z/OS



DFSORT Messages, Codes and Diagnosis Guide

SC26-7525-02

z/OS



DFSORT Messages, Codes and Diagnosis Guide

SC26-7525-02

Note
Before using this information and the product it supports, be sure to read the general information under "Notices" on page 133.

Third Edition, September 2006

This edition applies to Version 1 Release 8 of z/OS™ (5694–A01), Version 1 Release 8 of z/OS.e (5655-G52) and to all subsequent releases and modifications until otherwise indicated in new editions.

This edition replaces SC26-7525-01.

IBM welcomes your comments. A form for readers' comments may be provided at the back of this publication, or you may address your comments to the following address:
- International Business Machines Corporation
- Department 55JA, Mail Station P384
- 2455 South Road
- Poughkeepsie, NY 12601-5400
- United States of America
-
- FAX (United States & Canada): 1+845+432-9405
- FAX (Other Countries):
- Your International Access Code +1+845+432-9405
-
- IBMLink™ (United States customers only): IBMUSM10(MHVRCFS)
- Internet e-mail: mhvrcfs@us.ibm.com
- World Wide Web: http://www.ibm.com/servers/eserver/zseries/zos/webqs.html

If you would like a reply, be sure to include your name, address, telephone number, or FAX number.

Make sure to include the following in your comment or note:
- Title and order number of this document
- Page number or topic related to your comment

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 1995, 2006. All rights reserved.
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Figures v

Tables vii

About this document ix
How to use this document ix
Required product knowledge ix
Referenced documents x
Accessing z/OS DFSORT documents on the Internet x
Using LookAt to look up message explanations . . xi
Accessing z/OS licensed documents on the Internet xi

Summary of changes xiii
| Summary of Changes for SC26-7525-02 z/OS
| Version 1 Release 8 (PTFs - April, 2006) xiii
| New Information xiii
| Operational Changes that may Require User
| Action xvii
Summary of Changes for SC26-7525-01 z/OS
Version 1 Release 6 (PTFs - December, 2004) . . . xviii
New Information xviii
Operational Changes that may Require User
Action. xxii
Summary of Changes for SC26-7525-00 z/OS
Version 1 Release 5 xxiii
New Information xxiii
Summary of Changes for SC26-7050-05 DFSORT
Release 14 (PTFs - February, 2003) xxiv
New Information xxiv
Summary of Changes for SC26-7050-05 DFSORT
Release 14 (PTFs - March, 2002) xxvi
New Information xxvi

Part 1. Eliminating Common Sources of Error 1

Chapter 1. Eliminating Common Sources of Error 3
How to Get Messages 3
How to Get a Dump. 3
Eliminating DFSORT Installation as a Source of Error 4
Eliminating Your Invoking Program as a Source of
Error 4
Eliminating Your User Exit Routines as a Source of
Error 4
Errors in Using Registers 4
Errors in Using Main Storage 5
Errors in Record Contents 5
Being Aware of Data Management Rules 5
Correcting Common Errors 6
Common System Abends Related to Storage. . . 6
Miscellaneous System Abends 7
User Abends at Run Time 9
User Abends from DFSORT Panels 9
Common Problems Reported by DFSORT Users 10

Part 2. Messages and Codes 13

Chapter 2. Message Considerations . . 15
Printing Messages and Control Statements 15
Printing EFS Messages and Control Statements . . 16
Writing Messages to the Master Console. 16
Finding DFSORT Control Statement Coding Errors 17

Chapter 3. Informational and Error Messages 19

Chapter 4. Diagnostic Messages. . . . 83

Chapter 5. Return Codes and User Abends 87
DFSORT Return Codes 87
ICETOOL Return Codes 87
ICEGENER Return Codes 88

Part 3. Diagnosis Guide 91

Chapter 6. Resolving Failures in the DFSORT Program 93
Developing Your Keyword String 94
Constructing Keyword Strings 96
Component Identification Keyword Procedure. . . 97
FMID Keyword Procedure 98
Type-of-Technique Keyword Procedure 98
Type-of-Application Keyword Procedure 99
Type-of-Failure Keyword Procedures 99
Wait and Loop Keywords Procedure 100
System Abend Keyword Procedure 101
User Abend Keyword Procedure 101
Message Keyword Procedure 102
Incorrect-Output Keyword Procedure 102
Performance Keyword Procedure. 103
Document Keyword Procedure 103
Module Keyword Procedure 103
Offset Keyword Procedure 105
Message Variable Keywords 106
ICE061A Variable Keywords 106
ICE117A Variable Keywords 107

Chapter 7. Resolving Problems When Using DFSORT Panels 109
Obtaining Diagnostic Information 109
Determining Where the Failure Occurred 110
Resolving Failures in DFSORT Panels 110
DFSORT Panels Abend Panel 110

© Copyright IBM Corp. 1995, 2006

TSO Abend Message 111
Resolving Failures in ISPF or ISMF 112
Resolving Failures in the DFSORT Program . . . 113
Foreground Execution Failures 113
Background Execution Failures 114
Recording Information 114
Inter-Module Traces 115
ISMF Logging with SORTREC 115
Using the Profile Application 115
Directing Abend Processing 116
Obtaining Diagnostic Messages 116

Chapter 8. Searching the Problem Reporting Databases 119

Chapter 9. Fixing or Bypassing the Problem. 121
Fixing the Problem 121
Bypassing the Problem 121

Chapter 10. Reporting a Problem . . . 123

Part 4. Appendixes 125

Appendix A. Valid Input/Output Options at User Exits—ICE044I. . . . 127

Appendix B. User Exit Return Codes—ICE071A 129

Appendix C. Accessibility 131
Using assistive technologies 131
Keyboard navigation of the user interface 131
z/OS information 131

Notices 133
Programming Interface Information 134
Trademarks 134

Figures

1. Keyword String Flowchart 95
2. Keyword String Syntax. 97
3. Finding the System Abend Code and the PSW in the SYSUDUMP 100
4. Finding the Wait State Bit in the PSW 101
5. Finding the Abend Address 104
6. Finding the Module Name 105
7. Finding the Offset 106
8. DFSORT Panels Abend Panels Indicate the Problem Is in DFSORT Panels 111
9. TSO Messages Also Indicate That the Problem Is in DFSORT Panels 111
10. Two Symptom Dumps from TSO Appear on Your Screen 112
11. ISPF Abend Panels Indicate the Problem Is in ISPF. 113
12. ISMF Abend Panels Indicate the Problem Is in ISMF 113

© Copyright IBM Corp. 1995, 2006

Tables

1. Referenced documents x
2. Getting a Dump 3
3. Common System Abends Related to Storage 6
4. Miscellaneous System Abends 7
5. User Abends from DFSORT Panels 10
6. Program Failure Resolution 93
7. Types of DFSORT Failures 99
8. Abend Cause Location Procedure 110
9. DFSORT Reason Codes 115
10. Steps to Search the IBM Software Support Facility 119
11. User Exit Return Codes - ICE071A 129

© Copyright IBM Corp. 1995, 2006

About this document

This document is intended to help you:

- Eliminate common sources of errors. Part 1, "Eliminating Common Sources of Error," on page 1 explains how to get messages and dumps and how to identify and eliminate various problems.
- Interpret the informational and error messages issued by DFSORT™. Part 2, "Messages and Codes," on page 13 contains the messages returned from the program, explains their meanings, and suggests appropriate responses.
- Diagnose a program failure that you suspect is caused by DFSORT. Part 3, "Diagnosis Guide," on page 91 is a step-by-step guide that will assist you in searching for a solution and, if necessary, reporting the problem to the IBM Support Center.

For information about the accessibility features of z/OS, for users who have a physical disability, see Appendix C, "Accessibility," on page 131.

How to use this document

The Diagnosis Guide is designed to help you solve problems you encounter when running DFSORT. To use the information in this part most efficiently, first determine whether the problem is caused by:

A user error
: Read Chapter 1, "Eliminating Common Sources of Error," on page 3, for solutions to some of the most common errors users encounter when running DFSORT.

A program error
: Use the numbered steps listed in Chapter 6, "Resolving Failures in the DFSORT Program," on page 93, to describe the problem and to find out if corrections for it are already available.

A DFSORT Panels error
: A failure while working in DFSORT Panels might be due to an abend in the DFSORT Panels program, ISPF, or ISMF. Begin with Chapter 7, "Resolving Problems When Using DFSORT Panels," on page 109. If the problem actually is a DFSORT program failure, the steps in that section will direct you back to this page.

Required product knowledge

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

- Job control language (JCL)
- DFSORT and ICETOOL control statement syntax
- Data management
- Tape and disk hardware
- IBM support and how it is structured
- Diagnostic techniques

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

© Copyright IBM Corp. 1995, 2006

Document Title	Document Order Number
z/OS DFSORT Application Programming Guide	SC26-7523
z/OS MVS System Messages, Vol 7 (IEB-IEE)	SA22-7637
z/OS MVS System Messages, Vol 8 (IEF-IGD)	SA22-7638
z/OS MVS System Codes	SA22-7626

Referenced documents

This document refers to the following documents:

Table 1. Referenced documents

Document title	Order number
SMP/E Reference	SA22-7772
z/OS and z/OS.e Planning for Installation	GA22-7504
z/OS DFSMSdfp Diagnosis	GY27-7618
z/OS DFSMS Macro Instructions for Data Sets	SC26-7408
z/OS DFSMS Using Data Sets	SC26-7410
z/OS MVS JCL Reference	SA22-7597
z/OS MVS Programming: Authorized Assembler Services Guide	SA22-7608
z/OS MVS System Management Facilities (SMF)	SA22-7630
z/OS DCE Application Development Reference	SC24-5908
z/OS TSO/E Command Reference	SA22-7782

z/OS DFSORT Messages, Codes and Diagnosis Guide is a part of a more extensive DFSORT library. These documents can help you work with DFSORT more effectively.

Task	Publication	Order Number
Planning For and Customizing DFSORT	*z/OS DFSORT Installation and Customization*	SC26-7524
Learning to Use DFSORT	*z/OS DFSORT: Getting Started*	SC26-7527
Learning to Use DFSORT Panels	*z/OS DFSORT Panels Guide*	GC26-7037
Application Programming	*z/OS DFSORT Application Programming Guide*	SC26-7523
Tuning DFSORT	*z/OS DFSORT Tuning Guide*	SC26-7526

Accessing z/OS DFSORT documents on the Internet

In addition to making softcopy documents available on CD-ROM, IBM provides access to unlicensed z/OS softcopy documents on the Internet. To find z/OS documents on the Internet, first go to the z/OS home page: http://www.ibm.com/servers/eserver/zseries/zos

From this Web site, you can link directly to the z/OS softcopy documents by selecting the Library icon. You can also link to the IBM Publications Center to order printed documentation.

Using LookAt to look up message explanations

LookAt is an online facility that lets you look up explanations for most of the IBM® messages you encounter, as well as for some system abends and codes. Using LookAt to find information is faster than a conventional search because in most cases LookAt goes directly to the message explanation.

You can use LookAt from the following locations to find IBM message explanations for z/OS® elements and features, z/VM®, and VSE:

- The Internet. You can access IBM message explanations directly from the LookAt Web site at http://www.ibm.com/eserver/zseries/zos/bkserv/lookat/.
- Your z/OS TSO/E host system. You can install code on your z/OS or z/OS.e systems to access IBM message explanations, using LookAt from a TSO/E command line (for example, TSO/E prompt, ISPF, or z/OS UNIX® System Services running OMVS).
- Your Windows® workstation. You can install code to access IBM message explanations on the *z/OS Collection* (SK3T-4269), using LookAt from a Windows DOS command line.
- Your wireless handheld device. You can use the LookAt Mobile Edition with a handheld device that has wireless access and an Internet browser (for example, Internet Explorer for Pocket PCs, Blazer, or Eudora for Palm OS, or Opera for Linux handheld devices). Link to the LookAt Mobile Edition from the LookAt Web site.

You can obtain code to install LookAt on your host system or Windows workstation from a disk on your *z/OS Collection* (SK3T-4269), or from the LookAt Web site (click **Download**, and select the platform, release, collection, and location that suit your needs). More information is available in the LOOKAT.ME files available during the download process.

Accessing z/OS licensed documents on the Internet

z/OS licensed documentation is available on the Internet in PDF format at the IBM Resource Link™ Web site at: http://www.ibm.com/servers/resourcelink

Licensed documents are available only to customers with a z/OS license. Access to these documents requires an IBM Resource Link user ID and password, and a key code. With your z/OS order you received a Memo to Licensees, (GI10-0671), that includes this key code.

To obtain your IBM Resource Link user ID and password, log on to: http://www.ibm.com/servers/resourcelink

To register for access to the z/OS licensed documents:

1. Sign in to Resource Link using your Resource Link user ID and password.
2. Select **User Profiles** located on the left-hand navigation bar.

Note: You cannot access the z/OS licensed documents unless you have registered for access to them and received an e-mail confirmation informing you that your request has been processed.

Printed licensed documents are not available from IBM.

You can use the PDF format on either **z/OS Licensed Product Library CD-ROM** or IBM Resource Link to print licensed documents.

Summary of changes

This document contains terminology, maintenance, and editorial changes. Technical changes or additions to the text and illustrations are indicated by a vertical line to the left of each change.

You might notice changes in the style and structure of some content in this document—for example, more specific headings for notes, such as Tip and Requirement. The changes are ongoing improvements to the consistency and retrievability of information in DFSORT documents.

Summary of Changes for SC26-7525-02 z/OS Version 1 Release 8 (PTFs - April, 2006)

This document contains information that was previously presented in *z/OS DFSORT Messages, Codes and Diagnosis Guide* , SC26-7525-01.

The following sections summarize the changes to that information.

New Information

This edition includes the following new enhancements:

INCLUDE and OMIT Enhancements

COND now allows you to compare date fields in various formats to past and future dates (relative to the date of your DFSORT run) using new DATEn+r, DATEn-r, DATEn(c)+r, DATEn(c)-r, DATEnP+r, DATEnP-r, Y'DATEn'+r and Y'DATEn'-r constants. &DATEn+r, &DATEn-r , &DATEn(c)+r, &DATEn(c)-r, &DATEnP+r and &DATEnP-r can be used as aliases for DATEn+r, DATEn-r, DATEn(c)+r, DATEn(c)-r, DATEnP+r and DATEnP-r, respectively.

COND now allows you to test a field for numerics (field,EQ,NUM) or non-numerics (field,NE,NUM) in character (FS), zoned decimal (ZD) or packed decimal (PD) format.

COND now allows you to use &DATEn, &DATEn(c) and &DATEnP as aliases for DATEn, DATEn(c) and DATEnP, respectively.

INREC and OUTREC Enhancements

PARSE and IFTHEN PARSE are new options that allow you to extract variable position/length fields into fixed-length parsed fields defined as %nn fields. PARSE gives you powerful new capabilities for handling variable fields such as delimited fields, comma separated values (CSV), tab separated values, blank separated values, keyword separated fields, null-terminated strings, and many other types.

You can use various PARSE options to define the rules for extracting variable fields into up to one hundred %nn fixed-length parsed fields (%00-%99), and then use these %nn fields where you can use p,m fields in BUILD, OVERLAY, IFTHEN BUILD, IFTHEN OVERLAY and FIELDS. You can edit, convert, justify, squeeze, translate, and do arithmetic with %nn fields.

BUILD, OVERLAY, IFTHEN BUILD, IFTHEN OVERLAY and FIELDS now allow you to use a new JFY option to left-justify or right-justify the data in a field. For a

© Copyright IBM Corp. 1995, 2006

left-justified field, leading blanks are removed and the characters from the first nonblank to the last nonblank are shifted left, with blanks inserted on the right if needed. For a right-justified field, trailing blanks are removed and the characters from the last nonblank to the first nonblank are shifted right, with blanks inserted on the left if needed.

Optionally for JFY, specific leading and trailing characters can be changed to blanks before justification begins, a leading string can be inserted, a trailing string can be inserted, and the output length can be changed.

BUILD, OVERLAY, IFTHEN BUILD, IFTHEN OVERLAY and FIELDS now allow you to use a new SQZ option to left-squeeze or right-squeeze the data in a field. For a left-squeezed field, all blanks are removed and the characters from the first nonblank to the last nonblank are shifted left, with blanks inserted on the right if needed. For a right-squeezed field, all blanks are removed and the characters from the last nonblank to the first nonblank are shifted right, with blanks inserted on the left if needed.

Optionally for SQZ, specific characters can be changed to blanks before squeezing begins, a leading string can be inserted, a trailing string can be inserted, a string can be inserted wherever a group of blanks is removed between the first nonblank and the last nonblank, blanks can be kept as is between paired apostrophes or paired quotes, and the output length can be changed.

BUILD, OVERLAY, IFTHEN BUILD, IFTHEN OVERLAY and FIELDS now allow you to insert past and future dates (relative to the date of your DFSORT run) into your records in various forms using new DATEn+r, DATEn-r, DATEn(c)+r, DATEn(c)-r, DATEnP+r and DATEnP-r constants. &DATEn+r, &DATEn-r, &DATEn(c)+r, &DATEn(c)-r, &DATEnP+r and &DATEnP-r can be used as aliases for DATEn+r, DATEn-r, DATEn(c)+r, DATEn(c)-r, DATEnP+r and DATEnP-r, respectively.

BUILD, OVERLAY, IFTHEN BUILD, IFTHEN OVERLAY and FIELDS now allow you to use FL format to convert 4-byte or 8-byte hexadecimal floating-point values to integer values.

BUILD, OVERLAY, IFTHEN BUILD, IFTHEN OVERLAY and FIELDS now allow you to use new TO=PDF and TO=PDC options to convert numeric values to PD values with F or C for the positive sign, respectively. The TO=PDC option is equivalent to the existing TO=PD option.

BUILD, OVERLAY, IFTHEN BUILD, IFTHEN OVERLAY and FIELDS now allow you to use &DATEn, &DATEn(c), &DATEnP, &YDDD=(abc), &YDDDNS=(ab), &TIMEn, &TIMEn(c) and &TIMEnP as aliases for DATEn, DATEn(c), DATEnP, YDDD=(abc), YDDDNS=(ab), TIMEn, TIMEn(c) and TIMEnP, respectively.

IFTHEN WHEN now allows you to compare date fields in various formats to past and future dates (relative to the date of your DFSORT run) using new DATEn+r, DATEn-r, DATEn(c)+r, DATEn(c)-r, DATEnP+r, DATEnP-r, Y'DATEn'+r and Y'DATEn'-r constants. &DATEn+r, &DATEn-r, &DATEn(c)+r, &DATEn(c)-r, &DATEnP+r and &DATEnP-r can be used as aliases for DATEn+r, DATEn-r, DATEn(c)+r, DATEn(c)-r, DATEnP+r and DATEnP-r, respectively.

IFTHEN WHEN now allows you to test a field for numerics (field,EQ,NUM) or non-numerics (field,NE,NUM) in character (FS), zoned decimal (ZD) or packed decimal (PD) format.

IFTHEN WHEN now allows you to use &DATEn, &DATEn(c) and &DATEnP as aliases for DATEn, DATEn(c) and DATEnP, respectively.

OUTFIL Enhancements

BLKCCH1 is a new report option that allows you to avoid forcing a page eject at the start of the report header; the ANSI carriage control character of '1' (page eject) in the first line of the report header (HEADER1) is replaced with a blank.

BLKCCH2 is a new report option that allows you to avoid forcing a page eject at the start of the first page header; the ANSI carriage control character of '1' (page eject) in the first line of the first page header (HEADER2) is replaced with a blank.

BLKCCT1 is a new report option that allows you to avoid forcing a page eject at the start of the report trailer; the ANSI carriage control character of '1' (page eject) in the first line of the report trailer (TRAILER1) is replaced with a blank.

SPLIT1R is a new option that allows you to write contiguous groups of records in one rotation among multiple output data sets. A specified number of records is written to each output data set and extra records are written to the last output data set.

PARSE and IFTHEN PARSE are new options that allow you to extract variable position/length fields into fixed-length parsed fields defined as %nn fields. PARSE gives you powerful new capabilities for handling variable fields such as delimited fields, comma separated values (CSV), tab separated values, blank separated values, keyword separated fields, null-terminated strings, and many other types.

You can use various PARSE options to define the rules for extracting variable fields into up to one hundred %nn fixed-length parsed fields (%00-%99), and then use these %nn fields where you can use p,m fields in BUILD, OVERLAY, IFTHEN BUILD, IFTHEN OVERLAY and OUTREC. You can edit, convert, justify, squeeze, translate, and do arithmetic with %nn fields.

BUILD, OVERLAY, IFTHEN BUILD, IFTHEN OVERLAY and OUTREC now allow you to use a new JFY option to left-justify or right-justify the data in a field. For a left-justified field, leading blanks are removed and the characters from the first nonblank to the last nonblank are shifted left, with blanks inserted on the right if needed. For a right-justified field, trailing blanks are removed and the characters from the last nonblank to the first nonblank are shifted right, with blanks inserted on the left if needed.

Optionally for JFY, specific leading and trailing characters can be changed to blanks before justification begins, a leading string can be inserted, a trailing string can be inserted, and the output length can be changed.

BUILD, OVERLAY, IFTHEN BUILD, IFTHEN OVERLAY and OUTREC now allow you to use a new SQZ option to left-squeeze or right-squeeze the data in a field. For a left-squeezed field, all blanks are removed and the characters from the first nonblank to the last nonblank are shifted left, with blanks inserted on the right if needed. For a right-squeezed field, all blanks are removed and the characters from the last nonblank to the first nonblank are shifted right, with blanks inserted on the left if needed.

Optionally for SQZ, specific characters can be changed to blanks before squeezing begins, a leading string can be inserted, a trailing string can be inserted, a string can be inserted wherever a group of blanks is removed between the first nonblank

and the last nonblank, blanks can be kept as is between paired apostrophes or paired quotes, and the output length can be changed.

BUILD, OVERLAY, IFTHEN BUILD, IFTHEN OVERLAY and OUTREC now allow you to insert past and future dates (relative to the date of your DFSORT run) into your records in various forms using new DATEn+r, DATEn-r, DATEn(c)+r, DATEn(c)-r, DATEnP+r and DATEnP-r constants. &DATEn+r, &DATEn-r, &DATEn(c)+r, &DATEn(c)-r, &DATEnP+r and &DATEnP-r can be used as aliases for DATEn+r, DATEn-r, DATEn(c)+r, DATEn(c)-r, DATEnP+r and DATEnP-r, respectively.

TRAILERx, BUILD, OVERLAY, IFTHEN BUILD, IFTHEN OVERLAY and OUTREC now allow you to use FL format to convert 4-byte or 8-byte hexadecimal floating-point values to integer values.

TRAILERx, HEADERx, BUILD, OVERLAY, IFTHEN BUILD, IFTHEN OVERLAY and OUTREC now allow you to use new TO=PDF and TO=PDC options to convert numeric values to PD values with F or C for the positive sign, respectively. The TO=PDC option is equivalent to the existing TO=PD option.

BUILD, OVERLAY, IFTHEN BUILD, IFTHEN OVERLAY and OUTREC now allow you to use &DATEn, &DATEn(c), &DATEnP, &YDDD=(abc), &YDDDNS=(ab), &TIMEn, &TIMEn(c) and &TIMEnP as aliases for DATEn, DATEn(c), DATEnP, YDDD=(abc), YDDDNS=(ab), TIMEn, TIMEn(c) and TIMEnP, respectively.

INCLUDE, OMIT and IFTHEN WHEN now allow you to compare date fields in various formats to past and future dates (relative to the date of your DFSORT run) using new DATEn+r, DATEn-r, DATEn(c)+r, DATEn(c)-r, DATEnP+r, DATEnP-r, Y'DATEn'+r and Y'DATEn'-rconstants. &DATEn+r, &DATEn-r, &DATEn(c)+r, &DATEn(c)-r, &DATEnP+r and &DATEnP-r can be used as aliases for DATEn+r, DATEn-r, DATEn(c)+r, DATEn(c)-r, DATEnP+r and DATEnP-r, respectively.

INCLUDE, OMIT and IFTHEN WHEN now allow you to test a field for numerics (field,EQ,NUM) or non-numerics (field,NE,NUM) in character (FS), zoned decimal (ZD) or packed decimal (PD) format.

INCLUDE, OMIT and IFTHEN WHEN now allow you to use &DATEn, &DATEn(c) and &DATEnP as aliases for DATEn, DATEn(c) and DATEnP, respectively.

Symbol Enhancements

A symbol can now be used for a %nn parsed field. For example, if Account,%01 is defined in SYMNAMES, Account can be used for %01. A symbol for %nn can be used in DFSORT control statements where %nn can be used. A symbol for %nn results in substitution of %nn.

A symbol can now be used for an output column. For example, if Start_address,18 is defined in SYMNAMES, Start_address: can be used for 18:. symbol: can be used in DFSORT control statements where c: can be used. A symbol for p or p,m or p,m,f results in substitution of p: for symbol: (output column).

A symbol can now be used for a new system symbol string constant. symbol,S'string' can be used to define a string containing any combination of EBCDIC characters and system symbols you want to use to form a character string. For example, if whererun,S' &JOBNAME. on &SYSPLEX' is defined in SYMNAMES, whererun can be used for the resulting constant. You can use

dynamic system symbols such as &JOBNAME, &DAY, and so on, system-defined static system symbols such as &SYSNAME, &SYSPLEX, and so on, and installation-defined static system symbols specified by your installation in an IEASYM*xx* member of SYS1.PARMLIB.

A symbol for a system symbol string can be used in DFSORT and ICETOOL control statements where a symbol for a character string can be used. DFSORT will replace each system symbol in S'string' with its substitution text to create a character string in the format C'new_string'.

ICETOOL Enhancements

DISPLAY now allows you to use FL format to convert 4-byte or 8-byte hexadecimal floating-point values to integer values.

DISPLAY and OCCUR now allow you to use a new TBETWEEN(n) option to specify the number of blanks between title elements (title, page number, date, time).

SELECT and SPLICE now allow you to use an INREC statement to reformat your records before they are selected or spliced. All of the operands of the INREC statement (PARSE, BUILD, OVERLAY, IFTHEN, IFOUTLEN and FIELDS) are now available with SELECT and SPLICE.

SORT and MERGE Enhancements

The maximum length for a PD or ZD sort or merge field has been raised to 256.

SUM Enhancements

The maximum position for the end of a sum field has been raised to 32752.

Other Enhancements

DFSORT supports large physical sequential data sets for input, output and work data sets.

DSA can now be specified as a run-time option. This allows you to adjust the maximum amount of storage available to DFSORT for dynamic storage adjustment of individual Blockset sort applications when SIZE/MAINSIZE=MAX is in effect.

DFSORT now accepts and ignores zero values in the starting and ending address of the RECORD statement image in the 24-bit Parameter List. You can set these addresses to zero if you don't want to pass a control statement to DFSORT using the third and fourth words of the parameter list .

Operational Changes that may Require User Action

The following are operational changes that may require user action for existing DFSORT/ICETOOL applications that use certain functions as specified:

New reserved words for symbols

The following are new DFSORT/ICETOOL reserved words (uppercase only, as shown), which are no longer allowed as symbols: DATE1..., DATE2..., DATE3..., PDC and PDF.

If you used these words as symbols, you must change them to other words, such as lowercase or mixed case forms (for example, Date1p, date3(/) or pdc).

FL Conversion

DFSORT's INREC, OUTREC and OUTFIL statements, and ICETOOL's DISPLAY operator, can now convert FL (hexadecimal floating-point) values to integer values, providing you are running in z/Architecture mode. If you use FL in INREC, OUTREC, OUTFIL or DISPLAY when running in ESA/390 mode, FL will be recognized and the error messages issued may be different than those issued previously when FL was not allowed in INREC, OUTREC, OUTFIL or DISPLAY.

If you want to use FL in INREC, OUTREC, OUTFIL or DISPLAY, you must be running in z/Architecture mode.

OUTREC Statement with SELECT or SPLICE

DFSORT will now issue error message ICE652A and terminate if an OUTREC statement is used with ICETOOL's SELECT or SPLICE operator. This will make it easier to identify this error condition.

