETPAR,R11     Make TEXT parm list addressable
                L       R10,ICETSMFA    Load address of DFSORT SMF record
                USING  ICESMFH,R10     Make the SMF header section addr

```

Figure 38. A Nonexecutable ICETEXIT (Part 1 of 5)

```

. . . . . 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
. . . . .
. . . . . 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
*

```

Figure 38. A Nonexecutable ICETEXIT (Part 2 of 5)

Collecting Statistical Data

```

*****
*                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
*
*****
*                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
*

```

Figure 38. A Nonexecutable ICETEXIT (Part 3 of 5)

Collecting Statistical Data

```

*****
*          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 38. A Nonexecutable ICETEXIT (Part 4 of 5)

Collecting Statistical Data

```

*****
*          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       CLOSDOWN     If not, go perform closedown function
        USING    ICEDATAS,R10  Make data space stats addressable
        . . . . .
        . . . . .
        . . . . .
        DROP     R10           Drop data space statistics
*
*****
*          EXIT LINKAGE AND ANY NECESSARY CLOSE DOWN OPERATIONS      *
*****
*
CLOSDOWN DS      0H
        . . . . .
        . . . . .
        . . . . .
*
        END      ICETEXIT

```

Figure 38. A Nonexecutable ICETEXIT (Part 5 of 5)

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 39. An Executable ICETEXIT (Part 1 of 7)

Collecting Statistical Data

```

*
*****
*   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 TEXTIT 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 39. An Executable ICETEXIT (Part 2 of 7)

Collecting Statistical Data

```

*
*****
*          COLLECT ALL DESIRED STATISTICS          *
*****
*
      USING    ICETPAR,R10      Make TEXT 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,ICETVRT       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 TEXT 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 39. An Executable ICETEXIT (Part 3 of 7)

Collecting Statistical Data

```

*
*****
*          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 39. An Executable ICETEXIT (Part 4 of 7)

Collecting Statistical Data

```

*
*****
*           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,DALXTPS     Load address of text unit pointers
      ST      R4,DALXPT      Store pointer in request block
      LA      R4,DALDDN      Load address of DDNAME text unit
      ST      R4,DALXTPS     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,DALXTPS+4   Store address in list of text units
      MVI    DALXTPS+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 (Part 5 of 7)

Collecting Statistical Data

```

*
*****
*           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'RICHRDK.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 39. An Executable ICETEXIT (Part 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
OPENSIZ EQU     *-OPENCMD
CLOSECMD CLOSE   (COLLECT,,),MF=L
CLOSSIZ 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(OPENSIZ)  Space for OPEN list
CLOSLIST DS     XL(CLOSSIZ)  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 39. An Executable ICETEXIT (Part 7 of 7)

Chapter 8. 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 you do not want to pass a particular value to DFSORT, leave it as zero.

LRECL is the logical record length for the tape data set.

BLKSIZE is the block size for the tape data set.

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.

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'.

- 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.

Part 3. Appendixes

Appendix A. The ICEMSGSGS Macro

The following figure is a copy of the ICEMSGSGS macro. You can modify the message texts in this macro if you want DFSORT to issue messages in a language other than English, or if you want to change the wording of some of the messages. See “Modifying Message Texts” on page 51 for more information.

The ICEMSGSG Macro

```

MACRO                                00050000
ICEMSGSG                              00100000
/*PROPRIETARY V3 STATEMENT           00142800
/*LICENSED MATERIALS - PROPERTY OF IBM 00185600
/*"RESTRICTED MATERIALS OF IBM"      00228400
/*5740-SM1                            00271200
/*(C) COPYRIGHT 1973,1998 IBM CORP.   00314000
/*END PROPRIETARY V3 STATEMENT       00356800
/*/*/*/*/*/*                          00400000
/*      (C) COPYRIGHT WATSON M CONNER INC. 1983 00450000
/*      ALL RIGHTS RESERVED            00500000
/*/*/*/*/*/*                          00600000
* 00650000
$MAC(ICEMSGSG) PROD(5740-SM1)        * 00700000
* 00750000
FUNCTION: CONTAINS ALL THE MESSAGE TEXT FOR MESSAGES ISSUED
        BY THE STANDARD TECHNIQUES. EACH MESSAGE TEXT
        CONSISTS OF FIXED CHARACTERS AND INSERT FIELDS. THE
        INSERT FIELDS SUPPLY VARIABLE INFORMATION IN THE
        FOLLOWING FORMS:
        1. ? - REPRESENTS CHARACTER DATA
        2. % - REPRESENTS FIXED LENGTH NUMERIC DATA
        3. ! - REPRESENTS VARIABLE LENGTH NUMERIC DATA
* 00800000
* 00850000
* 00900000
* 00950000
* 01000000
* 01050000
* 01100000
* 01150000
* 01200000
* 01250000
* 01300000
NOTES:
* 01350000
DEPENDENCIES: THE FOLLOWING MODULES MUST BE RE-ASSEMBLED AND
        RE-LINKEDITED TO APPLY CHANGES MADE TO THIS
        MACRO:
        ICEMCRE,ICEMDEC,ICEMDED,ICEMDEF,ICEMDEV,
        ICEMERR,ICEMLM,ICEMMON,ICEMMOS,ICEMPAR,
        ICEMSGGA,ICEMTRY,ICEMUTL
* 01400000
* 01450000
* 01500000
* 01550000
* 01600000
* 01700000
RESTRICTIONS: THE FOLLOWING RULES APPLY WHEN MODIFYING MESSAGES:
        1. THE AND SIGN MUST NOT BE USED.
        2. MESSAGE MACROS MUST NOT BE ADDED OR DELETED.
        3. THE MACRO CALL NAME 'ICEMSET' AND THE
        MESSAGE NUMBER MUST NOT BE CHANGED .
        4. FIXED CHARACTERS MAY BE CHANGED, ADDED,
        DELETED, OR RELOCATED.
        5. INSERT FIELD CHARACTERS ?, %, AND ! MUST
        NOT BE CHANGED, ADDED, OR DELETED, OR USED
        AS NON-SPECIAL CHARACTERS.
        6. INSERT FIELDS MUST NOT BE MADE CONTIGUOUS.
        7. INSERT FIELDS MAY BE RELOCATED. HOWEVER,
        THE POSITION OF INSERT FIELDS RELATIVE TO
        EACH OTHER MUST NOT BE CHANGED.
        8. THE TOTAL LENGTH OF EACH TEXT STRING MUST NOT
        EXCEED 110 BYTES.
        9. ASSEMBLER MACRO CONTINUATION RULES APPLY IF
        CONTINUATION IS NECESSARY.
        10. EXPANSION OF MESSAGE TEXTS BY MORE THAN 20%
        MAY CAUSE THE RE-ASSEMBLY OF ONE OR MORE
        MESSAGE CSECTS TO FAIL.
* 01750000
* 01800000
* 01850000
* 01900000
* 01950000
* 02000000
* 02050000
* 02100000
* 02150000
* 02200000
* 02250000
* 02300000
* 02350000
* 02400000
* 02450000
* 02500000
* 02550000
* 02600000
* 02650000
* 02700000
* 02750000
* 02800000

```

Figure 41. ICEMSGSG Macro (Part 1 of 10)

```

* 02850000
* 02900000
ICEMSET 0,'- CONTROL STATEMENTS FOR 5740-SM1, DFSORT REL 14.0 *02950000
- ????:? ON ??? ??? %%, ??? -' 03000000
ICEMSET 1,'TEXT BEGINS IN WRONG COLUMN' 03050000
ICEMSET 2,'DUPLICATE OR CONFLICTING ???????? STATEMENT' 03100000
ICEMSET 3,'CONTINUATION LINE MISSING' 03150000
ICEMSET 5,'STATEMENT DEFINER ERROR' 03200000
ICEMSET 6,'OPERAND DEFINER ERROR' 03250000
ICEMSET 7,'SYNTAX ERROR' 03300000
ICEMSET 8,'FIELD OR VALUE EXCEEDS MAXIMUM ALLOWABLE CHARACTERS*03350000
' 03400000
ICEMSET 10,'NO SORT OR MERGE CONTROL STATEMENT' 03450000
ICEMSET 11,'DUPLICATE OR CONFLICTING OPERANDS ON THE OPTION ST*03500000
ATEMENT' 03550000
ICEMSET 12,'MISSING FIELDS OPERAND DEFINER' 03600000
ICEMSET 13,'INVALID SORT OR MERGE STATEMENT OPERAND' 03650000
ICEMSET 14,'DUPLICATE SORT OR MERGE STATEMENT OPERAND' 03700000
ICEMSET 15,'VARIABLE RECORD TOO SHORT' 03750000
ICEMSET 16,'INVALID FIELDS OPERAND VALUE' 03800000
ICEMSET 17,'CONTROL FIELD DISPLACEMENT OR LENGTH VALUE ERROR' 03850000
ICEMSET 18,'INVALID FORMAT' 03900000
ICEMSET 19,'MEMBER NAME NOT SPECIFIED FOR ????????' 03950000
ICEMSET 20,'INVALID RECORD STATEMENT OPERAND' 04000000
ICEMSET 21,'NO TYPE OPERAND' 04050000
ICEMSET 22,'RECORD FORMAT NOT F, V OR D' 04100000
ICEMSET 23,'NO LENGTH OPERAND' 04150000
ICEMSET 24,'RECORD LENGTH VALUE ERROR' 04200000
ICEMSET 25,'RECORD COUNT OFF' 04250000
ICEMSET 26,'SMF RECORD NOT WRITTEN TO THE SMF DATA SET(RC=!!!)*04300000
' 04350000
ICEMSET 27,'END OF ???????? FIELD BEYOND MAXIMUM RECORD LENGTH*04383300
' 04416600
ICEMSET 29,'IMPROPER EXIT' 04450000
ICEMSET 30,'MULTIPLY DEFINED EXITS' 04500000
ICEMSET 32,'EXIT E61 REQUIRED' 04550000
ICEMSET 33,'CONTROL FIELD SEQUENCE INDICATOR E REQUIRED' 04600000
ICEMSET 34,'MODS STATEMENT OPERAND ERROR' 04650000
ICEMSET 39,'INSUFFICIENT MAIN STORAGE - ADD AT LEAST !!!!!!!!!K *04700000
BYTES' 04750000
ICEMSET 40,'INSUFFICIENT WORK UNITS' 04800000
ICEMSET 42,'UNIT ASSIGNMENT ERROR ????????' 04850000
ICEMSET 43,'INVALID DATA SET ATTRIBUTES SPECIFIED ???????? ??*04900000
????' 04950000
ICEMSET 44,'EXIT E% INVALID OPTION' 05000000
ICEMSET 46,'SORT CAPACITY EXCEEDED - RECORD COUNT !!!!!!!!!!!!!*05050000
!!!!!!!!' 05100000
ICEMSET 47,'RECORD COUNT OFF, SPECIFIED !!!!!!!!!!!!!!!!!!!!! ,*05150000
RECEIVED !!!!!!!!!!!!!!!!!!!!!' 05200000
ICEMSET 51,'UNENDING MERGE' 05250000
ICEMSET 52,'END OF DFSORT' 05300000
ICEMSET 53,'OUT OF SEQUENCE' 05350000
ICEMSET 54,'RECORDS - IN: !!!!!!!!!!!!!!!!!!!!!!!!!!!!!, OUT: !!!!!!!!!!!!!*05400000
!!!!!!!!!!!!' 05450000
ICEMSET 55,'INSERT !!!!!!!!!!!!!!!!!!!!!!!!!!!!!, DELETE !!!!!!!!!!!!!!!!!!!!!*05500000
!!!!!!!!' 05550000
ICEMSET 56,'???????? NOT DEFINED' 05600000
ICEMSET 59,'RECORD LENGTH INVALID FOR ????????' 05650000
ICEMSET 60,'INVALID ?????????? PARAMETER' 05700000

```

Figure 41. ICEMSGS Macro (Part 2 of 10)

The ICEMSGSGS Macro

```

ICEMSET 61,'I/O ERROR, DD ????????, DEV ????, ECB ??, CSW ????,*05750000
    SENSE ????' 05800000
ICEMSET 62,'LINK-EDIT ERROR' 05850000
ICEMSET 63,'OPEN ERROR ????????' 05900000
ICEMSET 65,'PROBABLE DECK STRUCTURE ERROR' 05950000
ICEMSET 67,'INVALID PARAMETER IN JCL EXEC PARM OR INVOKED PARM*06000000
    LIST' 06050000
ICEMSET 68,'OUT OF SEQUENCE SORTIN??' 06100000
ICEMSET 69,'INVALID SIGN' 06150000
ICEMSET 70,'EXIT ??????? NOT FOUND' 06200000
ICEMSET 71,'INVALID RETURN CODE FROM EXIT E%% - RC=!!!!!!!!!!!!,*06250000
    CASE=??' 06300000
ICEMSET 72,'FIELD NOT WITHIN MINIMUM RECORD LENGTH' 06350000
ICEMSET 73,'VARIABLE RECORD TOO LONG' 06400000
ICEMSET 74,'RECORD LENGTH L1 OR L3 OVERRIDDEN' 06450000
ICEMSET 75,'VSAM CB ERROR (!!) AT ??????' 06500000
ICEMSET 76,'VSAM INPUT ERROR ?(!!!) ????????' 06550000
ICEMSET 77,'VSAM OUTPUT ERROR ?(!!!) ????????' 06600000
ICEMSET 78,'VSAM OPEN ERROR (!!!) ????????' 06650000
ICEMSET 79,'VSAM CLOSE ERROR (!!!) ????????' 06700000
ICEMSET 80,'IN MAIN STORAGE SORT' 06750000
ICEMSET 81,'FIRST COMMUNICATION AREA SPACE EXCEEDED' 06800000
ICEMSET 82,'CHECKPOINT CANCELLED' 06850000
ICEMSET 83,'RESOURCES WERE UNAVAILABLE FOR DYNAMIC ALLOCATION *06883300
    OF WORK DATA SETS (????)' 06916600
ICEMSET 84,'???? ACCESS METHOD USED FOR ????????' 06950000
ICEMSET 85,'SECOND COMMUNICATION AREA SPACE EXCEEDED - ADD AT *07000000
    LEAST !!!K BYTES TO REGION' 07050000
ICEMSET 88,'?????????.?????????.?????????, INPUT LRECL = !!!!!, B*07100000
    LKSIZE = !!!!!!!!!!!!!, TYPE = ?????' 07150000
ICEMSET 89,'?????????.?????????.?????????, INPUT LRECL = !!!!!, T*07200000
    YPE = ?????' 07250000
    ICEMSET 90,'OUTPUT LRECL = !!!!!, BLKSIZE = !!!!!!!!!!!!!, TYPE *07329990
    = ????? ??????' 07359980
ICEMSET 91,'OUTPUT LRECL = !!!!!, TYPE = ?????' 07400000
ICEMSET 92,'MAIN STORAGE = (!!!!!!!!!!!!!,!!!!!!!!!!!!,!!!!!!!!!!!!)' 07450000
ICEMSET 93,'MAIN STORAGE = (MAX,!!!!!!!!!!!!,!!!!!!!!!!!!)' 07500000
ICEMSET 94,'SMF RECORD NOT WRITTEN' 07550000
ICEMSET 95,'INVALID OPTION STATEMENT OPERAND' 07600000
ICEMSET 96,'SUCCESSFUL RECOVERY FROM B37 ABEND(S) FOR WORK DAT*07650000
    A SET(S)' 07700000
ICEMSET 98,'AVERAGE RECORD LENGTH = !!!!!!!!!!!!!!! BYTES' 07750000
ICEMSET 99,'BLDL FAILED FOR ??????? DATA SET' 07800000
ICEMSET 101,'???????????? STATEMENT NOT SUPPORTED FOR TECHNIQU*07900000
    E USED' 07950000
ICEMSET 102,'MISSING COND OPERAND DEFINER' 08000000
ICEMSET 103,'INVALID INCLUDE OR OMIT STATEMENT OPERAND' 08050000
ICEMSET 104,'INVALID INREC OR OUTREC STATEMENT OPERAND' 08100000
ICEMSET 105,'INVALID SORT, MERGE, OR SUM STATEMENT OPERAND' 08150000
ICEMSET 106,'DUPLICATE INCLUDE OR OMIT STATEMENT OPERAND' 08200000
ICEMSET 107,'DUPLICATE INREC OR OUTREC STATEMENT OPERAND' 08250000
ICEMSET 108,'DUPLICATE SUM STATEMENT OPERAND' 08300000
ICEMSET 109,'SUM FIELD DISPLACEMENT OR LENGTH VALUE ERROR' 08350000
ICEMSET 110,'VERIFY NOT USED WITH SUM' 08400000
ICEMSET 111,'REFORMATTING FIELD ERROR' 08450000
ICEMSET 112,'EQUALS NOT USED WITH SUM' 08500000
ICEMSET 113,'COMPARISON FIELD ERROR' 08550000

```

Figure 41. ICEMSGSGS Macro (Part 3 of 10)