Note: If ABEND is in effect, DFSORT will ABEND with U0652, but will not issue ICE652A.

If you want to reformat your records after SELECT or SPLICE processing, use an OUTFIL statement rather than an OUTREC statement.

Summary of Changes for SC26-7525-01 z/OS Version 1 Release 6 (PTFs - December, 2004)

This document contains information that was previously presented in *z/OS DFSORT Messages, Codes and Diagnosis Guide* , SC26-7525-00.

The following sections summarize the changes to that information.

New Information

This edition includes the following new enhancements:

ICETOOL Enhancements

ICETOOL's DISPLAY, OCCUR, RANGE, SELECT, SPLICE, STATS, UNIQUE and VERIFY operators now allow you to use larger numeric values for ON and BREAK fields. PD, ZD and FS fields can now be up to 31 digits (or more in some cases). BI and FI fields can now be up to 8 bytes (or more in some cases).

ICETOOL's RANGE operator now allows you to use larger decimal values for the HIGHER(n), LOWER(n), EQUAL(n) and NOTEQUAL(n) options. These values can now be up to 31 digits.

ICETOOL's DISPLAY, OCCUR, RANGE, SELECT, SPLICE, STATS and UNIQUE operators now allow you to use new UFF (unsigned free form) and SFF (signed free form) formats for ON and BREAK fields. UFF extracts a positive numeric value from a free form field (for example, ' $1234.56' is treated as +123456). SFF extracts a positive or negative value from a free form field (for example, '(1,234.56)' is treated as -123456).

ICETOOL's DISPLAY and OCCUR operators now allow you to use new DC1-DC3 (TOD date), DE1-DE3 (ETOD date), TC1-TC4 (TOD time) and TE1-TE4 (ETOD time) formats for ON and BREAK fields. These new formats produce meaningful representations of TOD and ETOD date and time values.

ICETOOL's DISPLAY and OCCUR operators now allow you to specify multiline headings for the columns of your reports. You can specify one, two or three line headings with the HEADER option.

ICETOOL's DISPLAY and OCCUR operators now allow you to use new edit masks G1-G6 to display numeric values with 4 decimal digits in various ways.

ICETOOL's DISPLAY and OCCUR operators now allow you to use new YDDD(abc) and YDDDNS(abc) options to insert the year (yyyy) and day of the year (ddd) of your ICETOOL run into your titles in various forms.

ICETOOL's COUNT operator now allows you to use a new RC4 option to set RC=4 (instead of RC=12) or RC=0 based on the count of records in a data set.

ICETOOL's SPLICE operator now allows you to use a new KEEPBASE option to keep the base records as well as the spliced records.

ICETOOL's SPLICE operator now allows you to use new VLENMAX or VLENOVLY options to set the length of spliced records to the maximum of the base or overlay record length, or to the overlay record length.

OUTFIL Enhancements

OUTFIL now allows you to reformat records in one of the following three ways using unedited, edited, or converted input fields and a variety of constants:

- BUILD or OUTREC: The existing OUTREC parameter, or its new alias of BUILD, allows you to reformat each record by specifying all of its items one by one. BUILD or OUTREC gives you complete control over the items you want in your reformatted OUTFIL records and the order in which they appear. You can delete, rearrange and insert fields and constants.
- OVERLAY: The new OVERLAY parameter allows you to reformat each record by specifying just the items that overlay specific columns. Overlay lets you change specific existing columns without affecting the entire record.
- IFTHEN clauses: The new IFTHEN clauses allow you to reformat different records in different ways by specifying how BUILD or OVERLAY items are applied to records that meet given criteria. IFTHEN clauses let you use sophisticated conditional logic to choose how different record types are reformatted.

OUTFIL OUTREC, as well as BUILD, OVERLAY, and IFTHEN, now allows you to use larger numeric values for fields and decimal constants to be edited, converted or used in arithmetic expressions. PD, ZD and FS fields, and decimal constants, can now be up to 31 digits. BI and FI fields can now be up to 8 bytes.

OUTFIL OUTREC, as well as BUILD, OVERLAY, and IFTHEN, now allows you to use new UFF (unsigned free form) and SFF (signed free form) formats for fields to be edited, converted or used in arithmetic expressions.

OUTFIL OUTREC, as well as BUILD, OVERLAY, and IFTHEN, now allows you to use new DC1-DC3 (TOD date), DE1-DE3 (ETOD date), TC1-TC4 (TOD time) and TE1-TE4 (ETOD time) formats for fields to be edited, converted or used in arithmetic expressions. These new formats produce meaningful representations of TOD and ETOD date and time values.

OUTFIL OUTREC, as well as BUILD, OVERLAY, and IFTHEN, now allows you to use a set field in the CHANGE option (for example, 1,2,CHANGE=(4,C'FY',C'0001',C'VV',21,4)).

OUTFIL OUTREC, as well as BUILD, OVERLAY and IFTHEN, now allows you to restart the sequence number when the binary value of a specified field changes (for example, SEQNUM,5,ZD,RESTART=(11,4)).

OUTFIL OUTREC, as well as BUILD, OVERLAY, and IFTHEN, now allows you to use DATE, DATE=(abcd), DATENS=(abc), YDDD=(abc), YDDDNS=(ab), TIME, TIME=(abc) and TIMENS=(ab) options to insert the date and time of your DFSORT run into your records in various forms.

OUTFIL OUTREC, as well as BUILD, OVERLAY, and IFTHEN, now allows you to use new TO=ZDC and TO=ZDF options to convert numeric values to ZD values with C or F for the positive sign, respectively. The TO=ZDF option is equivalent to the existing TO=ZD option.

OUTFIL INCLUDE and OMIT now allow you to use larger FS values for compare fields. These values can now be up to 32 digits.

OUTFIL INCLUDE and OMIT now allow you to use larger decimal constants for comparison to BI and FI fields. Decimal constants can now be up to +18446744073709551615 for comparison to BI fields. Decimal constants can now be between -9223372036854775808 and +9223372036854775807 for comparison to FI fields.

OUTFIL INCLUDE and OMIT now allow you to use new UFF (unsigned free form) and SFF (signed free form) formats for compare fields. A UFF or SFF field can be compared to a UFF, SFF, FS, CSL or CST field or to a decimal constant.

OUTFIL TRAILERx now allows you to use larger numeric values for statistical fields (total, maximum, minimum, average). PD, ZD and FS fields, and decimal constants, can now be up to 31 digits. BI and FI fields can now be up to 8 bytes.

OUTFIL TRAILERx now allows you to use new UFF (unsigned free form) and SFF (signed free form) formats for statistical fields (total, maximum,minimum, average).

OUTFIL TRAILERx now allows you to use TO=fo and fo (to) options to convert statistical fields (total, maximum, minimum, average) and counts to BI, FI, PD, ZD, ZDC, ZDF or FS output values.

OUTFIL TRAILERx now allows you to use COUNT+n=(edit), COUNT+n=(to), COUNT-n=(edit) and COUNT-n=(to) to add or subtract n from a count to be edited or converted (for example, COUNT+1=(TO=ZD)).

OUTFIL HEADERx and TRAILERx now allow you to insert hexadecimal strings (X'yy...yy' or nX'yy...yy') in your headers and trailers.

OUTFIL HEADERx and TRAILERx now allow you to use new YDDD=(abc) and YDDDNS=(ab) options to insert the year (yyyy) and day of the year (ddd) of your DFSORT run in your headers and trailers.

OUTFIL HEADERx and TRAILERx now allow you to use PAGE=(edit) and PAGE=(to) to edit or convert the page number (for example, PAGE=(M11,LENGTH=3)).

INREC and OUTREC Enhancements

INREC and OUTREC now allow you to reformat records in one of the following three ways using unedited, edited, or converted input fields and a variety of constants:

- BUILD or FIELDS: The existing FIELDS parameter, or its new alias of BUILD, allows you to reformat each record by specifying all of its items one by one. BUILD or FIELDS gives you complete control over the items you want in your reformatted INREC or OUTREC records and the order in which they appear. You can delete, rearrange and insert fields and constants.
- OVERLAY: The new OVERLAY parameter allows you to reformat each record by specifying just the items that overlay specific columns. Overlay lets you change specific existing columns without affecting the entire record.
- IFTHEN clauses: The new IFTHEN clauses allow you to reformat different records in different ways by specifying how BUILD or OVERLAY items are applied to records that meet given criteria. IFTHEN clauses let you use sophisticated conditional logic to choose how different record types are reformatted.

INREC and OUTREC now allow you to use new UFF (unsigned free form) and SFF (signed free form) formats for fields to be edited, converted or used in arithmetic expressions.

INREC and OUTREC now allow you to use new DC1-DC3 (TOD date), DE1-DE3 (ETOD date), TC1-TC4 (TOD time) and TE1-TE4 (ETOD time) formats for fields to be edited, converted or used in arithmetic expressions. These new formats produce meaningful representations of TOD and ETOD date and time values.

INREC and OUTREC now allow you to use a set field in the CHANGE option.

INREC and OUTREC now allow you to restart the sequence number when the binary value of a specified field changes.

INREC and OUTREC now allow you to use new TO=ZDC and TO=ZDF options to convert numeric values to ZD values with C or F for the positive sign, respectively. The TO=ZDF option is equivalent to the existing TO=ZD option.

INREC and OUTREC now allow you to use new DATE, DATE=(abcd), DATENS=(abc), YDDD=(abc), YDDDNS=(ab), TIME, TIME=(abc) and TIMENS=(ab) options to insert the date and time of your DFSORT run into your records in various forms.

SORT and MERGE Enhancements

SORT and MERGE now allow you to use larger FS values for control fields. These values can now be up to 32 digits.

SORT and MERGE now allow you to use new UFF (unsigned free form) and SFF (signed free form) formats for control fields.

SORT and MERGE now allow you to use other characters besides '+' (plus) as a positive sign in control field values for CSL, CST, ASL, and AST formats. A '–' (minus) sign is treated as negative and any other sign (for example, blank) is treated as positive.

INCLUDE and OMIT Enhancements

INCLUDE and OMIT now allow you to use larger FS values for compare fields. These values can now be up to 32 digits.

INCLUDE and OMIT now allow you to use larger decimal constants for comparison to BI and FI fields. Decimal constants can now be up to +18446744073709551615 for comparison to BI fields. Decimal constants can now be between -9223372036854775808 and +9223372036854775807 for comparison to FI fields.

INCLUDE and OMIT now allow you to use new UFF (unsigned free form) and SFF (signed free form) formats for compare fields. A UFF or SFF field can be compared to a UFF, SFF, FS, CSL or CST field or to a decimal constant.

SUM Enhancement

SUM now allows you to use larger ZD values for sum fields. These values can now be up to 31 digits.

Other Enhancements

For sort and copy applications with concatenated variable-length input data sets for SORTIN, DFSORT now uses the largest LRECL it finds in the concatenation.

DFSORT now allows control statements in SYSIN and SORTCNTL to contain labels up to 70 characters, and allows any character in the label. DFSORT now ignores statements with a label followed only by blanks.

Operational Changes that may Require User Action

The following are operational changes that may require user action for existing DFSORT/ICETOOL applications that use certain functions as specified:

Return Area for ICETOOL STATS Operator

In order to handle larger numbers, ICETOOL will now return 16-byte PD values rather than 8-byte PD values in the return area for the STATS operator. If you have programs that call ICETOOL and use the returned STATS values, change the DCs for those values from PL8 to PL16 and recompile the programs before running them against this level of DFSORT/ICETOOL.

16-Byte FS Fields

In order to handle larger FS fields (a sign and up to 31 digits), DFSORT and ICETOOL will now treat a 16-byte FS field as having a maximum of 16 digits, rather than 15 digits.

For ICETOOL's DISPLAY and OCCUR operators, ON and BREAK fields for 16-byte FS values may result in different formatting for the output reports. You can use the new U15 formatting item to limit these values to 15 digits instead of 16 digits. For example, ON(11,16,FS,U15).

For INREC, OUTREC and OUTFIL OUTREC, 16-byte FS values may be edited or converted differently for output. You can use EDIT=(pattern) or LENGTH=n to change the length of the output fields, if appropriate. For example, 11,16,FS,EDIT=(SIIIIIIIIIIIIIT),SIGNS=(+,-).

TO=BI and TO=FI for 10-digit values

In order to handle larger numbers, INREC, OUTREC and OUTFIL OUTREC will create an 8-byte field instead of a 4-byte field when TO=BI or TO=FI is used for a

10-digit value. You can use LENGTH=n to change the length of the output fields, if appropriate. For example, 11,10,ZD,TO=BI,LENGTH=4.

New reserved words for symbols

The following are new DFSORT/ICETOOL reserved words (uppercase only, as shown), which are no longer allowed as symbols: DC1, DC2, DC3, DE1, DE2, DE3, SFF, TC1, TC2, TC3, TC4, TE1, TE2, TE3, TE4, UFF, ZDC, and ZDF.

If you used these words as symbols, you must change them to other words, such as lowercase or mixed case forms (for example, dc1 or Dc1).

Summary of Changes for SC26-7525-00 z/OS Version 1 Release 5

This document contains information that was previously presented in *z/OS V1R5 DFSORT Messages, Codes and Diagnosis Guide*, SC26-7525-00.

The following sections summarize the changes to that information.

New Information

This edition includes the following new enhancements:

Improvements in Performance

Memory object sorting is a new DFSORT capability that uses a memory object on 64-bit real architecture to improve the performance of sort applications. A memory object is a data area in virtual storage that is allocated above the bar and backed by central storage. With memory object sorting, a memory object can be used exclusively, or along with disk space, for temporary storage of records. Memory object sorting can reduce I/O processing, elapsed time, EXCPs, and channel usage. When a memory object is used, Hiperspace™ and data space are not needed.

Installation and Run-Time Options

A new MOSIZE installation (ICEMAC) and run-time option allows you to specify the maximum size of a memory object to be used for memory object sorting. The existing EXPMAX, EXPOLD and EXPRES installation (ICEMAC) options now apply to memory object sorting as well as Hipersorting.

A new NULLOUT installation (ICEMAC) and run-time option allows you to specify what you want DFSORT to do when there are no records for the SORTOUT data set. This gives you control over the action (continue or end), type of message (informational or error) and return code (0, 4 or 16) for a SORTOUT data set with no records.

A new NULLOFL installation (ICEMAC) option and OUTFIL run-time option allows you to specify what you want DFSORT to do when there are no data records for an OUTFIL data set. This gives you control over the action (continue or end), type of message (informational or error) and return code (0, 4 or 16) for an OUTFIL data set with no data records.

The IBM-supplied default for ICEMAC option COBEXIT has been changed from COB1 to COB2.

The IBM-supplied default for ICEMAC option DSA has been changed from 32MB to 64MB.

The IBM-supplied default for ICEMAC option TMAXLIM has been changed from 4MB to 6MB.

The IBM-supplied default for ICEMAC option ZDPRINT has been changed from NO to YES.

Summary of Changes for SC26-7050-05 DFSORT Release 14 (PTFs - February, 2003)

This document contains information that was previously presented in *DFSORT Messages, Codes and Diagnosis Guide R14*, SC26-7050-04.

The following sections summarize the changes to that information.

New Information

This edition includes the following new enhancements:

ICETOOL Enhancements

A new SPLICE operator helps you to perform various file ″join″ and ″match″ operations. SPLICE allows you to create output records in a variety of ways by splicing together fields from records that have the same key, but different information. For example, for the same userid, you could create output records consisting of the division and department from one type of record (originating in input file1) with the account number and phone number from another type of record (originating in input file2). Spliced records can be created by combining the first duplicate and:

- one or more fields from the last duplicate, or
- one or more fields for each subsequent duplicate, or
- one field from each subsequent duplicate.

Non-duplicate records can be deleted or kept.

The USING(xxxx) option can now be used with ICETOOL's SELECT operator to process DFSORT control statements like INCLUDE, OMIT and OUTFIL for a SELECT operation.

New LISTSDB and LISTNOSDB options for ICETOOL's DEFAULTS, DISPLAY and OCCUR operators allow you to control the use of system-determined optimum blocksize for LIST data sets.

OUTFIL Enhancements

New SAMPLE=n and SAMPLE=(n,m) options of OUTFIL allow you to sample records in a variety of ways.

A new REPEAT=n option of OUTFIL allows you to write each output record multiple times.

A new SPLITBY=n option of OUTFIL allows you to write groups of records in rotation among multiple output data sets.

A new DATE4 option of OUTFIL OUTREC allows you to insert a timestamp for your DFSORT run in the form 'yyyy-mm-dd-hh.mm.ss' into your records.

A new DATE4 option of OUTFIL INCLUDE and OUTFIL OMIT allows you to compare fields to a timestamp for your DFSORT run in the form 'yyyy-mm-dd-hh.mm.ss' or to a portion of that timestamp truncated on the right.

OUTFIL OUTREC now allows you to insert decimal constants (+n and –n) in your records as BI, FI, PD, ZD, FS or edited CH values.

OUTFIL OUTREC now allows you to combine fields (p,m,f), decimal constants (+n and –n), operators (MIN, MAX, MUL, DIV, MOD, ADD, SUB) and parentheses to form arithmetic expressions, and place the results in your records as BI, FI, PD, ZD, FS or edited CH values.

The maximum length for an SS field used with OUTFIL INCLUDE and OUTFIL OMIT has been raised to 32752.

A PD0 field can now be compared to a hexadecimal constant or to another PD0 field for OUTFIL INCLUDE and OMIT.

OUTFILE can now be used as an alias for OUTFIL.

INREC and OUTREC Enhancements

A new DATE4 option of INREC and OUTREC allows you to insert a timestamp for your DFSORT run in the form 'yyyy-mm-dd-hh.mm.ss' into your records.

INREC and OUTREC now allow you to insert decimal constants (+n and –n) in your records as BI, FI, PD, ZD, FS or edited CH values.

INREC and OUTREC now allow you to combine fields (p,m,f), decimal constants (+n and –n), operators (MIN, MAX, MUL, DIV, MOD, ADD, SUB) and parentheses to form arithmetic expressions, and place the results in your records as BI, FI, PD, ZD, FS or edited CH values.

INCLUDE and OMIT Enhancements

A new DATE4 option of INCLUDE and OMIT allows you to compare fields to a timestamp for your DFSORT run in the form 'yyyy-mm-dd-hh.mm.ss' or to a portion of that timestamp truncated on the right.

The maximum length for an SS field used with INCLUDE and OMIT has been raised to 32752.

A PD0 field can now be compared to a hexadecimal constant or to another PD0 field for INCLUDE and OMIT.

FORMAT=f can now be used with mixed p,m and p,m,f fields in the COND operand for INCLUDE and OMIT. f from FORMAT=f will be used for p,m fields but not for p,m,f fields.

SORT, MERGE and SUM Enhancement

FORMAT=f can now be used with mixed p,m and p,m,f fields in the FIELDS operand for SORT, MERGE and SUM. f from FORMAT=f will be used for p,m fields but not for p,m,f fields.

Other Enhancements

Enhancements to DFSORT's control statement continuation rules allow you to continue a line that breaks at column 71 anywhere in columns 2 to 16 of the next line.

When PARMDDN=ddname is specified at installation-time, DFSORT will now use a //DFSPARM DD data set if a //ddname DD data set is not present. When

PARMDDN=DFSPARM is specified or defaulted at installation-time, DFSORT will continue to use a //$ORTPARM DD data set if a //DFSPARM DD data set is not present.

Summary of Changes for SC26-7050-05 DFSORT Release 14 (PTFs - March, 2002)

This document contains information that was previously presented in *DFSORT Messages, Codes and Diagnosis Guide R14*, SC26-7050-04.

The following sections summarize the changes to that information.

New Information

This edition includes the following new enhancements:

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 that 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 ends if the QNAME option is specified on an output DD statement.

Part 1. Eliminating Common Sources of Error

Chapter 1. Eliminating Common Sources of Error 3
How to Get Messages 3
How to Get a Dump 3
Eliminating DFSORT Installation as a Source of Error 4
Eliminating Your Invoking Program as a Source of Error 4
Eliminating Your User Exit Routines as a Source of Error 4
Errors in Using Registers 4
Errors in Using Main Storage 5
Errors in Record Contents 5
Being Aware of Data Management Rules 5
Correcting Common Errors 6
Common System Abends Related to Storage . . . 6
Miscellaneous System Abends 7
User Abends at Run Time 9
User Abends from DFSORT Panels 9
Common Problems Reported by DFSORT Users 10

© Copyright IBM Corp. 1995, 2006

Chapter 1. Eliminating Common Sources of Error

Some problems in running DFSORT are not caused by DFSORT failures, and using the information in this section can save you time. This section explains how to get messages and dumps as well as how to identify and eliminate problems caused by errors in your installation of DFSORT, your calling programs, or your user exit routines. The section also discusses other common errors.

How to Get Messages

When a message data set is required, but neither a SYSOUT DD statement nor an alternative message data set ddname is provided, DFSORT can either terminate or continue. Using the defaults with which it was delivered, DFSORT issues message ICE158A to the console, and terminates with either a return code of 20 or a user abend code of 158 (depending on the ERET, ABEND, or NOABEND option in effect at run-time). You can change the default value during installation with the ICEMAC option NOMSGDD to allow DFSORT to continue even when a required message data set is not supplied.

If you do not receive any DFSORT messages, make sure that you supply a message data set, and resubmit your job using a SORTDIAG DD statement with a DUMMY parameter. The SORTDIAG DD statement specifies that all messages, including diagnostic messages (ICE800I through ICE999I), and control statements be printed.

You can provide either a SYSOUT DD statement or a DDNAME DD statement (where DDNAME is the alternate message data set ddname specified at installation or execution time with MSGDDN=ddname).

For example:

```
//SORTDIAG DD DUMMY
//SYSOUT DD SYSOUT=A
```

If you use a temporary or permanent message data set, it is best to specify a disposition of MOD to ensure you see all messages and control statements in the message data set. For example:

```
//SYSOUT DD DSN=&MSG,DISP=(MOD,PASS),SPACE=(CYL,(2,2)),UNIT=SYSDA
```

How to Get a Dump

To get a dump at an abend, use the JCL and program control statements shown in Table 2. You get additional abend information when DFSORT's ESTAE recovery routine is in effect. At the beginning of each run, DFSORT establishes an ESTAE recovery routine to trap system or user abends for Blockset and Peerage/Vale applications. The ESTAE recovery routine remains in effect unless you use ICEMAC option ESTAE=NO or the run-time option NOESTAE to delete it.

Table 2. Getting a Dump

If you received	Then use
A user or system abend code	//SYSUDUMP DD SYSOUT=A

A condition code 16	//SYSUDUMP DD SYSOUT=A
	//DFSPARM DD *
	DEBUG ABEND

© Copyright IBM Corp. 1995, 2006

Table 2. Getting a Dump (continued)

If you received	Then use
Note: If your program invokes DFSORT and does not pass the DFSORT return code to the system, the listing's condition code might not be the actual DFSORT return code. To determine if the code in the listing is from DFSORT, find out how your program handles return codes.	

If you invoke DFSORT from a COBOL program, you might receive a COBOL dump and a DFSORT dump. For the procedures in this manual, use only the DFSORT dump.

If you are using PL/I, you must specify the NOSPIE and NOSTAE macro at run time to ensure that you receive a DFSORT dump. If you do not, you might receive only a PL/I dump.

Eliminating DFSORT Installation as a Source of Error

If you install a new DFSORT release or PTF, the standard DFSORT aliases will be installed automatically. However, you must re-install any aliases you previously installed that are affected by the new release or PTF. Otherwise, DFSORT might be executing mixed levels of program modules, which can cause an abnormal end (ABEND).

Eliminating Your Invoking Program as a Source of Error

Main storage space can be a problem when you invoke DFSORT from another program, especially if you use MAINSIZE/SIZE=MAX and invoke user exit E15 or E35 (or, from COBOL, use an input or output procedure). In this case, you usually receive a system ABEND 80A message.

If you invoked DFSORT from another program, and particularly if you opened a file in your user exit routine, check that you specified a sufficiently large amount of reserved storage with the RESALL, ARESALL, RESINV, or ARESINV run-time option.

For more detailed information on main storage, see *z/OS DFSORT Application Programming Guide*.

Eliminating Your User Exit Routines as a Source of Error

To eliminate your own routines or programs as possible sources of error, check the use of registers, the main storage space allocated to the routines or programs, and the record contents.

Errors in Using Registers

Make sure your routines or programs save and restore the appropriate registers. If the routines or programs return to DFSORT with, for example, a different value in register 12 the outcome is probably an abend.

Ensure that registers you use for loading or storing are not accidentally overlaying DFSORT code or work areas. In this case, DFSORT might work without errors with one technique but fail with another.

For more detailed information on register usage, see *z/OS DFSORT Application Programming Guide*.

Errors in Using Main Storage

Check whether your routines need more main storage space than you have allocated. Have you increased the size of your routines without increasing the corresponding exit size on your MODS statement? A change in the size of your user exit routines can lead to an abend in your own routine, or it can lead to insufficient space for DFSORT.

You can see if too little space was left for sorting by studying the information in messages ICE092I, ICE093I, ICE156I and ICE231I. If you did not receive any messages, refer to "How to Get Messages" on page 3.

Similar situations can occur if the MAINSIZE/SIZE=MAX option is specified or taken by default and a fairly large reserved value is passed to DFSORT through the RESALL, ARESALL, RESINV, or ARESINV option or taken by default. Another problem can arise if your E15 routine issues a GETMAIN without a corresponding FREEMAIN at the end. This can be done indirectly, for example, by leaving a data set open so that a buffer pool remains reserved.

For more detailed information, see *z/OS DFSORT Application Programming Guide*.

Errors in Record Contents

If the output records do not appear to contain the same data as the input records, and either E15 or E35 has been used, check that your routine is handling register 1 correctly; especially, check that it is correct on return to DFSORT. If, for example, you first load register 1 and then restore all registers (including register 1), register 1 probably has the wrong contents.

Also, if you first restore all registers and then try to load register 1 from a changed base register, you might pass the wrong information to DFSORT.

Being Aware of Data Management Rules

Standard system data management rules apply to all data set processing. In particular:

- Be aware that when using fixed standard record format for input data sets, the first short block is treated like an End of Volume. See *z/OS DFSMS Using Data Sets*, SC26-7410 for more details.
- Be aware that, in some cases, if a DD statement specifies a data set for output that is extended to a second or subsequent volume, and another DD statement within the same step requests the same data set, only the records on the first volume will be read, and incorrect output will result.

Specifically, when a new output data set is allocated with a unit count and volume count greater than one but specifies no volume serial numbers, one volume is allocated. If a second or succeeding DD statement within the same step requests the same data set, the same volume is allocated to it. If this job step extends the output data set to more volumes, this new volume information is not available to the second or succeeding DD statement.

Thus, you should not use different DDs for a data set to be used for output and then input in the same step, unless that data set cannot be extended to a second or

subsequent volume, or is allocated with the guaranteed space attribute in the storage class. See *z/OS MVS JCL Reference*, SA22-7597 and *z/OS DFSMS Using Data Sets*, SC26-7410 for more details.

Correcting Common Errors

You can encounter abends or performance problems not caused by failures in the DFSORT program or the DFSORT Panels program. For example, your job might terminate abnormally because of system errors beyond your control. Abends can also result from specifying inadequate storage or incorrect storage parameters. You can sometimes improve degraded performance by allocating resources differently.

This section lists abends you might encounter when running DFSORT, and suggests ways of correcting the problems. Not every possible problem can be listed, but the list includes those most commonly reported to IBM personnel. Use the information here in conjunction with Part 2, "Messages and Codes," on page 13 and *z/OS MVS System Codes*.

Common System Abends Related to Storage

Table 3 covers common system abend codes that might appear when abends result from errors in storage parameters. The items are listed by system abend code. For details on DFSORT's use of storage and storage parameters, see *z/OS DFSORT Installation and Customization* and *z/OS DFSORT Application Programming Guide*.

Table 3. Common System Abends Related to Storage

Abend Code	Problem
106	There was not sufficient storage to load a module. Ensure that the values for REGION, MAINSIZE, SIZE, RESALL or RESINV are sufficient.
804	The amount of storage requested exceeded that available. Ensure that the values for REGION, MAINSIZE, SIZE, RESALL or RESINV are sufficient.
878	User exits required more storage than was available. Ensure that the values for each user exit size in the MODS statement and for REGION, MAINSIZE, SIZE, RESALL or RESINV are sufficient.
80A	This abend can result from the following: • A DFSORT job required more storage than was available. Ensure that the values for REGION, MAINSIZE, SIZE, RESALL or RESINV are sufficient. • The '-1' flag required to mark the end of an extended parameter list was missing. Include an F'-1' to mark the last item in the parameter list. • The invoking program used AL3 to define addresses for DFSORT control statements. Be sure to use full 4-byte addresses to prevent DFSORT from "running off the end" while allocating storage.
D37	A SORTWK, SORTOUT, or OUTFIL data set was specified with insufficient primary space allocation and no secondary allocation. Increase the primary allocation, or include a value for secondary allocation.
E37	A SORTOUT or OUTFIL data set was allocated on a primary volume that had either insufficient storage or excessive fragmentation, and no secondary volume was specified. Specify more than one volume in the VOLSER parameter for the SORTOUT or OUTFIL data set or compress the volume to allow it to contain the full 16 extents.

Miscellaneous System Abends

Table 4 covers other common system abend codes. The items are listed by system abend code.

Table 4. Miscellaneous System Abends

Abend Code	Problem
001	This abend can result from the following: • The data control block (DCB) for SORTIN contained incorrect LRECL or BLKSIZE data set attributes. Check the SORTIN DD statement to ensure that the LRECL and BLKSIZE parameters are correct. • When a COBOL program invoked DFSORT: – The SELECT or ASSIGN statement specified an incorrect file name or DDNAME parameter. – The SORTIN or SORTOUT files specified DCB attributes different from those defined for the data set. – The FD file description for SORTIN or SORTOUT files contained inaccurate specifications (such as RECORD CONTAINS, BLOCK CONTAINS, RECORDING MODE, and so on).
002	This abend can result from the following: • For variable-length record SORTOUT or OUTFIL data sets, the record descriptor word was invalid (less than 4 bytes or greater than 32 KB) or, for nonspanned records, was greater than the specified block size. • For fixed-length record SORTOUT or OUTFIL data sets, the LRECL parameter was greater than the BLKSIZE parameter.
013	The SORTIN, SORTOUT, or OUTFIL BLKSIZE parameter: • Was greater than 32760 for a disk data set, or was greater than the maximum block size supported by the access method for a tape data set. • Was not an integer multiple of the LRECL parameter for fixed-length records, or was not at least four bytes longer than the LRECL for variable-length records.

Table 4. Miscellaneous System Abends (continued)

Abend Code	Problem
0Cx	These abends can result from the following: • Locale processing with a run-time library that contains down-level dynamically loadable routines. • An invalid PD or ZD sign (0-9) or digit (A-F) can result in a data exception (0C7 ABEND) for SUM processing. An invalid PD or ZD digit can result in a data exception for OUTFIL edit or statistics processing. ICETOOL's DISPLAY or VERIFY operators can be used to identify decimal values with invalid digits. ICETOOL's VERIFY operator can be used to identify decimal values with invalid signs. • A user exit passed back a variable-length record that was shorter than the record descriptor word indicated. • A user exit passed back a fixed-length record that was shorter than the length specified in the RECORD statement (L2 for E15 and E32, or L3 for E35). • The contents of register 1 (and possibly other registers) used by DFSORT were altered before returning to DFSORT. Ensure the registers are restored before returning to DFSORT. • The SORTLIB DD statement specified an incorrect library. • The DEBUG CTRx option was in effect and the criteria were met. • For COBOL: – DISPLAY was used for COBOL messages, but the message data set was missing. Supply a message data set DD statement. (See the discussion of MSGDDN in *z/OS DFSORT Application Programming Guide* for an alternative solution.) – An attempt was made to reopen an open data set, or a data set was already open prior to executing the SORT USING statement. – A STOP RUN or GOBACK statement was required but not supplied; therefore, the COBOL program did not exit as intended but processed the next instruction. – Invalid logic was used in a sort I/O procedure (possibly using the STOP RUN, GOBACK, CALL, or GO TO statements), or PERFORM referred to a paragraph outside the section. – The file name or DDNAME parameter of the SELECT/ASSIGN statement was incorrect. – The LINKAGE SECTION was specified incorrectly or misplaced. – The PROCEDURE DIVISION USING statement gave an invalid parameter list. – An attempt was made to move data from the COBOL SD area prior to execution of the SORT verb. – The FD entry RECORD CONTAINS clause (LRECL) for the USING data set is different than the SD entry RECORD CONTAINS clause for the SORTWK data set.
213	This abend can result from the following: • A specified data set was not found. • Space for the SORTOUT or OUTFIL data set was not allocated. • For COBOL: – The file name or DDNAME parameter of the SELECT or ASSIGN statement was incorrect. – The SORT verb required the SORTLIB DD statement.
30A	The SORTIN LRECL parameter was greater than the SORTOUT or OUTFIL LRECL parameter (for VSAM data sets).

Table 4. Miscellaneous System Abends (continued)

Abend Code	Problem
513	For COBOL, the SORT USING statement was used with an input tape data set that was already opened prior to invoking DFSORT.
806	DFSORT required a program, such as a user exit or invoking program, during execution and was unable to find it. Correct the job step or supply the necessary program. ABEND 806, along with message "CSV003I REQUESTED MODULE CEEBINIT NOT FOUND", may occur if during locale processing DFSORT does not have access to the library containing the dynamically loadable routines. For example, the data set containing these routines might be called SYS1.SCEERUN. If you are unsure of the name of the data set, contact your system administrator.
813	The SORTIN, SORTOUT, or OUTFIL data set was tape, and the data set name was incorrect or missing or DISP=MOD was specified inappropriately for a tape data set.
913	For COBOL, the file name or DDNAME parameter of the SELECT or ASSIGN statement was incorrect.
C03	A user exit opened data sets, but did not close them upon completion.
C13	For COBOL, the file name or DDNAME parameter of the SELECT or ASSIGN statement was incorrect.

User Abends at Run Time

DFSORT issues two types of user abends at run time:

- If ABEND or ABSTP is in effect and DFSORT terminates with an error message, you receive a user abend. With installation option ABCODE=MSG (the supplied default), the user abend code is equal to the error message (for example, U0046 for message ICE046A). With installation option ABCODE=*n*, the user abend code is equal to *n* (for example, U0010 with ABCODE=10 for any error message). For user abend codes that are error message numbers (ABCODE=MSG), you can refer to the explanation for the corresponding message (for example, the ICE046A Explanation for U0046).

  For a tape work data set sort or conventional merge, user abend 0 can be issued if DFSORT terminates with an error message and ABEND or ABSTP is in effect.
- If you receive a user abend with a code from 1000 through 1599, DFSORT detected an error in its internal logic. The abend is issued in this case to prevent infinite loops and to aid diagnosis of the problem. Please report such problems to your IBM representative.

  In some cases, you may be able to bypass a DFSORT user 1nnn abend by making more storage available to DFSORT; try adding 32 KB by changing the REGION parameter or the MAINSIZE or SIZE value. See *z/OS DFSORT Application Programming Guide* for ways to specify MAINSIZE and SIZE.

User Abends from DFSORT Panels

If you receive a user abend with a code of 400, 401, or 402, a failure has occurred in the DFSORT Panels interface with ISPF shown Table 5 on page 10.

Table 5. User Abends from DFSORT Panels

Abend Code	Problem
400	An unexpected return code was received from an ISPF service after DFSORT Panels called that service. Possible causes are: • A DFSORT run-time abend (system or user) occurred after foreground execution was selected from DFSORT Panels. DFSORT run-time abends can cause problems with the ISPF environment, with unpredictable results. To avoid DFSORT error message user abends when using DFSORT Panels, make sure that your site has selected ERET=RC16 for the ICEMAC TSO environment, and that the ABEND option is not specified at run time. • An ISPF service that was called by DFSORT Panels did not complete as expected. • An internal logic error in DFSORT Panels resulted in an incorrect call to an IBM service.
401	An ISPF SELECT service severe error (SELECT return code of 20) occurred after DFSORT Panels used the ISPF SELECT service to invoke a user-defined data entry routine. See *z/OS DFSORT Installation and Customization* for more information about customizing DFSORT Panels.
402	An ISPF SELECT service severe error (SELECT return code of 20) occurred after DFSORT Panels used the ISPF SELECT service to invoke a user-defined statement coder routine. See *z/OS DFSORT Installation and Customization* for more information about customizing DFSORT Panels.

If you receive a user abend with a code from 1600 through 1699, DFSORT Panels detected an error in its internal logic. The abend is issued in this case to prevent infinite loops and to aid diagnosis of the problem. Please report such problems to your IBM representative.

For information about other user abends displayed by ISPF, see *ISPF Dialog Management Guide*.

Common Problems Reported by DFSORT Users

The situations below are problems that are frequently reported to IBM personnel.

- System messages indicate DFSORT is using more storage than specified in the DFSORT parameters.

  DFSORT treats its storage parameters (for example, TMAXLIM, MAXLIM, SIZE, MAINSIZE) as flexible limits, and may, in fact, use more storage than the specified limits in some cases to optimize performance or prevent termination. To avoid storage fragmentation, DFSORT might request all storage allocated to it above and below 16 MB virtual, then release all storage exceeding the specified limits.

  DFSORT messages ICE092I and ICE093I list an estimate of the total amount of storage used by DFSORT. DFSORT message ICE156I lists an estimate of the storage used by DFSORT above 16 MB virtual. The difference between the two is an estimate of the storage used by DFSORT below 16 MB virtual. Message ICE231I lists an estimate of the storage used exclusively for OUTFIL processing above and below 16 MB virtual.
- Elapsed time for executing sort jobs appears excessive.

To achieve the shortest elapsed time for sort jobs, DFSORT tries to select the Blockset technique. Examine message ICE143I to see which technique was used. If DFSORT did not select Blockset, code a SORTDIAG DD DUMMY statement in the JCL to request a set of diagnostic messages and re-run the job. Diagnostic message ICE800I reports a reason code indicating why DFSORT did not select the Blockset technique. Correct the condition, if possible.

Excessive elapsed time can also be the result of forcing DFSORT to sort a large data set in a relatively small amount of storage. Ensure that you provide sufficient storage for the data to be sorted as discussed in *z/OS DFSORT Application Programming Guide* and *z/OS DFSORT Tuning Guide*.

Part 2. Messages and Codes

Chapter 2. Message Considerations 15
Printing Messages and Control Statements 15
Printing EFS Messages and Control Statements . . 16
Writing Messages to the Master Console. 16
Finding DFSORT Control Statement Coding Errors 17

Chapter 3. Informational and Error Messages . . 19

Chapter 4. Diagnostic Messages 83

Chapter 5. Return Codes and User Abends . . . 87
DFSORT Return Codes 87
ICETOOL Return Codes 87
ICEGENER Return Codes 88

© Copyright IBM Corp. 1995, 2006

Chapter 2. Message Considerations

This chapter discusses the types of messages DFSORT produces and explains how to print messages and control statements, write messages to the master console, and find control statement coding errors.

DFSORT produces three types of messages:
- **Critical error** messages, which report and describe problems requiring programmer action to correct (explained in Chapter 3, "Informational and Error Messages," on page 19).
- **Informational** messages, which provide information only and require no programmer action (explained in Chapter 3, "Informational and Error Messages," on page 19).
- **Diagnostic** messages, which provide additional information useful in debugging and tuning (explained in Chapter 4, "Diagnostic Messages," on page 83).

Different messages might be issued for the same error situation, depending upon which DFSORT technique is used.

See *z/OS DFSORT Installation and Customization* for information on the ICEMAC installation options discussed in this section. PARM options and program control statement options are explained in *z/OS DFSORT Application Programming Guide*.

Printing Messages and Control Statements

You can select the type of messages you want written to the message data set during installation or at run time. Messages ICE097I and ICE158A are written only to the master console.

If ICEMAC option NOMSGDD=QUIT is in effect, you must provide a DD statement using the name of the alternate message data set specified during installation or at run time, or a SYSOUT DD statement. Otherwise, DFSORT terminates with a return code of 20.

The following list outlines the hierarchy for message and control statements for the Blockset and Peerage/Vale sorting techniques. SORTDIAG DD, MSGPRT, and LIST, as they are used here, represent all methods of specifying the appropriate function. For example, FLAG (I) in the EXEC PARM field is equivalent to MSGPRT=ALL, and DIAGSIM=YES in ICEMAC is equivalent to specifying a SORTDIAG DD statement for every job.

SORTDIAG DD statement
: Prints all messages (including diagnostic messages); prints control statements. A message data set DD statement should be provided.

  When SORTDIAG DD is specified, Blockset error messages that are normally suppressed are printed. This may result in an error message being printed for a run that is actually successfully completed by another technique (as indicated by a return code of 0 or 4).

MSGPRT=ALL and **LIST** in effect
: Prints all messages except diagnostic messages; prints control statements. A message data set DD statement should be provided.

© Copyright IBM Corp. 1995, 2006

MSGPRT=ALL and **NOLIST** in effect
Prints all messages except diagnostic messages; does not print control statements. A message data set DD statement should be provided.

MSGPRT=CRITICAL and **LIST** in effect
Prints only critical error messages; prints control statements. A message data set DD statement should be provided.

Control statements are not printed if MSGPRT=CRITICAL is in effect and DFSORT is dynamically invoked using the 24-bit parameter list.

MSGPRT=CRITICAL and **NOLIST** in effect
Prints only critical error messages; does not print control statements. A message data set DD statement should be provided.

MSGPRT=NONE in effect
Does not print messages; does not print control statements.

Printing EFS Messages and Control Statements

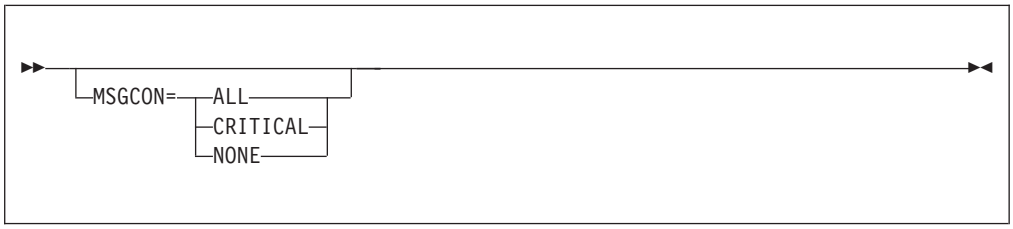
Both the Extended Function Support (EFS) program and its exit routines (EFS01 and EFS02) can send messages to DFSORT for printing. The EFS program can send messages to DFSORT for printing at Major Call 1, 2, 3, or 4. EFS01 and EFS02 can only send messages to DFSORT for printing at Major Call 4. DFSORT prints EFS messages to the message data set, but does not write messages to the master console.

Use the MSGPRT option to control the printing of messages returned by the EFS program. Use the LISTX and NOLISTX options to control the printing of control statements returned by the EFS program; use these two options in the same way you use LIST and NOLIST to control printing of DFSORT control statements. LISTX and NOLISTX can be used independently of LIST and NOLIST. When a SORTDIAG DD statement is specified, LISTX is used. If no verb request list is returned at Major Call 1, NOLISTX is used.

See *z/OS DFSORT Application Programming Guide* for more information about Extended Function Support.

Writing Messages to the Master Console

You can select the type of messages to be written to the master console with ICEMAC option MSGCON:



ALL Specifies that all messages except option-in-effect messages (ICE127I through ICE133I, and ICE235I) and diagnostic messages (ICE800I through 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.

Console message choices are independent of the message data set choices. Messages ICE097I and ICE158A are written only to the master console. Inclusion of a SORTDIAG DD statement has no effect on console messages.

Because of the addition of the job name and the step name, some DFSORT messages can be truncated from the right when they are written to the master console.

Messages returned by an EFS program cannot be written to the master console.

Finding DFSORT Control Statement Coding Errors

DFSORT scans control statements for syntax and content errors, such as invalid operators, operands, or operand values, missing operands, and missing or invalid delimiters.

As explained in "Printing Messages and Control Statements" on page 15 and "Printing EFS Messages and Control Statements" on page 16, DFSORT prints messages and control statements as requested by the SORTDIAG DD statement, MSGPRT, LIST, and LISTX operands. The following paragraphs assume that messages and control statements are to be printed.

DFSORT scans each control statement for errors, and prints an error message for the first error detected. Scanning stops at the first error, and then continues with the next control statement. If DFSORT detects an error, it usually terminates after all the control statements are scanned. However, in some cases, detection of an error by the Blockset technique results in the use of another technique that could avoid termination, and, in other cases, Blockset terminates immediately without completing control statement scanning.

If an error is associated with a specific operator, operand or value for a control statement in DFSPARM, SYSIN, and SORTCNTL, DFSORT prints a marker ($) directly below the control statement near the error. If an error is associated with a specific control statement, DFSORT prints the error message directly below the marker line or control statement.

If an error is associated with a specific control statement returned by an Extended Function Support (EFS) program, the EFS program can return a message to be printed with that control statement. The associated message must be returned at the same time the control statement is returned to DFSORT. The EFS program can also return an offset pointing to the error found in the control statement. DFSORT uses this offset to place a marker ($) below the control statement. Any message associated with the control statement is printed directly below the marker line or control statement. See the *z/OS DFSORT Application Programming Guide* for more information about Extended Function Support.

Chapter 3. Informational and Error Messages

DFSORT messages can appear within your printed output or on the master console. The formats are:

For printed messages

ICEnnns c text

For console messages

ICEnnns c jjjjjjjj.ssssssss text

Where	Represents
nnn	Message number.
s	Severity code. Values for *s* are:
	A Critical error message; requires programmer action.
	I Informational message; usually requires no programmer action.
c	Diagnostic trace character. This character is usually not needed, but might be requested by your IBM representative for diagnostic purposes.
jjjjjjjj	Job name.
ssssssss	Step name.
text	Message text.

ICE000I - CONTROL STATEMENTS FOR 5694-A01, Z/OS DFSORT *vvrr - hh:mm* **ON** *day mmm dd, yyyy* **-**

Explanation: Identifies the following for this DFSORT run:

- z/OS version level (*vv*) and release level (*rr*)
- Time as hours (*hh*) and minutes (*mm*)
- Day of the week (*day*)
- Date as month (*mmm*), day of the month (*dd*) and year (*yyyy*)

System action: None.

Programmer response: None.

ICE001A TEXT BEGINS IN WRONG COLUMN

Explanation: Critical. A continuation line started in column 1.

System action: The program terminates.

Programmer response: Use one of the following recommended valid methods for continuing a statement:

- Break the previous line at a comma-blank or semicolon-blank or colon-blank before column 72 and start the continuation line in column 2-71
- Break the previous line at column 71, put a nonblank in column 72 and start the continuation line in column 16.

See *z/OS DFSORT Application Programming Guide* for other valid methods of continuing a statement.

ICE002I DUPLICATE OR CONFLICTING *xxxxxxxx* **STATEMENT**

Explanation: The same operation definer, or mutually exclusive operation definers (SORT and MERGE, or INCLUDE and OMIT), appeared more than once in the same source (for example, SORTCNTL).

System action: DFSORT does not analyze duplicate or

© Copyright IBM Corp. 1995, 2006

conflicting statements. The first one encountered is used.

Programmer response: No action is necessary, if the parameters on the duplicate (second and subsequent) control statements are not needed. Otherwise, consolidate the parameters from the duplicate control statements into a single control statement.

ICE003A CONTINUATION LINE MISSING

Explanation: Critical. A continuation line was indicated by the previous line ending with a comma, or with a nonblank entry in column 72, but no line followed.

System action: The program terminates.

Programmer response: Check for an overflow of parameters into column 72 or a missing continuation line.

ICE005A STATEMENT DEFINER ERROR

Explanation: Critical. A control statement did not contain one of the acceptable operation definers (ALTSEQ, DEBUG, END, INCLUDE, INREC, MERGE, MODS, OMIT, OPTION, OUTFIL, OUTREC, RECORD, SORT, SUM). You can also receive this message for continuation lines after a line that has an error.

System action: The program terminates.

Programmer response: Check all statements for incorrect, misplaced, or misspelled operation definers. Check that no definer begins in column 1 (in which case it was treated as a label). If you have a label, check that it begins in column 1 (otherwise, it was treated as an operation definer). Check that no definer or DFSPARM PARM option is continued using the continuation column.

ICE006A OPERAND DEFINER ERROR

Explanation: Critical. The first operand of a control statement did not begin on the same line as the operation definer, or an operand or operand value was not valid, or a required operand was not specified.

System action: The program terminates.

Programmer response: Check for statements with invalid operands, invalid operand values, missing operands, or no operands.

ICE007A SYNTAX ERROR

Explanation: Critical. A control statement contained an error in syntax.

System action: The program terminates.

Programmer response: Check the control statements for syntax errors. Some of the more common syntax errors are:

- Unbalanced parenthesis
- Missing comma
- Embedded blank
- Invalid format type
- Invalid operator
- Invalid constant
- Continuation of DFSPARM PARM options using the continuation column.
- Symbol used where it is not allowed
- Parsed field (%nn) used where it is not allowed.

ICE008A FIELD VALUE EXCEEDS MAXIMUM ALLOWABLE CHARACTERS

Explanation: Critical. A specified parameter value was too long. Most parameter values have a maximum length of eight characters, but some can have longer or shorter maximum lengths.

System action: The program terminates.

Programmer response: Check control statements for parameter values that are too long. Refer to the description of each parameter to verify the maximum length for its values.

ICE010A NO SORT OR MERGE CONTROL STATEMENT

Explanation: Critical. All control statements have been processed and either:

- No SORT or MERGE control statement or OPTION COPY statement was found, or
- A 24-bit parameter list did not contain a SORT or MERGE control statement.

System action: The program terminates.

Programmer response: Make sure that the 24-bit parameter list contains a SORT or MERGE control statement. If you are not using a 24-bit parameter list, make sure you have supplied a SORT or MERGE control statement or an OPTION COPY statement.

ICE011A DUPLICATE OR CONFLICTING OPERANDS ON THE OPTION STATEMENT

Explanation: Critical. On an OPTION control statement, one of the following errors was found:

- A keyword was specified twice.
- A keyword and a variation of it were both specified. CKPT and CHKPT are variations, as are FILSZ and SIZE.
- A keyword and its opposite were both specified. EQUALS and NOEQUALS are examples of this.

System action: The program terminates.

Programmer response: Check the OPTION control statement for the errors indicated in the explanation and correct the errors.

ICE012A MISSING FIELDS OPERAND DEFINER

Explanation: Critical. A SORT, MERGE, or SUM control statement did not contain the FIELDS operand.

System action: The program terminates.

Programmer response: Check for a SORT, MERGE, or SUM control statement that does not have the FIELDS operand.

ICE013A INVALID SORT OR MERGE STATEMENT OPERAND

Explanation: Critical. An invalid keyword operand was detected on a SORT or MERGE control statement.

System action: The program terminates.

Programmer response: Make sure the SORT or MERGE control statement does not contain an invalid keyword operand.

ICE014A DUPLICATE SORT OR MERGE STATEMENT OPERAND

Explanation: Critical. On a SORT or MERGE control statement, one of the following errors was found:
- A keyword was specified twice.
- A keyword and a variation of it were both specified. CKPT and CHKPT are variations, as are FILSZ and SIZE.
- A keyword and its opposite were both specified. EQUALS and NOEQUALS are examples of this.

System action: The program terminates.

Programmer response: Check the SORT or MERGE control statement for the errors indicated in the explanation and correct the errors.

ICE015A VARIABLE RECORD TOO SHORT

Explanation: Critical. This message was issued for one of the following reasons:
- An input data set contained an invalid RDW or BDW.
- A record passed to DFSORT from user exit E15 or E35 contained an invalid RDW.
- A variable-length record was too short to contain all fields or, if Blockset was not selected, shorter than L4.
- A record did not contain the first byte of the first control field when VLSHRT was in effect and Blockset was not selected.

System action: The program terminates.

Programmer response: If the records are valid but too short to contain all fields, use the VLSCMP option (INCLUDE, OMIT) or VLSHRT option (INCLUDE, OMIT, SORT, MERGE) to prevent DFSORT from terminating. Otherwise, take one or more of the following actions:
- Check that the input data set contains valid BDWs.
- Check that the input data set and all records passed from exits contain variable-length records with valid RDWs.
- Decrease L4 if it is too large.
- Check the input data set and all records inserted at user exits E15 and E35 to see that all records contain all fields.
- Remove any records that are too short.
- Check your E15 and E35 routines and correct any errors.

If ICETOOL was used, check each ON field of the failing operator for incorrectly specified displacement (position) or length.

You can use the ICETOOL STATS operator with ON(VLEN) to show the length of the shortest variable length record. You can use the ICETOOL DISPLAY operator with ON(VLEN) to show the length of each variable length record.

ICE016A INVALID FIELDS OPERAND VALUE

Explanation: Critical. The FIELDS operand of a SORT or MERGE statement contained an invalid f or s value in a p,m,s or p,m,f,s field.

System action: The program terminates.

Programmer response: Check for invalid f and s values in the control fields.

ICE017A CONTROL FIELD DISPLACEMENT OR LENGTH VALUE ERROR

Explanation: Critical. The message was issued for one of the following reasons:
- The length or position of a control field was specified incorrectly.
- The position plus length for a control field exceeded 32753.
- A bit position or length specified a control field whose format is not BI.
- A bit position or length exceeded 7.
- VLSHRT was in effect and control fields were too complex.

System action: The program terminates.

Programmer response: Check the position, length, and format of each control field for the situations described above. Specify NOVLSHRT if possible.

ICE018A INVALID OR MISSING FORMAT

Explanation: Critical. This message was issued for one of the following reasons:

1. More than 112 control fields were specified, and Blockset was not selected.
2. A SORT, MERGE, OUTFIL, SUM, INCLUDE, or OMIT statement contained an invalid format type. For example:

```
 SORT FIELDS=(5,4,NG,A)
*  NG IS AN INVALID FORMAT

 SUM FIELDS=(12,2),FORMAT=CH
*  CH IS AN INVALID FORMAT FOR SUM
```

3. For a SORT or MERGE control statement, the format was invalid for the length specified.
4. CSF, FS, UFF, SFF, Y2x or PD0 format was specified for SORT or MERGE and Blockset was not selected.
5. D2 format was specified in the INCLUDE or OMIT operand of an OUTFIL statement, or in the IFTHEN WHEN operand of an INREC, OUTREC, or OUTFIL statement.
6. The INCLUDE or OMIT operand of an OUTFIL statement, or an IFTHEN WHEN operand of an INREC, OUTREC, or OUTFIL statement, contained an invalid format type or a field without a format. For example:

```
  OUTFIL FNAMES=OUT1,
    INCLUDE=(5,2,NG,EQ,C'AB')
*    NG IS AN INVALID FORMAT

  OUTFIL FNAMES=OUT2,
    IFTHEN=(WHEN=(21,2,EQ,35,2),OVERLAY=(21:C'OK'))
*    FORMATS ARE MISSING IN WHEN CONDITION
```

7. A SORT, MERGE, SUM, INCLUDE, or OMIT statement without a FORMAT=f operand contained a field without a format (that is, p,m instead of p,m,f). For example:

```
 SORT FIELDS=(5,4,BI,A,21,2,D)
*  FORMAT IS MISSING FOR SECOND FIELD
```

System action: The program terminates.

Programmer response: For cases 1 through 3, check that each format type is valid for the control statement specified.

For case 4, rerun the job with a SORTDIAG DD statement to get message ICE800I, which indicates the reason Blockset could not be used. If possible, remove the condition preventing the use of Blockset.