```

ICEMSET 114,'INVALID COMPARISION'                                08600000
ICEMSET 115,'INSUFFICIENT MAIN STORAGE'                          08650000
ICEMSET 116,'SORT CAPACITY EXCEEDED'                             08700000
ICEMSET 117,'I/O ERROR ?????????????????????????????????????*08750000
?????????????????????????????????????????????????????????' 08800000
ICEMSET 118,'UNKNOWN FILE SIZE - FILSZ=EN MAY IMPROVE RESOURCE*08809990
USAGE AND PERFORMANCE'                                         08819980
ICEMSET 119,'SUM FIELD OVERLAPS CONTROL FIELD'                   08850000
ICEMSET 120,'EXIT E?? IGNORED'                                   08900000
ICEMSET 121,'FILE SIZE IS TOO LARGE'                             08950000
ICEMSET 122,'????????? VALUE INVALID'                           09000000
ICEMSET 123,'CKPT OR CHKPT OPTION IGNORED'                      09050000
ICEMSET 124,'SUM FIELD OVERLAPS RECORD DESCRIPTOR WORD'        09100000
ICEMSET 125,'SUM FIELD OVERLAPS SUM FIELD'                       09150000
ICEMSET 126,'INCONSISTENT ???????? REFORMATTING FIELDS FOUND' 09200000
ICEMSET 127,'OPTIONS: OVFL=RC??,PAD=RC??,TRUNC=RC??,SPANINC=R*09216600
C??,VLSMCP=?,SZERO=?,RESET=?,VSAMEMT=?,DYNMPC=!!!!!'          09233200
ICEMSET 128,'OPTIONS: SIZE=!!!!!!!!!!!!,MAXLIM=!!!!!!!!!!!!,MINLIM*09250000
=!!!!!!!!!!!!,EQUALS=?,LIST=?,ERET=?????,MSGDDN=?????????'    09300000
ICEMSET 129,'OPTIONS: VIO=?,RESDNT=????,SMF=?????,WRKSEC=?,OUT*09350000
SEC=?,VERIFY=?,CHALT=?,DYNALOC=????????????????,ABCODE=???*09383300
'                                                                09416600
ICEMSET 130,'OPTIONS: RESALL=!!!!!!!!!!!!,RESINV=!!!!!!!!!!!!,SVC*09450000
????,CHECK=?,WRKREL=?,OUTREL=?,CKPT=?,STIMER=?,COBEXIT=C*09500000
OB?'                                                            09550000
ICEMSET 131,'OPTIONS: TMAXLIM=!!!!!!!!!!!!,ARESALL=!!!!!!!!!!!!,AR*09600000
ESINV=!!!!!!!!!!!!,OVERRGN=!!!!!!!!!!!!,CINV=?,CFW=?,DSA=!!!*09650000
!'                                                            09700000
ICEMSET 132,'OPTIONS: VLSHRT=?,ZDPRINT=?,IEXIT=?,TEXTIT=?,LISTX*09750000
=?,EFS=????????,EXITCK=?,PARMDDN=????????,FSZEST=?'          09800000
ICEMSET 133,'OPTIONS: HIPRMAX=????????,DSPSIZE=????,ODMAXBF=!!!*09833300
!!!!,SOLRF=?,VLLONG=?,VSAMIO=?'                                09866600
ICEMSET 134,'NUMBER OF BYTES SORTED: !!!!!!!!!!!!!!!!!!!!!!!' 09900000
ICEMSET 136,'SPACE FOR ???????? WAS READJUSTED TO CYLINDERS'   09950000
ICEMSET 137,'VSAM INPUT ERROR ?(!!!) ?????????????????????????*10000000
?????????????????????????????????????????????'             10050000
ICEMSET 138,'VSAM OUTPUT ERROR ?(!!!) ?????????????????????????*10100000
?????????????????????????????????????????????'             10150000
ICEMSET 139,'UNSUPPORTED VSAM DATA SET ATTRIBUTE FOR ????????' 10200000
ICEMSET 140,'END OF PARAMETERS FROM ???????? - SYSIN OR ???CN*10250000
TL/PARAMETER LIST CONTROL STATEMENTS FOLLOW'                   10300000
ICEMSET 141,'SPANNED RECORD ON ???????? COULD NOT BE ASSEMBLED' 10350000
ICEMSET 142,'????????? NOT FOUND - SYSOUT USED'                 10400000
ICEMSET 143,'????????????? ?????? TECHNIQUE SELECTED'         10450000
ICEMSET 144,'TOO MANY SUM FIELDS'                                10500000
ICEMSET 145,'SVC !!! IS AN INVALID USER SVC'                   10550000
ICEMSET 146,'END OF STATEMENTS FROM ???CNTL - PARAMETER LIST *10600000
STATEMENTS FOLLOW'                                             10650000
ICEMSET 147,'OPTION STATEMENT OPERAND SORTIN OR SORTOUT NOT AL*10700000
LOWED'                                                         10750000
ICEMSET 148,'????????? CONCATENATION NOT ALLOWED'              10800000
ICEMSET 149,'DFSORT IS NOT LICENSED FOR USE ON THIS SYSTEM - R*10812500
ETURN CODE !!!! AND REASON CODE !!!!'                          10825000
ICEMSET 150,'VLSHRT NOT USED FOR SORT, MERGE, INCLUDE, OMIT OR*10857490
SUM STATEMENT FIELDS'                                          10877480
ICEMSET 151,'TOO MANY ???????? COMPARISON FIELDS'              10912500
ICEMSET 152,'OVERFLOW DURING SUMMATION - RC=?'                 10956200
ICEMSET 153,'COBOL EXIT NOT SUPPORTED FOR TECHNIQUE USED'      11000000
ICEMSET 154,'STOPAFT NOT SUPPORTED FOR TECHNIQUE USED'         11050000
ICEMSET 155,'STOPAFT OR SKIPREC NOT APPLICABLE TO MERGE'       11100000
ICEMSET 156,'MAIN STORAGE ABOVE 16MB = (!!!!!!!!!!!!,!!!!!!!!!!!!)*11150000

```

Figure 41. ICEMSGS Macro (Part 4 of 10)

The ICEMSGSG Macro

```

' 11200000
ICEMSET 157,'EXEC PARM E15=COB OR E35=COB AND NO MODS EXIT' 11250000
ICEMSET 159,'MODULE ICECOB2 NOT FOUND' 11300000
ICEMSET 160,'COPY FUNCTION COULD NOT BE USED - REASON CODE IS *11350000
%%' 11400000
ICEMSET 161,'COBEXIT=COB2 AND COBOL E15 AND E35 EXITS WERE FOU*11450000
ND' 11500000
ICEMSET 162,'ICEIEXIT CHANGED ONE OR MORE OPTIONS IN EFFECT' 11550000
ICEMSET 164,'????? ABEND OCCURRED DURING EXECUTION OF ICETEXIT' 11600000
ICEMSET 165,'TOTAL WORK DATA SET TRACKS ALLOCATED: !!!!!!!! , *11650000
TRACKS USED: !!!!!!!!' 11700000
ICEMSET 168,'CENTURY WINDOW FOR Y2 FORMAT FIELDS IS FROM ????' *11716600
TO ????' 11733200
ICEMSET 170,'FORMAT OPERAND IGNORED' 11750000
ICEMSET 171,'SORTOUT LRECL OF !!!!! IS DIFFERENT FROM SORTIN(N*11758300
N) LRECL OF !!!!! - RC=?' 11766600
ICEMSET 172,'XSUM IS NOT SUPPORTED - USE ICETOOL SELECT IF APP*11768590
ROPRIATE' 11770580
ICEMSET 176,'????????? SERVICE ERROR (!!!!!!!!!!!!!-!!!!!!!!!!!!)!' 11775000
ICEMSET 177,'DASD CALCULATION SERVICES ERROR (!!!-!!!!!!!!!!!!)!' 11800000
ICEMSET 178,'REALLOCATING VIO SORTWK DATA SETS TO REAL SORTDK *11810000
DATA SETS USING UNIT=?????' 11820000
ICEMSET 179,'RESOURCES WERE UNAVAILABLE FOR VIO REALLOCATION O*11830000
F WORK DATA SETS (?????)' 11840000
ICEMSET 180,'HIPERSPACE STORAGE USED = !!!!!!!!!!!K BYTES' 11850000
ICEMSET 181,'HIPERSPACE READ ERROR (!!) ??????????' 11900000
ICEMSET 182,'HIPERSPACE WRITE ERROR (!!) ??????????' 11950000
ICEMSET 183,'CONDITIONS NOT MET TO USE THE SAME VSAM DATA SET *11959990
FOR ?????????? AND ??????????' 11969980
ICEMSET 184,'THE NUMBER OF RECORDS SORTED WITH EQUAL KEYS IS !*12000000
!!!!!!!!!!!!!!!!!!!!!!!!!!!!' 12050000
ICEMSET 185,'DFSORT TERMINATED WITH ?????? ABEND DURING PHASE ?x12100000
?' 12150000
ICEMSET 186,'DFSORT CONTINUED AFTER ERROR AND OUTPUT IS USEABLx12200000
E' 12250000
ICEMSET 187,'DFSORT SVC LEVEL INCOMPATIBLE WITH DFSORT PROGRAx12300000
M LEVEL' 12350000
ICEMSET 188,'DATA SPACE STORAGE USED = !!!!!!!!!!!K BYTES' 12400000
ICEMSET 189,'BLOCKSET REQUIRED BUT COULD NOT BE USED - REASON *12416600
CODE IS %%' 12433200
ICEMSET 190,'????????? DYNALLOC CALL FAILED - RETURN CODE !!!! *12438190
AND REASON CODE !!!!!' 12443180
ICEMSET 191,'DFSORT SVC NOT AVAILABLE - PERFORMANCE MAY BE DEG*12450000
RADED' 12500000
ICEMSET 192,'INVALID RETURN CODE FROM EXIT E%% WAS ACCEPTED' 12550000
ICEMSET 193,'ICEAM? ENVIRONMENT IN EFFECT - ICE??? INSTALLATIO*12566600
N MODULE SELECTED' 12583200
ICEMSET 195,'OVERFLOW DURING SUMMATION - RC=16' 12612400
ICEMSET 196,'SORTOUT LRECL OF !!!!! IS DIFFERENT FROM SORTIN(N*12635100
N) LRECL OF !!!!! - RC=16' 12657800
ICEMSET 197,'ONE OR MORE INCOMPLETE SPANNED RECORDS DETECTED A*12680500
ND ELIMINATED - RC=?' 12690200
ICEMSET 198,'DEBUG EQUCOUNT CANNOT BE USED WITHOUT WORK DATA S*12700000
ET' 12750000
ICEMSET 200,'IDENTIFIER FROM CALLING PROGRAM IS ????' 12800000
ICEMSET 201,'RECORD TYPE IS ? - DATA STARTS IN POSITION ?' 12819990
ICEMSET 202,'SYSUT2 LRECL OF !!!!! IS DIFFERENT FROM SYSUT1 LR*12850000
ECL OF !!!!! - RC=?' 12900000
ICEMSET 203,'ONE OR MORE RECORDS HAD SHORT SUM FIELDS' 12907990
ICEMSET 204,'INCOMPLETE SPANNED RECORD DETECTED ON ?????????? - *12916600
RC=16' 12933200

```

Figure 41. ICEMSGSG Macro (Part 5 of 10)

```

ICEMSET 205,'CONTROL WAS NOT PASSED TO ICETEXIT' 12950000
ICEMSET 210,'???????? : ???? USED, LRECL = !!!!!, BLKSIZE = !!*12951200
!!!!!!!, TYPE = ???? ?????' 12952400
ICEMSET 211,'OLD OUTFIL STATEMENT PROCESSING USED' 12953600
ICEMSET 212,'MATCH NOT FOUND FOR ???????? CHANGE FIELD AT POSI*12954800
TION !!!!!' 12956000
ICEMSET 213,'INVALID OUTFIL STATEMENT OPERAND' 12957200
ICEMSET 214,'DUPLICATE OR CONFLICTING OUTFIL STATEMENT OPERAND*12958790
S' 12959180
ICEMSET 215,'SPECIFIED FEATURE IS ONLY SUPPORTED BY OUTFIL STA*12959600
TEMENT' 12960800
ICEMSET 217,'!!!!!! BYTE VARIABLE RECORD IS LONGER THAN !!!!! B*12964400
YTE MAXIMUM FOR ????????' 12965600
ICEMSET 218,'!!!!!! BYTE VARIABLE RECORD IS SHORTER THAN !!!!! *12966800
BYTE MINIMUM FOR ????????' 12968000
ICEMSET 219,'DUPLICATE DDNAME ???????? IGNORED' 12969200
ICEMSET 220,'???????? COULD NOT BE USED - REASON CODE IS %%' 12970400
ICEMSET 221,'INVALID FIELD OR CONSTANT FOUND IN ???????? CONDI*12971990
TION !!!!!' 12972380
ICEMSET 222,'!!!!!! BYTE FIXED RECORD LENGTH IS NOT EQUAL TO !!*12972800
!!! BYTE LRECL FOR ????????' 12974000
ICEMSET 223,'REPORT FIELD ERROR' 12975200
ICEMSET 224,'???????? CANNOT BE USED FOR A REPORT - RECFM WITH*12976790
OUT 'A' SPECIFIED, OR VSAM DATA SET' 12977180
ICEMSET 225,'???????? : LRECL = !!!!!!!!!!! IS INVALID' 12977600
ICEMSET 226,'???????? : BLKSIZE = !!!!!!!!!!! IS INVALID (LREC*12978800
L = !!!!!)' 12980000
ICEMSET 227,'???????? : DELETED = !!!!!!!!!!!!!!!!!!!!!!!', REPORT*12981200
= !!!!!!!!!!!!!!!!!!!!!!!', DATA = !!!!!!!!!!!!!!!!!!!!!!!' 12982400
ICEMSET 228,'???????? : TOTAL IN = !!!!!!!!!!!!!!!!!!!!!!!', TOTAL*12983600
OUT = !!!!!!!!!!!!!!!!!!!!!!!' 12984800
ICEMSET 229,'???????? : LINES = !!! IN EFFECT, BUT !!!!!!!!!!! *12986000
LINES REQUIRED' 12987200
ICEMSET 230,'!!!!!! BYTE HEADER/TRAILER RECORD EXCEEDS !!!!! BY*12988400
TE LRECL FOR ????????' 12989600
ICEMSET 231,'STORAGE USED FOR OUTFIL : BELOW 16M = !!!!!!!!!!!,*12990800
ABOVE 16M = !!!!!!!!!!!' 12992000
ICEMSET 232,'???????? : A REPORT CANNOT BE SPLIT' 12993200
ICEMSET 233,'OUTFIL STATEMENTS CANNOT BE PASSED TO OR FROM AN *12994400
EFS PROGRAM' 12995600
ICEMSET 234,'STARTREC AND ENDREC VALUES ARE INCONSISTENT' 12996800
ICEMSET 270,'PROCESSING SYMNames STATEMENTS' 12996900
ICEMSET 271,'SYMNames IS EMPTY - SYMBOL SUBSTITUTION WILL NOT *12997000
BE PERFORMED' 12997100
ICEMSET 272,'SYMBOL, VALUE OR SYNTAX IS INVALID' 12997200
ICEMSET 273,'SYMBOL OR VALUE IS TOO LONG' 12997300
ICEMSET 274,'NEXT POSITION TO BE USED FOR * IS LARGER THAN 327*12997400
52' 12997500
ICEMSET 275,'DUPLICATE SYMBOL' 12997600
ICEMSET 276,'RESERVED WORD - NOT ALLOWED FOR SYMBOL' 12997700
ICEMSET 279,'ONE OR MORE ERRORS ENCOUNTERED DURING SYMNames PR*12997800
OCESSING' 12997900
ICEMSET 280,'ORIGINAL STATEMENTS FROM ???????? FOLLOW' 12998000
ICEMSET 281,'ORIGINAL OPERATOR STATEMENTS FOLLOW' 12998100
ICEMSET 282,'PERFORMING SYMBOL SUBSTITUTION AS NEEDED' 12998200
ICEMSET 283,'SYMBOL, SYNTAX OR DELIMETER ERROR' 12998300
ICEMSET 284,'SYMBOL SUBSTITUTION CAUSES OPERAND TO OVERFLOW CO*12998400
LUMN 72' 12998500
ICEMSET 287,'ONE OR MORE ERRORS ENCOUNTERED DURING SYMBOL SUBS*12998600
TITUTION' 12998700

```

Figure 41. ICEMSGS Macro (Part 6 of 10)

The ICEMSGSGS Macro