For case 5, use an INCLUDE or OMIT statement, or change the D2 format to a format that is valid for the INCLUDE or OMIT operand of an OUTFIL statement, or for the IFTHEN WHEN operand of an INREC, OUTREC or OUTFIL statement, as appropriate.

For case 6, check that each field is specified as p,m,f with a valid format for f. For example:

```
OUTFIL FNAMES=OUT1,
 INCLUDE=(5,2,CH,EQ,C'AB')

OUTFIL FNAMES=OUT2,
     IFTHEN=(WHEN=(21,2,BI,EQ,35,2,BI),
       OVERLAY=(21:C'OK'))
```

For case 7, check that each field is specified as p,m,f or that a FORMAT=f operand is specified. For example:

```
SORT FIELDS=(5,4,BI,A,21,2,D),FORMAT=FI
```

ICE019A MEMBER NAME NOT SPECIFIED FOR *ddname*

Explanation: Critical. A data set indicated by the specified ddname is partitioned, but the member name was missing.

System action: The program terminates.

Programmer response: Ensure that a member name is specified for each partitioned data set or PDSE defined for the indicated ddname. If the ddname is a concatenation of data sets, then ensure that each partitioned data set or PDSE in the concatenation has a member name specified with it.

ICE020A INVALID RECORD STATEMENT OPERAND

Explanation: Critical. An invalid keyword was found in a RECORD control statement.

System action: The program terminates.

Programmer response: Check for invalid keywords.

ICE021A NO TYPE OPERAND

Explanation: Critical. RECORD TYPE was not specified for a tape work data set sort or Conventional merge application that uses one of the following:

- VSAM input
- an E15 exit that supplies all of the input records
- an E32 exit that supplies all of the input records.

System action: The program terminates.

Programmer response: Specify RECORD TYPE=F or RECORD TYPE=V as appropriate.

ICE022A RECORD FORMAT NOT F, V OR D

Explanation: Critical. The TYPE operand of a RECORD control statement contained an invalid value.

System action: The program terminates.

Programmer response: Check that the TYPE operand value on the RECORD statement is F, FB, V, VB, D, or DB. Check also for a conflict between the SORTIN/SORTOUT DCB RECFM parameter and the RECORD control statement.

ICE023A NO LENGTH OPERAND

Explanation: Critical. The input record length was not available. A RECORD statement with an L1 or L2 value is required when an E15 or E32 user exit supplies all of the input records, or when a temporary empty VSAM data set defined without a RECSZ value is used for input.

This message can also be issued if input processing is bypassed because of a previous error.

System action: The program terminates.

Programmer response: Check for:

- A missing RECORD statement
- A RECORD statement without a LENGTH operand or without an L1 value
- A missing SORTIN DD statement or SORTINnn DD statement
- A temporary empty input VSAM data set defined without a RECSZ value

ICE024A RECORD LENGTH VALUE ERROR

Explanation: Critical. An incorrect or unsupported value was associated with either:

- The LENGTH operand of a RECORD control statement, or
- The input or output length values obtained from SORTIN/SORTINnn or SORTOUT/OUTFIL respectively.

System action: The program terminates.

Programmer response: Some of the more common errors are:

- Entry errors. Length values must not contain nonnumeric characters, negative numbers, more than eight characters, a nonprintable character, and so forth.
- Specified or calculated minimum length (L4) was greater than maximum length (L2) or average length (L5).
- Average length (L5) is greater than maximum length (L2).
- No LENGTH was specified either for the operand or the logical record length on the SORTIN DD statement.
- All records are supplied by an E15 or E32 user exit and no L1 or L2 operand is specified.
- The input record length (L1), the maximum length (L2), or the output record length (L3) exceeds the maximum record length that DFSORT can handle.
- The OUTFIL input record length was greater than 32767.
- SOLRF was in effect and the OUTREC or INREC record length was greater than 32760 for fixed-length output records.
- SOLRF was in effect and the OUTREC or INREC record length was greater than 32767.
- SOLRF was in effect and the OUTREC or INREC record length was greater than 32756 for variable-length records, but RECFM=VBS was not specified for SORTOUT.

ICE025A RECORD COUNT OFF

Explanation: Critical. The program has compared the count of input records and output records (shown in message ICE054I), taken into account the number of records inserted or deleted (shown in message ICE055I), if any, and found a discrepancy.

The message is issued when the entire output data set has been written. The message is suppressed if NOCHECK was in effect and you had an E35 user exit and no SORTOUT or OUTFIL data set.

System action: The program terminates.

Programmer response: The most probable cause is that you have not specified a SORTOUT or OUTFIL data set, have specified E35, and from your E35 routine have passed a return code of 8 (do not return) too early, when there are still output records remaining. If this is the case, you can avoid receiving this message by specifying OPTION NOCHECK. If this is not the case, examine any user exit routines (especially E15 and E35) for possible return code or other errors.

If a COBOL-invoking program contains an output procedure, make sure the RETURN statement is iterated until the AT END condition is executed. This occurs the next time the RETURN statement is executed *after* the last record was returned to you.

If the iteration of the RETURN is controlled by a PERFORM statement, the PERFORM logic should be controlled by the execution of the AT END clause of the RETURN statement.

It is possible but less likely that the error was caused by an internal sort problem.

ICE026I SMF RECORD NOT WRITTEN TO THE SMF DATA SET (RC=*nn*)

Explanation: A nonzero return code was returned from system management facilities (SMF). nn is the return code from the SMFWTM macro, in decimal.

System action: The SMF type 16 record is not successfully written.

Programmer response: Have your system programmer determine whether SMF is properly initialized on your system by referring to *z/OS MVS System Management Facilities (SMF)* for the meaning of the return code from SMFWTM.

ICE027A END OF [*ddname*] FIELD BEYOND MAXIMUM RECORD LENGTH

Explanation: Critical. If ddname is blank, then:
- The OUTREC or INREC length exceeded 32767 bytes, or
- if INREC was specified, a SORT, MERGE, OUTREC, or SUM field was defined as extending beyond the maximum reformatted record length.
- otherwise, a SORT, MERGE, INREC, OUTREC, SUM, INCLUDE, or OMIT field was defined as extending beyond the maximum input record length

Otherwise, *ddname* indicates the first data set in the OUTFIL group for which either:
- an OUTFIL field was defined as extending beyond the maximum OUTFIL input record length, or
- the OUTFIL output record length exceeded 32767 bytes

System action: The program terminates.

Programmer response: Check all relevant control statements for incorrectly specified field position or length values. Check the record statement for incorrectly specified length values.

If an EFS field is specified in the SORT or MERGE control statement, make sure the position begins within the record.

If ICETOOL is used, check each ON field of the failing operator for incorrectly specified position or length values, and for a DISPLAY, OCCUR, RANGE, STATS, UNIQUE, or VERIFY operator, check that the LRECL of the FROM data set is at least 4.

ICE028A TOO MANY EXITS

Explanation: Critical. A MODS statement specified more than the maximum number of user exits.

System action: The program terminates.

Programmer response: Make sure that routines are specified for valid user exits only, and that each exit is associated with only one routine. Exits that can be specified in the MODS statement are E11, E15, E16, E17, E18, E19, E31, E35, E37, E38, E39, and E61.

Restriction: For a merge-only application you can only specify exits E31, E35, E37, E38, E39, and E61.

ICE029A IMPROPER EXIT

Explanation: Critical. This message was issued for one of the following reasons:
- An incorrect user exit was specified on a MODS control statement.
- A user exit in the sort or intermediate merge phase of the program was specified for a merge application.

System action: The program terminates.

Programmer response: Make sure that the MODS control statement does not contain errors that resulted in the specification of an invalid program user exit number. Numbers that can be specified are E11, E15, E16, E17, E18, E19, E31, E35, E37, E38, E39, and E61.

Restriction: For a merge-only application, you can only specify exits E31, E35, E37, E38, E39, and E61.

ICE030A MULTIPLY DEFINED EXITS

Explanation: Critical. This message was issued for one of the following reasons:
- A user exit was defined twice in a MODS control statement.
- An E15, E18, E35, or E39 user exit was specified both in the invoking program parameter list and in a MODS statement.

System action: The program terminates.

Programmer response: Check MODS statement for multiply defined user exits.

ICE031A INVALID MODS OP CHAR

Explanation: Critical. A parameter of a MODS control statement contained an invalid character.

System action: The program terminates.

Programmer response: Check the parameters of the MODS control statement for a length field containing something other than numeric data, a source or name field beginning with something other than an alphabetic character, or containing a special character other than $, @, or #.

ICE032A EXIT E61 REQUIRED

Explanation: Critical. A SORT or MERGE control statement defined a control field to be modified by a user-written routine (this was done by specifying E for the control field sequence indicator), and user exit E61 was not activated by a MODS control statement.

System action: The program terminates.

Programmer response: Check for conflicts in the SORT or MERGE and MODS statements with respect to E and E61.

ICE033A CONTROL FIELD SEQUENCE INDICATOR E REQUIRED

Explanation: Critical. User exit E61 was activated and no control fields were specified for user modification (an E control field sequence parameter was missing on a SORT or MERGE control statement).

System action: The program terminates.

Programmer response: Check for conflicts in the SORT or MERGE and MODS statements with respect to E and E61.

ICE034A MODS STATEMENT OPERAND ERROR

Explanation: Critical. This message was issued for one of the following reasons:
- An incorrect number of parameters followed an operand defined on a MODS control statement.
- SYSIN was specified as the third parameter for a user exit whose fourth parameter was N or C.
- For those user exits pre-link-edited by the user (fourth parameter was N or C, or null), the third parameter (library ddname) was different for two or more exits.
- An invalid value was specified for the fourth parameter of the statement.
- E15=COB or HILEVEL=YES was specified and E15 was specified with T as the fourth parameter.
- E35=COB or HILEVEL=YES was specified and E35 was specified with T as the fourth parameter.
- S was specified for the fourth parameter, but the user exit was not E11 or E31.

System action: The program terminates.

Programmer response: Make sure that:
- Each exit operand of the MODS statement is specified using correct syntax.
- SYSIN is not specified for any user exit with the fourth parameter as N or C.
- T has not been specified as the fourth parameter for an E15 or E35 user exit along with E15=COB or E35=COB, respectively, or HILEVEL=YES.
- S has not been specified for the fourth parameter for an user exit other than E11 or E31.
- The dynamic link-edit parameters (SYSIN, T, and S) have not been specified for a copy application.
- The third parameter is specified if S or T is used for the fourth parameter.
- The third parameter is specified for a conventional merge or tape work data set sort.

ICE035A DUPLICATE MODS ROUTINE OPERAND

Explanation: Critical. The same user-written routine was used for more than one user exit in a DFSORT program phase, or two or more routines had the same name.

System action: The program terminates.

Programmer response: Make sure that the MODS control statement does not use duplicate names.

ICE036I B = a

Explanation: DFSORT diagnostic message for use by IBM software support personnel.

ICE0371 G = a

Explanation: DFSORT diagnostic message for use by IBM software support personnel.

ICE038I NMAX APPROXIMATELY = *n*

Explanation: This message gives an estimate of the maximum number of records that can be sorted using the intermediate storage and main storage available to DFSORT for the current application. The number replaces n in the text of the message as shown above. Nmax is calculated assuming that 2400-foot tapes are used. For variable-length records, the value is based on maximum record length.

System action: None.

Programmer response: None.

ICE039A INSUFFICIENT MAIN STORAGE - ADD AT LEAST *n*K BYTES

Explanation: Critical. There was not enough main storage available for DFSORT to run, or main storage was fragmented.

System action: The program terminates.

Programmer response: The message gives an estimate of how much more main storage is needed. Add at least that amount to the main storage already allocated to the program by recoding the REGION parameter or the MAINSIZE/SIZE value (see *z/OS DFSORT Application Programming Guide* for ways to specify this).

This message may sometimes be issued again even after you have provided more storage because DFSORT re-estimates the amount of storage it needs at several points in the program.

Storage requirements can be reduced by decreasing either the input block size or the number of intermediate storage areas. See also messages ICE092I, ICE093I, ICE156I and ICE231I.

Make sure that storage allocation is permitted above 16 MB virtual and enough storage is available above 16 MB virtual to load the DFSORT modules.

If you are using OUTFIL processing, increase the REGION value or decrease the number of OUTFIL data sets or their block sizes; do not increase the MAINSIZE/SIZE value.

ICE040A INSUFFICIENT WORK UNITS

Explanation: Critical. There were not enough work data sets to allow your application to run. This can occur when work data sets are on tape. In a merge-only application, this message can result from incorrectly specifying one or more input data sets (SORTIN01, and so forth) or from using an E32 user exit without specifying the number of data sets.

System action: The program terminates.

Programmer response: Make sure that the DD statements do not contain errors. For a sort using tape work files, the SORTWKdd numbers must be in sequence, starting with SORTWK01, and at least three work data sets must be assigned to the program. For a Conventional merge application, make sure that the numbers for the SORTINnn DD statements are in sequence, starting with SORTIN01.

ICE041A N GT NMAX

Explanation: Critical. The exact number of records specified in a FILSZ or SIZE operand was greater than the maximum sort capacity calculated by the program.

System action: The program terminates.

Programmer response: Check all FILSZ or SIZE operands for errors. If all operands are correct, check DD statements for an error in assigning work data sets. If DD statements are correct, assign more work data sets to the program.

ICE042A UNIT ASSIGNMENT ERROR *ddname*

Explanation: Critical. This message was issued for one of the following reasons:

1. Both a VSAM SORTINnn DD statement and a non-VSAM SORTINnn DD statement were specified for a merge application.
2. Input consisted of concatenated VSAM and non-VSAM data sets.
3. A dummy, spool or pipe data set, an HFS data set, or an HFS file, was specified as a work data set.
4. If ddname is DYNALLOC, the specified device type or number of work data sets was invalid.
5. DFSORT was invoked by the ICEGENER facility, but SYSIN (or the alternate ddname for SYSIN) was not DUMMY. Control was automatically transferred to the IEBGENER system utility.
6. Both tape and disk work devices were specified.
7. An HFS file was specified for input or output, but the system does not support HFS files.
8. An HFS data set (not file) was specified for input or output.
9. QNAME was specified on an output DD statement.

System action: The program terminates.

Programmer response: For case 1, use either VSAM data sets or non-VSAM data sets for a merge application, but not both.

For case 2, use either a single VSAM data set or concatenated non-VSAM data sets for a copy or merge application.

For Case 3, remove or replace the invalid work data set.

For case 4, check that the device type specified is supported by the program and available at your site (see *z/OS DFSORT Application Programming Guide* for a list of disk devices). Also check whether you have exceeded the maximum number of areas permitted for the storage type used.

For case 5, no action is necessary.

For case 6, specify either all tape or all disk work data sets.

For Case 7, remove the HFS file or ensure that the system supports HFS files.

For Case 8, remove the HFS data set or specify an HFS file.

For case 9, replace the output DD statement containing QNAME with an output DD statement for a data set supported by DFSORT.

ICE043A INVALID DATA SET ATTRIBUTES SPECIFIED *ddname yyyyyyyy*

Explanation: Critical. ddname is the name of the DD statement in error; yyyyyyyy is the error description. One of the following conditions was encountered:

1. DD statements that defined input and output data sets contained information conflicting with each other, with information on the data set labels, or with the default values assumed for DCB subparameters by the program (see *z/OS DFSORT Application Programming Guide* for information on DCB subparameters).
2. The RECFM, LRECL or BLKSIZE was not available for an input data set.
3. A DD statement for input or output specified a cataloged disk data set that does not exist on the volume pointed to by the catalog entry.
4. Input data sets for a merge contained different record lengths.
5. Blockset was not selected for a merge, and the largest block size was not specified for SORTIN01.
6. For a tape work data set sort, the first data set of the concatenation did not specify the largest block size.
7. A tape data set with the largest block size was not first in the SORTIN concatenation, BLKSIZE was not specified in the DD statement for that tape data set, and that tape data set was not managed by DFSMSrmm™ or by a tape management system using ICETPEX.
8. The record format of the input or output data set was U (undefined).

9. For a tape work data set sort of fixed length records, the LRECL of the input or output data set was larger than 32752 (or 32748 with EQUALS in effect).
10. LRECL=X was used for an input or output data set.
11. The track capacity of the SORTOUT device was smaller than the SORTOUT block size.
12. A variable-length VSAM data set was processed with the wrong RECFM.
13. The RECFM, LRECL and BLKSIZE were not available for a DUMMY or NULLFILE input data set.

System action: The program terminates.

Programmer response: For case 1, check DD statements for input and output data sets for conflict in the BLKSIZE (block size), RECFM (record format), and LRECL (logical record length) subparameters.

If OUTFIL's VTOF or CONVERT parameter is used, an OUTFIL data set must not have a variable-length record format (for example, RECFM = VB).

If OUTFIL's FTOV parameter is used, an OUTFIL data set must not have a fixed-length record format (for example, RECFM = FB).

If OUTFIL's VTOF, CONVERT or FTOV parameters are not used, an OUTFIL data set must not have a different record type (fixed or variable) than the input data set (for example, RECFM = FB for input and RECFM = VB for output, or RECFM = VB for input and RECFM = FB for output).

When using fixed-length records without E15 or E35 user exits, the following are true:
- For a merge application, SORTOUT LRECL (or RECORDSIZE for VSAM data sets) must not exceed either the SORTINnn LRECL (RECORDSIZE for VSAM data sets) or the reformatted record length.
- For a sort application, except with Blockset, SORTOUT LRECL (or RECORDSIZE for VSAM data sets) must not exceed SORTIN LRECL (or RECORDSIZE for VSAM data sets).

When using fixed-length records for a sort application, except with Blockset, the SORTOUT LRECL (or RECORDSIZE for VSAM data sets) must not exceed the reformatted record length.

For case 2, specify the RECFM, LRECL or BLKSIZE as appropriate on the input DD statement.

For case 3, check the volume serial numbers of the input data sets.

For case 4, ensure that all input data sets have the same record length.

For case 5, check that the SORTIN01 data set has the largest block size.

For case 6, ensure that the first data set of the concatenation has the largest block size.

For case 7, ensure that BLKSIZE is specified in the DD statement for the tape data set with the largest block size.

For case 8, ensure that the record format of the input and output data sets is not U (undefined).

For case 9, either reduce the input/output LRECL size, or replace the tape work data sets with disk work data sets so the Blockset technique can be used.

For case 10, specify the actual maximum LRECL instead, or if the LRECL is greater than the maximum allowed, decrease it.

For case 11, ensure that the SORTOUT block size is appropriate for the SORTOUT device

For case 12, specify a RECORD statement with TYPE=F to process the VSAM records as fixed-length or TYPE=V to process the VSAM records as variable-length, as appropriate.

For case 13, specify the RECFM, LRECL and BLKSIZE on the input DD statement for the DUMMY or NULLFILE input data set.

ICE044I EXIT E*nn* INVALID OPTION

Explanation: An invalid input/output option was passed to DFSORT at user exit E18, E19, E38, or E39. The nn value in the above message text is replaced by the number of the user exit at which the error occurred.

System action: The invalid option is ignored.

Programmer response: Check the parameter list passed by the user-written routine against the table in Appendix A, "Valid Input/Output Options at User Exits—ICE044I," on page 127 before rerunning the application. An *x* in the table indicates an option that is allowed with the user exit in question.

ICE045I END SORT PH

Explanation: The sort (input) phase completed successfully. This message appears only when the BALN or POLY tape technique was used.

ICE046A SORT CAPACITY EXCEEDED - RECORD COUNT: *n*

Explanation: Critical. DFSORT was not able to complete processing with the intermediate storage available (Hiperspace or disk work data sets). For work data sets with secondary allocation allowed, DFSORT overrides system B37 abends and continues processing; this message is issued only when no more space is

available in Hiperspace or on any allocated work data set.

Note: DFSORT uses only the first volume of multi-volume work data sets.

The count n is either an approximation of the number of records or is the total number of records that DFSORT was able to read in before it used all of the available intermediate storage.

The amount of intermediate storage required can vary depending on many factors including:
- The amount of Hiperspace DFSORT is able to use at the time the sort is run
- The amount of main storage available
- The degree of randomness of the records to be sorted
- The values specified (or defaulted) for options such as DYNALOC, DYNAUTO, DYNSPC, FILSZ/SIZE, AVGRLEN, OR DYNALLOC..
- The amount of padding required for short records when VLSHRT is in effect.

System action: The program terminates.

Programmer response: Take one or more of the following actions:
- If appropriate, increase the amount of main storage available to DFSORT using the options MAINSIZE/SIZE or the JCL option REGION. Increasing the amount of main storage available to DFSORT can help DFSORT use less intermediate storage. Avoid running a large sort in a small amount of main storage.
- If dynamic allocation was used, ensure that the values for the options DYNALOC, DYNAUTO, DYNSPC, DYNALLOC, FILSZ/SIZE, and AVGRLEN are appropriate. If necessary, specify these options or change their values.

  If message ICE118I was received, specify FILSZ=En with a reasonably accurate estimate of the number of records to be sorted. If you cannot specify FILSZ=En, and you are using dynamically allocated work data sets, use DYNSPC=n to increase the primary space allocated.

  If your average input record length is significantly different from one-half of the LRECL, specify AVGRLEN=n with a reasonably accurate estimate of the average record length.
- If VLSHRT was in effect and the total size of all control fields was significantly larger than the average LRECL for the data set, you may be able to reduce the amount of work space required by reducing the total size of the control fields.
- If JCL work data sets were used, increase the amount of work space available to DFSORT.

ICE047A RECORD COUNT OFF, SPECIFIED *n*, RECEIVED *n*

RCD COUNT OFF

Explanation: Critical. Installation option FSZEST=NO was specified or defaulted, and the number of records received was NOT equal to the value specified in the FILSZ=x or SIZE=y option in effect.

The SPECIFIED field contains the value specified in the FILSZ=x or SIZE=y option. The RECEIVED field contains the actual number of records in the input data sets (for SIZE=y) or the actual number of records to be sorted or merged (for FILSZ=x). See the explanation of SIZE=y or FILSZ=x on the OPTION statement in the *z/OS DFSORT Application Programming Guide* for details.

System action: The program terminates.

Programmer response: Make sure that the FILSZ=x or SIZE=y value is accurate. Message ICE047A can be suppressed by specifying installation option FSZEST=YES, or by using FILSZ=Ex, SIZE=Ey, FILSZ=Ux, or SIZE=Uy, thus ensuring that the value will be treated as an estimate.

If an E15 or E32 exit passes all of the records (no SORTIN or SORTINxx), use FILSZ=x rather than SIZE=y.

ICE048I NMAX EXCEEDED

Explanation: DFSORT exceeded the calculated sort capacity while processing the input data set, and user exit E16 was specified.

System action: The user-written routine at E16 is entered.

Programmer response: No response necessary. (The number of records sorted is equal to the NMAX calculated by DFSORT. See message ICE038I.)

ICE049I SKIP MERGE PH

Explanation: DFSORT diagnostic message for use by IBM software support personnel.

ICE050I END MERGE PH

Explanation: DFSORT diagnostic message for use by IBM software support personnel.

ICE051A UNENDING MERGE

Explanation: Critical. There was not enough main storage or work data set space available for DFSORT to run with.

System action: The program terminates.

Programmer response: Assign more intermediate storage or main storage and rerun the job. Reverse

sequence files can require more space.

ICE052I END OF DFSORT

Explanation: The program completed.

System action: Returns control to the operating system or invoking program.

Programmer response: None.

ICE053A OUT OF SEQUENCE

Explanation: Critical. The current record leaving the intermediate or output phase was not in collating sequence with the last record blocked for output.

System action: The program terminates.

Programmer response: Contact IBM for programming support. If message ICE143I indicates the Blockset technique was selected, specifying OPTION NOBLKSET temporarily may allow you to bypass the DFSORT error.

ICE054I RECORDS-IN: *n*, OUT: *n*

RCD IN *n*,OUT *n*

Explanation: This message lists the number of records read by DFSORT from the input data sets, and the number of records written to SORTOUT or used as input for OUTFIL processing. The numbers replace n in the text of the message as shown above. For a Conventional merge application, if an exact file size was not specified, the IN field is zero; if an exact file size was specified, it appears in the IN field.

If DEBUG CTRx was used and Blockset was not selected, the n values could be meaningless.

For a tape work data set sort or Conventional merge application, record counts that exceed 4294967295 will be displayed as asterisks.

System action: None.

Programmer response: If you are using user exit E15 or E35 and have any reason to suspect that you are "losing" or "gaining" records, check message ICE055I. The sum of RECORDS IN plus INSERT should always be equal to the sum of RECORDS OUT plus DELETE. If it is not, you should also receive message ICE025A.

ICE055I INSERT *n*, DELETE *n*

Explanation: The number of records inserted or deleted (prior to OUTFIL processing, if used) replaces the values shown as n in the above format.

If DEBUG CTRx was used and Blockset was not selected, the n values could be meaningless.

For a tape work data set sort or Conventional merge application, record counts that exceed 4294967295 will be displayed as asterisks.

System action: None.

Programmer response: See message ICE054I.

ICE056A *ddname* NOT DEFINED

Explanation: Critical. A required DD statement was not supplied. This message appears for:

- Sort or copy applications without a SORTIN DD statement or E15 user exit
- Merge applications without a SORTINnn DD statement or E32 user exit
- Sort, merge, or copy applications without an E35 user exit, a SORTOUT DD statement or an OUTFIL data set.
- Tape work data set sort applications without a SORTIN DD statement or a SORTOUT DD statement
- The same OPTION SORTIN and OPTION SORTOUT ddname
- CKPT in effect with no work data set specified
- Conventional merge applications without a SORTIN01 DD statement (Conventional merge ignores SORTIN1) or a SORTOUT DD statement.
- A DD statement for a specified OUTFIL data set was not present.

System action: The program terminates.

Programmer response: Check the DD statements for errors.

ICE057A SORTIN NOT SORTWK01

Explanation: Critical. A work data set other than SORTWK01 was assigned to the same tape drive as SORTIN.

System action: The program terminates.

Programmer response: Check the DD statements for errors.

ICE058A SORTOUT A WORK UNIT

Explanation: Critical. SORTOUT was specified on the same tape drive as a work data set.

System action: The program terminates.

Programmer response: Check the DD statements for errors.

ICE059A RECORD LENGTH INVALID FOR {DEVICE | *ddname*}

Explanation: Critical. The intermediate record created by DFSORT from the input record was either less than 18 bytes or more than 32756 bytes when work units were tape (indicated by DEVICE) or was too large for the assigned work data set device with the indicated ddname. The error can exist for other devices in

addition to the one indicated.

System action: The program terminates.

Programmer response: If the intermediate record is too small, redefine input record length to at least 18 bytes. If the length is too large, assign work data set devices with larger track capacity.

If EQUALS is in effect, the maximum record length is reduced by 4 bytes.

ICE060I INVALID *ddname* PARAMETER

Explanation: A PARM parameter specified in the ddname data set contained an error. If a parameter was specified more than once, the first entry was used (if valid).

System action: Processing continues. Invalid and duplicate parameters are ignored.

Programmer response: No action is necessary. For later runs, be sure that the optional parameters are valid. Valid PARM parameters are described in *z/OS DFSORT Application Programming Guide*.

ICE061A I/O ERROR, DD *ddname*, DEV *address*, ECB *completion code*, CSW *status bytes*, SENSE *sense bytes*

Explanation: Critical. This message was issued for one of the following reasons:

- The job control statements incorrectly specified record length or blocking information for the data set located on the device indicated by the DEV field in the message.
- A permanent error occurred during an I/O operation on the indicated device.

The most likely causes include:

- The output data set of a merge, copy, or ICEGENER application is the same as one of the input data sets.
- An output data set is the same as another output data set.
- Hardware-related error.

System action: If no user options are specified, the program terminates.

Programmer response: Make sure that the DD statement for the data set assigned to this device contains the correct DCB information. In a merge application, if the device in error holds an input data set, make sure that the DCB information (except for BLKSIZE) specified in the SORTIN01 DD statement correctly describes the data in this device.

If ICE061A contains ECB=41 and CSW=0006, and you receive an accompanying IOS000I simulated I/O error message, you could be experiencing a missing interrupt time-out on the corresponding channel due to a large data transfer. If this is the case, either reduce the value set for the IOMAXBF installation option or raise the missing interrupt handler (MIH) value for the device associated with the error.

If the error persists, you can bypass it, in some cases, by specifying the NOBLKSET option temporarily to force DFSORT to use a different sorting technique.

ICE062A LINK-EDIT ERROR

Explanation: Critical. The linkage editor found a serious error.

System action: The program terminates.

Programmer response: Check that all the DD statements used by the linkage editor are included (SYSPRINT, SYSLIN, SYSUT1, and SYSLMOD), and that they are correct.

If the DD statements are correct, make sure that:

- All user routines in libraries are valid object decks or load modules.
- All user routines in the system input stream (SYSIN) are valid object decks.
- Modules to be link-edited together do not contain duplicate entry point names.
- A SORTLIB DD statement is specified if a tape work data set or conventional merge is used and the SORTLIB=SYSTEM installation option is specified.

ICE063A OPEN ERROR *ddname*

Explanation: Critical. An error occurred while trying to open the data set with the indicated ddname.

System action: The program terminates.

Programmer response: Check for any of the following:

- A missing or invalid DD statement
- Conflicting DCB information; for example, the records are fixed-blocked, but the block size is not a multiple of the record length
- Concatenated mixed disk/tape or tape-only input without a DCB BLKSIZE parameter for the tape data set containing the largest block size
- Concatenated, fixed-length input with different LRECL specifications
- Data sets required for dynamic link-editing cannot be opened.

ICE064A DELETE ERR

Explanation: Critical. DFSORT was unable to delete either itself or a user exit routine. This message appears only when user exit routines are used.

System action: The program terminates.

Programmer response: Make sure that the user exit routines are not modifying the DFSORT code and information areas, and rerun the job.

ICE065A PROBABLE DECK STRUCTURE ERROR

Explanation: Critical. The end of the SYSIN data set was found before all needed user exit modules were read or the end of the SYSIN data set was not found after all specified modules were read.

System action: The program terminates.

Programmer response:
- Check that the MODS statement specifies the correct routines.
- Make sure the SYSIN data set contains all user exit routines specified by the MODS statement, and no others.
- Check for misplaced job control language statements, especially preceding a user exit routine on SYSIN.
- Make sure nothing follows the last object deck in SYSIN.

ICE066I APROX RCD CNT *n*

Explanation: Sort capacity was reached. The count n is an approximation of the number of records DFSORT can handle with the available work data sets.

System action: The program terminates.

Programmer response: Respond as indicated in the accompanying message, ICE046A.

ICE067I INVALID PARAMETER IN JCL EXEC PARM OR INVOKED PARMLIST

Explanation: An error was found in the PARM field parameters of the EXEC statement, or in the optional parameters of the parameter list passed to DFSORT using ATTACH, LINK, or XCTL. If a parameter was specified more than once, the first entry was used (if valid).

System action: Processing continues. Invalid and duplicate parameters are ignored.

Programmer response: No action is necessary. For later runs, be sure that the optional parameters are valid. Valid parameters are described in *z/OS DFSORT Application Programming Guide*.

ICE068A OUT OF SEQUENCE SORTIN*nn*

OUT OF SEQ SORTIN*nn*

Explanation: Critical. During a merge, a data set was found to be out of sequence.

If Blockset was selected:

- nn is the data set identification (00 to 99).
- If input was supplied through user exit E32, then 00 signifies the first input file, 01 the second, and so on.

If Blockset was not selected:

- nn is the data set identification (01 to 16).
- If input was supplied through user exit E32, then 01 signifies the first input file, 02 the second, and so on.

System action: The program terminates.

Programmer response: If a user-written routine was modifying the records, check the routine thoroughly. It should not modify control fields at user exit E35. If a user-written routine is not being used, make sure that all input data sets have been sorted on the same control fields, in the same order, and that they all have a similar format. Check whether you have also received message ICE072I.

If input was supplied through user exit E32, check your routine to make sure records are passed to the merge from the correct file.

If you were reading in variable-length VSAM records through user exit E32, check the format and accuracy of the record descriptor word (RDW) you built at the beginning of each record.

If Blockset was not selected, rerun the job with a SORTDIAG DD statement to get message ICE800I, which indicates the reason Blockset could not be used. If possible, remove the condition preventing the use of Blockset.

ICE069A INVALID SIGN

Explanation: Critical. For a tape work data set sort or conventional merge, the first byte of signed numeric data with leading separate sign (that is, CSL or ASL data), or the last byte of signed numeric data with trailing separate sign (that is, CST or AST), did not contain a valid sign character of '+' (plus) or '–' (minus).

System action: The program terminates.

Programmer response: Rerun the job with a SORTDIAG DD statement to get message ICE800I, which indicates the reason Blockset could not be used. If possible, remove the condition preventing the use of Blockset. Blockset treats a '–' (minus) sign as negative and any other sign (for example, '+' (plus) or blank) as positive. If you must use tape work data sets or conventional merge, use '+' for all of the positive signs in your CSL, CST, ASL and AST data.

ICE070A EXIT *xxxxxxxx* NOT FOUND

Explanation: Critical. This message was issued for one of the following reasons:
1. If xxxxxxxx is CSECT, the CSECT name of the user exit specified for a dynamically linkedited user exit was incorrect.
2. If xxxxxxxx is not CSECT, it is a member name specified in the MODS statement that could not be found in the appropriate library data set.

System action: The program terminates.

Programmer response: For case 1, change the CSECT name to be the same as the member name specified in the MODS statement.

For case 2:
- If you specified an exit ddname for the member, make sure the member is present in the library data set with that ddname.
- If you did not specify an exit ddname for the member, make sure the member is present in a library identified by the STEPLIB/JOBLIB or link library.

ICE071A INVALID RETURN CODE FROM EXIT E*nn* [RC=*x*, CASE=*id*]

Explanation: Critical. An E15, E16, E32, or E35 user exit passed back an invalid return code to DFSORT. The user exit can be part of an invoking program, such as a COBOL input or output procedure.

nn is the number of the user exit that passed back return code *x*.

x is the return code passed to DFSORT from the user exit; it prints as the decimal equivalent of the 32-bit value in register 15.

id is an identifier that helps you determine why return code *x* is invalid and what the valid return codes are for the situation.

System action: The program terminates.

Programmer response: Check the exit number (*nn*), return code (*x*), and case identification (*nn*) against the table in Appendix B, "User Exit Return Codes—ICE071A," on page 129. Change your user exit to pass an appropriate valid return code (that is, one marked with an x in the table) for the identified situation. As shown in the table, ICEMAC option EXITCK=WEAK can be used to avoid this message and continue processing for certain invalid return codes from the E15 and E35 user exits.

In a COBOL-invoking program containing an input or output procedure, check that control passes through the end of the input procedure before the output procedure is entered, and through the output procedure before the sort is ended.

Also check that the user exit did not overlay register 15, for example, by inadvertently including the register in a load multiple (LM) instruction.

For complete details of how return codes are interpreted in various situations for the E15, E16, E32, and E35 user exits, see the chapter on using your own user exit routines in the *z/OS DFSORT Application Programming Guide*. In particular, see the detailed tables in the section on the E15/E35 return codes and EXITCK.

ICE072I FIELD NOT WITHIN MINIMUM RECORD LENGTH

Explanation: A RECORD statement specified a minimum record length (L4) that cannot contain all fields specified in the SORT, MERGE, INCLUDE, OMIT, INREC, OUTREC, or SUM statements.

System action: The L4 value is adjusted. Processing continues.

Programmer response: None.

ICE073A VARIABLE RECORD TOO LONG

Explanation: Critical. This message was issued for one of the following reasons:
1. A variable length record was larger than the maximum length specified or defaulted.
2. The sum of the record length and the control field lengths was larger than 32767 bytes.
3. For a tape work data set sort, a variable length record was larger than 32752 bytes (or 32748 with EQUALS in effect).
4. OPTCD=Q was not specified on the DD statement for an ASCII tape input data set.
5. An invalid RDW or BDW was found.

System action: The program terminates.

Programmer response: For case 1, check the input both at E15 and E35, if used, and in SORTIN. Then delete the extra long records or increase the SORTIN or SORTOUT DD statement DCB LRECL value or the RECORD statement L1, L2, or L3 value, as appropriate.

If you are using INREC or OUTREC, check that a SORTOUT DD LRECL value or L3 value is at least as large as the reformatted output record length.

If you have VSAM records, remember that they are increased in length by the 4-byte record descriptor word (RDW) added when they enter DFSORT. If you are inserting or modifying records through E15, check the format of the RDW you are building at the beginning of each record.

For case 2, if the Blockset technique was selected, you may be able to run the application by specifying OPTION NOBLKSET.

You can use the ICETOOL STATS operator with ON(VLEN) to show the length of the longest variable length record. You can use the ICETOOL DISPLAY operator with ON(VLEN) to show the length of each variable length record.

For case 3, either reduce the record lengths, or replace the tape work data sets with disk work data sets so the Blockset technique can be used.

For case 4, ensure that OPTCD=Q is specified on all ASCII tape input DD statements.

For case 5, ensure that all RDWs and BDWs are valid.

ICE074I RECORD LENGTH L1 OR L3 OVERRIDDEN

Explanation: The L1 value for the LENGTH parameter of the RECORD statement was not the same as the LRECL value for SORTIN or SORTIN01, or the L3 value was not the same as the SORTOUT LRECL value. For VSAM, the equivalent of LRECL is maximum RECSZ.

System action: Processing continues with the L values overridden.

Programmer response: For subsequent runs, check all the record lengths. Take special note of the L2 value. If you did not specify one, it defaulted to the value you specified for L1 (and was not overridden by the LRECL value). If the L2 value was too small, it could have caused your application to terminate at any of a number of points, and the error can be difficult to detect.

The input record type and maximum record length is shown in message ICE088I or ICE089I. If you have variable-length input records, check that the L1 value specified is actually a maximum.

The output record type and maximum record length is shown in message ICE090I or ICE091I. If you have variable-length output records, check that the L3 value specified is actually a maximum.

ICE075A VSAM CB ERROR (*nn*) AT *aaaaaa*

Explanation: Critical. *aaaaaa* represents the storage address of the control block at which the error was detected. *nn* is the VSAM return code, in decimal, from a VSAM macro.

System action: The program terminates, unless the error is detected during close, when the program tries to close all remaining VSAM data sets before terminating.

Programmer response: Refer to *z/OS DFSMS Macro Instructions for Data Sets* for the meaning of the error code and, if possible, take appropriate action.

ICE076A VSAM INPUT ERROR *i* (*nnn*) *ddname*

Explanation: Critical. i is either P (physical) or L (logical), describing the type of error encountered for the data set with the indicated ddname. nnn is the VSAM feedback code (RPLERRCD) from a GET macro, in decimal.

System action: The program terminates.

Programmer response: Refer to *z/OS DFSMS Using Data Sets* or *z/OS DFSMS Macro Instructions for Data Sets* for the meaning of the error code and, if possible, take appropriate action.

ICE077A VSAM OUTPUT ERROR *i* (*nnn*) [*ddname*]

Explanation: Critical. *i* is either P (physical) or L (logical), describing the type of error encountered. *nnn* is the VSAM feedback code (RPLERRCD) from a PUT macro, in decimal. The *ddname* of the data set in error is given, if available.

System action: The program terminates.

Programmer response: Refer to *z/OS DFSMS Using Data Sets* or *z/OS DFSMS Macro Instructions for Data Sets* for the meaning of the error code and, if possible, take appropriate action.

A VSAM logical error, reason code 8, can occur for a non-empty KSDS, RRDS, or VRRDS output data set if NORESET is in effect, or if the data set is defined without the REUSE option.

A VSAM logical error, reason code 12, can occur if the output data set is a KSDS and either the VSAM keys are not in ascending binary sequence or there are duplicate keys. If so, ensure that the keys are in ascending binary sequence and that there are no duplicate keys.

Refer to *z/OS DFSORT Application Programming Guide* for more information about VSAM data sets.

ICE078A VSAM OPEN ERROR (*nnn*) *ddname*

Explanation: Critical. *nnn* is the VSAM OPEN ERROR return code (ACBERFLG) in decimal, for the data set with the indicated *ddname*.

System action: The program terminates.

Programmer response: Refer to *z/OS DFSMS Macro Instructions for Data Sets* for the meaning of the error code and, if possible, take appropriate action.

Some of the more common reasons for a VSAM OPEN error are as follows:

- An empty data set is used for input, but NVSAMEMT is in effect.
- Concatenated VSAM data sets are used.
- The same VSAM data set defined without REUSE is used for both input and output, but RESET is in effect.

ICE079A VSAM CLOSE ERROR (*nnn*) *ddname*

Explanation: Critical. *nnn* is the VSAM CLOSE ERROR return code (ACBERFLG), in decimal, for the data set with the indicated *ddname*.

System action: The program terminates.

Programmer response: Refer to *z/OS DFSMS Using Data Sets* or *z/OS DFSMS Macro Instructions for Data Sets* for the meaning of the error code and, if possible, take appropriate action.

ICE080I IN MAIN STORAGE SORT

Explanation: All records were sorted in main storage or data space. Work data sets were not used. Hiperspace was not used.

System action: None.

Programmer response: None.

ICE081A FIRST COMMUNICATION AREA SPACE EXCEEDED

Explanation: Critical. The complexity of the application caused dynamic areas and routines to exceed the total fixed amount of storage allowed for them in the first communication area. This situation normally occurs due to specification of one or more of the following (in approximate order of importance):

- A large number of SORT, MERGE, INREC, or OUTREC fields
- Use of an E61 user exit
- Use of the VLSHRT option
- Use of the NOSZERO option
- Use of an ICETEXIT routine
- Very large records.
- An EFS program requesting statements with a large number of fields.

System action: The program terminates.

Programmer response: Reduce the complexity of the application. For example, reduce the number of SORT fields or specify OPTION NOVLSHRT, if possible.

ICE082I CHECKPOINT CANCELLED

Explanation: No more work data set tracks were available; therefore, the tracks allocated for CKPT were given back to the sort work data sets.

System action: The program continues, but no checkpoints are taken.

Programmer response: Increase the work space allocation for the next run.

ICE083A RESOURCES WERE UNAVAILABLE FOR DYNAMIC ALLOCATION OF WORK DATA SETS (*xxxx*)

Explanation: Critical.

- If *xxxx* is 064K:

  The work space needed for each work data set exceeds the limit of 64K tracks for a single, physical data set. Dynamic allocation could not be attempted.
- If *xxxx* is not 064K:

  DFSORT attempted to dynamically allocate work data sets using the DYNALOC, DYNSPC, and DYNALLOC values in effect. Dynamic allocation failed; xxxx indicates the failure code returned by the system's dynamic allocation facility. See *Application Development Guide for Authorized Programs* for an explanation of the failure code.

  Two commonly received return codes are:

  0218 The device type is defined to the system, but the requested device has insufficient space available. For example, the device type is defined to the system but no devices of that type have been installed, or all devices of that type are mounted as private.

  021C The device type is not defined to the system. For example, SYSDX was specified instead of SYSDA.

  However, in a DFSMS environment, the return code is usually 970C regardless of the specific error encountered.

System action: The program terminates.

Programmer response:

- If *xxxx* is 064K:

  Use the DYNALLOC=(,n) parameter to increase the maximum number of work data sets (n) such that the work space needed for each work data set does not exceed 64K tracks.
- If *xxxx* is not 064K:

  Ensure that work data sets can be dynamically allocated.

If message ICE118I was received, specify FILSZ=En with a reasonably accurate estimate of the number of records to be sorted. If you cannot specify FILSZ=En, use DYNSPC=n to decrease the primary space allocated.

ICE084I {BSAM|EXCP|VSAM} ACCESS METHOD USED FOR *ddname*

Explanation: This message identifies the access method used for the identified data set.

System action: None.

Programmer response: None, unless you have any problems reading or writing the data set. If EXCP was used, you can force DFSORT not to use EXCP by specifying BSAM on the DEBUG control statement or as an EXEC PARM.

ICE085A SECOND COMMUNICATION AREA SPACE EXCEEDED - ADD AT LEAST *n*K BYTES TO REGION

Explanation: Critical. The complexity of the application caused dynamic areas and routines to exceed the total fixed amount of storage allowed for them in the second communication area. This situation normally occurs due to specification of one or more of the following (in approximate order of importance):

- A large number of SORT, MERGE, INREC, or OUTREC fields
- Use of an E61 user exit
- Use of the VLSHRT option
- Use of the NOSZERO option
- Use of an ICETEXIT routine
- A large number of SUM fields.
- A large number of OUTFIL fields.

System action: The program terminates.

Programmer response:
- If n is not zero, increase the REGION value by at least n KB, and ensure that the MAINSIZE value in effect is at least 200 KB.
- If n is zero and the MAINSIZE value in effect is less than 200 KB, increase the MAINSIZE value to at least 200 KB.
- If n is zero and the MAINSIZE value in effect is at least 200 KB, reduce the complexity of the application. For example, reduce the number of sort fields or specify OPTION NOVLSHRT, if possible.

ICE088I *jobname.stepname.procstepname*, **INPUT LRECL =** *n*, **BLKSIZE =** *n*, **TYPE =** *t*

Explanation: Gives current job name, step name, procedure step name, and information related to the input records when SORTIN or SORTINnn is present:
- LRECL is the largest logical record length of the input data sets.
- BLKSIZE is the largest block size (non-VSAM) or largest control interval size (VSAM) used by DFSORT.
- TYPE is the record format of one of the input data sets, and can be one or more of the following:
  - F - fixed-length input records
  - V - EBCDIC or ASCII variable-length input records
  - B - blocked records
  - S - standard for fixed-length records or spanned for variable-length records
  - A - ANSI control characters.

System action: None.

Programmer response: None.

ICE089I *jobname.stepname.procstepname*, **INPUT LRECL =** *n*, **TYPE =** *t*

Explanation: Gives the current job name, step name, procedure step name, and information related to the input records when all input is supplied by user exit E15 or E35 or when input is VSAM and DFSORT cannot obtain the control interval size:
- LRECL is the logical record length of the input records.
- TYPE is F for fixed-length input records, or V for EBCDIC or ASCII variable-length input records.

System action: None.

Programmer response: None.

ICE090I **OUTPUT LRECL =** *n*, **BLKSIZE =** *n*, **TYPE =** *t* **[(SDB)]**

Explanation: Gives information related to the output records when SORTOUT is present:
- LRECL is the logical record length of the output records.
- BLKSIZE is the block size (non-VSAM) or control interval size (VSAM) of the output data set.
- TYPE is the record format of the output data set and can be one or more of the following:
  - F - fixed-length output records
  - V - EBCDIC or ASCII variable-length output records
  - B - blocked records
  - S - standard for fixed-length records or spanned for variable-length records
  - A - ANSI control characters.
- (SDB) indicates that the system-determined optimum block size was used.

System action: None.

Programmer response: None.

ICE091I **OUTPUT LRECL =** *n*, **TYPE =** *t*

Explanation: Gives information related to the output records when neither SORTOUT nor an OUTFIL data set is present (that is, when all output records are deleted by user exit E35):
- LRECL is the logical record length of the output records.
- TYPE is F for fixed-length output records or V for EBCDIC or ASCII variable-length output records.

System action: None.

Programmer response: None.

ICE092I **MAIN STORAGE = (***x***,***y***,***z***)**

Explanation: Gives information related to the use of main storage for this DFSORT application:

x — The main storage value specified (subject to the SIZE/MAINSIZE limit), or supplied by default.

y — The main storage theoretically available to DFSORT, considering the MINLIM value specified when the program was installed.

z — The main storage actually available to DFSORT, after any other program took what it needed from the region (invoking program or exit routines).

The value for z is not meaningful if the storage taken by the invoking program or user exit is greater than REGION or SIZE/MAINSIZE.

The x, y, and z references to main storage apply to the total storage above and below 16 MB virtual.

System action: None.

Programmer response: None, unless DFSORT subsequently terminated abnormally. In that case, check the z value to see how much storage was really available to DFSORT. If space was the problem, you have probably also received message ICE039A. If storage was heavily fragmented, the result might instead be a system 80A abend in either DFSORT or one of your own routines.

You can need considerably more than the normal minimum if the region is fragmented.

ICE093I MAIN STORAGE = (MAX,*y*,*z*)

Explanation: Gives information related to the use of main storage for this DFSORT application:

MAX The value MAX was in effect.

y The main storage theoretically available to DFSORT, considering the TMAXLIM or MAXLIM values specified when the program was installed.

z The main storage actually available to DFSORT, after any other program took what it needed from the region (invoking program or user exit routines). RESALL and RESINV are not considered.

The value for z is not meaningful if the storage taken by the invoking program or user exit is greater than REGION or SIZE/MAINSIZE.

The y and z references to main storage apply to the total storage above and below 16 MB virtual.

System action: None.

Programmer response: None, unless DFSORT subsequently terminated abnormally. In that case, check the z value to see how much storage was really available to DFSORT. If space was the problem, you probably also received message ICE039A. If storage was heavily fragmented, the result might instead be a system 80A abend in either DFSORT or one of your own routines.

You can need considerably more than the normal minimum if the region is fragmented.

ICE094I SMF RECORD NOT WRITTEN

Explanation: A DFSORT SMF record was requested, but system management facilities (SMF) was not active in the system.

System action: Writing of the SMF record to the DFSORT SMF data set is bypassed.

Programmer response: Have your system programmer determine if SMF is properly initialized on your system.

ICE095A INVALID OPTION STATEMENT OPERAND

Explanation: Critical. An OPTION control statement contained an invalid keyword operand.

System action: The program terminates.

Programmer response: Make sure that the OPTION control statement does not contain an invalid keyword operand. For valid keywords, refer to *z/OS DFSORT Application Programming Guide*.

ICE096I SUCCESSFUL RECOVERY FROM B37 ABEND(S) FOR WORK DATA SET(S)

Explanation: DFSORT successfully recovered from one or more system B37 abends. These abends occurred when DFSORT attempted to allocate more disk space than was available on one or more work data sets.

System action: Processing continues.

Programmer response: None.

ICE097I SORT RECOVERING FROM B37 ABEND ON SORTWK DATASET

Explanation: A system B37 abend occurred when DFSORT attempted to allocate more disk space than was available on a work data set. This message is issued only to the master console.

System action: Processing continues.

Programmer response: None.

ICE098I AVERAGE RECORD LENGTH = *n* BYTES

Explanation: *n* is the number of bytes in the variable-length records (including the record descriptor word) divided by the number of sorted records. The number of sorted records includes all records received, added, or deleted before the E35 user exit was entered.

System action: None.

Programmer response: None.

ICE099A BLDL FAILED FOR *ddname* DATA SET

Explanation: Critical. An error was detected during member verification when the identified data set was defined as a member of a partitioned data set or a PDSE. The BLDL system function was used for verification. The most likely cause is that the member was not found in the data set.

System action: The program terminates.

Programmer response: Ensure that the specified partitioned data set or PDSE member exists.

ICE101A *xxxxxxxxxxx* **STATEMENT NOT SUPPORTED FOR TECHNIQUE USED**

Explanation: Critical. The control statement indicated by xxxxxxxxxxx is not supported for the technique used (tape work data set sort or Conventional merge). xxxxxxxxxxx is replaced by INCLUDE/OMIT, SUM, INREC, or OUTREC.

System action: The program terminates.

Programmer response: Rerun the job with a SORTDIAG DD statement to get message ICE800I. ICE800I indicates the reason Blockset could not be used. If possible, remove the condition preventing the use of Blockset. Alternatively, remove the indicated statement.

ICE102A MISSING COND OPERAND DEFINER

Explanation: Critical. An INCLUDE or OMIT control statement did not contain a logical expression definition.

System action: The program terminates.

Programmer response: Check the INCLUDE or OMIT control statement for lack of a logical expression definition (COND operand).

ICE103A INVALID INCLUDE OR OMIT STATEMENT OPERAND

Explanation: Critical. An INCLUDE or OMIT control statement contained an invalid keyword operand.

System action: The program terminates.

Programmer response:
- Make sure that the INCLUDE or OMIT control statement does not contain an invalid keyword operand.
- Make sure that the D2 format is not specified in the FORMAT=f operand.

ICE104A INVALID INREC OR OUTREC STATEMENT OPERAND

Explanation: Critical. An INREC or OUTREC control statement contained an invalid keyword operand.

System action: The program terminates.

Programmer response: Make sure that the INREC or OUTREC control statement does not contain an invalid keyword operand.

ICE105A INVALID SORT, MERGE, OR SUM STATEMENT OPERAND

Explanation: Critical. A SORT, MERGE, or SUM control statement contained an invalid keyword operand.

System action: The program terminates.

Programmer response:
- Make sure that the SORT, MERGE, or SUM control statement does not contain an invalid keyword operand.
- Make sure that FIELDS=COPY and FORMAT=f are not both specified for the SORT or MERGE control statement.
- Make sure that FIELDS=NONE and FORMAT=f are not both specified for the SUM control statement.
- Make sure that the D1 format is not specified in the FORMAT=f operand.

ICE106A DUPLICATE INCLUDE OR OMIT STATEMENT OPERAND

Explanation: Critical. An INCLUDE or OMIT control statement contained a duplicate keyword operand.

System action: The program terminates.

Programmer response: Check the INCLUDE or OMIT control statement for a duplicate keyword operand.

ICE107A DUPLICATE, CONFLICTING, OR MISSING INREC OR OUTREC STATEMENT OPERAND

Explanation: Critical. One of the following errors was found in an INREC or OUTREC statement:

- A PARSE, FIELDS, BUILD, or OVERLAY operand was specified twice. Example:

```
INREC FIELDS=(5,4,C'***',40:X),FIELDS=(1,60)
```

- PARSE and IFTHEN, FIELDS and BUILD, FIELDS and OVERLAY, FIELDS and IFTHEN, BUILD and OVERLAY, BUILD and IFTHEN, or OVERLAY and IFTHEN were specified. Example:

```
OUTREC BUILD=(1,20),OVERLAY=(10:C'A')
```

- IFOUTLEN and PARSE, IFOUTLEN and BUILD, IFOUTLEN and FIELDS, or IFOUTLEN and OVERLAY were specified. Example:

```
INREC OVERLAY=(21:C'A'),IFOUTLEN=50
```

- For an IFTHEN clause, WHEN was not specified. Example:

```
OUTREC IFTHEN=(OVERLAY=(10:C'A'))
```

- For an IFTHEN clause, WHEN=INIT, WHEN=(logexp), or WHEN=NONE was specified without PARSE, BUILD, or OVERLAY. Example:

```
INREC IFTHEN=(WHEN=(5,1,CH,EQ,C'1'),HIT=NEXT)
```

- For an IFTHEN clause, WHEN=(logexp), WHEN=ANY, or WHEN=NONE was specified with PARSE, but without BUILD or OVERLAY. Example:

```
INREC IFTHEN=(WHEN=NONE,
      PARSE=(%01=(FIXLEN=5,ENDBEFR=BLANKS)))
```

- An IFTHEN clause with WHEN=INIT was preceded by an IFTHEN clause with WHEN=(logexp), WHEN=ANY or WHEN=NONE. Example:

```
OUTREC IFTHEN=(WHEN=(5,2,CH,EQ,C'AA'),
               OVERLAY=(10:C'A')),
       IFTHEN=(WHEN=INIT,BUILD=(1,80))
```

- An IFTHEN clause with WHEN=NONE was followed by an IFTHEN clause with WHEN=INIT, WHEN=(logexp), or WHEN=ANY. Example:

```
INREC IFTHEN=(WHEN=NONE,OVERLAY=(10:C'A')),
      IFTHEN=(WHEN=ANY,BUILD=(1,80))
```

- The first IFTHEN clause with WHEN=ANY was not preceded by an IFTHEN clause with WHEN=(logexp). Example:

```
OUTREC IFTHEN=(WHEN=INIT,OVERLAY=(10:C'A')),
       IFTHEN=(WHEN=ANY,BUILD=(1,80))
```

- An IFTHEN clause with WHEN=ANY and without HIT=NEXT was followed by an IFTHEN clause with WHEN=ANY. Example:

```
OUTREC IFTHEN=(WHEN=(5,1,CH,EQ,C'1'),
         OVERLAY=(10:C'A'),HIT=NEXT),
       IFTHEN=(WHEN=(5,1,CH,EQ,C'2'),
         OVERLAY=(10:C'B'),HIT=NEXT),
       IFTHEN=(WHEN=ANY,
         OVERLAY=(28:C'ABC')),
       IFTHEN=(WHEN=ANY,BUILD=(1,80))
```

System action: The program terminates.