```

ICEMSET 600,'DFSORT ICETOOL UTILITY RUN STARTED' 13000000
ICEMSET 601,'DFSORT ICETOOL UTILITY RUN ENDED - RETURN CODE: *13050000
??' 13100000
ICEMSET 602,'OPERATION RETURN CODE: ??' 13150000
ICEMSET 603,'INFORMATION PRINTED IN ???????? DATA SET' 13200000
ICEMSET 604,'ERROR IN KEYWORD, PARAMETER, OR DELIMITER' 13250000
ICEMSET 605,'REQUIRED DD STATEMENT NOT FOUND: ????????' 13300000
ICEMSET 606,'DFSORT CALL ??? FOR ??? FROM ???????? TO ??????*13350000
?? USING ???CNTL ??????????' 13400000
ICEMSET 607,'STATISTICS FOR ??????????????:' 13450000
ICEMSET 608,' MINIMUM: ??????????????, MAXIMUM: ??????????*13500000
?????????' 13550000
ICEMSET 609,' AVERAGE: ?????????????????, TOTAL : ??????????*13600000
?????????' 13650000
ICEMSET 610,'NUMBER OF UNIQUE VALUES FOR ??????????????: ???*13700000
?????????????' 13750000
ICEMSET 611,'TOTAL FOR ?????????????? OVERFLOWED 15 DECIMAL DI*13800000
GITS' 13850000
ICEMSET 612,'NO ERRORS FOUND IN STATEMENT' 13900000
ICEMSET 613,'REQUIRED KEYWORD MISSING: ??????????????????*13950000
?????????????????????????????????????????????????' 14000000
ICEMSET 614,'INVALID OPERATOR' 14050000
ICEMSET 615,'DDNAME USED MORE THAN ONCE IN STATEMENT: ??????*14100000
?' 14150000
ICEMSET 616,'STATEMENT DOES NOT END AT OR BEFORE COLUMN 72' 14200000
ICEMSET 617,'RECORD COUNT OVERFLOWED 15 DECIMAL DIGITS' 14250000
ICEMSET 618,'INVALID ?????????????? VALUE - RECORD: ??????????*14300000
?????, HEX VALUE ??????????????????????????????????????????' 14350000
ICEMSET 619,'INVALID LENGTH, FORMAT, OR COMBINATION FOR ??????*14400000
?? OPERATION' 14450000
ICEMSET 620,'SUM OF POSITION AND LENGTH GREATER THAN ?????' 14500000
ICEMSET 621,'DFSORT DETECTED AN ERROR AFTER E35 EXIT PROCESSIN*14550000
G COMPLETED' 14600000
ICEMSET 622,'EXPECTED CONTINUATION LINE NOT FOUND' 14650000
ICEMSET 623,'MAXIMUM NUMBER OF ???????? KEYWORDS EXCEEDED' 14700000
ICEMSET 624,'MAXIMUM NUMBER OF TO DDNAMES EXCEEDED' 14750000
ICEMSET 625,'OPEN FAILED FOR ???????? DATA SET' 14800000
ICEMSET 626,'LIMIT FOR INVALID VALUES REACHED' 14850000
ICEMSET 627,'DFSORT CALL ??? FOR ??? FROM ???????? TO ??????*14900000
?? ??????????????' 14950000
ICEMSET 628,'RECORD COUNT: ??????????????????' 15000000
ICEMSET 629,'HIGHER AND LOWER VALUES EXCLUDE ALL RECORDS' 15050000
ICEMSET 630,'MODE IN EFFECT: ??????????' 15100000
ICEMSET 631,'NUMBER OF VALUES IN RANGE FOR ??????????????: ??*15150000
?????????????????' 15200000
ICEMSET 632,'SOURCE FOR ICETOOL STATEMENTS: ??????????' 15250000
ICEMSET 633,'RETURN AREA IS ?????? BYTES, BUT ?????? BYTES ARE R*15300000
EQUIRED' 15350000
ICEMSET 634,'VALUE FOR ???????????????? EXCEEDS 15 DIGITS' 15400000
ICEMSET 635,'NUMBER OF HEADER KEYWORDS DOES NOT MATCH NUMBER O*15450000
F ON KEYWORDS' 15500000
ICEMSET 636,'INVALID VALUE IN DFSORT INSTALLATION MODULE ICE?*15550000
?' 15600000
ICEMSET 637,'????????? RECORD LENGTH OF ?????? BYTES EXCEEDS MAX*15679990
IMUM OF ?????? BYTES' 15709980
ICEMSET 638,'NUMBER OF RECORDS RESULTING FROM CRITERIA: ?????*15750000
?????????????' 15800000

```

Figure 41. ICEMSGSGS Macro (Part 7 of 10)


```

ICEMSET 639,'INSUFFICIENT MAIN STORAGE - ADD AT LEAST ???K BY*15850000
TES TO REGION' 15900000
ICEMSET 640,'INVALID FORMATTING ITEM' 15925000
ICEMSET 641,'???????? SERVICE ERROR (????????-????????)' 15937500
ICEMSET 642,'DFSORT RELEASE AND ICE??? RELEASE DO NOT MATCH' 15943700
ICEMSET 643,'WIDTH OF REPORT IS ???? BYTES' 15944590
ICEMSET 644,'TOTAL FOR ?????????????? OVERFLOWED ?? DECIMAL DI*15945480
GITS' 15946370
ICEMSET 645,'???????? OVERFLOWED ?? DECIMAL DIGITS' 15947260
ICEMSET 646,'RECORD COUNT MEETS CRITERIA - RC=12 SET' 15948150
ICEMSET 647,'RECORD COUNT DOES NOT MEET CRITERIA - RC=0 SET' 15949040
ICEMSET 700,'EFS PROCESSING IS NOT SUPPORTED FOR THE TECHNIQUE*15950000
SELECTED' 16000000
ICEMSET 701,'EFS PROCESSING WAS TERMINATED BY ???????? AT MAJO*16050000
R CALL ?' 16100000
ICEMSET 702,'INVALID RETURN CODE FROM EFS PROGRAM ???????? AT *16150000
MAJOR CALL ?' 16200000
ICEMSET 703,'INVALID RETURN CODE FROM ?????? OF EFS PROGRAM ???*16250000
?????' 16300000
ICEMSET 704,'LOAD FAILED FOR EFS PROGRAM ????????' 16350000
ICEMSET 705,'EFS PROCESSING WAS TERMINATED BY ?????? OF EFS PRO*16400000
GRAM ????????' 16450000
ICEMSET 706,'END OF SUPPLIED CONTROL STATEMENTS - EFS RETURNED*16500000
CONTROL STATEMENTS FOLLOW' 16550000
ICEMSET 707,'EXIT E61 IS NOT SUPPORTED FOR EFS FIELDS' 16600000
ICEMSET 708,'AN EFS PROGRAM IS REQUIRED' 16650000
ICEMSET 709,'????? EXIT ROUTINE IS REQUIRED' 16700000
ICEMSET 710,'THE ???????? CONTROL STATEMENT WAS NOT SENT TO EF*16750000
S PROGRAM ????????' 16800000
ICEMSET 711,'END OF EFS RETURNED CONTROL STATEMENTS FROM ???C*16850000
NTL - PARAMETER LIST CONTROL STATEMENTS FOLLOW' 16900000
ICEMSET 712,'END OF EFS RETURNED PARAMETERS FROM ???????? - SYSI*16950000
N OR ???CNTL/PARAMETER LIST CONTROL STATEMENTS FOLLOW' 17000000
ICEMSET 713,'LANGUAGE ENVIRONMENT INITIALIZATION FAILED' 17002300
ICEMSET 714,'LOAD FAILED FOR ACTIVE LOCALE ?????????????????*17004600
?????????????????' 17006900
ICEMSET 715,'LOCALE PROCESSING CONFLICTS WITH ????????' 17010300
ICEMSET 716,'A CONTROL FIELD CONTAINS AN INVALID CHARACTER' 17013800
ICEMSET 717,'CONTROL FIELDS FOR LOCALE PROCESSING ARE TOO COMP*17019900
LEX' 17026000
ICEMSET 721,'A COMPARE FIELD CONTAINS AN INVALID CHARACTER' 17032200
ICEMSET 722,'ACTIVE LOCALE ?????????????????????????????????*17034500
S USED FOR LOCALE PROCESSING' 17036800
ICEMSET 725,'LANGUAGE ENVIRONMENT SERVICE ERROR (%%%)' 17043700
ICEMSET 726,'LOAD FAILED FOR LOCALE PROCESSING' 17046000
ICEMSET 727,'EFS PROCESSING CANNOT BE USED WITH SYMBOL PROCESS*17047300
ING' 17048600
ICEMSET 750,'DC !!!!!!!!!!!!!!!!!!!!!!! TC !!!!!!!!!!!!!!!!!!!!!!! C*17048690
? DSV?? KSZ !!!! VSZ !!!!' 17048780
ICEMSET 751,'????????????????????????????????????????????*17048870
?????????????????????????????????????????' 17048960
ICEMSET 752,'FSZ=!!!!!!!!!!!!!!!!!! ?? IGN=!!!!!!!!!!!!!!!!!! ? *17049050
AVG=!!!!!! !!!!! WSP=!!!!!!!!!!!!!!!!!! ? DYN=!!!!!!!!!!!! *17049140
!!!!!!' 17049230
ICEMSET 800,'BLOCKSET TECHNIQUE COULD NOT BE USED - REASON COD*17050000
E IS %%' 17100000
ICEMSET 802,'????????????? TECHNIQUE IN CONTROL' 17150000
ICEMSET 804,'????????? EXCP COUNT: !!!!!!!!!!!' 17200000
ICEMSET 805,'JOBNAME: ????????, STEPNAME: ????????' 17250000
ICEMSET 816,'DFSORT SVC CALL RESULTED IN ???? ABEND' 17700000
ICEMSET 818,'LOCATE CALL FAILED' 17799990
ICEMSET 820,'RL=!!!!!! B=!!!!!! IL=!!!!!! IS=!!!!!! IB=!!!!!! *17850000
RM=!!!!!! EM=!!!!!! BA=!!!!!! IX=!!!!!! OX=!!!!!!' 17900000
ICEMSET 821,'BN=!!!!!! X=!!!!!! TO=!!!!!! SN=!!!!!! G=!!!!!!' 17950000
ICEMSET 822,'BN=!!!!!! X=!!!!!! G=!!!!!! PN=!!!!!! BT=!!!!!! TO*18000000
=!!!!!!' 18050000

```

Figure 41. ICEMSGS Macro (Part 8 of 10)

The ICEMSGSGS Macro

```

ICEMSET 823,'BN=!!!!!! X=!!!!!! G=!!!!!! TO=!!!!!! BT=!!!!!!' 18100000
ICEMSET 824,'PE=!!!!!! RP=!!!!!! CX=!!!!!! CO=!!!!!! C0=!!!!!!*18150000
CR=!!!!!! G=!!!!!! WB=!!!!!!' 18200000
ICEMSET 825,'GP=!!!!!! SA=!!!!!! X=!!!!!!' 18250000
ICEMSET 826,'BN=!!!!!! X=!!!!!! RM=!!!!!!' 18300000
ICEMSET 827,'RD=!! BL=!!!!!! BY=!!!!!!' 18302900
ICEMSET 855,'???????? : TX=?, R=?, L=?, B=?, BL=?, BR=?, DCT=!*18305800
!! , VS=?, RU=?, SB=!!!' 18308700
ICEMSET 857,'C=!!!!!! , LA=!!!!!! , DA=!!!!!! , LB=!!!!!! , DB=!!!*18311600
!! , MB=!!!!!! , CB=!!!!!! , UB=!!!!!!' 18314500
ICEMSET 858,'LA=!!!!!! , DA=!!!!!! , AA=!!!!!! , BA=!!!!!! , T*18317400
A=!!!!!!' 18320300
ICEMSET 859,'LB=!!!!!! , DB=!!!!!! , AB=!!!!!! , BB=!!!!!! , CP=!!!!!! ,*18323200
RS=!!!!!! , TB=!!!!!!' 18326100
ICEMSET 860,'F=?? , P=?, M=?, B=!!!!!!' 18329000
ICEMSET 880,'QP=!!!! QA=!!!!!! HI=!!!! LI=!!!! MI=!!!! *18331900
TZ=!!!!!! N1=!!!!!! N2=!!!!!! SZ=!!!! HN=!!' 18334800
ICEMSET 881,'EQ=?? DX=!!!! D2=!!!! D3=!!!! D4=!!!! AS=!!!!!!*18337700
!! SA=!!!!!! SB=!!!!!! SC=!!!!!! HN=!!' 18340600
ICEMSET 882,'?? PLE ????????' 18342000
ICEMSET 883,'IX=?? ,IFC=???? ,INC=!!!!!! ,OX=?? ,OFC=???? ,ONC=!*18343500
!!!!!! ,CRT=???????? ,NCR=????????' 18346400
ICEMSET 884,'?? ? !!!!!!! AVK !!!!!!! AVR !!!!!!!' 18350000
ICEMSET 885,'DAT ?? DSR ??? BINS !!!!!!! BSZ !!!!! RCP !*18400000
!!! AJC !!!!! RLC !!!!! DUNIT !!!!!' 18450000
ICEMSET 886,'SYS !!!!!!! TSTG !!!!!!! FS !!!!!!! INIT*18529990
!!!!!! MAX !!!!!!! LEN !!!!!!!' 18559980
ICEMSET 887,'CSES !!!!!!! ,!!!!!! ,!!!!!! ES !!!!!!! ,!!!!*18600000
!!!! ,!!!!!!' 18650000
ICEMSET 888,'INIO1 !!!!! AXIO1 !!!!! W1 !!!!! INIO3 !!!!! *18700000
AXIO3 !!!!! BRUNS !!!!! VD?' 18750000
ICEMSET 889,'CT=???????? , SB=!!! , L=?' 18800000
ICEMSET 891,'!!!!!! WMAIN, !!!!! CMAIN, ??? CALLOC, ? SCN,*18950000
? BA, !!!!! AZ, !!!!! BZ, ?? QC' 19029990
ICEMSET 892,'!!!! RIN !!!!!!! BLI !!!!!!! BLO !!!!! *19059980
RUN !!!!! BUN ??? CPU ?? CVC' 19089970
ICEMSET 893,'!!!! XIN !!!!! WIN !!!!!!! GIN NDE? PFP?? B?? C*19150000
M?? CI? UPT? LM? VS?? RU?' 19200000
ICEMSET 894,'!!!! STR !!! MOR !!!!! IPB !!!!! OPB !!!!! C*19250000
YL ??? MN' 19300000
ICEMSET 895,'!!!! MUNIT !!!!! SUNIT !!!!! OUNIT' 19350000
ICEMSET 896,'!!!!!! SET !!!!!!! DEXTOT !!!!!!! BLK *19400000
!!!!!! CSZ !!!!!!! WE' 19450000
ICEMSET 897,'AVAL=!!!!!! ,BVAL=!!!!!! ,CVAL=!!!!!! ,A*19459990
SV=!!!!!! ,HSZ=!!!!!! ,HM=?? ,HN=!!' 19469980
ICEMSET 898,'OMAX=!!!!!! ,NMAX=!!!!!! ,TMAX=!!!!!! ,C*19500000
MAX=!!!!!! ,HU=!!! ,BUN=!!!!!! ,MD=?? ,DU=!!! ,DR=!!!!!! ,HN=*19559990
!' 19569980
ICEMSET 899,'HSR=!!!!!! ,HSW=!!!!!! ,HRE=!!!!!! ,HWE=!!!!!!*19600000
!! ,HRP=!!!!!! ,HWP=!!!!!! ,HWM=!!!!!! ,HNM=!!' 19650000
ICEMSET 900,'CON=!!!! ,MUV=!!!! ,VOL=!!!! ,ENU=!!!! ,SBK=!!!!!!*19700000
! ,SRC=!!!!!! ,VEM=!!!!' 19750000
ICEMSET 901,'? ?????? ?????? ?????? ?????? ?????? ?????? ?????*19800000
? ??????' 19850000
ICEMSET 902,'O ???? I ???? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ?*19900000
? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? ? *19950000
?? ??' 20000000
ICEMSET 905,'? : RF=!!! ,LR=!!!!!! ,BLK=!!!!!! ,BCT=!!!!!!*20001990
!!!!!!' 20003980

```

Figure 41. ICEMSGSGS Macro (Part 9 of 10)