Programmer response: Check the INREC or OUTREC control statement for the errors indicated in the explanation and correct the errors.

ICE108A DUPLICATE SUM STATEMENT OPERAND

Explanation: Critical. A SUM control statement contained a duplicate keyword operand.

System action: The program terminates.

Programmer response: Check the SUM control statement for a duplicate keyword operand.

ICE109A SUM FIELD DISPLACEMENT OR LENGTH VALUE ERROR

Explanation: Critical. A sum field definition on a SUM control statement contained an invalid length or displacement (position) value.

System action: The program terminates.

Programmer response: Make sure that the length and position values in the FIELDS operand of the SUM control statement were specified correctly. For BI and FI, length must be 2, 4, or 8 bytes; for PD, length must be 1 through 16 bytes; for ZD, length must be 1 through 31 bytes; for FL, length must be 4, 8 or 16 bytes. Make sure that the position plus length of each field does not exceed 32753.

ICE110I VERIFY NOT USED WITH SUM

Explanation: A SUM control statement was specified and VERIFY was in effect, but the Blockset technique was not selected. VERIFY with SUM is not supported by the other techniques.

System action: Processing continues with NOVERIFY in effect.

Programmer response: Rerun the job with a SORTDIAG DD statement to get message ICE800I. ICE800I indicates the reason Blockset could not be used. If possible, remove the condition preventing the use of Blockset.

ICE111A REFORMATTING FIELD ERROR

Explanation: Critical. The FIELDS, BUILD, OVERLAY, IFTHEN BUILD or IFTHEN OVERLAY operand of an INREC or OUTREC statement, or the OUTREC, BUILD, OVERLAY, IFTHEN BUILD or IFTHEN OVERLAY operand of an OUTFIL statement, contained an invalid column, separator, position, length, keyword, pattern, sign, constant or value. Some common errors are:

- A 0 value was used.
- A null value was used where it was not permitted.
- A null string, pattern, or sign was used.
- A column was greater than 32752, or was followed by another column.
- A position plus length was greater than 32753.
- DATE=(abcd) or DATENS(abc) was specified with a, b or c not M, D, Y or 4, with M, D, Y or 4 specified more than once, or with Y and 4 both specified.
- YDDD=(abc) or YDDDNS=(ab) was specified with a or b not D, Y or 4, with D, Y or 4 specified more than once, or with Y and 4 both specified.
- TIME=(abc) or TIMENS=(ab) was specified with ab not 12 or 24.
- The length (m for p,m or FIXLEN=m for %nn) for a hexadecimal field was greater than 16376.
- A repetition factor was greater than 4095 for a separator, or a character or hex constant was longer than 256 bytes.
- An invalid digit or an odd number of digits was specified for a hexadecimal constant.
- The length (m for p,m or FIXLEN=m for %nn) for a Y2 format field was not 2 for Y2C, Y2Z, Y2P, Y2S or Y2PP, or 1 for Y2D, Y2B or Y2DP, or 3–6 for Y2T, Y2W, Y2TP or Y2WP, or 2–3 for Y2U, Y2X, Y2UP or Y2XP, or 3–4 for Y2V, Y2Y, Y2VP or Y2YP.
- The length (m for p,m or FIXLEN=m for %nn) for an edit field was less than 2 or greater than 8 for PD0.
- The length (m for p,m or FIXLEN=m for %nn) for an edit field was greater than 8 for BI or FI, 16 for PD, 31 for ZD, 32 for CSF/FS or 44 for UFF or SFF.

- The length (m for p,m or FIXLEN=m for %nn) for an edit field was not 4 for DT1, DT2, DT3, TM1, TM2, TM3, or TM4.
- The length (m for p,m or FIXLEN=m for %nn) for an edit field was not 8 for DC1, DC2, DC3, TC1, TC2, TC3, TC4. DE1, DE2, DE3, TE1, TE2, TE3, or TE4.
- The length (m for p,m or FIXLEN=m for %nn) for an edit field was not 4 or 8 for FL.
- %nn,H or %nn,F or %nn,D was specified.
- More than 31 digits or 44 characters were specified in an edit pattern.
- SIGNz (where z is not S) was specified with Mn or without EDIT or EDxy.
- x, y, or z in EDxy or SIGNz were the same character.
- The value for LENGTH was greater than 44.
- NOMATCH was specified after p,m or %nn rather than after CHANGE.
- The length for a lookup input field was greater than 64 with character or hexadecimal find constants, or greater than 1 with find bit constants.
- The length for a lookup output field was greater than 64.
- The length for a find constant was greater than the lookup input field length.
- A find constant was not a character, hexadecimal, or bit constant.
- The length for a set constant or set field (m for p,m or FIXLEN=m for %nn) was greater than the lookup output field length.
- An invalid character was specified in a find bit constant, or the number of bits for a find bit constant was not 8.
- A set constant was not a character or hexadecimal constant.
- The length for a sequence number was greater than 16.
- The value for START was greater than 100000000000.
- The value for INCR was greater than 10000000.
- The length (m for p,m or FIXLEN=m for %nn) for a RESTART field was greater than 256 bytes.
- A position without length (p without m) was specified for OVERLAY or IFTHEN OVERLAY.
- A / was specified for OVERLAY or IFTHEN OVERLAY.

System action: The program terminates.

Programmer response: Correct the invalid value.

ICE112I EQUALS NOT USED WITH SUM

Explanation: A SUM control statement was specified and EQUALS was in effect, but the Blockset technique was not selected. EQUALS with SUM is not supported by the other techniques.

System action: Processing continues with NOEQUALS in effect.

Programmer response: Rerun the job with a SORTDIAG DD statement to get message ICE800I. ICE800I indicates the reason Blockset could not be used. If possible, remove the condition preventing the use of Blockset.

ICE113A COMPARISON FIELD ERROR

Explanation: Critical. The COND operand of an INCLUDE or OMIT statement, or the INCLUDE or OMIT operand of an OUTFIL statement, or the IFTHEN WHEN operand of an INREC, OUTREC, or OUTFIL statement, contained an invalid position, length, format, constant, or mask.

System action: The program terminates.

Programmer response: Make sure that all fields and constants are specified correctly. Make sure that the position plus the length of each field does not exceed 32753. Make sure that a shift-out (X'0E') is not embedded within double-byte data in a character constant.

ICE114A INVALID COMPARISON

Explanation: Critical. One of the following situations exists:

- The COND operand of an INCLUDE or OMIT statement, or the INCLUDE or OMIT operand of an OUTFIL statement, or the IFTHEN WHEN operand of an INREC, OUTREC, or OUTFIL statement, contained an invalid field-to-field, field-to-mask, or field-to-constant comparison, or used a field with NUM that was not FS, CSF, PD, or ZD.
- FORMAT=SS was specified after COND in an INCLUDE or OMIT statement.
- Locale processing was required, but a character (CH) field to binary (BI) field comparison was specified. Locale processing does not support the comparison of a CH to BI field.

System action: The program terminates.

Programmer response: Make sure that all of the comparisons are valid. If you specified FORMAT=SS after COND in an INCLUDE or OMIT statement, respecify it before COND. DFSORT's locale processing might eliminate the need for CH to BI comparisons. See *z/OS DFSORT Application Programming Guide* for information relating to locale processing. If a CH to BI comparison is needed, specify run time option LOCALE=NONE.

ICE115A INSUFFICIENT MAIN STORAGE

Explanation: Critical. Storage was fragmented, the reserved storage value was too large, or the sizes of the user exits were too large compared to the total storage available to DFSORT.

Main storage refers to the storage *below* 16 MB virtual.

System action: The program terminates.

Programmer response: Add more main storage in 32 KB increments by recoding the REGION parameter or the MAINSIZE/SIZE value. See *z/OS DFSORT Application Programming Guide* for ways to specify MAINSIZE/SIZE.

If routines are used at user exits, their size should be added to this minimum value. For efficient sorting, allow at least 50 percent more storage than the minimum required.

This message can be issued again even after you have provided more storage because DFSORT re-estimates the amount of storage it needs at several points in the program.

Storage requirements can be reduced by decreasing either the input block size or the number of work data sets. See also message ICE092I or ICE093I.

ICE116A SORT CAPACITY EXCEEDED

Explanation: Critical. DFSORT was not able to complete processing with the available disk or tape work data sets. For disk work data sets with secondary allocation allowed, DFSORT overrode system B37 abends and continued processing; this message was issued only when no space was available on any allocated work data set.

Note: DFSORT uses only the first volume of multi-volume work data sets.

The amount of work data set space required can vary depending on many factors, including:
- The amount of main storage available
- The degree of randomness of the records to be sorted
- The values specified (or defaulted) for options such as FILSZ/SIZE and DYNALLOC.

System action: The program terminates.

Programmer response: Take one or more of the following actions:
- If appropriate, increase the amount of main storage available to DFSORT by using the options MAINSIZE/SIZE or the JCL option REGION. Increasing the amount of main storage available to DFSORT can help DFSORT use less work data set space. Avoid running a large sort in a small amount of main storage.
- If dynamic allocation was used, check that the values for options DYNALLOC and FILSZ/SIZE were appropriate. If necessary, specify these options.
- If JCL disk work data sets were used, increase the amount of disk work space available to DFSORT.
- If tape work data sets were used, check that all reels contained full-length tapes. (A bad tape may appear short because of many write errors.) If all reels contained full-length tapes, rerun the application and specify more work data sets.

ICE117A I/O ERROR, *jobname*, *stepname*, *unit address*, *device type*, *ddname*, *operation attempted*, *error description*, *last seek address or block count*, *access method*. **(SYNADAF)**

Explanation: Critical. This message was issued for one of the following reasons:
- The job control statements incorrectly specified record length or blocking information for the data set located on the device indicated by the "unit address" field in the message.
- A permanent error occurred during an I/O operation on the indicated device.

The most probable cause is a hardware-related error.

System action: If no user options are specified, the program terminates.

Operator response: If the "error description" field in the message does not contain

```
"WRNG. LEN. RECORD"
```

run the job again with the indicated unit offline, using an alternative unit or volume in its place at run time.

Programmer response: Make sure that the DD statement for the data set assigned to this device contains the correct DCB information. In a merge application, if the device in error holds an input data set, make sure that the DCB information (except for BLKSIZE) specified in the SORTIN01 DD statement correctly describes the data in this device.

ICE118I UNKNOWN FILE SIZE - FILSZ=EN MAY IMPROVE RESOURCE USAGE AND PERFORMANCE

Explanation: DFSORT could not determine the input file size, and the number of records was not supplied by a FILSZ or SIZE value, that is, the input file size is unknown. Generally, DFSORT can automatically determine the input file size. However, in a few cases, such as when an E15 supplies all of the input records or when information about a tape data set is not available from a tape management system, DFSORT cannot determine an accurate file size.

If dynamic allocation of work space is used, DFSORT allocates the primary space according to the DYNSPC value in effect. This can result in an underallocation or overallocation, possibly leading to wasted space or an out-of-space condition, respectively. Performance may also be degraded.

System action: None.

Programmer response: None required, unless you receive message ICE046A or ICE083A, in which case you should specify FILSZ=En with a reasonably

accurate estimate of the number of records to be sorted. If you receive message ICE046A with dynamically allocated work space and you cannot specify FILSZ=En, use DYNSPC=n to increase the primary space allocated. If you receive message ICE083A and you cannot specify FILSZ=En, use DYNSPC=n to decrease the primary space allocated.

You may want to specify FILSZ=En even if you don't receive message ICE046A or ICE083A, because doing so may improve resource usage and performance.

Note: FILSZ=E0 is ignored.

ICE119A SUM FIELD OVERLAPS CONTROL FIELD

Explanation: Critical. A sum field in a SUM control statement overlapped a control field in a SORT or MERGE control statement.

System action: The program terminates.

Programmer response: Make sure that the sum fields in the FIELDS operand of the SUM control statement do not overlap control fields in the FIELDS operand of the SORT or MERGE control statement.

ICE120I EXIT E*nn* IGNORED

Explanation: User exit E14 or E2n was specified in the MODS statement, as identified in the message by E*nn*. Exit Enn is not supported and cannot be entered.

System action: The Enn operand is checked for correct syntax, but the user exit is not used.

Programmer response: Optional. Correct the MODS statement by removing the identified Enn operand.

ICE121A FILE SIZE IS TOO LARGE

Explanation: Critical. The amount of data to be sorted exceeded a DFSORT implementation limit for the maximum number of records that can be sorted. DFSORT cannot handle the amount of data regardless of environmental factors, such as the amount of work data set space provided. Specific implementation limits include:

1. Variable-length records - 4294967295 records
2. EQUALS in effect - 4294967295 records
3. VLSHRT in effect - 2147483647 records
4. Blockset technique not selected - 2147483647 records

System action: The program terminates.

Programmer response: For cases 1 through 3, perform the sort on subsets of the data set containing less than the indicated maximum number of records. Then merge the sorted subsets into a single data set.

For case 4, rerun the job with a SORTDIAG DD statement to get message ICE800I, which indicates the reason Blockset could not be used. If possible, remove the condition preventing the use of Blockset. Alternatively, perform the sort on subsets of the data set containing less than the indicated maximum number of records. Then merge the sorted subsets into a single data set.

ICE122A *keyword* VALUE INVALID

Explanation: Critical. The numeric value specified for the indicated keyword exceeded the DFSORT processing limit of 562949953421311 records (Blockset selected) or 2147483647 records (Blockset could not be used).

System action: The program terminates.

Programmer response: Specify a numeric value for the indicated keyword that is less than the processing limit.

ICE123I CKPT OR CHKPT OPTION IGNORED

Explanation: CKPT or CHKPT was specified in the SORT or OPTION control statement for a sort application, IGNCKPT=YES was specified during installation, and the Blockset technique was selected. The Blockset technique does not support the automatic checkpoint/restart facility, so the CKPT or CHKPT option was ignored.

System action: The program continues, but no checkpoints are taken.

Programmer response: Specifying NOBLKSET on the OPTION control statement will allow you to take checkpoints, except in cases where only the Blockset technique can be used (for example, when an OUTFIL control statement is specified).

ICE124A SUM FIELD OVERLAPS RECORD DESCRIPTOR WORD

Explanation: Critical. A sum field in a SUM control statement overlapped the record descriptor word (RDW) for variable-length record processing.

System action: The program terminates.

Programmer response: Make sure that the sum fields in the FIELDS operand of the SUM control statement do not overlap the RDW.

ICE125A SUM FIELD OVERLAPS SUM FIELD

Explanation: Critical. A sum field in a SUM control statement overlapped another sum field.

System action: The program terminates.

Programmer response: Make sure that the sum fields in the FIELDS operand of the SUM control statement do not overlap.

ICE126A INCONSISTENT {*INREC|*OUTREC|*ddname*} IFTHEN *n* REFORMATTING FIELD FOUND

Explanation: Critical. The FIELDS, BUILD, or IFTHEN BUILD operand of an INREC or OUTREC statement, or the OUTREC, BUILD or IFTHEN BUILD operand of an OUTFIL statement, contained a field that was inconsistent with the format (fixed or variable) of the input or output records, or with other fields.

OUTFIL data sets for which VTOF or CONVERT is used have variable-length input records and fixed length output records. OUTFIL data sets for which FTOV is used have fixed-length input records variable-length output records.

The specific cause of the error is identified as follows:

- *INREC and n=0 indicates that the FIELDS or BUILD operand of the INREC statement caused the error.
- *OUTREC and n=0 indicates that the FIELDS or BUILD operand of the OUTREC statement caused the error.
- ddname and n=0 indicates that the OUTREC or BUILD operand of an OUTFIL statement caused the error. ddname identifies the first data set in the associated OUTFIL group.
- *INREC and n>0 indicates that an IFTHEN BUILD operand of the INREC statement caused the error. n identifies the number of the associated IFTHEN clause (starting at 1 for the first IFTHEN clause in the INREC statement).
- *OUTREC and n>0 indicates that an IFTHEN BUILD operand of the OUTREC statement caused the error. n identifies the number of the associated IFTHEN clause (starting at 1 for the first IFTHEN clause in the OUTREC statement).
- ddname and n>0 indicates that an IFTHEN BUILD operand of an OUTFIL statement caused the error. ddname identifies the first data set in the associated OUTFIL group. n identifies the number of the associated IFTHEN clause (starting at 1 for the first IFTHEN clause in the OUTFIL statement).

The inconsistency is one of the following:

1. The first field was not 1,n where n is equal to or greater than 4, for variable-length input records, that is, the first field did not contain the record descriptor word (RDW).

   Example (variable-length input:):

   ```
   INREC FIELDS=(6,8)
   ```

2. The first field was 1,n, but was modified in some way (for example, 1,n,f or 1,n,TRAN=UTOL), for variable-length input records.

   Example (variable-length input):

   ```
   INREC BUILD=(1,6,HEX)
   ```

3. A single field containing only bytes from the RDW was specified for variable-length records.

   Example (variable-length input):

   ```
   OUTREC BUILD=(1,4)
   ```

4. The last position was specified without a corresponding length for fixed-length input records.

   Example (fixed-length input):

   ```
   OUTFIL IFTHEN=(WHEN=(5,1,CH,EQ,C'A'),
            BUILD=(1,5,8)),
          IFTHEN=(WHEN=NONE,OVERLAY=(21:C'OK'))
   ```

5. For variable-length input records, the last position was specified without a corresponding length for one, but not all, of the FIELDS or BUILD operand of the INREC statement, the FIELDS or BUILD operand of the OUTREC statement, or the OUTREC or BUILD operand of an OUTFIL statement (without a VTOF or CONVERT operand):

   Example (variable-length input):

   ```
   INREC FIELDS=(1,20,31)
   OUTREC FIELDS=(1,25)
   ```

6. For variable-length input records, the last position was specified with a corresponding length for one, but not all, of the FIELDS or BUILD operand of the INREC statement, the FIELDS or BUILD operand of the OUTREC statement, or the OUTREC or BUILD operand of an OUTFIL statement (without a VTOF or CONVERT operand):

   Example (variable-length input):

   ```
   OUTREC BUILD=(1,20,31,10)
   OUTFIL BUILD=(1,25,27)
   ```

7. A column overlapped the previous output field in the reformatted record.

   Example (fixed-length input):

   ```
   INREC FIELDS=(3,20,20:C'ABC')
   ```

8. A VTOF or CONVERT operand was specified without an OUTREC or BUILD operand for variable-length input record to fixed-length output record conversion in an OUTFIL statement.

   Example (variable-length input):

   ```
   OUTFIL VTOF
   ```

   Note: VTOF cannot be used with an IFTHEN or OVERLAY operand.

9. A VLFILL = C'x' or VLFILL = X'yy' operand was specified without an OUTREC or BUILD operand for variable-length input records in an OUTFIL statement.

   Example (variable-length input):

   ```
   OUTFIL VLFILL=C'*'
   ```

   Note: VLFILL cannot be used with an IFTHEN or OVERLAY operand.

10. An OUTREC or BUILD operand did not specify any input or separation fields.

    Example (fixed-length input):

    ```
    OUTFIL OUTREC=(//)
    ```

11. An IFOUTLEN=n operand was specified without an IFTHEN operand.

    Example (fixed-length input):

    ```
    OUTREC IFOUTLEN=50
    ```

System action: The program terminates.

Programmer response:

1. For case 1, specify 1,n for the first field, where n is equal to or greater than 4.

   Example (variable-length input):

   ```
   INREC FIELDS=(1,4,6,8)
   ```
2. For case 2, specify 1,4 before the modified field

   Example (variable-length input):

   ```
   INREC BUILD=(1,4,1,6,HEX)
   ```
3. For case 3, specify at least one data byte.

   Example (variable-length input):

   ```
   OUTREC BUILD=(1,5)
   ```
4. For case 4, specify the last position with a corresponding length.

   Example (fixed-length input):

   ```
   OUTFIL IFTHEN=(WHEN=(5,1,CH,EQ,C'A'),
                  BUILD=(1,5,8,25)),
          IFTHEN=(WHEN=NONE,OVERLAY=(21:C'OK'))
   ```
5. For case 5, specify the last position without a corresponding length for the INREC, OUTREC, and OUTFIL statements, as appropriate.

   Example (variable-length input):

   ```
   INREC FIELDS=(1,20,31)
   OUTREC FIELDS=(1,20,5X,21)
   ```
6. For case 6, specify the last position with a corresponding length for the INREC, OUTREC, and OUTFIL statements, as appropriate.

   Example (variable-length input):

   ```
   OUTREC BUILD=(1,20,31,10)
   OUTFIL BUILD=(1,20,5X,21,10)
   ```
7. For case 7, ensure that columns do not overlap previous fields in the reformatted record.

   Example (fixed-length input):

   ```
   INREC FIELDS=(3,20,25:C'ABC')
   ```
8. For case 8, specify an OUTREC or BUILD operand with the VTOF or CONVERT operand.

   Example (variable-length input):

   ```
   OUTFIL VTOF,BUILD=(5,60)
   ```
9. For case 9, specify an OUTREC or BUILD operand with the VLFILL=C'x' or VLFILL=X'yy operand'.

   Example (variable-length input):

   ```
   OUTFIL VLFILL=C'*',BUILD=(1,4,8,45)
   ```
10. For case 10, specify an input or separation field.

    Example (fixed-length input):

    ```
    OUTFIL OUTREC=(//,X)
    ```
11. For case 11, specify an IFTHEN operand with an IFOUTLEN=n operand.

    Example:

    ```
    OUTREC IFOUTLEN=50,
      IFTHEN=(WHEN=(5,1,CH,EQ,C'A'),
         OVERLAY=(21:C'N')),
      IFTHEN=(WHEN=NONE,OVERLAY=(21:C'Y'))
    ```

ICE127I **OPTIONS: OVFLO=*s*,PAD=*s*,TRUNC=*s*, SPANINC=*s*,VLSCMP=*x*,SZERO=*x*, RESET=*x*,VSAMEMT=*x*,DYNSPC=*n***

Explanation: See ICE133I.

ICE128I **OPTIONS: SIZE=*n*,MAXLIM=*n*,MINLIM=*n*, EQUALS=*x*,LIST=*x*,ERET=*a*, MSGDDN=*b***

Explanation: See ICE133I.

ICE129I **OPTIONS: VIO=*x*,RESDNT=*c*, SMF=*e*,WRKSEC=*x*,OUTSEC=*x*, VERIFY=*x*,CHALT=*x*,DYNALOC=*d*, ABCODE=*j***

Explanation: See ICE133I.

ICE130I **OPTIONS: RESALL=*n*,RESINV=*n*,SVC=*g*, CHECK=*x*,WRKREL=*x*,OUTREL=*x*, CKPT=*x*,STIMER=*x*,COBEXIT=*f***

Explanation: See ICE133I.

ICE131I **OPTIONS: TMAXLIM=*n*,ARESALL=*n*, ARESINV=*n*,OVERRGN=*n*,CINV=*x*, CFW=*x*, DSA=*n***

Explanation: See ICE133I.

ICE132I

OPTIONS: VLSHRT=*x*,ZDPRINT=*x*, IEXIT=*x*,TEXIT=*x*,LISTX=*x*, EFS=*h*,EXITCK=*k*,PARMDDN=*q*, FSZEST=*x*

Explanation: See ICE133I.

ICE133I **OPTIONS: HIPRMAX=*p*, DSPSIZE=*r*,ODMAXBF=*n*,SOLRF=*x*, VLLONG=*x*, VSAMIO=*x*,MOSIZE=*r***

Explanation: Messages ICE127I through ICE133I and ICE235I indicate the options in effect for a Blockset sort, copy, or merge application or a Peerage/Vale sort application.

When DFSORT is invoked by JCL, these options can come from the EXEC PARM field, the SYSIN data set, the DFSPARM data set, or the installation defaults. When DFSORT is invoked dynamically, these options can come from the SORTCNTL data set, the DFSPARM

data set, the invocation parameter list, or the installation defaults.

DFSORT can change the options due to conflicting specifications or for performance reasons. Some options are not meaningful for all applications (for example, HIPRMAX applies to a sort application, but not to a copy or merge application.

For information on where the options can be specified and on the order of override, see *z/OS DFSORT Application Programming Guide*. Message values are as follows:

x	Y for YES; N for NO
n	A decimal value
a	RC16 or ABEND
b	Data set ddname or SYSOUT
c	ALL for eligible DFSORT modules loaded into LPA; NONE for eligible DFSORT modules not loaded into LPA. DFSORT automatically determines the RESDNT status; you cannot override this status nor specify it as an option.
d	N for NO, or (y,n) where y is the dynamic allocation device name and n is a decimal value
e	SHORT, FULL, or NO
f	COB1 or COB2
g	n for SVC number n, or 109A for SVC 109 using the alternate SVC module
h	EFS program name or NONE
j	MSG or nnn for ABCODE number
k	S for STRONG; W for WEAK
m	ALL, NOWRK, or NONE
p	A decimal value or OPTIMAL
q	Control statement/PARM option DDNAME or DFSPARM
r	A decimal value or MAX
s	RC0, RC4 or RC16

System action: None.

Programmer response: None.

Notes:

1. Do not expect the values in ICE127I through ICE133I and ICE235I to be identical to the installation-option values selected at your site. You can use the ICETOOL DEFAULTS operator to show all the installation option values.
2. Messages ICE131I and ICE133I are printed only for blockset.

ICE134I NUMBER OF BYTES SORTED: *n*

Explanation: *n* is the total number of bytes "sorted"; that is, the number of bytes in records for which control field processing was performed. The following insert/delete/alter processing is performed before control field processing and is taken into account when determining n: SORTIN, SKIPREC, STOPAFT, E15, and INCLUDE/OMIT. Other processing that can affect the number of bytes actually manipulated by DFSORT (for example, INREC) is done after control field processing, and is not taken into account when determining *n*.

If the input data includes incomplete spanned segments, they will be eliminated before "sorting". However, these incomplete spanned segments may be taken into account when determining *n*, in order to avoid impacting performance. If you want to ensure that n does not reflect incomplete spanned segments, you can eliminate them with copy, and sort the resulting data.

System action: None.

Programmer response: None.

ICE135A [SORTIN=*ddname* **| SORTOUT=***ddname***] MATCHES POSSIBLE WORK DDNAME**

Explanation: Critical. The ddname specified for OPTION statement operand SORTIN or SORTOUT in DFSPARM or in the extended parameter list was *cccc*WK*dd*, *cccc*WK*dd*, *cccc*DK*d*, or *cccc*DK*dd*. *cccc* is the specified or defaulted value for the SORTDD operand and *d* is any character. These ddnames can be used for work data sets and thus cannot be used as input or output ddnames. For example, each of the following OPTION statements in DFSPARM will result in this error message:

```
OPTION SORTOUT=SORTWK02
OPTION SORTIN=SORTDKA
OPTION SORTDD=ABCD,SORTIN=ABCDWK2
OPTION SORTDD=ABCD,SORTOUT=ABCDDKW1
```

System action: The program terminates.

Programmer response: Specify a ddname for SORTIN or SORTOUT that cannot be used for a work data set. For example, none of the following OPTION statements in DFSPARM will result in this error message:

```
OPTION SORTOUT=MYOUT
OPTION SORTIN=SORTWR01
OPTION SORTDD=ABCD,SORTIN=MYIN
OPTION SORTDD=ABCD,SORTOUT=SORTWK1
```

ICE136I SPACE FOR *ddname* **WAS READJUSTED TO CYLINDERS**

Explanation: The indicated work data set was not allocated in cylinders. DFSORT adjusted its usage of the allocated space to cylinders and could have skipped several tracks in the primary extents of the data set.