```

ICEMSET 995, 'PN=?,HN=!!' 20017190
ICEMSET 996, 'ESM=!!!!!!!!!!!!,ESO=!!!!!!!!!!!!,ESR=!!!!!!!!!!!!,ESP=*20021300
!!!!!!!!!!!!,ESS=!!!!!!!!!!!!,CES=!!!!!!!!!!!!,HSZ=!!!!!!!!!!!!' 20028400
ICEMSET 997, 'HWSP=!!!!!!!!!!!!,HMAX=!!!!!!!!!!!!,HES=!!!!!!!!!!!!,AS*20035500
V=!!!!!!!!!!!!,EQ=??,HN=!!' 20042600
ICEMSET 999, 'PWK=!!! PSP=!!!!!!!!!!!! SWK=!!! SSP=!!!!!!!!!!!! TWK=!!!*20200000
TSP=!!!!!!!!!!!! RWK=!!! RSP=!!!!!!!!!!!! AWK=!!' 20250000
MEND 20300000

```

Figure 41. ICEMSGSGS Macro (Part 10 of 10)

Appendix B. 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 *Messages and Codes*.

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 "Changing the DFSORT Installation Defaults" on page 48 for information on what to do if you receive an error message.

12,*ICESM1000 JCL, INV, TSO, TSOINV, TD1, TD2, 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,*ICESM1001 SIZE VALUE x INVALID**

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,*ICESM1002 MAXLIM VALUE x INVALID**

Explanation: The value *x* specified for MAXLIM is not a valid decimal number or is not in the valid range for MAXLIM.

8,*ICESM1003 MINLIM VALUE x INVALID**

Explanation: The value *x* specified for MINLIM is not a valid decimal number or is not in the valid range for MINLIM.

8,*ICESM1004 MSGPRT VALUE x INVALID**

Explanation: The value *x* specified for MSGPRT is not a valid keyword.

8,*ICESM1005 EQUALS VALUE x INVALID**

Explanation: The value *x* specified for EQUALS is not a valid keyword.

8,*ICESM1006 LIST VALUE x INVALID**

Explanation: The value *x* specified for LIST is not a valid keyword.

Messages

8,*ICESM1007 ERET VALUE x INVALID**

Explanation: The value x specified for ERET is not a valid keyword.

8,*ICESM1009 MSGDDN VALUE x GREATER THAN 8 CHARACTERS**

Explanation: The value x specified for MSGDDN is too long.

8,*ICESM1011 VIO VALUE x INVALID**

Explanation: The value x specified for VIO is not a valid keyword.

8,*ICESM1012 PARMDDN VALUE x GREATER THAN 8 CHARACTERS**

Explanation: The value x specified for PARMDDN is too long.

8,*ICESM1013 ESTAE VALUE x INVALID**

Explanation: The value x specified for ESTAE is not a valid keyword.

8,*ICESM1014 IGNCKPT VALUE x INVALID**

Explanation: The value x specified for IGNCKPT is not a valid keyword.

8,*ICESM1015 WRKSEC VALUE x INVALID**

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,*ICESM1016 VERIFY VALUE x INVALID**

Explanation: The value x specified for VERIFY is not a valid keyword.

8,*ICESM1017 SVC VALUE x INVALID**

Explanation: The value x specified for the user SVC number is not decimal 109 or decimal 200-255.

8,*ICESM1018 DYNALOC VALUE x GREATER THAN 8 CHARACTERS**

Explanation: The value x specified for DYNALOC device type is too long.

8,*ICESM1019 DYNALOC VALUE x INVALID**

Explanation: The value x specified for DYNALOC number of devices is not a valid decimal number.

8,*ICESM1020 ALTSEQ VALUE x INVALID**

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,*ICESM1021 RESALL VALUE x INVALID**

Explanation: The value x specified for RESALL is not a valid decimal number or is not in the valid range for RESALL.

8,*ICESM1022 RESINV VALUE x INVALID**

Explanation: The value x specified for RESINV is not a valid decimal number.

8,*ICESM1023 SMF VALUE x INVALID**

Explanation: The value x specified for SMF is not a valid keyword.

8,*ICESM1024 CHALT VALUE x INVALID**

Explanation: The value x specified for CHALT is not a valid keyword.

8,*ICESM1025 CHECK VALUE x INVALID**

Explanation: The value x specified for CHECK is not a valid keyword.

8,*ICESM1026 WRKREL VALUE x INVALID**

Explanation: The value x specified for WRKREL is not a valid keyword.

8,*ICESM1027 OUTREL VALUE x INVALID**

Explanation: The value x specified for OUTREL is not a valid keyword.

8,*ICESM1028 MINLIM VALUE x IS GREATER THAN SIZE VALUE**

Explanation: The value x specified for MINLIM is larger than the value specified for SIZE.

8,*ICESM1029 MINLIM VALUE x IS GREATER THAN MAXLIM VALUE**

Explanation: The value x specified for MINLIM is larger than the value specified for MAXLIM.

8,*ICESM1030 OUTSEC VALUE x INVALID**

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,*ICESM1031 STIMER VALUE x INVALID**

Explanation: The value x specified for STIMER is not a valid keyword.

8,*ICESM1032 OVERRGN VALUE x INVALID**

Explanation: The value x specified for OVERRGN is not a valid decimal number or is not in the valid range for OVERRGN.

8,*ICESM1033 TMAXLIM VALUE x INVALID**

Explanation: The value x specified for TMAXLIM is not a valid decimal number or is not in the valid range for TMAXLIM.

8,*ICESM1034 ARESALL VALUE x INVALID**

Explanation: The value x specified for ARESALL is not a valid decimal number or is not in the valid range for ARESALL.

8,*ICESM1035 ARESINV VALUE x INVALID**

Explanation: The value x specified for ARESINV is not a valid decimal number or is not in the valid range for ARESINV.

8,*ICESM1036 COBEXIT VALUE x INVALID**

Explanation: The value x specified for COBEXIT is not valid.

8,*ICESM1037 TMAXLIM VALUE IS LESS THAN MAXLIM VALUE**

Explanation: The value specified for TMAXLIM is less than the MAXLIM value.

8,*ICESM1038 MSGCON VALUE x INVALID**

Explanation: The value x specified for MSGCON is not a valid keyword.

8,*ICESM1039 VLSHRT VALUE x INVALID**

Explanation: The value x specified for VLSHRT is not a valid keyword.

8,*ICESM1040 CINV VALUE x INVALID**

Explanation: The value x specified for CINV is not a valid keyword.

8,*ICESM1041 EFS VALUE x GREATER THAN 8 CHARACTERS**

Explanation: The value x specified for the EFS program name is too long.

8,*ICESM1042 LISTX VALUE x INVALID**

Explanation: The value x specified for LISTX is not a valid keyword.

8,*ICESM1043 VSAMBSP VALUE x INVALID**

Explanation: The value x specified for VSAMBSP is not a valid keyword.

8,*ICESM1044 DIAGSIM VALUE x INVALID**

Explanation: The value x specified for DIAGSIM is not a valid keyword.

8,*ICESM1046 SORTLIB VALUE x INVALID**

Explanation: The value x specified for SORTLIB is not a valid keyword.

8,*ICESM1058 DYNSPC VALUE x INVALID**

Explanation: The value x specified for DYNSPC is not a valid decimal number or is not in the valid range for DYNSPC.

8,*ICESM1059 SVC VALUE x IS NOT ALT**

Explanation: The value x is specified instead of the keyword ALT.

8,*ICESM1060 ALT IS INVALID WITH SVC x**

Explanation: The value x specified for the user SVC number is not decimal 109 and is therefore invalid with ALT.

8,*ICESM1061 ZDPRINT VALUE x INVALID**

Explanation: The value x specified for ZDPRINT is not a valid keyword.

8,*ICESM1062 IEXIT VALUE x INVALID**

Explanation: The value x specified for IEXIT is not a valid keyword.

Messages

8,*ICESM1063** TEXT VALUE x INVALID

Explanation: The value x specified for TEXT is not a valid keyword.

8,*ICESM1073** GENER VALUE x GREATER THAN 8 CHAR

Explanation: The value x specified for GENER is too long.

8,*ICESM1077** DYNAUTO VALUE x INVALID

Explanation: The value x specified for DYNAUTO is not a valid keyword.

8,*ICESM1078** ABCODE VALUE x INVALID

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,*ICESM1081** EXITCK VALUE x INVALID

Explanation: The value x specified for EXITCK is not a valid keyword.

8,*ICESM1082** NOMSGDD VALUE x INVALID

Explanation: The value x specified for NOMSGDD is not a valid keyword.

8,*ICESM1083** TABLE IS INVALID WITH OTHER 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,*ICESM1085** HIPRMAX VALUE x INVALID

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,*ICESM1086** FSZEST VALUE x INVALID

Explanation: The value x specified for FSZEST is not a valid keyword.

8,*ICESM1087** SDB VALUE x INVALID

Explanation: The value x specified for SDB is not a valid keyword.

8,*ICESM1089** IDRCPCT VALUE x INVALID

Explanation: The value x specified for IDRCPCT is not a valid decimal number or NONE, or is not in the valid range for IDRCPCT.

8,*ICESM1090** ODMAXBF VALUE x INVALID

Explanation: The value x specified for ODMAXBF is not a valid decimal number or is not in the valid range for ODMAXBF.

8,*ICESM1091** CFW VALUE x INVALID

Explanation: The value x specified for CFW is not a valid keyword.

8,*ICESM1092** GNPAD VALUE x INVALID

Explanation: The value x specified for GNPAD is not a valid keyword.

8,*ICESM1093** GNTRUNC VALUE x INVALID

Explanation: The value x specified for GNTRUNC is not a valid keyword.

8,*ICESM1094** DSPSIZE VALUE x INVALID

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,*ICESM1096** SDBMSG VALUE x INVALID

Explanation: The value x specified for SDBMSG is not a valid keyword.

8,*ICESM1097** LOCALE VALUE x GREATER THAN 32 CHARACTERS

Explanation: The value x specified for the LOCALE name is too long.

8,*ICESM1098** DSA VALUE x INVALID

Explanation: The value x specified for DSA is not a valid decimal number or is not in the valid range for DSA.

8,*ICESM1099** Y2PAST VALUE x INVALID

Explanation: The value x specified for Y2PAST is not a valid decimal number or is not in the valid range for Y2PAST.

8,*ICESM1100 EXPMAX VALUE x INVALID**

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,*ICESM1101 EXPOLD VALUE x INVALID**

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,*ICESM1102 EXPRES VALUE x INVALID**

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,*ICESM1103 IOMAXBF VALUE x INVALID**

Explanation: The value x specified for IOMAXBF is not a valid decimal number or is not in the valid range for IOMAXBF.

8,*ICESM1104 ENABLE VALUE x INVALID**

Explanation: The value x specified for ENABLE is not a valid keyword.

8,*ICESM1105 ENABLE VALUE IS NULL**

Explanation: A null value was specified for ENABLE instead of a valid keyword.

8,*ICESM1106 ENABLE VALUE x DUPLICATED**

Explanation: The value x was specified for ENABLE more than once.

8,*ICESM1107 day VALUE x INVALID**

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,*ICESM1108 day VALUE IS NULL**

Explanation: A null value was specified for the indicated *day* parameter instead of a valid hhmm value or keyword.

8,*ICESM1109 day CONFLICTS WITH ANOTHER PARAMETER**

Explanation: The indicated *day* parameter was specified with another day parameter with which it is not allowed.

8,*ICESM1110 day VALUES ARE UNPAIRED**

Explanation: A starting hhmm value was specified without an ending hhmm value for the indicated *day* parameter.

8,*ICESM1111 PAD VALUE x INVALID**

Explanation: The value x specified for PAD is not a valid keyword.

8,*ICESM1112 TRUNC VALUE x INVALID**

Explanation: The value x specified for TRUNC is not a valid keyword.

8,*ICESM1113 OVFLO VALUE x INVALID**

Explanation: The value x specified for OVFLO is not a valid keyword.

8,*ICESM1114 SPANINC VALUE x INVALID**

Explanation: The value x specified for SPANINC is not a valid keyword.

8,*ICESM1115 SOLRF VALUE x INVALID**

Explanation: The value x specified for SOLRF is not a valid keyword.

8,*ICESM1117 VLSCMP VALUE x INVALID**

Explanation: The value x specified for VLSCMP is not a valid keyword.

8,*ICESM1118 SZERO VALUE x INVALID**

Explanation: The value x specified for SZERO is not a valid keyword.

8,*ICESM1119 VLLONG VALUE x INVALID**

Explanation: The value x specified for VLLONG is not a valid keyword.

8,*ICESM1120 VSAMEMT VALUE x INVALID**

Explanation: The value x specified for VSAMEMT is not a valid keyword.

8,*ICESM1121 RESET VALUE x INVALID**

Explanation: The value x specified for RESET is not a valid keyword.

Messages

8,***ICESM1122 VSAMIO VALUE *x* INVALID

Explanation: The value *x* specified for VSAMIO is not a valid keyword.

Message Module Error Messages

The messages in the following pages are produced during expansion of the message text modules after you have changed the ICEMSGS macro. See “Modifying Message Texts” on page 51 for information on what to do if you receive an error message.

12,***ICESM1050 TEXT FOR MESSAGE *n* IS NOT PRESENT IN ICEMSGS

Explanation: The message text requested was not found in the ICEMSGS macro; that is, a statement of the form:

```
ICEMSET n,'textn'
```

was not found in ICEMSGS.

Programmer Response: Check to make sure ICEMSGS was updated correctly. RESTORE the SMP/E usermod to see the original ICEMSGS macro in SICEUSER.

Locate the statement:

```
ICEMSET n,'textn'
```

in the original version of ICEMSGS. Add this statement to the SMP/E usermod version of this statement, and modify it if appropriate.

If the ICEMSGS macro in the SMP/E target zone has the same RMID as the ICEMSGS macro in the distribution zone, you can also use the ICEMSGS macro in the AICEUSER library for reference.

12,***ICESM1051 MESSAGE *n* TEXT LENGTH IS *m* BYTES TOO LONG

Explanation: The message text requested is *m* bytes too long in the ICEMSGS macro.

Programmer Response: Check to make sure ICEMSGS was updated correctly. RESTORE the SMP/E usermod to see the original ICEMSGS macro in SICEUSER.

Locate the statement:

```
ICEMSET n,'textn'
```

in the SMP/E usermod and shorten *textn* to 110 characters or fewer.

If the ICEMSGS macro in the SMP/E target zone has the same RMID as the ICEMSGS macro in the distribution zone, you can also use the ICEMSGS macro in the AICEUSER library for reference.

8,*** GENERATION TERMINATED ***

Explanation: The generation procedure was unsuccessful. Correct errors and restart generation.

12,***ICESM1052 MESSAGE TEXTS LENGTH IS *n* BYTES TOO LONG

Explanation: The total length of the message texts requested is too long as they appear in the ICEMSGS macro. The message texts requested are indicated by the statements:

```
ICEDMSG a,...  
ICEDMSG b,...
```

```
.  
.  
.
```

in this CSECT.

Programmer Response: Check to make sure ICEMSGS was updated correctly. RESTORE the SMP/E usermod to see the original ICEMSGS macro in SICEUSER.

The ICEDMSG statements in the CSECT identify the messages requested in ICEMSGS. Locate the corresponding statements in ICEMSGS:

```
ICEMSET a,'texta'  
ICEMSET b,'textb'
```

```
.  
.  
.
```

and shorten the total length of the text in the SMP/E usermod version of these statements by *n* bytes.

If the ICEMSGS macro in the SMP/E target zone has the same RMID as the ICEMSGS macro in the distribution zone, you can also use the ICEMSGS macro in the AICEUSER library for reference.

12,***ICESM1053 TEXT FOR MESSAGE *n* HAS INCORRECT INSERT FIELD(S)

Explanation: The message text requested has incorrect insert field(s) as it appears in the ICEMSGS 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 ICEMSGS was updated correctly. RESTORE the SMP/E usermod to see the original ICEMSGS macro in SICEUSER.

Locate the statement:

```
ICEMSET n,'textn'
```

in the original version of ICEMSGSGS. 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 ICEMSGSGS macro in the SMP/E target zone has the same RMID as the ICEMSGSGS macro in the distribution zone, you can also use the ICEMSGSGS macro in the AICEUSER library for reference.

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 listed below to describe the default you have selected (or the IBM-supplied default).

*** , 5740-SM1 RELEASE 14.0**

*** , ICEAM1 DEFAULTS: BATCH JCL DIRECT ENVIRONMENT MODULE**

*** , ICEAM2 DEFAULTS: BATCH PROGRAM ENVIRONMENT MODULE**

*** , ICEAM3 DEFAULTS: TSO DIRECT ENVIRONMENT MODULE**

*** , ICEAM4 DEFAULTS: TSO PROGRAM ENVIRONMENT MODULE**

*** , ICETD1 DEFAULTS: FIRST TIME-OF-DAY MODULE**

*** , ICETD2 DEFAULTS: SECOND TIME-OF-DAY MODULE**

*** , ICETD3 DEFAULTS: THIRD TIME-OF-DAY MODULE**

*** , ICETD4 DEFAULTS: FOURTH TIME-OF-DAY MODULE**

*** , ONE OR MORE ICETDX MODULES ARE ENABLED**

*** , ICETDX MODULES ARE NOT ENABLED**

*** , ENABLE IGNORED FOR ICETDX MODULE**

*** , *day* START AND END TIMES:
hhmm TO *hhmm***

*** , *day* START AND END TIMES: ALL**

*** , *day* START AND END TIMES: NONE**

* , ONE OR MORE DAY OPERANDS IGNORED FOR ICEAMX MODULE

* , EXCPVR IGNORED

* , VIRTDSP IGNORED

* , SIZE IS x BYTES

* , SIZE IS LIMITED BY TMAXLIM/MAXLIM

* , x BYTES IS LIMIT FOR MAXLIM

* , 1024K BYTES IS LIMIT FOR MAXLIM

* , MIN MAIN STORAGE DFSORT

* ,WILL TRY TO USE: x BYTES

* , MIN MAIN STORAGE DFSORT

* ,WILL TRY TO USE: 440K BYTES

* , x BYTES RESERVED FOR SYSTEM USE

* , 4096 BYTES RESERVED FOR SYSTEM USE

* , x BYTES RESERVED FOR PROGRAM INVOKED DFSORT

* , NO BYTES RESERVED FOR PROGRAM INVOKED DFSORT

* , RESINV IGNORED FOR DIRECTLY INVOKED DFSORT

* , DYNAMIC ALLOCATION NAME IS x

Messages

***, DYNAMIC ALLOCATION NAME IS SYSDA**

***, DYNAMIC ALLOCATION NUMBER IS x**

***, DYNAMIC ALLOCATION DEFAULT NUMBER IS 4**

***, DYNAMIC ALLOCATION NUMBER IS x , DEFAULTED TO 255 (MAXIMUM)**

***, AUTOMATIC DYNAMIC ALLOCATION WILL NOT BE USED**

***, AUTOMATIC DYNAMIC ALLOCATION WILL BE USED**

***, SORTWKDD DATA SETS WILL BE DEALLOCATED**

***, ABEND CODE WILL BE MESSAGE NUMBER**

***, ABEND CODE WILL BE x**

***, NO AUTOMATIC SECONDARY ALLOCATION FOR WORK DATA SETS**

***, AUTOMATIC SECONDARY ALLOCATION FOR WORK DATA SETS**

***, NO AUTOMATIC SECONDARY ALLOCATION FOR OUTPUT DATA SETS**

***, AUTOMATIC SECONDARY ALLOCATION FOR OUTPUT DATA SETS**

***, DO NOT PRINT MESSAGES**

***, PRINT ALL MESSAGES**

***, PRINT CRITICAL MESSAGES**

***, ORDER OF EQUALS WILL BE PRESERVED**

* , ORDER OF EQUALS WILL NOT BE PRESERVED

* , ORDER OF EQUALS WILL BE PRESERVED FOR VLR BLOCKSET

* , CONTROL STATEMENTS WILL NOT BE LISTED

* , CONTROL STATEMENTS WILL BE LISTED

* , ABEND IF DFSORT UNSUCCESSFUL

* , RC=16 IF DFSORT UNSUCCESSFUL

* , X IS ALTERNATE MESSAGE DDNAME

* , SYSOUT IS ALTERNATE MESSAGE DDNAME

* , ALLOW VIRTUAL I/O FOR WORK DATA SETS

* , AVOID VIRTUAL I/O FOR WORK DATA SETS

* , UNUSED SORTWKDD SPACE WILL BE RELEASED

* , UNUSED SORTWKDD SPACE WILL NOT BE RELEASED

* , UNUSED OUTPUT DATA SET SPACE WILL BE RELEASED

* , UNUSED OUTPUT DATA SET SPACE WILL NOT BE RELEASED

* , STIMER WILL BE USED

* , STIMER WILL NOT BE USED

Messages

***, OUTPUT RECORDS WILL BE VERIFIED**

***, OUTPUT RECORDS WILL NOT BE VERIFIED**

***, ALTSEQ TABLE USED ON FORMAT AQ CONTROL FIELDS**

***, ALTSEQ TABLE USED ON FORMAT AQ AND CH CONTROL FIELDS**

, SVC NUMBER IS *x

***, THE DEFAULT SVC 109 MODULE IGX00017 WILL BE USED**

***, THE ALTERNATE SVC 109 MODULE IGX00038 WILL BE USED**

***, RECORD COUNTERS NOT CHECKED WHEN E35 IS PRESENT**

***, RECORD COUNTERS CHECKED WHEN E35 IS PRESENT**

***, NO SMF RECORDS WILL BE WRITTEN TO SMF DATA SET**

***, FULL SMF RECORDS WILL BE WRITTEN TO SMF DATA SET**

***, SHORT SMF RECORDS WILL BE WRITTEN TO SMF DATA SET**

***, BLOCKSET TECHNIQUE MAY BE USED IF CKPT IS SPECIFIED**

***, BLOCKSET TECHNIQUE MAY NOT BE USED IF CKPT IS SPECIFIED**

***, COBOL EXITS USE OS/V5 COBOL**

***, COBOL EXITS USE VS COBOL II**

***, DO NOT WRITE MESSAGES TO MASTER CONSOLE**

*** , WRITE ALL MESSAGES TO MASTER CONSOLE**

*** , WRITE CRITICAL MESSAGES TO MASTER CONSOLE**

*** , SHORT VARIABLE RECORDS WILL BE ACCEPTED**

*** , SHORT VARIABLE RECORDS WILL NOT BE ACCEPTED**

*** , ICEIEXIT WILL NOT BE USED**

*** , ICEIEXIT WILL BE USED**

*** , ICETEXIT WILL NOT BE USED**

*** , ICETEXIT WILL BE USED**

*** , X BYTES OVER REGION FOR BLOCKSET**

*** , 64K BYTES OVER REGION FOR BLOCKSET**

*** , 16K BYTES OVER REGION FOR BLOCKSET**

*** , X BYTES IS LIMIT FOR TMAXLIM**

*** , 4096K BYTES IS LIMIT FOR TMAXLIM**

*** , X BYTES RESERVED ABOVE 16 MB VIRTUAL FOR SYSTEM USE**

*** , NO BYTES RESERVED ABOVE 16 MB VIRTUAL FOR SYSTEM USE**

*** , X BYTES RESERVED ABOVE 16 MB VIRTUAL**

*** , FOR PROGRAM INVOKED DFSORT**

Messages

*** , NO BYTES RESERVED ABOVE 16 MB VIRTUAL**

*** , FOR PROGRAM INVOKED DFSORT**

*** , ARESINV IGNORED FOR DIRECTLY INVOKED DFSORT**

*** , POSITIVE ZD SUMS WILL NOT BE PRINTABLE AS NUMBERS**

*** , POSITIVE ZD SUMS WILL BE PRINTABLE AS NUMBERS**

*** , E15/E35 RETURN CODE CHECKING WILL BE STRONG**

*** , E15/E35 RETURN CODE CHECKING WILL BE WEAK**

*** , TERMINATE IF MESSAGE DATA SET MISSING**

*** , CONTINUE IF MESSAGE DATA SET MISSING (MSGCON=ALL)**

*** , CONTINUE IF MESSAGE DATA SET MISSING (MSGCON=CRITICAL)**

*** , CONTINUE IF MESSAGE DATA SET MISSING (MSGCON=NONE)**

*** , EFS MODIFIED CONTROL STATEMENTS WILL NOT BE LISTED**

*** , EFS MODIFIED CONTROL STATEMENTS WILL BE LISTED**

*** , EFS PROGRAM x WILL BE CALLED**

*** , NO EFS PROGRAM WILL BE CALLED**

*** , x WILL BE USED FOR IEBCGENR MODULE NAME**

*** , IEBCGENR WILL BE USED FOR IEBCGENR MODULE NAME**

*** , GENER VALUE IGNORED FOR DIRECTLY INVOKED DFSORT**

* , HIPRMAX IS LIMITED TO x MEGABYTES

* , HIPRMAX IS LIMITED BY OPTIMAL

* , HIPRMAX IS LIMITED TO x PERCENT

* , REFORMATTED LENGTH WILL BE USED FOR SORTOUT LRECL

* , REFORMATTED LENGTH WILL NOT BE USED FOR SORTOUT LRECL

* , ESTAE RECOVERY ROUTINE WILL NOT BE DELETED EARLY

* , ESTAE RECOVERY ROUTINE WILL BE DELETED EARLY

* , x IS THE ALTERNATE PARAMETER DATA SET DDNAME

* , DFSPARM IS THE ALTERNATE PARAMETER DATA SET DDNAME

* , VSAM CONTROL INTERVAL ACCESS CAN BE USED

* , VSAM CONTROL INTERVAL ACCESS CANNOT BE USED

* , THE CONVENTIONAL TECHNIQUE MODULES RESIDE IN A SYSTEM LIBRARY

* , THE CONVENTIONAL TECHNIQUE MODULES RESIDE IN A PRIVATE LIBRARY

* , ALL FILESIZE VALUES WILL BE TREATED AS ESTIMATES

* , EXACT FILESIZE VALUES WILL BE ALLOWED

* , SYSTEM-DETERMINED BLOCK SIZE FOR OUTPUT DATA SETS

* , CAN BE GREATER THAN 32760 BYTES

* , IF INPUT BLOCK SIZE IS GREATER THAN 32760 BYTES

Messages

***, SYSTEM-DETERMINED BLOCK SIZE FOR OUTPUT DATA SETS**

***, CAN BE GREATER THAN 32760 BYTES**

***, SYSTEM-DETERMINED BLOCK SIZE FOR OUTPUT DATA SETS**

***, CANNOT BE GREATER THAN 32760 BYTES**

***, SYSTEM-DETERMINED BLOCK SIZE WILL NOT BE USED FOR**

***, OUTPUT DATA SETS**

***, ICEGENER WILL RETURN RC=0 FOR LRECL PADDING**

***, ICEGENER WILL RETURN RC=4 FOR LRECL PADDING**

***, ICEGENER WILL TRANSFER TO IEBGENER FOR LRECL PADDING**

***, GNPAD VALUE IGNORED FOR DIRECTLY INVOKED DFSORT**

***, ICEGENER WILL RETURN RC=0 FOR LRECL TRUNCATION**

***, ICEGENER WILL RETURN RC=4 FOR LRECL TRUNCATION**

***, ICEGENER WILL TRANSFER TO IEBGENER FOR LRECL TRUNCATION**

***, GNTRUNC VALUE IGNORED FOR DIRECTLY INVOKED DFSORT**

***, CACHE FAST WRITE CAN BE USED**

***, CACHE FAST WRITE CANNOT BE USED**

***, THE COMPACTION PERCENTAGE FOR IDRC TAPES IS UNKNOWN**

***, A COMPACTION FACTOR OF x PERCENT WILL BE USED FOR IDRC TAPES**

***, MAXIMUM VSAM BUFFER SPACE CAN BE USED**

*** , MINIMUM VSAM BUFFER SPACE WILL BE USED**

*** , OPTIMAL VSAM BUFFER SPACE CAN BE USED**

*** , SORTDIAG WILL BE SIMULATED**

*** , SORTDIAG WILL NOT BE SIMULATED**

*** , MAXIMUM DATA SPACE SIZE WILL BE x MEGABYTES**

*** , MAXIMUM DATA SPACE SIZE OBTAINABLE WILL BE USED**

. *SHORT VARIABLE COMPARE FIELDS WILL NOT BE PADDED

. *SHORT VARIABLE COMPARE FIELDS WILL BE PADDED

*** , DYNAMIC ALLOCATION DEFAULT SPACE IS x MEGABYTES**

*** , DYNAMIC ALLOCATION DEFAULT SPACE IS 256 MEGABYTES**

*** , SYSTEM-DETERMINED BLOCK SIZE WILL BE USED**

*** , FOR MESSAGE AND LIST DATA SETS**

*** , SYSTEM-DETERMINED BLOCK SIZE WILL NOT BE USED**

*** , FOR MESSAGE AND LIST DATA SETS**

*** , ZERO WILL BE TREATED AS SIGNED**

*** , ZERO WILL BE TREATED AS UNSIGNED**

*** , DFSORT WILL RETURN RC=0 FOR INCOMPLETE SPANNED RECORDS**

Messages

*** , DFSORT WILL RETURN RC=4 FOR INCOMPLETE SPANNED RECORDS**

*** , DFSORT WILL RETURN RC=16 AND TERMINATE FOR**

*** , INCOMPLETE SPANNED RECORDS**

*** , DFSORT WILL RETURN RC=0 FOR LRECL PADDING**

*** , DFSORT WILL RETURN RC=4 FOR LRECL PADDING**

*** , DFSORT WILL RETURN RC=16 AND TERMINATE FOR LRECL PADDING**

*** , DFSORT WILL RETURN RC=0 FOR LRECL TRUNCATION**

*** , DFSORT WILL RETURN RC=4 FOR LRECL TRUNCATION**

*** , DFSORT WILL RETURN RC=16 AND TERMINATE FOR LRECL TRUNCATION**

*** , DFSORT WILL RETURN RC=0 FOR SUM OVERFLOW**

*** , DFSORT WILL RETURN RC=4 FOR SUM OVERFLOW**

*** , DFSORT WILL RETURN RC=16 AND TERMINATE FOR SUM OVERFLOW**

*** , THE CENTURY WINDOW STARTS 80 YEARS BEFORE THE CURRENT YEAR**

*** , THE CENTURY WINDOW STARTS x YEARS BEFORE THE CURRENT YEAR**

*** , THE CENTURY WINDOW STARTS AT x**

*** , x BYTES IS LIMIT FOR SORTIN/SORTOUT BUFFER SPACE**

*** , 32768K BYTES IS LIMIT FOR SORTIN/SORTOUT BUFFER SPACE**

*** , x MEGABYTES IS THE LIMIT FOR DSA**

* , 0 MEGABYTES IS THE LIMIT FOR DSA BECAUSE

* , DSA IS LESS THAN OR EQUAL TO TMAXLIM

* , 32 MEGABYTES IS THE LIMIT FOR DSA

* , x BYTES IS LIMIT FOR BUFFER SPACE PER OUTFIL DATA SET

* , 2048K BYTES IS LIMIT FOR BUFFER SPACE PER OUTFIL DATA SET

* , EXPMAX IS LIMITED TO x MEGABYTES

* , EXPMAX IS LIMITED BY MAX

* , EXPMAX IS LIMITED TO x PERCENT

* , EXPOLD IS LIMITED TO x MEGABYTES

* , EXPOLD IS LIMITED BY MAX

* , EXPOLD IS LIMITED TO x PERCENT

* , EXPRES IS LIMITED TO x MEGABYTES

* , EXPRES IS LIMITED BY MAX

* , EXPRES IS LIMITED TO x PERCENT

* , LOCALE PROCESSING WILL USE x AS THE ACTIVE LOCALE

* , LOCALE PROCESSING WILL USE THE CURRENT ACTIVE LOCALE

* , LOCALE PROCESSING WILL NOT BE USED

Messages

***, LONG VARIABLE OUTPUT RECORDS WILL NOT BE ACCEPTED**

***, LONG VARIABLE OUTPUT RECORDS WILL BE TRUNCATED**

***, SAME VSAM REUSE DATA SET CANNOT BE USED FOR INPUT/OUTPUT**

***, SAME VSAM REUSE DATA SET CAN BE USED FOR INPUT/OUTPUT**

***, VSAM OUTPUT REUSE DATA SET WILL BE PROCESSED AS NEW**

***, VSAM OUTPUT REUSE DATA SET WILL BE PROCESSED AS MOD**

***, EMPTY VSAM INPUT DATA SET WILL BE ACCEPTED**

***, EMPTY VSAM INPUT DATA SET WILL NOT BE ACCEPTED**

***, CHANGED ALTSEQ CODE GENERATED:**

***, ALTSEQ CODE GENERATED:**

***, SUPPLIED 256-BYTE ALTSEQ TABLE MUST FOLLOW:**

, ** GENERATION SUCCESSFUL *

Appendix C. 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 7, "Collecting Statistical Data" on page 137 for more details on these parameters). 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 9 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 9. 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 0CYYDDDF 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.	

SMF Type-16 Record

Table 9. 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: 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

Table 9. SMF Type-16 Record (continued)

Offsets		Name	Length	Format	Description	Notation
Dec.	Hex					
+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	ICERES1B	2		Reserved.	
+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
+120	+78	ICERES1D	8		Reserved.	

Product Section

Table 10. 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: "5740SM1b"	a
+10	+A	ICERELNM	4	EBCDIC	DFSORT release level, such as "14.0"	a
+14	+E	ICERES1C	2		Reserved.	

Data Section

Table 11. 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

Data Section

Table 11. SMF Type-16 Record: Data Section (continued)

Offsets		Name	Length	Format	Description	Notation
Dec.	Hex					
+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
+34	+22	ICEFLBYT	1	binary	<p>Bit Meaning when set</p> <p>0 Reserved n</p> <p>1–2³ 00=Fixed-length records n 01=Variable-length records n 10=Variable-length spanned records n</p> <p>3–4 00=Blockset technique t 01=Peerage technique t 10=Vale technique t 11=Conventional technique t</p> <p>5 DFSORT was invoked through a program. a</p> <p>6 Sorting was completed in memory (work space was not needed) n</p> <p>7 Reserved</p>	
+35	+23	ICENDYNA	1	binary	Number of allocated work data sets.	t
+36	+24	ICEFLBY2	1	binary	<p>Bit Meaning when set</p> <p>0 A sort was specified. t</p> <p>1 A merge was specified. t</p> <p>2 A copy was specified. t</p> <p>3 DFSORT was called by ICETOOL. t</p> <p>4–7 Reserved. t</p>	

Table 11. 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 Reserved.	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 ended processing (local time, in hundredths of a second).	a
+52	+34	ICEDATEE	4	packed	Date DFSORT ended processing, in the form 0CYDDDF where F is the sign (local date).	a

Data Section

Table 11. SMF Type-16 Record: Data Section (continued)

Offsets		Name	Length	Format	Description	Notation
Dec.	Hex					
+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
+108	+6C	ICESRBTE	4	binary	Cumulative value of the SRB time charged to the address space when DFSORT ended 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

Table 11. SMF Type-16 Record: Data Section (continued)

Offsets		Name	Length	Format	Description	Notation
Dec.	Hex					
+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
+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

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 13. 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 SmartBatch 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
+3	+3	ICEINTYP	1	binary	Data set type flags byte: Bit Meaning when set 0 Tape data set. 1 DASD data set. 2 Spool, dummy or SmartBatch 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

Input Data Set Section

Table 13. SMF Type-16 Record: Input Data Set Section (continued)

Offsets		Name	Length	Format	Description	Notation
Dec.	Hex					
+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 14. SMF Type-16 Record: SORTOUT Data Set Section

Offsets		Name	Length	Format	Description	Notation
Dec.	Hex					
+0	+0	ICEOTFL1	1	binary	<p>Bit Meaning when set</p> <p>0 SmartBatch pipe data set.</p> <p>1 Striped data set.</p> <p>2 Compressed data set.</p> <p>3 VSAM extended addressability data set.</p> <p>4 HFS file.</p> <p>5-7 Reserved.</p>	f
+1	+1	ICEOTAMB	1	binary	<p>Access method flags byte:</p> <p>Bit Meaning when set</p> <p>0 EXCP was used for data set.</p> <p>1 VSAM was used for data set.</p> <p>2 BSAM was used for data set.</p> <p>3-7 Reserved.</p>	f