System action: None.

Programmer response: To optimize performance and minimize wasted disk space for future runs, change your JCL to use cylinder allocation for the indicated work data set.

ICE137A **VSAM INPUT ERROR *i* (*nnn*) text from VSAM SYNAD**

Explanation: Critical. i is either P (physical) or L (logical), describing the type of error encountered. nnn is the VSAM feedback code from a GET macro, in decimal, and is followed by the VSAM SYNAD message.

System action: The program terminates.

Programmer response: Refer to *z/OS DFSMS Using Data Sets* or *z/OS DFSMS Macro Instructions for Data Sets* for the meaning of the error code and, if possible, take appropriate action.

ICE138A **VSAM OUTPUT ERROR *i* (*nnn*) text from VSAM SYNAD**

Explanation: Critical. i is either P (physical) or L (logical), describing the type of error encountered. nnn is the VSAM feedback code from a PUT macro, in decimal, and is followed by the VSAM SYNAD message.

System action: The program terminates.

Programmer response: Refer to *z/OS DFSMS Using Data Sets* or *z/OS DFSMS Macro Instructions for Data Sets* for the meaning of the error code and, if possible, take appropriate action.

ICE139A **UNSUPPORTED VSAM DATA SET ATTRIBUTE FOR *ddname***

Explanation: Critical. An attribute specified for the VSAM data set with the indicated ddname is not supported by DFSORT. DFSORT supports the following attributes:

- If Blockset is selected: SPAN, KSDS, ESDS, RRDS, and VRRDS
- If Blockset is not selected: SPAN, KSDS, ESDS, and RRDS

System action: The program terminates.

Programmer response: Remove any unsupported attributes, other than VRRDS, from the indicated VSAM data set. To use the VRRDS attribute, you must remove any condition preventing the use of Blockset. Rerun the job with a SORTDIAG DD statement to get message ICE800I, which indicates the reason why Blockset could not be used.

ICE140I **END OF PARAMETERS FROM *ddname* - SYSIN OR *xxxx*CNTL/PARAMETER LIST CONTROL STATEMENTS FOLLOW**

Explanation: An alternate parameter data set was specified with the indicated ddname. This message separates the listing of its parameters (control statements or PARM parameters) from the listing of control statements specified from other sources:

- The SYSIN DD statement
- The SORTCNTL DD statement
- The invoking parameter list.

The data set's parameters are listed only if LIST is in effect (see message ICE128I).

A parameter in a statement other than the OPTION or DEBUG control statement completely overrides the same or corresponding control statement in other sources.

A DEBUG control statement in the alternate parameter data set overrides only the same or corresponding parameters in a DEBUG control statement in the other sources. This selective override does not affect the other parameters in a DEBUG control statement in the alternate parameter data set or the other sources.

An OPTION control statement in the alternate parameter data set overrides only the same or corresponding parameters in an OPTION, SORT, or MERGE control statement in the other sources. This selective override does not affect the other parameters in an OPTION, SORT, or MERGE control statement in the alternate parameter data set or the other sources.

See *z/OS DFSORT Application Programming Guide* for complete details on the order of override.

System action: None.

Programmer response: None.

ICE141A **SPANNED RECORD ON *ddname* COULD NOT BE ASSEMBLED**

Explanation: Critical. A spanned record on the indicated data set could not be properly assembled. This message was issued for one of the following reasons:

- A segment length was greater than the LRECL.
- A segment length was less than 4 bytes.
- The total length of segments was greater than the LRECL.
- Segments are blocked incorrectly for VBS data sets (that is, a first and a last segment are in the same block).

System action: The program terminates.

Programmer response: Check the indicated data set

for incorrect spanned records.

ICE142I *ddname* **NOT FOUND - SYSOUT USED**

Explanation: A DD statement for ddname (the specified alternate message data set) was not found. SYSOUT was used instead.

System action: Processing continues; messages are written to SYSOUT.

Programmer response: If you want to use an alternate message data set, provide a DD statement for ddname.

ICE143I *t a* **TECHNIQUE SELECTED**

Explanation: t indicates the DFSORT technique chosen for the run, and a indicates the application chosen for the DFSORT technique. Message values are as follows:

t BLOCKSET – disk work data set sort, standard merge or copy
PEERAGE — disk work data set sort
VALE — disk work data set sort
CONVENTIONAL — tape work data set sort or nonstandard merge

a SORT
MERGE
COPY

System action: None.

Programmer response: None required. However, if Blockset was not selected, you may want to remove the condition preventing the use of Blockset. Rerun the job with a SORTDIAG DD statement to get message ICE800I, which indicates the reason Blockset could not be used.

ICE144A TOO MANY SUM FIELDS

Explanation: Critical. The complexity of the application caused dynamic areas to exceed the storage allowed for them.

System action: The program terminates.

Programmer response: Reduce the number of summary fields.

ICE145A SVC *nnn* **IS AN INVALID USER SVC**

Explanation: Critical. The DFSORT ICEMAC installation parameters for your site indicate that SVC nnn is the DFSORT SVC. SVC nnn is not compatible with the DFSORT release currently running. Either SVC nnn is for a release of DFSORT other than this one, or SVC nnn is a non-DFSORT SVC.

System action: The program terminates.

Programmer response: Report this message to your system programmer. It indicates that the DFSORT SVC was incorrectly installed.

ICE146I END OF STATEMENTS FROM *xxxx***CNTL - PARAMETER LIST STATEMENTS FOLLOW**

Explanation: When a data set specified with a DD statement for xxxxCNTL is present, this message separates the listing of its statements from the listing of the statements specified in the parameter list. The statements for xxxxCNTL (if any) and the parameter list (if any) are listed only if LIST is in effect (see message ICE128I).

A statement other than OPTION or DEBUG in the xxxxCNTL data set completely overrides the same or corresponding statement in the parameter list.

A DEBUG statement in the xxxxCNTL data set overrides only the same or corresponding operands in a DEBUG statement in the parameter list. This selective override does not affect the other operands in a DEBUG statement in the xxxxCNTL data set or parameter list.

An OPTION statement in the xxxxCNTL data set overrides only the same or corresponding operands in an OPTION, SORT, or MERGE statement in the parameter list. This selective override does not affect the other operands in an OPTION, SORT, or MERGE statement in the xxxxCNTL data set or parameter list. (OPTION statement operands LIST, NOLIST, LISTX, NOLISTX, EFS, MSGDDN, MSGPRT, SORTDD, SORTIN, and SORTOUT are not used if specified in the xxxxCNTL data set.)

For complete details on the order of override, see *z/OS DFSORT Application Programming Guide*.

System action: None.

Programmer response: None.

ICE147A OPTION STATEMENT OPERAND SORTIN OR SORTOUT NOT ALLOWED

Explanation: Critical. The OPTION statement operand SORTOUT was specified for a Conventional merge, or the operand SORTIN or SORTOUT was specified for a tape work data set sort, or ICETOOL was used for a tape work data set sort.

System action: The program terminates.

Programmer response: Remove the SORTIN or SORTOUT operand. You can use the SORTDD operand of the OPTION statement to provide alternate ddnames for SORTIN and SORTOUT.

If ICETOOL was used, do not use tape work data sets.

ICE148A *ddname* **CONCATENATION NOT ALLOWED**

Explanation: Critical. SORTINnn or SORTWKdd data sets were concatenated.

System action: The program terminates.

Programmer response: Use separate DD statements for SORTINnn or SORTWKdd data sets.

ICE149A **DFSORT IS NOT LICENSED FOR USE ON THIS SYSTEM - RETURN CODE** *x* **AND REASON CODE** *y*

Explanation: SYS1.PARMLIB member IFAPRD*xx* does not define DFSORT as a feature licensed for use on this system, or an error was encountered while registering DFSORT for use on this system.

System action: The program terminates.

Programmer response: If the return code (*x*) is 12, the reason code (*y*) is 4, and you are licensed to use DFSORT, then you need to update the IFAPRD*xx* member of parmlib. See *z/OS and z/OS.e Planning for Installation* for information about adding the appropriate entry to IFAPRD*xx*.

After you have added the appropriate entry to IFAPRD*xx*, you must activate it by using a SET PROD command or by IPLing the system.

For any other return and reason codes, contact the IBM Support Center.

ICE150I **VLSHRT NOT USED FOR SORT, MERGE, INCLUDE, OMIT OR SUM STATEMENT FIELDS**

Explanation: The VLSHRT option was in effect, and one or more of the following situations exists:

- An INREC or OUTREC statement was specified.
- A copy application was specified without an INCLUDE or OMIT statement.
- More than one control field was specified, and Blockset was not selected.
- The application had EFS processing in effect, and an EFS01 or EFS02 user exit routine was supplied by the EFS program.
- Locale processing was required for sort or merge fields.

System action: Processing continues; VLSHRT is not used for SORT, MERGE, INCLUDE, OMIT or SUM statement fields, but is used for OUTFIL INCLUDE or OMIT parameter fields, if specified.

Programmer response: None, unless message ICE015A or ICE218A was received, in which case you can either respecify your fields to be within the shortest record in the input data set, or use equivalent OUTFIL parameters (for example, the INCLUDE and OUTREC parameters of OUTFIL instead of the INCLUDE and OUTREC statements) for the application.

If Blockset was not selected, rerun the job with a SORTDIAG DD statement to get message ICE800I, which indicates the reason Blockset could not be used. If possible, remove the condition preventing the use of Blockset.

ICE151A **TOO MANY {*INCLUDE|*OMIT|*INREC|*OUTREC|***ddname***} IFTHEN** *n* **CONDITIONS**

Explanation: Critical. The complexity of the conditions in a COND, INCLUDE, OMIT or IFTHEN WHEN operand caused dynamic areas to exceed the storage allowed for them. The specific cause of the error is identified as follows:

- *INCLUDE indicates that the COND operand of the INCLUDE statement caused the error. *n* is 0.
- *OMIT indicates that the COND operand of the OMIT statement caused the error. *n* is 0.
- *ddname* and *n*=0 indicates that the INCLUDE or OMIT operand of an OUTFIL statement caused the error. *ddname* identifies the first data set in the associated OUTFIL group.
- *INREC indicates that an IFTHEN WHEN operand of the INREC statement caused the error. *n* identifies the number of the associated IFTHEN clause (starting at 1 for the first IFTHEN clause in the INREC statement).
- *OUTREC indicates that an IFTHEN WHEN operand of the OUTREC statement caused the error. *n* identifies the number of the associated IFTHEN clause (starting at 1 for the first IFTHEN clause in the OUTREC statement).
- *ddname* and *n*>0 indicates that an IFTHEN WHEN operand of an OUTFIL statement caused the error. *ddname* identifies the first data set in the associated OUTFIL group. *n* identifies the number of the associated IFTHEN clause (starting at 1 for the first IFTHEN clause in the OUTFIL statement).

System action: The program terminates.

Programmer response: Reduce the number of conditions, or the size of the constants, in the COND, INCLUDE, OMIT or IFTHEN WHEN operand that caused the problem. Alternatively, you may be able to avoid reducing the number of conditions, or the size of the constants, by using one of the following techniques:

- If NOSZERO is in effect and you can treat numeric –0 and +0 values as signed (that is, different) for this application, use the SZERO option.
- If possible, rewrite the conditions to use substring comparison tests (see *z/OS DFSORT Application Programming Guide* for details)
- Use an INREC or OUTREC statement with multiple IFTHEN clauses to "OR" sets of conditions together and set a "flag" that an OUTFIL statement can use to delete the appropriate records. In the following

example, the input data set has RECFM=FB and LRECL=80, and a1 to an and b1 to bn represent relational conditions:

```
//MULT  EXEC  PGM=ICEMAN
//SYSOUT    DD  SYSOUT=*
//SORTIN DD DSN=...
//SORTOUT DD DSN=...
//SYSIN    DD    *
  OPTION COPY
* If a1,OR,a2,OR,...,an is true,
* set flag in 81 to 'Y'.
  INREC IFTHEN=(WHEN=(a1,OR,a2,OR,...,an),
            OVERLAY=(81:C'Y')),
* If b1,OR,b2,OR,...,bn is true,
* set flag in 81 to 'Y'.
     IFTHEN=(WHEN=(b1,OR,b2,OR,...,bn),
       OVERLAY=(81:C'Y')),
* If none of the conditions is true,
* set flag in 81 to 'N'.
     IFTHEN=(WHEN=NONE,OVERLAY=(81:C'N'))
* If the flag in 81 is 'Y', include the
* records.
  OUTFIL INCLUDE=(81,1,CH,EQ,C'Y'),
* Remove the flag byte.
    OUTREC=(1,80)
/*
```

ICE152I OVERFLOW DURING SUMMATION - RC=*n*

Explanation: The application was successfully processed, but overflow occurred for one or more pairs of summary fields. DFSORT set a return code of 0 or 4 as indicated by n. DFSORT only set a return code of 4 if OVFLO=RC4 was in effect and summary fields overflowed.

System action: Summation continues for pairs of records for which overflow does not occur. Summation is not performed for pairs of records for which overflow would produce more than one record with the same control field. DFSORT passes back a return code of n unless it sets a higher return code for some other reason.

Programmer response: If appropriate, redesign the records so that summary fields do not overflow, or, if possible, use INREC to increase the size of the summary fields (see *z/OS DFSORT Application Programming Guide*), or specify the OVFLO=RC0, OVFLO=RC4, or OVFLO=RC16 run-time option to provide a different return code for this situation.

ICE153A COBOL EXIT NOT SUPPORTED FOR TECHNIQUE USED

Explanation: Critical. An E15 or E35 user exit written in COBOL was specified, but is not supported for the technique used (tape work data set sort or Conventional merge).

System action: The program terminates.

Programmer response: Rerun the job with a SORTDIAG DD statement to get message ICE800I. ICE800I indicates the reason Blockset could not be used. If possible, remove the condition preventing the use of Blockset. Alternatively, remove the COBOL user exit.

ICE154A STOPAFT NOT SUPPORTED FOR TECHNIQUE USED

Explanation: Critical. STOPAFT was in effect but is not supported for the technique used (tape work data set sort).

System action: The program terminates.

Programmer response: Use disk work data sets, if possible, or remove the STOPAFT option.

ICE155I STOPAFT OR SKIPREC NOT APPLICABLE TO MERGE

Explanation: A MERGE control statement was specified and STOPAFT or SKIPREC was in effect. Neither STOPAFT nor SKIPREC can be used with MERGE.

System action: Processing continues; STOPAFT or SKIPREC is ignored.

Programmer response: None.

ICE156I MAIN STORAGE ABOVE 16-MEGABYTE = (*y*,*z*)

Explanation: Supplies information on the amount of storage available above 16 MB virtual:

y The maximum amount of main storage available to DFSORT above 16 MB virtual.

z The actual amount of main storage available to DFSORT above 16 MB virtual, after DFSORT has released the ARESALL and ARESINV space.

This message is only printed when Blockset is selected.

System action: None.

Programmer response: None.

ICE157I EXEC PARM E15=COB OR E35=COB AND NO MODS EXIT

Explanation: A MODS statement contained one of the following:
- E15=COB without a corresponding E15 operand
- E35=COB without a corresponding E35 operand.

System action: Processing continues. The EXEC parameter E15=COB or E35=COB is ignored.

Programmer response: Specify an E15 or E35 operand on a MODS statement or remove the EXEC parameter.

ICE158A SYSOUT DD STATEMENT MISSING

Explanation: Critical. Issued only to the master console to indicate that ICEMAC option NOMSGDD=QUIT was used and a message data set was required, but not specified. Corresponds to DFSORT return code 20.

System action: The program terminates.

Programmer response: Specify a DD statement for the message data set, using the ddname specified by MSGDDN (if any) or SYSOUT.

ICE159A MODULE ICECOB2 NOT FOUND

Explanation: Critical. Module ICECOB2 was not installed as part of DFSORT. An E15 or E35 user exit routine written in COBOL was to be used and COBEXIT=COB2 was in effect. Module ICECOB2 is required for COBEXIT=COB2.

System action: The program terminates.

Programmer response: Ask your system programmer to ensure that module ICECOB2 is available.

ICE160A COPY FUNCTION COULD NOT BE USED - REASON CODE IS *nn*

Explanation: Critical. The COPY function was specified but could not be processed. Reason code values are as follows:

1 The SORTIN or SORTOUT data set was BDAM.

2 FREE=CLOSE was specified (ICEGENER).

3 An attempt to open a data set caused a system error.

4 The SORTIN or SORTOUT data set resided on an unsupported device.

5 ASCII tapes had the following parameters:

(LABEL=AL or OPTCD=Q) and RECFM=D and BUFOFF¬=L

or

(LABEL=AL or OPTCD=Q) and RECFM¬=D and BUFOFF¬=0

6 An attempt to read the DSCB for the SORTIN data set caused an error.

7 An attempt to read the DSCB for the SORTOUT data set caused an error.

8 The SORTIN data set had keyed records (ICEGENER).

9 User labels were present (ICEGENER).

10 A MODS statement Exx operand had SYSIN in the third parameter or T or S in the fourth parameter (that is, dynamic link-editing was requested.)

Note: For ICEGENER, SYSUT1 is equivalent to SORTIN and SYSUT2 is equivalent to SORTOUT.

System action: The program terminates.

Programmer response: Correct the situation indicated by the reason code.

ICE161A COBEXIT=COB2 AND COBOL E15 AND E35 EXITS WERE FOUND

Explanation: Critical. COBEXIT=COB2 was in effect; therefore, separately compiled COBOL E15 and E35 user exits were not allowed together for COPY processing.

System action: The program terminates.

Programmer response: When COBEXIT=COB2 is in effect, use either a COBOL E15 user exit or a COBOL E35 user exit for copy applications, but not both.

ICE162I ICEIEXIT CHANGED ONE OR MORE OPTIONS IN EFFECT

Explanation: During installation, the initialization exit (ICEIEXIT) was used to change one or more of the options that apply to your application. DFSORT used the changed options where appropriate. Messages ICE127I-ICE133I and ICE235I indicate which options DFSORT used.

System action: None.

Programmer response: No action is necessary. If appropriate, see your system programmer for an explanation of the changes made with the ICEIEXIT routine.

ICE163A DFSORT CANNOT FIND THE IEBGENER MODULE

Explanation: Critical. This message, issued only to the master console, indicates the DFSORT ICEGENER facility attempted to transfer control to the IEBGENER module, but the DFSORT installation option for that name was the same as the name for the ICEGENER module, thus resulting in a loop.

System action: The program terminates with user abend 163.

Programmer response: Ensure that the name being used for the GENER installation option is an alias of the IEBGENER module and not an alias of the ICEGENER module.

ICE164A *xnnnn* ABEND OCCURRED DURING EXECUTION OF ICETEXIT

Explanation: Critical. ESTAE was in effect and an abend occurred after the installation exit ICETEXIT was called but before control was returned to DFSORT.

x The abend type, either S for system or U for user

nnnn The abend code

System action: None.

Programmer response: Notify your system programmer that the installation exit, ICETEXIT, has abended so that they can determine the cause of the abend and correct the error. You might need to rerun the job with a SYSUDUMP DD statement to get an abend dump. With ESTAE in effect, DFSORT tries to call ICETEXIT (if it has not been called already) after an earlier abend. In this case, only the information about the first abend appears in the dump. Rerun the job with DEBUG ABSTP to get the dump for the secondary abend.

ICE165I TOTAL DATA SET TRACKS ALLOCATED: *a*, TRACKS USED: *b*

Explanation: This message gives work data set space usage information for the sort run. Message values are as follows:

a Total primary and secondary disk work space allocated and available throughout sort processing (in tracks). This space is not available for use by other applications.

b Total primary and secondary disk work space actually used during sort processing (in tracks).

Note: In some cases, DFSORT may free up JCL-specified work space if it is determined that the space is not needed for the sort.

System action: None.

Programmer response: None.

ICE168I CENTURY WINDOW FOR Y2 FORMAT FIELDS IS FROM *ssss* TO *eeee*

Explanation: Indicates the starting year (ssss) and ending year (eeee) for the century window to be used in conjunction with the specified Y2 format fields.

System action: None

Programmer response: None

ICE170I FORMAT OPERAND IGNORED

Explanation: A FORMAT=f operand was specified on a SORT, MERGE, SUM, INCLUDE or OMIT statement in which p,m,f was used for all of the fields. For example:

```
SORT FORMAT=BI,FIELDS=(5,4,CH,A,12,2,PD,A,21,8,BI,D)
```

System action: f from p,m,f is used for each field. f from FORMAT=f is not used for any of the fields.

Programmer response: Optional. Remove FORMAT=f, or remove any or all f values in FIELDS that match the f value in FORMAT=f. For example, the SORT statement above could be changed to either of the following SORT statements to eliminate the ICE170I message:

```
SORT FIELDS=(5,4,CH,A,12,2,PD,A,21,8,BI,D)
```

```
SORT FORMAT=BI,FIELDS=(5,4,CH,A,12,2,PD,A,21,8,D)
```

ICE171I SORTOUT LRECL OF *x* IS DIFFERENT FROM SORTIN(NN) LRECL OF *y* - RC=*n*

Explanation: The application was successfully processed, but the SORTOUT LRECL (*x*) and SORTIN/SORTINnn LRECL (*y*) were not equal. DFSORT set a return code of 0 or 4 as indicated by *n*. DFSORT only sets a return code of 4 if:

- PAD=RC4 was in effect and *x* was larger than *y* (LRECL padding), or
- TRUNC=RC4 was in effect and *x* was smaller than *y* (LRECL truncation).

System action: Processing continues and DFSORT passes back a return code of *n*, unless it sets a higher return code for some other reason.

Programmer response: If appropriate, change the LRECL of SORTOUT or SORTIN/SORTIN*nn* so that the LRECLs are equal, or specify the PAD=RC0, PAD=RC4, or PAD=RC16 run-time option to provide a different return code for this situation.

ICE172A XSUM IS NOT SUPPORTED - USE ICETOOL SELECT IF APPROPRIATE

Explanation: Critical. XSUM was specified on the SUM statement. DFSORT does not support the XSUM parameter provided by a competitive sort product to write records deleted by SUM processing to a SORTXSUM DD data set.

System action: The program terminates.

Programmer response: If you are using SUM FIELDS = NONE with XSUM, you can perform the same function with ICETOOL's SELECT operator. For example, if you are using XSUM as follows:

```
//S1 EXEC PGM= ICEMAN
//SYSOUT DD SYSOUT=*
//SORTIN DD DSN=...
//SORTOUT DD DSN=...
//SORTXSUM DD DSN=...
//SYSIN DD *
  SORT FIELDS=(5,4,CH,A)
  SUM FIELDS=NONE,XSUM
/*
```

You can use this ICETOOL job instead:

```
//S1 EXEC PGM=ICETOOL
//TOOLMSG DD SYSOUT=*
//DFSMSG DD SYSOUT=*
//SORTIN DD DSN=...
```

```
//SORTOUT DD DSN=...
//SORTXSUM DD DSN=...
//TOOLIN DD *
SELECT FROM(SORTIN) TO(SORTOUT)-
  ON(5,4,CH) FIRST DISCARD(SORTXSUM)
/*
```

Related reading: You can also perform additional functions with ICETOOL's SELECT operator that are not available with XSUM. See *z/OS DFSORT Application Programming Guide* for complete details of ICETOOL's SELECT operator.

ICE173I NO RECORDS FOR THE SORTOUT DATA SET - RC=*n*

Explanation: The application was successfully processed, but there were no records for the SORTOUT data set, as indicated by an OUT count of 0 in message ICE054I. DFSORT set a return code of 0 or 4 as indicated by *n*. DFSORT only sets a return code of 4 if NULLOUT=RC4 was in effect, but there were no records for the SORTOUT data set.

System action: Processing continues and DFSORT passes back a return code of *n*, unless it sets a higher return code for some other reason.

Programmer response: If appropriate, change the application to produce records for the SORTOUT data set. Alternatively, specify the NULLOUT=RC0, NULLOUT=RC4, or NULLOUT=RC16 run-time option to provide a different return code for this situation.

ICE174I NO DATA RECORDS FOR AN OUTFIL DATA SET - RC=*n*

Explanation: The application was successfully processed, but there were no data records for one or more OUTFIL data sets, as indicated by a DATA count of 0 in message ICE227I. DFSORT set a return code of 0 or 4 as indicated by *n*. DFSORT only sets a return code of 4 if NULLOFL=RC4 was in effect for an OUTFIL data set, but there were no data records for that OUTFIL data set.

System action: Processing continues and DFSORT passes back a return code of *n* unless it sets a higher return code for some other reason.

Programmer response: If appropriate, change the application to produce records for the relevant OUTFIL data sets (which you can identify by a DATA count of 0 in message ICE227I). Alternatively, specify the NULLOFL=RC0, NULLOFL=RC4, or NULLOFL=RC16 run-time option on the relevant OUTFIL statements to provide a different return code for this situation.

ICE176A *service* SERVICE ERROR (*x*-*y*)

Explanation: Critical. The indicated system service failed when DFSORT called it. Message values are as follows:

x The return code from the service, in decimal.

y The reason code from the service, in decimal, or zero if unavailable.

System action: The program terminates.

Programmer response: Contact IBM for programming support.

ICE177A DASD CALCULATION SERVICES ERROR (*x*-*y*)

Explanation: Critical. DASD Calculation Services returned an unexpected return or reason code when DFSORT called it to determine the optimum block size for an output data set.

System action: The program terminates.

Programmer response: Contact IBM for programming support. You can bypass the use of system-determined block size for this application by specifying either SDB=NO or a BLKSIZE value on the DD statement for the output data set.

ICE178I REALLOCATING VIO SORTWK DATA SETS TO REAL SORTDK DATA SETS USING UNIT=*device*

Explanation: SORTWKdd data sets specified in the JCL were allocated on virtual devices (VIO). Because the VIO=NO installation option was specified or defaulted, DFSORT will attempt to reallocate the SORTWKdd data sets as SORTDKdd data sets using devices of the type indicated by device.

System action: Processing continues; DFSORT attempts to reallocate the data sets.

Programmer response: If appropriate, allocate the SORTWKdd data sets on real devices to avoid reallocation.

ICE179A RESOURCES WERE UNAVAILABLE FOR VIO REALLOCATION OF WORK DATA SETS (*xxxx*)

Explanation: Critical. SORTWKdd data sets specified in the JCL were allocated on virtual devices (VIO). Because the VIO=NO installation option was specified or defaulted, DFSORT attempted to reallocate the SORTWKdd data sets as SORTDKdd data sets using a real device of the type indicated in message ICE178I. The reallocation failed; xxxx indicates the failure code returned by the system's dynamic allocation facility. See *Application Development Guide for Authorized Programs* for an explanation of the failure code.

Two commonly received return codes are:

0218 The real device type is defined to the system, but the requested device has insufficient space available. For example, the real device type is defined to the system but no devices of that type have been installed, or all devices of that type are mounted as private.

021C The real device type is not defined to the system. For example, 3380 was specified instead of 3390.

However, in a DFSMS environment, the return code is usually 970C regardless of the specific error encountered.

System action: The program terminates.

Programmer response: Ensure that the VIO SORTWKdd data sets can be reallocated on the real device indicated in message ICE178I, or if appropriate, allocate the SORTWKdd data sets on real devices to avoid reallocation.

Alternatively, specify installation option VIO=YES (can degrade performance). See *z/OS DFSORT Installation and Customization* for details on the VIO installation option.

ICE180I HIPERSPACE STORAGE USED = *n*K BYTES

Explanation: n is the number of Kilobytes of Hiperspace storage used during this sort. If *n* is zero, Hiperspace storage was not used.

System action: None.

Programmer response: None.

ICE181A HIPERSPACE READ ERROR (*xx*) [*yyyyyyyy*]

Explanation: Critical. The HSPSERV SREAD service request for a Hiperspace read failed. *xx* is the HSPSERV return code (in decimal). *yyyyyyyy* is the reason code (set only when the return code is 8).

System action: The program terminates.