Table 14. SMF Type-16 Record: SORTOUT Data Set Section (continued)

Offsets		Name	Length	Format	Description	Notation
Dec.	Hex					
+2	+2	ICEOTTYP	1	binary	Data set type flags byte: Bit Meaning when set 0 Tape data set. 1 DASD data set. 2 Spool, dummy or SmartBatch 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
+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
+9A	+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.

OUTFIL Data Set Section

Table 15. 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 SmartBatch 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
+2	+2	ICEOFTYP	1	binary	Data set type flags byte: Bit Meaning when set 0 Tape data set. 1 DASD data set. 2 Spool, dummy or SmartBatch 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

Table 15. SMF Type-16 Record: OUTFIL Data Set Section (continued)

Offsets		Name	Length	Format	Description	Notation
Dec.	Hex					
+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
+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

OUTFIL Data Set Section

Table 15. SMF Type-16 Record: OUTFIL Data Set Section (continued)

Offsets		Name	Length	Format	Description	Notation
Dec.	Hex					
+9A	+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

1. ICECPUT is set to zero if STIMER=NO.
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 139 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/WITES (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.

Appendix D. 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 7, “Collecting Statistical Data” on page 137 for a description of the parameters.

Notes to all the tables are listed after “Sorting with Data Space Statistics” on page 229.

SMF Statistics

Table 16. 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. ¹⁸																		
+6	+6	ICESMFS1	1	SMF status information: <table border="0"> <tr> <td>Bit</td> <td>Meaning</td> </tr> <tr> <td>0</td> <td>An attempt was made to write an SMF record.</td> </tr> <tr> <td>1</td> <td>The DFSORT SMF record was successfully transferred to an SMF buffer.</td> </tr> <tr> <td>2</td> <td>SMF is not active in the system.</td> </tr> <tr> <td>3</td> <td>SMF=SHORT is in effect.</td> </tr> <tr> <td>4</td> <td>SMF=FULL is in effect.</td> </tr> <tr> <td>5</td> <td>SMF=NO is in effect.</td> </tr> <tr> <td>6</td> <td>STIMER=YES is in effect.</td> </tr> <tr> <td>7</td> <td>Reserved.</td> </tr> </table>	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	STIMER=YES is in effect.	7	Reserved.
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	STIMER=YES is in effect.																					
7	Reserved.																					
+7	+7	ICESMFR1	1	Reserved.																		

General Statistics

Table 17. 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.
+8	+8	ICEGPDNM	8	Product name of control program. This information is copied from CVT field CVTPRODN (EBCDIC). For z/OS and OS/390, 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”).

ICETEXIT Parameters

Table 17. DFSORT General Statistics (continued)

Offsets		Name	Length	Description												
Dec.	Hex															
+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 "14.0").												
+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). ³⁷ 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: <table border="0"> <tr> <td>Bit</td> <td>Meaning</td> </tr> <tr> <td>0</td> <td>DFSORT modules for the technique in use are resident.</td> </tr> <tr> <td>1</td> <td>DFSORT was loaded above 16MB virtual. ²⁷</td> </tr> <tr> <td>2</td> <td>The DFSORT SVC is installed.</td> </tr> <tr> <td>3-7</td> <td>Reserved.</td> </tr> </table>	Bit	Meaning	0	DFSORT modules for the technique in use are resident.	1	DFSORT was loaded above 16MB virtual. ²⁷	2	The DFSORT SVC is installed.	3-7	Reserved.		
Bit	Meaning															
0	DFSORT modules for the technique in use are resident.															
1	DFSORT was loaded above 16MB virtual. ²⁷															
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: <table border="0"> <tr> <td>17</td> <td>=</td> <td>SVC 109</td> <td>MODULE IGX00017</td> </tr> <tr> <td>38</td> <td>=</td> <td>SVC 109</td> <td>MODULE IGX00038</td> </tr> <tr> <td>N</td> <td>=</td> <td>SVC N</td> <td>(N = 200 - 255)</td> </tr> </table>	17	=	SVC 109	MODULE IGX00017	38	=	SVC 109	MODULE IGX00038	N	=	SVC N	(N = 200 - 255)
17	=	SVC 109	MODULE IGX00017													
38	=	SVC 109	MODULE IGX00038													
N	=	SVC N	(N = 200 - 255)													
+66	+42	ICEGRES3	6	Reserved.												

Options Statistics

Table 18. 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.

Table 18. DFSORT Option Statistics (continued)

Offsets		Name	Length	Description
Dec.	Hex			
+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 OUTFIL 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.
+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.
+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.

ICETEXIT Parameters

Table 18. DFSORT Option Statistics (continued)

Offsets		Name	Length	Description
Dec.	Hex			
+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.
+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: ³¹ 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. ³¹
+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.

Table 18. DFSORT Option Statistics (continued)

Offsets		Name	Length	Description
Dec.	Hex			
+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 19. 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.
+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 PD0 format was used. 5-7 Reserved.

ICETEXIT Parameters

Table 19. 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. 5-7 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 20. DFSORT Virtual Storage Statistics

Offsets		Name	Length	Description
Dec.	Hex			
+0	+0	ICEVSTAT	4	Length of the virtual storage statistics, including this field.
+4	+4	ICEVARSA	4	ARESALL value in effect. ²⁰
+8	+8	ICEVARIV	4	ARESINV value in effect. ²⁰
+12	+C	ICEVOVER	4	OVERRGN value in effect. ²⁰
+16	+10	ICEVTMAX	4	TMAXLIM value in effect. ²⁰
+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. ²¹
+52	+34	ICEVTHRY	4	Total amount of virtual storage theoretically available to DFSORT considering MINLIM. ²¹

Table 20. DFSORT Virtual Storage Statistics (continued)

Offsets		Name	Length	Description
Dec.	Hex			
+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. ^{21, 22}
+60	+3C	ICEVTABV	4	Amount of virtual storage theoretically available to DFSORT above 16MB virtual. ²³
+64	+40	ICEVAABV	4	Amount of virtual storage available above 16MB virtual after DFSORT has released the ARESALL and ARESINV space. ²³
+68	+44	ICEVRES3	4	Reserved.

Phase-Timing Statistics

Note: Because DFSORT makes phase-timing measurements internally, consider the values provided in Table 21 to be *approximate timings only*.

Table 21. 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.																
+8	+8	ICEPTST	1	Phase-timing status flags: <table border="0"> <tr> <td>Bit</td> <td>Meaning when set</td> </tr> <tr> <td>0</td> <td>Timing information exists for phase 0.</td> </tr> <tr> <td>1</td> <td>Timing information exists for phase 1.</td> </tr> <tr> <td>2</td> <td>Timing information exists for phase 2.</td> </tr> <tr> <td>3</td> <td>Sort timing information exists for phase 3.</td> </tr> <tr> <td>4</td> <td>Merge or copy timing information exists for phase 3.</td> </tr> <tr> <td>5</td> <td>Timing information exists for phase 4.</td> </tr> <tr> <td>6-7</td> <td>Reserved.</td> </tr> </table>	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.	6-7	Reserved.
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.																			
6-7	Reserved.																			
+9	+9	ICEPRES2	3	Reserved.																

Phase 0 Timings:

Table 21. DFSORT Phase-Timing Statistics

Offsets		Name	Length	Description
Dec.	Hex			
+12	+C	ICEP0TCB	4	TCB time for this phase, in hundredths of seconds. ³²
+16	+10	ICEP0ELT	4	Elapsed time, in hundredths of seconds.
+20	+14	ICEP0SRB	4	SRB time for this phase, in hundredths of seconds.

Phase 1 Timings:

ICETEXIT Parameters

Table 21. DFSORT Phase-Timing Statistics

Offsets		Name	Length	Description
Dec.	Hex			
+24	+18	ICEP1TCB	4	TCB time for this phase, in hundredths of seconds. ³²
+28	+1C	ICEP1ELT	4	Elapsed time, in hundredths of seconds.
+32	+20	ICEP1SRB	4	SRB time for this phase, in hundredths of seconds.

Phase 2 Timings:

Table 21. DFSORT Phase-Timing Statistics

Offsets		Name	Length	Description
Dec.	Hex			
+36	+24	ICEP2TCB	4	TCB time for this phase, in hundredths of seconds. ³²
+40	+28	ICEP2ELT	4	Elapsed time, in hundredths of seconds.
+44	+2C	ICEP2SRB	4	SRB time for this phase, in hundredths of seconds.

Phase 3 SORT Timings:

Table 21. DFSORT Phase-Timing Statistics

Offsets		Name	Length	Description
Dec.	Hex			
+48	+30	ICEP3TCB	4	TCB time for this phase, in hundredths of seconds. ³²
+52	+34	ICEP3ELT	4	Elapsed time, in hundredths of seconds.
+56	+38	ICEP3SRB	4	SRB time for this phase, in hundredths of seconds.

Phase 3 MERGE or COPY Timings:

Table 21. DFSORT Phase-Timing Statistics

Offsets		Name	Length	Description
Dec.	Hex			
+60	+3C	ICEP3TBM	4	TCB time for this phase, in hundredths of seconds. ³²
+64	+40	ICEP3ETM	4	Elapsed time, in hundredths of seconds.
+68	+44	ICEP3SBM	4	SRB time for this phase, in hundredths of seconds.

Phase 4 Timings:

Table 21. DFSORT Phase-Timing Statistics

Offsets		Name	Length	Description
Dec.	Hex			
+72	+48	ICEP4TCB	4	TCB time for this phase, in hundredths of seconds. ³²
+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 22. 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). ³⁶
+52	+34	ICESINCM	1	SORTIN information flags: Bit Meaning when set 0 SORTIN is concatenated. ²⁹ 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: ³⁶ 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 23. 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 SmartBatch pipe data set, or an HFS file. 1 SORTIN is tape. 2 SORTIN is DASD. 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.

ICETEXIT Parameters

Table 23. SORTIN first volume or first data set information:24 (continued)

Offsets		Name	Length	Description
Dec.	Hex			
+68	+44	ICESINDA	1	SORTIN DASD flags, byte 1: Bit Meaning when set 0 SORTIN resides on a shared DASD. 1 SORTIN resides on a non-synchronous DASD. 2 SORTIN resides on a DASD connected to a 3880 Model 13 control unit. 3 SORTIN resides on a DASD connected to a 3880 Model 23 control unit. 4 SORTIN resides on a DASD connected to a 3990 Model 3 control unit. 5 SORTIN resides on a DASD connected to a 3990 Model 2 control unit. 6 SORTIN resides on a DASD connected to an uncached 9343 or 9341 control unit. 7 Reserved.
+69	+45	ICESIND2	1	SORTIN DASD flags, byte 2: Bit Meaning when set 0 Reserved. 1 SORTIN resides on a DASD connected to a cached 9343 control unit. 2 SORTIN resides on a DASD connected to a 3990 Model 6 control unit. 3 SORTIN resides on a DASD connected to a 2105 control unit. 4-7 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. ³⁸
+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. ³³ 3-7 Reserved. This value is provided when available. This value is provided only for a DASD data set.
+80	+50	ICESINBS	4	Average block size specified for block allocation requests. This value is provided only for a DASD 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. ³⁸
+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 DASD data set.
+92	+5C	ICESINEX	2	Total number of extents allocated. This value is provided only for a DASD data set.

Table 24. Additional SORTIN Information

Offsets		Name	Length	Description
Dec.	Hex			
+94	+5E	ICESNCDV	1	Additional SORTIN information, byte 1: ^{25, 30} Bit Meaning when set 0 A tape was contained in the SORTIN concatenation list. 1 A DASD 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: ¹⁹ 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 DASD. 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 DASD. 2 A concatenated SORTIN data set, or one or more volumes of a multivolume SORTIN data set (other than the first volume), resides on a DASD connected to a 3880 Model 13 control unit. 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 DASD connected to a 3880 Model 23 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 DASD connected to a 3990 Model 3 control unit. 5 A concatenated SORTIN data set, or one or more volumes of a multivolume SORTIN data set (other than the first volume), resides on a DASD connected to a 3990 Model 2 control unit. 6 A concatenated SORTIN data set, or one or more volumes of a multivolume SORTIN data set (other than the first volume), resides on DASD connected to an uncached 9343 or 9341 control unit. 7 Reserved.

ICETEXIT Parameters

Table 24. Additional SORTIN Information (continued)

Offsets		Name	Length	Description
Dec.	Hex			
+96	+60	ICESNCD2	1	Additional SORTIN information, byte 3: ¹⁹ Bit Meaning when set 0 Reserved. 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 DASD connected to a cached 9343 control unit. 2 A concatenated SORTIN data set, or one or more volumes of a multivolume SORTIN data set (other than the first volume), resides on a DASD connected to a 3990 Model 6 control unit. 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 DASD connected to a 2105 control unit. 4-7 Reserved.
+97	+61	ICESINR9	7	Reserved.

SORTOUT Statistics

Table 25. 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). ³⁶
+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.
+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 26. SORTOUT first volume or first data set information:26

Offsets		Name	Length	Description																		
Dec.	Hex																					
+56	+38	ICESOTDV	1	<p>SORTOUT device flags:</p> <table border="0"> <tr> <td>Bit</td> <td>Meaning when set</td> </tr> <tr> <td>0</td> <td>SORTOUT is a spool, dummy or SmartBatch pipe data set, or an HFS file.</td> </tr> <tr> <td>1</td> <td>SORTOUT is tape.</td> </tr> <tr> <td>2</td> <td>SORTOUT is DASD.</td> </tr> <tr> <td>3-7</td> <td>Reserved.</td> </tr> </table>	Bit	Meaning when set	0	SORTOUT is a spool, dummy or SmartBatch pipe data set, or an HFS file.	1	SORTOUT is tape.	2	SORTOUT is DASD.	3-7	Reserved.								
Bit	Meaning when set																					
0	SORTOUT is a spool, dummy or SmartBatch pipe data set, or an HFS file.																					
1	SORTOUT is tape.																					
2	SORTOUT is DASD.																					
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.																		
+68	+44	ICESOTDA	1	<p>SORTOUT DASD flags, byte 1:</p> <table border="0"> <tr> <td>Bit</td> <td>Meaning when set</td> </tr> <tr> <td>0</td> <td>SORTOUT resides on a shared DASD.</td> </tr> <tr> <td>1</td> <td>SORTOUT resides on a non-synchronous DASD.</td> </tr> <tr> <td>2</td> <td>SORTOUT resides on a DASD connected to a 3880 Model 13 control unit.</td> </tr> <tr> <td>3</td> <td>SORTOUT resides on a DASD connected to a 3880 Model 23 control unit.</td> </tr> <tr> <td>4</td> <td>SORTOUT resides on a DASD connected to a 3990 Model 3 control unit.</td> </tr> <tr> <td>5</td> <td>SORTOUT resides on a DASD connected to a 3990 Model 2 control unit.</td> </tr> <tr> <td>6</td> <td>SORTOUT resides on a DASD connected to an uncached 9343 or 9341 control unit.</td> </tr> <tr> <td>7</td> <td>Reserved.</td> </tr> </table>	Bit	Meaning when set	0	SORTOUT resides on a shared DASD.	1	SORTOUT resides on a non-synchronous DASD.	2	SORTOUT resides on a DASD connected to a 3880 Model 13 control unit.	3	SORTOUT resides on a DASD connected to a 3880 Model 23 control unit.	4	SORTOUT resides on a DASD connected to a 3990 Model 3 control unit.	5	SORTOUT resides on a DASD connected to a 3990 Model 2 control unit.	6	SORTOUT resides on a DASD connected to an uncached 9343 or 9341 control unit.	7	Reserved.
Bit	Meaning when set																					
0	SORTOUT resides on a shared DASD.																					
1	SORTOUT resides on a non-synchronous DASD.																					
2	SORTOUT resides on a DASD connected to a 3880 Model 13 control unit.																					
3	SORTOUT resides on a DASD connected to a 3880 Model 23 control unit.																					
4	SORTOUT resides on a DASD connected to a 3990 Model 3 control unit.																					
5	SORTOUT resides on a DASD connected to a 3990 Model 2 control unit.																					
6	SORTOUT resides on a DASD connected to an uncached 9343 or 9341 control unit.																					
7	Reserved.																					
+69	+45	ICESOTD2	1	<p>SORTOUT DASD flags, byte 2:</p> <table border="0"> <tr> <td>Bit</td> <td>Meaning when set</td> </tr> <tr> <td>0</td> <td>Reserved</td> </tr> <tr> <td>1</td> <td>SORTOUT resides on a DASD connected to a cached 9343 control unit.</td> </tr> <tr> <td>2</td> <td>SORTOUT resides on a DASD connected to a 3990 Model 6 control unit.</td> </tr> <tr> <td>3</td> <td>SORTOUT resides on a DASD connected to a 2105 control unit.</td> </tr> <tr> <td>4-7</td> <td>Reserved.</td> </tr> </table>	Bit	Meaning when set	0	Reserved	1	SORTOUT resides on a DASD connected to a cached 9343 control unit.	2	SORTOUT resides on a DASD connected to a 3990 Model 6 control unit.	3	SORTOUT resides on a DASD connected to a 2105 control unit.	4-7	Reserved.						
Bit	Meaning when set																					
0	Reserved																					
1	SORTOUT resides on a DASD connected to a cached 9343 control unit.																					
2	SORTOUT resides on a DASD connected to a 3990 Model 6 control unit.																					
3	SORTOUT resides on a DASD connected to a 2105 control unit.																					
4-7	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. ³⁸																		
+76	+4C	ICESOTR7	3	Reserved.																		
+79	+4F	ICESOTSP	1	<p>SORTOUT space allocation request:</p> <table border="0"> <tr> <td>Bit</td> <td>Meaning when set</td> </tr> <tr> <td>0</td> <td>Cylinder request.</td> </tr> <tr> <td>1</td> <td>Track request.</td> </tr> <tr> <td>2</td> <td>Block request.³⁴</td> </tr> <tr> <td>3-7</td> <td>Reserved.</td> </tr> </table> <p>This value is provided when available. This value is provided only for a DASD data set.</p>	Bit	Meaning when set	0	Cylinder request.	1	Track request.	2	Block request. ³⁴	3-7	Reserved.								
Bit	Meaning when set																					
0	Cylinder request.																					
1	Track request.																					
2	Block request. ³⁴																					
3-7	Reserved.																					

ICETEXIT Parameters

Table 26. SORTOUT first volume or first data set information:26 (continued)

Offsets		Name	Length	Description
Dec.	Hex			
+80	+50	ICESOTBS	4	Average block size specified for block allocation requests. This value is provided only for a DASD 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. ³⁸
+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 DASD data set.
+92	+5C	ICESOTEX	2	Total number of extents allocated. This value is provided only for a DASD data set.

Table 27. SORTOUT Multivolume Information:19,28

Offsets		Name	Length	Description																		
Dec.	Hex																					
+94	+5E	ICESTCDA	1	<p>SORTOUT multivolume flags, byte 1:</p> <table border="0"> <thead> <tr> <th>Bit</th> <th>Meaning when set</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>One or more volumes (but not the first volume) of the multivolume SORTOUT data set resides on a shared DASD.</td> </tr> <tr> <td>1</td> <td>One or more volumes (but not the first volume) of the multivolume SORTOUT data set resides on a non-synchronous DASD.</td> </tr> <tr> <td>2</td> <td>One or more volumes (but not the first volume) of the multivolume SORTOUT data set resides on a DASD connected to a 3880 Model 13 control unit.</td> </tr> <tr> <td>3</td> <td>One or more volumes (but not the first volume) of the multivolume SORTOUT data set resides on a DASD connected to a 3880 Model 23 control unit.</td> </tr> <tr> <td>4</td> <td>One or more volumes (but not the first volume) of the multivolume SORTOUT data set resides on a DASD connected to a 3990 Model 3 control unit.</td> </tr> <tr> <td>5</td> <td>One or more volumes (but not the first volume) of the multivolume SORTOUT data set resides on a DASD connected to a 3990 Model 2 control unit.</td> </tr> <tr> <td>6</td> <td>One or more volumes (but not the first volume) of the multivolume SORTOUT data set resides on a DASD connected to an uncached 9343 or 9341 control unit.</td> </tr> <tr> <td>7</td> <td>Reserved.</td> </tr> </tbody> </table>	Bit	Meaning when set	0	One or more volumes (but not the first volume) of the multivolume SORTOUT data set resides on a shared DASD.	1	One or more volumes (but not the first volume) of the multivolume SORTOUT data set resides on a non-synchronous DASD.	2	One or more volumes (but not the first volume) of the multivolume SORTOUT data set resides on a DASD connected to a 3880 Model 13 control unit.	3	One or more volumes (but not the first volume) of the multivolume SORTOUT data set resides on a DASD connected to a 3880 Model 23 control unit.	4	One or more volumes (but not the first volume) of the multivolume SORTOUT data set resides on a DASD connected to a 3990 Model 3 control unit.	5	One or more volumes (but not the first volume) of the multivolume SORTOUT data set resides on a DASD connected to a 3990 Model 2 control unit.	6	One or more volumes (but not the first volume) of the multivolume SORTOUT data set resides on a DASD connected to an uncached 9343 or 9341 control unit.	7	Reserved.
Bit	Meaning when set																					
0	One or more volumes (but not the first volume) of the multivolume SORTOUT data set resides on a shared DASD.																					
1	One or more volumes (but not the first volume) of the multivolume SORTOUT data set resides on a non-synchronous DASD.																					
2	One or more volumes (but not the first volume) of the multivolume SORTOUT data set resides on a DASD connected to a 3880 Model 13 control unit.																					
3	One or more volumes (but not the first volume) of the multivolume SORTOUT data set resides on a DASD connected to a 3880 Model 23 control unit.																					
4	One or more volumes (but not the first volume) of the multivolume SORTOUT data set resides on a DASD connected to a 3990 Model 3 control unit.																					
5	One or more volumes (but not the first volume) of the multivolume SORTOUT data set resides on a DASD connected to a 3990 Model 2 control unit.																					
6	One or more volumes (but not the first volume) of the multivolume SORTOUT data set resides on a DASD connected to an uncached 9343 or 9341 control unit.																					
7	Reserved.																					
+95	+5F	ICESTCD2	1	<p>SORTOUT multivolume flags, byte 2:</p> <table border="0"> <thead> <tr> <th>Bit</th> <th>Meaning when set</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Reserved.</td> </tr> <tr> <td>1</td> <td>One or more volumes (but not the first volume) of the multivolume SORTOUT data set resides on a DASD connected to a cached 9343 control unit.</td> </tr> <tr> <td>2</td> <td>One or more volumes (but not the first volume) of the multivolume SORTOUT data set resides on a DASD connected to a 3990 Model 6 control unit.</td> </tr> <tr> <td>3</td> <td>One or more volumes (but not the first volume) of the multivolume SORTOUT data set resides on a DASD connected to a 2105 control unit.</td> </tr> <tr> <td>4-7</td> <td>Reserved.</td> </tr> </tbody> </table>	Bit	Meaning when set	0	Reserved.	1	One or more volumes (but not the first volume) of the multivolume SORTOUT data set resides on a DASD connected to a cached 9343 control unit.	2	One or more volumes (but not the first volume) of the multivolume SORTOUT data set resides on a DASD connected to a 3990 Model 6 control unit.	3	One or more volumes (but not the first volume) of the multivolume SORTOUT data set resides on a DASD connected to a 2105 control unit.	4-7	Reserved.						
Bit	Meaning when set																					
0	Reserved.																					
1	One or more volumes (but not the first volume) of the multivolume SORTOUT data set resides on a DASD connected to a cached 9343 control unit.																					
2	One or more volumes (but not the first volume) of the multivolume SORTOUT data set resides on a DASD connected to a 3990 Model 6 control unit.																					
3	One or more volumes (but not the first volume) of the multivolume SORTOUT data set resides on a DASD connected to a 2105 control unit.																					
4-7	Reserved.																					

Work Data Set Statistics

Table 28. 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.

EXCP Counts per DFSORT phase:

Table 28. DFSORT Work Data Set Statistics

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.

Work Data Set Entries: ³⁵

Table 28. DFSORT Work Data Set Statistics

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 29. 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.

ICETEXIT Parameters

Table 29. 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 DASD. 1 This data set resides on a non-synchronous DASD. 2 This data set resides on a DASD connected to a 3880 Model 13 control unit. 3 This data set resides on a DASD connected to a 3880 Model 23 control unit. 4 This data set resides on a DASD connected to a 3990 Model 3 control unit. 5 This data set resides on a DASD connected to a 3990 Model 2 control unit. 6 This data set resides on a DASD connected to an uncached 9343 or 9341 control unit. 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 DASD connected to a cached 9343 control unit. 2 This data set resides on a DASD connected to a 3990 Model 6 control unit. 3 This data set resides on a DASD connected to a 2105 control unit. 4-7 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 ended.
+36	+24	ICESER3	4	Reserved.
+40	+28	ICESEEXI	2	Number of extents allocated initially.
+42	+2A	ICESEEXF	2	Number of extents allocated when sorting ended.
+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 30. 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 storage theoretically available to DFSORT for this run (for Hiperspace).
+12	+C	ICEHPGSP	4	Pages of storage available at run time that will cause some paging (for Hiperspace).
+16	+10	ICEHPGNI	4	Pages of 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 31. 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 and expanded storage not in use ³⁹ .
+12	+C	ICEDSBTB	4	Pages of central and expanded storage not in use plus some inactive pages of central and expanded storage ³⁹ .
+16	+10	ICEDSBTC	4	Pages of central and expanded storage not in use plus some inactive pages of central and expanded storage plus some auxiliary storage pages ³⁹ .
+20	+14	ICEDSBKA	4	Pages of expanded storage not in use ³⁹ .
+24	+18	ICEDSBKB	4	Pages of expanded storage not in use plus some inactive pages of expanded storage ³⁹ .
+28	+1C	ICEDSBKC	4	Pages of expanded storage not in use plus some inactive pages of expanded storage plus some auxiliary storage pages ³⁹ .
+32	+20	ICEDSCR	4	Pages of data space created.
+36	+24	ICEDSUS	4	Pages of data space used.

-
18. The SMF return code is supplied only when SMF is active and either "SMF=FULL" or "SMF=SHORT" is in effect.
 19. The information in this field is provided for data sets and volumes containing active data only.
 20. This value is provided only when Blockset is selected.
 21. This value is the same as the corresponding value in message ICE092I or ICE093I.
 22. This value is not meaningful when the storage taken by an invoking program or exit is greater than REGION or SIZE/MAINSIZE.
 23. This value is the same as the corresponding value in message ICE156I. It is provided only when Blockset is selected.
 24. 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.
 25. 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.
 26. 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.
 27. This bit will not be set if DFSORT modules for the technique in use are resident.
 28. 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.
 29. If this bit is set, then the additional SORTIN information bytes below contains valid information.
 30. No information is provided in this area when execution is not in the Blockset technique.
 31. This field is zero if FILSZ/SIZE is not specified.
 32. This value is not provided if STIMER=NO is in effect.
 33. When this bit is set, ICESINBS contains the average block length space request.
 34. When this bit is set, ICESOTBS contains the average block length space request.
 35. All work data set entries, if any, are ordered by most heavily used to least heavily used.
 36. 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.
 37. 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.
 38. 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.
 39. In 64-bit real mode, there are no expanded storage pages.

Appendix E. Notices

This information was developed for products and services offered in the U.S.A. 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 give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, 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.

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
Information Enabling Requests
Dept. DZWA

5600 Cottle Road
San Jose, CA 95193 U.S.A.

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 information 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.

Programming Interface Information

This book primarily documents information that is NOT intended to be used as a Programming Interface of DFSORT.

This book 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:

```

_____ Programming Interface information _____
Programming Interface information
_____ End of Programming Interface information _____
```

Trademarks

The following terms are trademarks of the IBM Corporation in the United States or other countries or both:

DFSMS	MVS
DFSMSHsm	MVS/DFP
DFSMS/MVS	MVS/ESA
DFSMSrmm	OS/390
DFSORT	RACF
Hipersorting	VM/ESA
IBM	z/OS
Language Environment	z/VM

Other company, product, and service names may be trademarks or service marks of other companies.

Summary of Changes

New Programming Support for Release 13 (PTFs after April, 1996)

Additional Year 2000 Features

A new Y2S format can order and transform two-digit character or zoned decimal year data according to the century window, while handling binary zeros, blanks and binary ones in the year field as special indicators.

A new Y2B format can order and transform two-digit binary year data according to the century window.

FREE=CLOSE support for DFSPARM makes it possible to override the SORT statements generated by multiple COBOL SORT verbs in the same COBOL program.

OS/390 Registration

With OS/390 R2 and above, a check is performed to ensure that the DFSORT product is licensed for use, either as a feature of OS/390 or as a separate program product.

New Programming Support for Release 13 (PTFs – April, 1996)

Year 2000 Features

New Y2C, Y2Z, Y2P and Y2D formats, in conjunction with a new Y2PAST installation and run-time option, allow you to handle two-digit year data in the following ways:

- Set the appropriate century window for your applications (for example, 1915-2014 or 1950-2049).
- Order two-digit character, zoned decimal, packed decimal or decimal year data according to the century window using Blockset SORT or MERGE (for example, order 96 representing 1996 before 00 representing 2000 in ascending sequence, or order 00 before 96 in descending sequence).
- Transform two-digit character, zoned decimal, packed decimal or decimal year data to four-digit character year data according to the century window using OUTFIL OUTREC (for example, transform 96 to 1996 and 00 to 2000).

A new PD0 format allows you to order and transform parts of packed decimal fields (for example, month and day in date fields) using SORT, MERGE and OUTFIL.

Performance Improvements for FLR and VLR Blockset Sorts

Performance improvements for FLR and VLR Blockset sorts include the following:

- Dataspace sorting can now be used for variable-length record sort applications.
- DFSORT data processing methods have been improved.
- Dynamic storage adjustment is a new feature that allows DFSORT to automatically use more storage than the TMAXLIM value for a Blockset sort application if DFSORT determines that doing so should improve performance. New installation option DSA=n has been added to enable you to specify the dynamic storage adjustment limit.

- The upper limit for the amount of main storage that can be specified and used by DFSORT has been raised from 32M to 2000M. Specifying more main storage can provide the following benefits:
 - It allows DFSORT to sort very large data sets more efficiently.
 - It allows more sort applications to be done entirely in main storage, eliminating the need for intermediate work space and greatly reducing the EXCP counts for those applications.
 - It increases the maximum amount of data DFSORT can process in a single sort application.
- New installation option IOMAXBF=n has been added to enable you to specify the upper limit for the amount of storage to be used for SORTIN and SORTOUT data set buffers, which in turn limits the amount of data that can be transferred in a single I/O operation.
- The upper limit for the number of JCL and dynamically allocated work data sets that can be specified and used by DFSORT's Blockset technique has been raised from 32 to 100. The use of more work data sets increases the maximum amount of data DFSORT can process in a single sort application.
- Changes to the DFSORT SVC provide caching selection enhancements that improve storage control caching performance, especially for SORTIN and SORTOUT devices.
- DFSORT can now use NOEQUALS for VLR Blockset applications if EQUALS=NO is specified at installation or NOEQUALS is specified at run-time. The use of NOEQUALS can improve performance and is recommended for applications for which the order of records that collate identically need not be preserved from input to output. To minimize migration concerns, the IBM-supplied default for the ICEMAC EQUALS option is the new value VLBLKSET, which is equivalent to EQUALS=YES for VLR Blockset applications and to EQUALS=NO for all other applications.

Floating Point for SUM

FL format can now be used with the SUM control statement for short (4-byte), long (8-byte) and extended (16-byte) floating point data.

Security Improvements

Changes to the DFSORT SVC provide security improvements that bring DFSORT up to B1 security standards.

EXCPVR Processing Removed

To enhance DFSORT's protection of system integrity, EXCPVR processing will no longer be used. EXCPVR parameter values will continue to be accepted, but will have no effect on DFSORT processing. In general, the performance improvements provided by EXCPVR processing have diminished with newer technologies and will be more than offset by the performance improvements listed above. Please ignore any references to EXCPVR in this book; all such references will be deleted when the book is updated.

New Device Support for Release 13 (PTFs)

The IBM 3590 Magnetic Tape Subsystem is supported for input, output and work data sets.

New Programming Support for Release 13

DFSORT's Performance Booster for The SAS** System

DFSORT Release 13 provides significant CPU time improvements for SAS applications. To take advantage of this new feature, contact SAS Institute Inc. for details of the support they provide to enable this enhancement.

Dynamic Hipersorting

Dynamic Hipersorting is a new, automatic feature that eliminates the unintended system paging activity and expanded storage and paging data set space shortages that sometimes resulted from a large amount of Hipersorting activity, especially from multiple concurrent Hipersorting applications.

Dynamic Hipersorting allows for more optimal DFSORT and system performance and provides installation options that allow you to customize HIPRMAX=OPTIMAL to your own criteria. With the advent of this feature, we recommend that you use HIPRMAX=OPTIMAL as your site default.

Performance

Performance enhancements for DFSORT applications that use the Blockset technique include the following:

- Dataspace sorting, introduced in R12 for fixed-length record sort applications, now available for variable-length record sort applications (MVS/ESA only)
- Improved data processing methods for fixed-length record sort applications
- OUTFIL processing for producing multiple output data sets using a single pass over one or more input data sets.

OUTFIL Processing

OUTFIL is a new DFSORT control statement that allows you to create one or more output data sets for a sort, copy, or merge application from a single pass over one or more input data sets. You can use multiple OUTFIL statements, with each statement specifying the OUTFIL processing to be performed for one or more output data sets. OUTFIL processing begins after all other processing ends (that is, after processing for exits, options, and other control statements). OUTFIL statements support a wide variety of output data set tasks, including:

- Creation of multiple output data sets containing unedited or edited records from a single pass over one or more input data sets.
- Creation of multiple output data sets containing different ranges or subsets of records from a single pass over one or more input data sets. In addition, records that are not selected for any subset can be saved in another output data set.
- Conversion of variable-length record data sets to fixed-length record data sets.
- Sophisticated editing capabilities such as hexadecimal display and control of the way numeric fields are presented with respect to length, leading or suppressed zeros, symbols (for example, the thousands separator and decimal point), leading and trailing positive and negative signs, and so on. Twenty-six pre-defined editing masks are available for commonly used numeric editing patterns, encompassing many of the numeric notations used throughout the world. In addition, a virtually unlimited number of numeric editing patterns are available via user-defined editing masks.

- Selection of a character or hexadecimal string for output from a lookup table, based on a character, hexadecimal, or bit string as input (that is, lookup and change).
- Highly detailed three-level (report, page, and section) reports containing a variety of report elements you can specify (for example, current date, current time, page number, character strings, and blank lines) or derive from the input records (for example, character fields, edited numeric input fields, record counts, and edited totals, maximums, minimums, and averages for numeric input fields).

National Language Support

Cultural Sort and Merge

DFSORT will allow the selection of an active locale at installation or run time and 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.

Cultural Include and Omit

DFSORT will allow the selection of an active locale at installation or run time and 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.

UTFIL Reports

UTFIL allows date, time, and numeric values in reports to be formatted in many of the notations used throughout the world.

ICETOOL Reports

ICETOOL's DISPLAY operator allows date, time, and numeric values in reports to be formatted in many of the notations used throughout the world.

ICETOOL Enhancements

ICETOOL is now even more versatile as a result of enhancements to the existing operators. The improvements to ICETOOL include:

- Allowing more data to be displayed with the DISPLAY and OCCUR operators. DISPLAY now allows up to 20 fields (increased from 10) and a line length of up to 2048 characters (increased from 121). OCCUR now allows a line length of up to 2048 characters (increased from 121).
- More extensive formatting capabilities for numeric fields with the DISPLAY operator. Formatting items can be used to change the appearance of individual numeric fields in reports with respect to separators, decimal point, decimal places, signs, division, leading strings, floating strings and trailing strings. Thirty-three pre-defined editing masks are available for commonly used numeric editing patterns, encompassing many of the numeric notations used throughout the world. Leading and trailing strings can also be used with character fields.
- Display of the four-digit or two-digit year with the DISPLAY and OCCUR operators.
- Division of reports into sections with the DISPLAY operator, based on the values in a character or numeric break field. Statistics (total, maximum, minimum and/or average) can be displayed for each section as well as for the entire report.
- Automatic use of UTFIL processing for a list of TO ddnames with the COPY and SORT operators, resulting in creation of multiple TO (output) data sets from a single pass over the FROM (input) data set.

- Allowing OUTFIL statements to be specified in the USING data set in addition to or instead of the TO operand with the COPY and SORT operators.
- Allowing the active locale to be specified for the COPY, COUNT and SORT operators, in order to override the installation default for the active locale. Thus, multiple active locales can be used in the same ICETOOL job step for these operators.
- Allowing the last record for each unique field value to be kept with the SELECT operator.

INCLUDE/OMIT Substring Search

INCLUDE and OMIT function enhancements provide powerful substring search capability to allow inclusion or omission of records when:

- A specified character or hexadecimal constant is found anywhere within a specified input field (that is, a constant is a substring within a field) or
- A specified input value is found anywhere within a specified character or hexadecimal constant (that is, a field is a substring within a constant).

SMF Type-16 Record Enhancements

New fields, such as information pertaining to each DFSORT run about SORTIN, SORTINnn, SORTOUT and OUTFIL data sets, control statements, record counts, specified values for E15, E35, HIPRMAX, DSPSIZE, FILSZ, LOCALE and AVGRLEN, have been added to DFSORT's SMF type-16 record.

SMF=FULL, SMF=SHORT, and SMF=NO can now be specified in an OPTION statement in DFSPARM or the extended parameter list, to produce or suppress the SMF type-16 record for an individual application.

Note: The offsets of fields ICESPGN, ICEUSER, and ICEGROUP have changed in the Release 13 SMF record. If you have programs that reference those fields, recompile them using the Release 13 version of the ICESMF macro, before attempting to run them against Release 13 SMF records.

Other Enhancements

Several ICEMAC installation options have been added or changed:

- The IBM-supplied default for EXCPVR has been changed from ALL to NONE.
- The IBM-supplied default for DYNAUTO has been changed from NO to YES.
- SDBMSG enables you to specify whether DFSORT should use the system-determined optimum block size for DFSORT message data sets and ICETOOL message and list data sets.
- LOCALE enables you to select an active locale.
- ODMAXBF enables you to specify the maximum buffer space DFSORT can use for each OUTFIL data set.
- EXPMAX enables you to specify the maximum total amount of available storage to be used for all Hipersorting applications.
- EXPOLD enables you to specify the maximum total amount of old expanded storage to be used at any one time by all Hipersorting applications.
- EXPRES enables you to specify the minimum amount of available expanded storage to be reserved by DFSORT for use by non-Hipersorting applications.

Several run-time options have been added or changed:

- LOCALE enables you to select an active locale.

- SMF enables you to specify whether DFSORT is to produce SMF type-16 records.
- ODMAXBF enables you to specify the maximum buffer space DFSORT can use for each OUTFIL data set.
- NZDPRINT enables you to indicate that positive ZD summation results are not to be converted to printable numbers (overrides ZDPRINT).
- HILEVEL=YES on the MODS statement enables you to indicate that the E15 and E35 routines are to be treated as COBOL exits.
- DEBUG options BUFFERS=ANY and BUFFERS=BELOW will now be recognized but not used.

DFSORT will now ignore any DD statements not needed for the application (for example, a SORTIN DD statement will be ignored for a merge application).

For unsuccessful completion due to an unsupported operating system, DFSORT, ICEGENER, and ICETOOL will now pass back a return code of 24 to the operating system or invoking program.

The installation initialization exit, ICEIEXIT, enables you to specify the maximum buffer space DFSORT can use for each OUTFIL data set.

The installation termination exit, ICETEXIT, contains additional fields such as a flag to indicate that OUTFIL processing was used.

For INREC and OUTREC:

- The upper limit for columns and the end of fields has been raised from 32000 to 32752.
- 1: before the RDW field of variable-length records will be accepted and ignored.

For INCLUDE and OMIT, COND=ALL, COND=(ALL), COND=NONE, and COND=(NONE) enable you to include or omit all records.

The L2 value from the RECORD statement will be used if the L1 value is not specified when an E15 or E32 user exit passes all of the input records.

When input is a VSAM data set and output is a non-VSAM data set with RECFM not specified, DFSORT will now set the output RECFM as blocked rather than unblocked, when doing so will allow the use of the system-determined optimum block size for output.

New Programming Support for Release 12 (PTFs)

ICEGENER, copy, and Blockset sort and merge can now be used when a tape output data set is specified with DISP=MOD or DISP=OLD, without specifying the RECFM, LRECL, or BLKSIZE in the DD statement.

Sequential striping is supported for input and output data sets.

Compression is supported for input and output data sets.

BatchPipes/MVS input and output pipes are supported.

New Device Support for Release 12 (PTFs)

Four-digit device numbers are supported.

The IBM 3390-9 DASD is supported for input, output, and work data sets, although it is not recommended for work data sets for performance reasons.

The IBM RAMAC Array DASD and RAMAC Array Subsystem are supported for input, output, and work data sets.

The IBM 3990 Model 6 control unit is supported.

The IBM cached 9343 control unit models are supported.

Index

A

ABCODE parameter 67
ABEND parameter 76
accessibility 9
allocating ISPF data sets with LIBDEF 39
allocating storage
 temporary libraries 12
alternate collating sequence 67, 70
alternate message data set 92
ALTSEQ parameter 67, 70
applying SMP/E usermods 53
ARESALL parameter 22, 28, 69
ARESINV parameter 22, 28, 69

B

BCOPY 33
block size, system determined 100
buffer space, OUTFIL data set 93

C

cache fast write 69
century window 113
CFW parameter 69
CHALT parameter 70
CHECK parameter 70
checking record count 70
checkpoint/restart 86
CINV parameter 70
CLISTs, to allocate DFSORT Panels 38
COBEXIT parameter 70
COBOL
 invoking DFSORT 97
 user exits 70
collating sequence
 EBCDIC 67
 equal records 75
 modifying 67, 70
control statements, printing modified 88
converting zoned-decimal (ZD) fields 114
cultural environment
 considerations 49
 storage considerations 90
 using locales 89

D

data compaction percentage 84
Data Facility Sort (DFSORT)
 verifying installation of 12
data set
 block size, system determined 100
 ISPF, allocating with LIBDEF 39
 maximum buffer space, OUTFIL 93
dataspace sorting 72
dataspace sorting, changing use with ICEIEXIT 127

day parameter 64
ddname for message data set 92
default primary space allocation 74
defaults 48, 114, 118
DEFAULTS operator
 listing installation defaults with 119
deleting private libraries 16
description of ICEMAC parameters 118
devices required for storage 6
DFSMSrmm 159
DFSORT home page 3
DFSORT Panels
 distribution libraries 36
 target libraries 36
 testing 40
 verifying installation 40
DFSORT program messages 42
DIAGSIM parameter 71
distribution tape
 contents 3
 examples for storage administrators 3
DSA parameter 71
DSPSIZE parameter 72
DYNALLOC parameter 73, 116
DYNALOC parameter 73
dynamic allocation
 DYNALOC parameter 73
 DYNAUTO parameter 73
 example, ICEMAC macro 116
Dynamic Storage Adjustment (DSA) 71
DYNAUTO parameter 73
DYNSPC parameter 74

E

EBCDIC collating sequence, modifying 67, 70
EFS parameter 75
EFS program 50
ENABLE parameter 64
environment installation modules 55
equally collating records 75
EQUALS parameter 75
ERET parameter 76
error messages 177
ESTAE parameter 76
examples for storage administrators 3
EXITCK parameter 77
exits, user
 COBOL 70
 E15 70
 E35 70
expanded storage
 required 18
 tailoring 28, 57
EXPMAX parameter 77
EXPOLD parameter 78
EXPRES parameter 79

Extended Function Support (EFS) program
 invoking 50
 parameter 75

F

format
 error messages 177
 ICEMAC macro 57
FSZEST parameter 81

G

GCOPY1 33
GCOPY2 33
GEN 33
GENER parameter 81
GNPAD parameter 81
GNTRUNC parameter 82

H

Hipersorting
 changing use of Hiperspace with ICEIEXIT 127
 ICEMAC parameters 57
 maximum expanded storage 77
 maximum old expanded storage 78
 minimum expanded storage 79
 SMF record, statistics 229
 statistics, ICETEXIT 143
HIPRMAX parameter 83
home page (web) 3

I

ICEGENER
 discontinuing use 33
 GENER parameter 81
 verifying correct installation 32
ICEIEXIT
 general register 1 129
 IEXIT parameter 86
 installing 128
 interface points 129
 overview of installation control 127
 parameter list for installation-time options 129
 parameter list for run-time options 131
 requirements 128
 sample routine 133
 using 50
 writing 128
ICEMAC macro
 coding 57
 examples
 batch JCL direct invocation defaults 115
 batch program invocation defaults 115
 time-of-day defaults 118
 TSO direct invocation defaults 116
 TSO program invocation defaults 117
 parameters and defaults 118

ICEMAC macro (*continued*)
 tailoring expanded storage 57
 tailoring main storage 19
 tailoring storage 57
ICEMSGS macro 165
ICETEXIT
 designing 144
 example, executable 151
 example, nonexecutable 146
 installing 144
 mapping statistics 145
 parameters 213
 statistical data collection 137
 TEXT parameter 106
 using 50
ICETOOL
 examples for storage administrators 3
 verifying installation
 in temporary libraries 14
 with sample job, ICETOOLJ 12
ICETPEX
 Information passed 159
 Installing 159
 Sample 161
 Writing 159
identically collating records 75
IDRCPT parameter 84
IEBGENER, using ICEGENER as substitute for 30, 81
IEXIT parameter 86
IGNCKPT parameter 86
informative messages 184
initialization exit (ICEIEXIT) 50
installation
 defaults
 ICEMAC parameters 118
 listing 119
 modifying 48
 temporary libraries 9
 verifying 30
 ICECOBJ 16
 ICEJCLJ and ICEINVJ 13
 ICETOOL 14
 Locale Processing 15
installation-time options
 parameters changed by ICEIEXIT 129
installation-written exits
 ICEIEXIT 50
 ICETEXIT 50
Internet 3
invoking DFSORT 62
IOMAXBF parameter 87
ISPF Panels
 See DFSORT Panels 35

L

LIBDEF 39
libraries
 deleting 16
 required 7
link-pack area (LPA) 18

- LIST parameter 88
- listing installation defaults 119
- LISTX parameter 88
- LOCALE parameter 89
- locale processing
 - considerations for installation-wide default settings 90
 - environment considerations 49
 - LOCALE parameter 89
 - storage considerations 90
- Locale Processing
 - verifying installation in temporary libraries 15

M

- machine requirements 6
- main storage
 - allocating 21
 - considerations 22
 - factors affecting performance 17
 - limiting 21
 - reserving 22
 - tailoring 18
 - varying 17
- maximum buffer space, OUTFIL data sets 93
- MAXLIM parameter 21, 25, 90
- message data set
 - LIST parameter 88
 - LISTX parameter 88
 - MSGDDN parameter 92
 - MSGPRT parameter 92
 - NOMSGDD parameter 92
- message module error messages 182
- message texts
 - modifying 51
 - translating 51
- messages
 - error 177
 - explanation 177
 - format 177
 - informative 184
 - message module error 182
- migrating from other sort products 56
- MINLIM parameter 21, 91
- MNOTE messages 184
- modified control statements, printing 88
- modifying
 - installation defaults 47
 - message texts 51
- MSGCON parameter 91
- MSGDDN parameter 92
- MSGPRT parameter 92
- multiple releases of DFSORT
 - running 9

N

- NOMSGDD parameter 92

O

- ODMAXBF parameter 93
- operating system
 - conventions 5
 - requirements 5
 - standards 5
- OS/VS COBOL
 - invoking DFSORT 97
 - user exits 70
- OUTFIL processing
 - changing maximum buffer space with ICEIEXIT 127
 - examples for storage administrators 3
 - maximum buffer space 93
- OUTREL parameter 94
- OUTSEC parameter 94
- OVERRGN parameter 21, 28, 29, 94
- OVFLO parameter 95

P

- PAD parameter 95
- panels
 - selecting distribution and target libraries 36
 - verifying installation 40
- PARMDDN parameter 96
- performance
 - factors affecting, main storage 17
 - performance considerations 17
 - PL/I 97
 - prerequisite products 5, 6
 - previous release of DFSORT
 - replacing 10
 - primary space allocation, default 74
 - printing all control statements and messages 71
 - printing modified control statements 88
- procedures
 - modifying 47
 - SORT standard procedure 47
 - SORTD standard procedure 47
- processor time 104

R

- receiving SMP/E usermods 53
- record count 70
- reinstalling DFSORT 30
- replacing previous release of DFSORT 10
- require
 - machine 6
- required
 - expanded storage 18
 - libraries 7
 - main storage 17
- RESALL parameter 22, 26, 96
- RESET parameter 97
- RESINV parameter 22, 27, 97
- return codes, ERET parameter 76
- run time
 - changing options 127
 - examining information 128

- run time (*continued*)
 - options, changed by ICEIEXIT 131
- running DFSORT nonresident 9
- running DFSORT resident 8, 9
- running multiple releases of DFSORT 9
- running resident
 - required storage 18

S

- sample jobs
 - ICECOBJ 12
 - ICECSRTJ 12
 - ICEINVJ 12
 - ICEJCLJ 12
 - ICETOOLJ 12
- SDB parameter 98
- SDBMSG parameter 100
- simulating SORTDIAG DD statement 71
- SIZE parameter 19, 21, 22, 101
- SMF parameter 102
- SMF record
 - collecting data from 137
 - format 197
 - mapping 139
 - type-16 139, 197
- SMF statistics for ICETEXIT 213
- SMF type-16 record 139, 197
- SMF=FULL 138
- SMF=NO 138
- SMF=SHORT 138
- SMP/E usermod
 - applying 53
 - receiving 53
- software requirements 5, 6
- SOLRF parameter 102
- sorting instructions 17
- SORTLIB parameter 103
- SPANINC parameter 104
- specifying maximum storage 71
- statistical data collection
 - overview 137
 - using ICETEXIT 137
 - using SMF records 137
- STIMER parameter 104
- storage
 - amount required for running resident 18
 - decreasing main 17
 - devices 6
 - expanded 18, 28, 57
 - for temporary private libraries 12
 - modifying with ICEIEXIT 127
 - required 16, 18
 - required, main 16
 - tailoring expanded
 - with ICEMAC 57
 - tailoring main 19
 - with ICEMAC 57
 - varying main 17
- storage administrator examples 3

- SVC
 - parameter 104
 - renaming 10
- SVC installation
 - verifying 16
- SYSDA default 73
- system determined block size 100
- system management facilities (SMF) record 137
- SZERO parameter 105

T

- tailoring main storage 19
- Tape processing 159
- target libraries 9
- termination exit (ICETEXIT) 50
- testing DFSORT 9
- TEXT parameter 106
- time data 104
- time-of-day installation modules 55
- TMAXLIM parameter 21, 23, 106
- TRUNC parameter 106
- TSO logon procedure 38
- TSO messages 42

U

- user exits
 - COBOL 70
 - E15 70
 - E35 70

V

- VERIFY parameter 107
- verifying installation 30
 - DFSORT Panels 40
 - ICECOBJ 16
 - ICEINVJ 13
 - ICEJCLJ 13
 - ICETOOL 14
 - Locale Processing 15
- verifying SVC installation 16
- verifying that DFSORT was installed correctly 12
- VIO parameter 107
- virtual allocation 107
- VLLONG parameter 108
- VLSCMP parameter 109
- VLSHRT parameter 110
- VS COBOL II
 - invoking DFSORT 97
 - user exits 70
- VSAM buffers, specifying the number of 111
- VSAMBSP parameter 111
- VSAMEMT parameter 111
- VSAMIO parameter 111

W

Web 3

web site 3

work data sets

choice of 6

dedicated 47

dynamic allocation 73

releasing unused space 112

secondary allocation 112

virtual allocation 107

WRKREL parameter 112

WRKSEC parameter 112

Y

Y2PAST parameter 113

Year 2000 113

Z

ZDPRINT parameter 114

Readers' Comments — We'd Like to Hear from You

DFSORT
Installation and Customization
Release 14

Publication No. SC33-4034-20

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Complete	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Easy to find	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Easy to understand	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Well organized	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Applicable to your tasks	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Please tell us how we can improve this book:

Thank you for your responses. May we contact you? Yes No

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

Name

Address

Company or Organization

Phone No.



Fold and Tape

Please do not staple

Fold and Tape



NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation
RCF Processing Department
G26/M86 050
5600 Cottle Road
SAN JOSE, CA 95193-0001



Fold and Tape

Please do not staple

Fold and Tape



Program Number: 5740-SM1



Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

SC33-4034-20



Spine information:



DFSORT

DFSORT Installation and Customization R14

Release 14