Programmer response: Refer to the HSPSERV service documentation in the *z/OS DCE Application Development Reference* for a description of HSPSERV return codes and reason codes. If possible, take appropriate action.

The use of Hiperspace can be bypassed by specifying HIPRMAX=0 on the OPTION control statement.

ICE182A HIPERSPACE WRITE ERROR (*xx*) [*yyyyyyyy*]

Explanation: Critical. The HSPSERV SWRITE service request for a Hiperspace write failed. *xx* is the HSPSERV return code (in decimal). *yyyyyyyy* is the reason code (set only when the return code is 8).

System action: The program terminates.

Programmer response: Refer to the HSPSERV service documentation in the *z/OS DCE Application Development Reference* for a description of HSPSERV SWRITE return codes and reason codes. If possible, take appropriate action.

The use of Hiperspace can be bypassed by specifying HIPRMAX=0 on the OPTION control statement.

ICE183A CONDITIONS NOT MET TO USE THE SAME VSAM DATA SET FOR indd AND outdd

Explanation: Critical. The indd DD statement and outdd DD statement both specified the same VSAM data set, but one of the following conditions prevented DFSORT from using the same VSAM data set for input and output:

- The application is a copy or merge
- NOVSAMIO is in effect
- NORESET is in effect
- The VSAM data set is not defined with REUSE

System action: The program terminates.

Programmer response: If you want to sort the VSAM data set in-place, ensure that VSAMIO and RESET are in effect, and that the VSAM data set is defined with REUSE. Otherwise, ensure that the indd DD and outdd DD statements do not specify the same VSAM data set.

ICE184I THE NUMBER OF RECORDS SORTED WITH EQUAL KEYS IS *n*

Explanation: n is the number of records sorted having equal control keys.

System action: None.

Programmer response: None.

ICE185A AN *xnnnn* ABEND WAS ISSUED BY DFSORT, ANOTHER PROGRAM OR AN EXIT (PHASE *m y*)

Explanation: Critical. DFSORT detected a system or user abend while it's ESTAE routine was in effect. The abend may have been issued by DFSORT, another program that called DFSORT, or a user exit called by DFSORT. Although DFSORT detected the abend, DFSORT is not necessarily the cause of the abend.

This message gives details about the abend as follows:

x The abend type, either S for system or U for user

nnnn The abend code

This message gives details about DFSORT's application mode (if known at the time of the error) and phase at the time DFSORT detected the abend.

m S, M, or C for a Blockset sort, merge, or copy, respectively. m is P when the Peerage/Vale technique was used. m can also be blank if the abend occurred before DFSORT determined the application type.

y Phase number 0, 1, 2, 3, or 4.

System action: The program terminates.

Programmer response: Use the information in the abend dump to determine if DFSORT, a calling program, or a user exit caused the abend, and take appropriate action.

For system abends, see the appropriate systems codes document.

User abends are program specific. If U*nnnn* is reported in the message, *nnnn* in the range 1000-1675 or 2222 may or may not have been issued by DFSORT. *nnnn* outside that range was not issued by DFSORT.

ICE186A DFSORT CONTINUED AFTER ERROR AND OUTPUT IS USABLE

Explanation: Critical. An error was detected or an abend occurred; however, the sort, copy, or merge was completed successfully. The SORTOUT data was written successfully or an E35 user exit disposed of all of the output records. The SORTOUT data set written by DFSORT was closed successfully.

System action: The program terminates with the abend code.

Programmer response: The SORTOUT data set can be used (provided the system did not delete it) even though the program terminated abnormally. Because of the error, one of the following DFSORT functions below was not completed successfully:

- The writing of final and statistical messages
- The writing of an SMF record, if appropriate
- The calling of a termination exit, if present
- The closing of data sets, releasing of space, and other cleanup functions.

ICE187I DFSORT SVC LEVEL INCOMPATIBLE WITH DFSORT PROGRAM LEVEL

Explanation: The running level of DFSORT did not match the installed level of the DFSORT SVC. This incompatibility had no effect on the DFSORT run if the SVC was not needed. If the SVC was needed, the run could have terminated with a system abend code of 56D or other failure. DFSORT uses the SVC with certain disk devices and to write the SMF record. See *z/OS DFSORT Installation and Customization* for more information about installing the SVC. Message ICE816I was also printed if DFSORT abended trying to call the SVC.

System action: Processing continues.

Programmer response: Ask your system programmer to check that the DFSORT SVC is installed properly for this level of DFSORT and that the correct routing code for the SVC is selected. If necessary, rerun the job with a SORTDIAG DD dummy statement to get message ICE816I.

ICE188I DATA SPACE STORAGE USED = *n*K BYTES

Explanation: n is the number of Kilobytes of data space storage used during this sort. If *n* is zero, data space storage was not used.

System action: None.

Programmer response: None.

ICE189A BLOCKSET REQUIRED BUT COULD NOT BE USED - REASON CODE IS *nn*

Explanation: Critical. Blockset was required for one of the following:

- LOCALE processing
- OUTFIL processing
- Y2x, Y2xx, PD0, FS, CSF, UFF, or SFF format
- PARSE, OVERLAY, IFTHEN, or IFOUTLEN processing
- INREC or OUTREC processing with one of the following:
  - p,m,HEX
  - p,HEX
  - p,m,TRAN=LTOU
  - p,TRAN=LTOU
  - p,m,TRAN =UTOL
  - p,TRAN=UTOL
  - p,m,TRAN=ALTSEQ
  - p,TRAN=ALTSEQ
  - p,m,f
  - p,m,lookup
  - p,m,JFY=(...)
  - p,m,SQZ=(...)
  - %nn or %n
  - SEQNUM
  - DATE1, DATE1(c), DATE1P, DATE2, DATE2(c), DATE2P, DATE3, DATE3(c), DATE3P, or +n and -n variations (for example, DATE1-20 or DATE3P+20)
  - DATE4, DATE, DATE=(abcd), DATENS=(abc), YDDD=(abc) or YDDDNS=(ab),
  - TIME1, TIME1(c), TIME1P, TIME2, TIME2(c), TIME2P, TIME3, TIME3P, TIME, TIME=(abc), or TIMENS=(ab)
  - +n
  - –n
  - (...)
- A VSAM extended addressability data set
- To set the SORTOUT LRECL from the L3 length (without E35, INREC or OUTREC), the OUTREC length or the INREC length, with SOLRF in effect
- an FL format sort field with NOSZERO in effect

- VLLONG in effect and SORTOUT present
- VSAMEMT in effect for a sort or merge with VSAM input
- The same VSAM data set was specified for both input and output
- An HFS file was specified for input or output
- A tape data set with a block size greater than 32760 bytes was specified for input or output
- SDB=LARGE or SDB=INPUT was in effect and DFSORT selected a block size greater than 32760 bytes for a tape output data set
- VLSHRT in effect with a SUM statement
- Position plus length for a control field exceeded 4093
- Position plus length for a summary field exceeded 4093
- ICETOOL called DFSORT for an operation involving SORTOUT, and NULLOUT=RC16 is in effect
- DB2 Utilities called DFSORT.
- INCLUDE or OMIT processing with NUM
- A 33 to 256 byte PD, ZD, CSL, CST, CLO, CTO, ASL or AST format merge field
- A 256 byte PD format sort field with NOSZERO in effect

However, Blockset could not be used due to the reason indicated by reason code nn. See message ICE800I for the meaning of nn.

System action: The program terminates.

Programmer response: Correct the situation indicated by the reason code so Blockset can be used. Alternatively, you can remove the source of the requirement to use Blockset. However, this will result in the use of a less efficient technique.

ICE190A *ddname* **DYNALLOC CALL FAILED - RETURN CODE** *x* **AND REASON CODE** *y*

Explanation: Critical. For the data set with the associated ddname, the DYNALLOC SVC failed while attempting to obtain the pathname for an HFS file, or while attempting to determine if the data set was an HFS data set. *x* and *y* are the return code and reason code, respectively, from the DYNALLOC SVC.

System action: The program terminates.

Programmer response: Refer to the explanation for DYNALLOC Return Codes in *z/OS MVS Programming: Authorized Assembler Services Guide* for more information about the indicated return code and reason code. If possible, take appropriate action.

ICE191I DFSORT SVC NOT AVAILABLE - PERFORMANCE MAY BE DEGRADED

Explanation: An input, output, or work data set was on a cached disk control unit, but the DFSORT SVC was either not installed on the system or installed incorrectly. DFSORT did not use its SVC to set the optimal caching mode for the data set, which could have lengthened DFSORT's elapsed time. Message ICE816I was also printed if DFSORT abended while trying to call the SVC.

System action: Processing continues.

Programmer response: Have your system programmer check to see that the DFSORT SVC has been installed properly for this level of the DFSORT program and that the correct routing code for the SVC has been selected. If necessary, rerun the job with a SORTDIAG DD dummy statement to get message ICE816I.

ICE192I INVALID RETURN CODE FROM EXIT E*nn* **WAS ACCEPTED**

Explanation: User exit E*nn* (E15 or E35) inappropriately passed DFSORT a return code of 0, 4, 12 or 20. Because EXITCK=WEAK was in effect, the invalid return code was accepted and interpreted as a valid return code. See *z/OS DFSORT Application Programming Guide* for an explanation of how DFSORT interprets E15 and E35 return codes when EXITCK=WEAK is in effect.

This message is issued once for E15 if it passes one or more inappropriate return codes to DFSORT or once for E35 if it passes one or more inappropriate return codes to DFSORT.

System action: Processing continues.

Programmer response: None.

ICE193I ICEAM*x* **ENVIRONMENT IN EFFECT - ICE***yyy* **INSTALLATION MODULE SELECTED**

Explanation: ICEAM*x* indicates the environment in effect for this run as follows:

- ICEAM1 - batch JCL directly invoked environment
- ICEAM2 - batch program invoked environment
- ICEAM3 - TSO directly invoked environment
- ICEAM4 - TSO program invoked environment

ICE*yyy* indicates the time-of-day installation module (ICETD*x*) or environment installation module (ICEAM*x*) selected for this run. The set of installation defaults for the selected installation module are used for the run.

An ICETDx module can only be selected if:

- it is specified by the ENABLE parameter of the ICEAM*x* module in effect for the run-time environment, and
- it specifies a day and time range that includes the day and time of the run (as listed in DFSORT's ICE000I message)

If an ICETD*x* module is not selected, the ICEAM*x* module in effect for the run-time environment is selected.

Note: You can use ICETOOL's DEFAULTS operator to list the ENABLE values for the ICEAM*x* modules, the day and time ranges for the ICETD*x* modules and all of the installation defaults for all of the ICEAM*x* and ICETD*x* modules.

System action: None.

Programmer response: None.

ICE195A OVERFLOW DURING SUMMATION - RC=16

Explanation: Critical. OVFLO=RC16 was in effect and a pair of summary fields overflowed.

System action: The program terminates.

Programmer response: Redesign the records so that summary fields do not overflow, or if possible, use INREC to increase the size of the summary fields (see *z/OS DFSORT Application Programming Guide*). Alternatively, specify the OVFLO=RC0 or OVFLO=RC4 run-time option to override OVFLO=RC16 so summary overflow will be allowed for this application.

ICE196A SORTOUT LRECL OF *x* IS DIFFERENT FROM SORTIN(NN) LRECL OF *y* - RC=16

Explanation: Critical. One of the following errors occurred:

- PAD=RC16 was in effect and the SORTOUT LRECL (*x*) was larger than the SORTIN/SORTIN*nn* LRECL (*y*)
- TRUNC=RC16 was in effect and the SORTOUT LRECL (*x*) was smaller than the SORTIN/SORTINnn LRECL (*y*).

System action: The program terminates.

Programmer response: Change the LRECL of SORTOUT or SORTIN/SORTIN*nn* so that the LRECLs are equal. Alternatively, specify the PAD=RC0, PAD=RC4, TRUNC=RC0, or TRUNC=RC4 run-time option to override PAD=RC16 or TRUNC=RC16.

ICE197I ONE OR MORE INCOMPLETE SPANNED RECORDS DETECTED AND ELIMINATED - RC=*n*

Explanation: The application was successfully processed, but one or more incomplete spanned records was detected. DFSORT set a return code or 0 or 4, as indicated by *n*, according to whether SPANINC=RC0 or SPANINC=RC4 was in effect.

System action: Processing continues and DFSORT passes back a return code of *n* unless it sets a higher return code for some other reason. Incomplete spanned records are eliminated and valid records are recovered.

Programmer response: If appropriate, specify the SPANINC=RC0, SPANINC=RC4, or SPANINC=RC16 run-time option to provide a different return code for this situation.

ICE198I DEBUG EQUCOUNT CANNOT BE USED WITHOUT WORK DATA SET

Explanation: DEBUG EQUCOUNT was not used for this variable-length-record Blockset sort because no work data set was supplied.

System action: Processing continues. DEBUG EQUCOUNT is not used.

Programmer response: To use EQUCOUNT for this application, at least one work data set must be supplied.

ICE199I MEMORY OBJECT STORAGE USED = *n*M BYTES

Explanation: *n* is the number of megabytes DFSORT used for a memory object during this sort. If *n* is zero, DFSORT did not use a memory object during this sort.

System action: None.

Programmer response: None.

ICE200I IDENTIFIER FROM CALLING PROGRAM IS *cccc*

Explanation: *cccc* is the identifier passed by the calling program for this invocation of DFSORT.

System action: None.

Programmer response: If appropriate, use *cccc* to relate this set of DFSORT messages to the call to DFSORT that passed this identifier.

ICE201I RECORD TYPE IS *x* - DATA STARTS IN POSITION *y*

Explanation: Indicates whether DFSORT processes the input records as fixed-length (without an RDW) or variable-length (with an RDW).

- For fixed-length record processing, *x* is F and *y* is 1. Because the input records are processed without an RDW, the data starts in position 1. Positions in your control statements should be specified accordingly.
- For variable-length record processing, *x* is V and *y* is 5. Because the input records are processed with an RDW in positions 1-4, the data starts in position 5. Positions in your control statements should be specified accordingly.

System action: None.

Programmer response: None, unless you are using VSAM input, or supplying all input with an E15 or E32 exit, and the selected record type is not what you want DFSORT to use. In that case, specify RECORD TYPE=F or RECORD TYPE=V as appropriate.

ICE202I SYSUT2 LRECL OF *x* IS DIFFERENT FROM SYSUT1 LRECL OF *y* - RC=*n*

Explanation: The ICEGENER application was successfully processed, but the SYSUT2 LRECL *(x)* and SYSUT1 LRECL *(y)* were not equal. ICEGENER set a return code of 0 or 4 as indicated by n. ICEGENER only set a return code of 4 if:

- ICEMAC option GNPAD=RC4 was specified and *x* was larger than *y* (LRECL padding), or
- ICEMAC option GNTRUNC=RC4 was specified and *x* was smaller than *y* (LRECL truncation).

System action: Processing continues and ICEGENER passes back a return code of n unless it sets a higher return code for some other reason.

Programmer response: If appropriate, change the LRECL of SYSUT1 or SYSUT2 so that the LRECLs are equal.

ICE203I ONE OR MORE RECORDS HAD SHORT SUM FIELDS

Explanation: The application was successfully processed, but one or more records had short SUM summary fields, that is, a summary field extended beyond the end of the variable-length record.

System action: Records with short summary fields are excluded from summation, that is, if either one of a pair of records being summed has a short SUM field, the records are left unsummed and neither record is deleted.

Programmer response: If records with short summary fields have equal control fields and summation of these records is required, redesign the records to eliminate short summary fields.

ICE204A INCOMPLETE SPANNED RECORD DETECTED ON *ddname* - RC=16

Explanation: Critical. SPANINC=RC16 was in effect and a spanned record in the indicated data set was incomplete.

System action: The program terminates.

Programmer response: Delete all incomplete spanned records from the indicated data set. Alternatively, specify the SPANINC=RC0 or SPANINC=RC4 run-time option to override SPANINC=RC16; DFSORT can then eliminate incomplete spanned records and recover valid records for this application.

ICE205A CONTROL WAS NOT PASSED TO ICETEXIT

Explanation: Critical. Control was not passed to ICETEXIT because an error occurred while collecting statistical information for the termination installation exit (ICETEXIT).

System action: The program terminates.

Programmer response: Notify your system programmer of the problem.

ICE206A NO RECORDS FOR THE SORTOUT DATA SET - RC=16

Explanation: Critical. NULLOUT=RC16 was in effect, but there were no records for the SORTOUT data set, as indicated by an OUT count of 0 in message ICE054I.

System action: The program terminates.

Programmer response: Change the application to produce records for the SORTOUT data set. Alternatively, specify the NULLOUT=RC0 or NULLOUT=RC4 run-time option to override NULLOUT=RC16.

ICE209A NO DATA RECORDS FOR AN OUTFIL DATA SET - RC=16

Explanation: Critical. NULLOFL=RC16 was in effect for one or more OUTFIL data sets for which there were no data records, as indicated by a DATA count of 0 in message ICE227I for those OUTFIL data sets.

System action: The program terminates.

Programmer response: Change the application to produce records for the relevant OUTFIL data sets (which you can identify by a DATA count of 0 in message ICE227I). Alternatively, specify the NULLOFL=RC0 or NULLOFL=RC4 run-time option on the relevant OUTFIL statements to override NULLOFL=RC16.

ICE210I *xxxx* USED, LRECL = *n*, BLKSIZE = *n*, TYPE = *t* [(SDB)]

Explanation: Gives attribute and access method information for the OUTFIL data set with the indicated ddname, as follows:

- xxxx shows the access method used: EXCP, BSAM, or VSAM.
- LRECL=n shows the logical record length.
- BLKSIZE=n shows the block size (non-VSAM) or control interval size (VSAM).
- TYPE=t shows the record format where t is one or more of the following format indicators:
  - F - fixed-length records
  - V - EBCDIC or ASCII variable-length records
  - B - blocked records
  - S - standard for fixed-length records or spanned for variable-length records

  - A - ANSI control characters
- (SDB) shows that the system-determined optimum block size was used.

System action: None.

Programmer response: None.

ICE211I OLD OUTFIL STATEMENT PROCESSING USED

Explanation: This OUTFIL statement did not have any of the following operands: FNAMES, FILES, STARTREC, ENDREC, SAMPLE, INCLUDE, OMIT, SAVE, PARSE, OUTREC, BUILD, VTOF, CONVERT, VLFILL, OVERLAY, IFTHEN, FTOV, VLTRIM, REPEAT, SPLIT, SPLITBY, SPLIT1R, NULLOFL, LINES, HEADER1, TRAILER1, HEADER2, TRAILER2, SECTIONS, NODETAIL, BLKCCH1, BLKCCH2, BLKCCT1 or REMOVECC. For compatibility, this OUTFIL statement was treated as an ″old″ OUTFIL statement and all of its operands were ignored.

System action: Processing continues, but OUTFIL data sets are not associated with this OUTFIL statement. If the Blockset technique is not selected, control statement errors could result from continuation of this OUTFIL statement.

Programmer response: None, unless this is not an old OUTFIL statement, in which case valid operands from the list above should be specified.

ICE212A MATCH NOT FOUND FOR {*INREC|*OUTREC|*ddname*} IFTHEN *n* CHANGE FIELD: {POSITION *p*|PARSE *x*}

Explanation: Critical. In the FIELDS, BUILD, OVERLAY, IFTHEN BUILD or IFTHEN OVERLAY operand of an INREC or OUTREC statement, or the OUTREC, BUILD, OVERLAY, IFTHEN BUILD or IFTHEN OVERLAY operand of an OUTFIL statement, a CHANGE parameter was specified without a NOMATCH parameter and a fixed field (p,m) value or parsed field (%x) value did not match any of the find constants.

POSITION *p* indicates that a fixed field (p,m) did not match any of the find constants. *p* is the starting position of the input field. PARSE *x* indicates that a parsed field (%x) did not match any of the find constants.

The specific cause of the error is identified as follows:

- *INREC and n=0 indicates that a change field in the FIELDS, BUILD or OVERLAY operand of the INREC statement caused the error.
- *OUTREC and n=0 indicates that a change field in the FIELDS, BUILD or OVERLAY operand of the OUTREC statement caused the error.
- *ddname* and n=0 indicates that a change field in the OUTREC, BUILD or OVERLAY operand of an OUTFIL statement caused the error. *ddname* identifies the first data set in the associated OUTFIL group.
- *INREC and n>0 indicates that a change field in an IFTHEN BUILD or IFTHEN OVERLAY operand of the INREC statement caused the error. *n* identifies the number of the associated IFTHEN clause (starting at 1 for the first IFTHEN clause in the INREC statement).
- *OUTREC and n>0 indicates that a change field in an IFTHEN BUILD or IFTHEN OVERLAY operand of the OUTREC statement caused the error. *n* identifies the number of the associated IFTHEN clause (starting at 1 for the first IFTHEN clause in the OUTREC statement).
- *ddname* and n>0 indicates that a change field in an IFTHEN BUILD or IFTHEN OVERLAY operand of an OUTFIL statement caused the error. *ddname* identifies the first data set in the associated OUTFIL group. *n* identifies the number of the associated IFTHEN clause (starting at 1 for the first IFTHEN clause in the OUTFIL statement).

System action: The program terminates when the first change field is encountered for which a match is not found.

Programmer response: Correct the lookup table specified with the CHANGE parameter, or use the NOMATCH parameter to specify a constant, input field, or parsed field to be used as the output field if a match is not found. The use of a constant such as NOMATCH=(C'**') can be helpful in identifying all fixed field values or parsed field values for which a match is not found.

ICE213A INVALID OUTFIL STATEMENT OPERAND

Explanation: Critical. An OUTFIL control statement contained an invalid keyword operand.

System action: The program terminates.

Programmer response: Make sure that the OUTFIL control statement does not contain an invalid keyword operand.

ICE214A DUPLICATE, CONFLICTING, OR MISSING OUTFIL STATEMENT OPERANDS

Explanation: Critical. One of the following errors was found in an OUTFIL statement:

- An operand, other than IFTHEN, was specified twice.

  Example:

  ```
  OUTFIL STARTREC=5,STARTREC=10
  ```
- INCLUDE and OMIT, INCLUDE and SAVE, or OMIT and SAVE were specified.

  Example:

  ```
  OUTFIL INCLUDE=ALL,SAVE
  ```

- VTOF and CONVERT were specified.

  Example:

  ```
  OUTFIL VTOF,CONVERT
  ```

- FTOV and VTOF, FTOV and CONVERT, or FTOV and VLFILL were specified.

  Example:

  ```
  OUTFIL FTOV,VLFILL=C'*'
  ```

- PARSE and IFTHEN, OUTREC and BUILD, OUTREC and OVERLAY, OUTREC and IFTHEN, BUILD and OVERLAY, BUILD and IFTHEN, or OVERLAY and IFTHEN were specified.

  Example:

  ```
  OUTFIL BUILD=(1,20),OVERLAY=(10:C'A')
  ```

- IFOUTLEN and PARSE, IFOUTLEN and BUILD, IFOUTLEN and OUTREC or IFOUTLEN and OVERLAY were specified.

  Example:

  ```
  OUTFIL OVERLAY=(21:C'A'),IFOUTLEN=50
  ```

- For an IFTHEN clause, WHEN was not specified.

  Example:

  ```
  OUTFIL IFTHEN=(OVERLAY=(10:C'A'))
  ```

- For an IFTHEN clause, WHEN=INIT, WHEN=(logexp), or WHEN=NONE was specified without PARSE, BUILD, or OVERLAY

  Example:

  ```
  OUTFIL IFTHEN=(WHEN=(5,1,CH,EQ,C'1'),HIT=NEXT)
  ```

- For an IFTHEN clause, WHEN=(logexp), WHEN=ANY, or WHEN=NONE was specified with PARSE, but without BUILD or OVERLAY.

  Example:

  ```
  OUTFIL IFTHEN=(WHEN=NONE,
         PARSE=(%01=(FIXLEN=5,ENDBEFR=BLANKS)))
  ```

- For an IFTHEN clause, WHEN=INIT and BUILD with / were specified

  Example:

  ```
  OUTFIL IFTHEN=(WHEN=INIT,BUILD=(1,25,/,26,25))
  ```

- For an IFTHEN clause, BUILD with / and HIT=NEXT were specified.

  Example:

  ```
  OUTFIL IFTHEN=(WHEN=(21,1,CH,EQ,C'A'),
       BUILD=(1,25,/,26,25),HIT=NEXT)
  ```

- An IFTHEN clause with WHEN=INIT was preceded by an IFTHEN clause with WHEN=(logexp), WHEN=ANY or WHEN=NONE.

  Example:

  ```
  OUTFIL IFTHEN=(WHEN=(5,2,CH,EQ,C'AA'),
                 OVERLAY=(10:C'A')),
         IFTHEN=(WHEN=INIT,BUILD=(1,80))
  ```

- An IFTHEN clause with WHEN=NONE was followed by an IFTHEN clause with WHEN=INIT, WHEN=WHEN=(logexp), or WHEN=ANY.

  Example:

  ```
  OUTFIL IFTHEN=(WHEN=NONE,OVERLAY=(10:C'A')),
         IFTHEN=(WHEN=ANY,BUILD=(1,80))
  ```

- The first IFTHEN clause with WHEN=ANY was not preceded by an IFTHEN clause with WHEN=(logexp).

  Example:

  ```
  OUTFIL IFTHEN=(WHEN=INIT,OVERLAY=(10:C'A')),
         IFTHEN=(WHEN=ANY,BUILD=(1,80))
  ```

- An IFTHEN clause with WHEN=ANY and without HIT=NEXT was followed by an IFTHEN clause with WHEN=ANY.

  Example:

  ```
  OUTFIL IFTHEN=(WHEN=(5,1,CH,EQ,C'1'),
            OVERLAY=(10:C'A'),HIT=NEXT),
         IFTHEN=(WHEN=(5,1,CH,EQ,C'2'),
            OVERLAY=(10:C'B'),HIT=NEXT),
         IFTHEN=(WHEN=ANY,OVERLAY=(28:C'ABC')),
         IFTHEN=(WHEN=ANY,BUILD=(1,80))
  ```

System action: The program terminates.

Programmer response: Check the OUTFIL control statement for the errors indicated in the explanation and correct the errors.

ICE215A SPECIFIED FEATURE IS ONLY SUPPORTED BY OUTFIL STATEMENT

Explanation: Critical. One of the following, which is supported only by the OUTFIL control statement, was specified on an INREC or OUTREC control statement:

- A / in a FIELDS, BUILD, or IFTHEN BUILD operand
- A VTOF or CONVERT operand
- An FTOV operand
- A VLFILL=byte operand
- A VLTRIM=byte operand

System action: The program terminates.

Programmer response: If a / was specified, use a / in a BUILD or IFTHEN BUILD operand in an OUTFIL statement instead. If a VTOF, CONVERT, or VLFILL=byte operand was specified, use the same operand and BUILD in an OUTFIL statement instead. For example:

```
OUTFIL BUILD=(1,60),FTOV
```

If an FTOV or VLTRIM=byte operand was specified, use the same operand and BUILD or IFTHEN BUILD in an OUTFIL statement instead. For example:

```
OUTFIL VLTRIM=X'40',
  IFTHEN=(WHEN=(5,1,CH,EQ,C'A'),BUILD=(1,40,45))
```

ICE216A TOTAL LENGTH OF CONTROL FIELDS AND SUM FIELDS IS TOO LONG

Explanation: Critical. The total length of the SORT or MERGE control fields and SUM summary fields is too long for DFSORT to process, or the complexity of the application caused dynamic areas to exceed the storage allowed for them. Note that locale processing can significantly decrease the total length of the SORT or

MERGE fields DFSORT can process.

System action: The program terminates.

Programmer response: Remove one or more control fields or summary fields, or make them shorter.

ICE217A *n* **BYTE VARIABLE RECORD IS LONGER THAN** *m* **BYTE MAXIMUM FOR {***ddname* **| E15 | E32 | E35}**

Explanation: Critical. A variable-length record was larger than the maximum length specified or defaulted. The values shown in the message are as follows:

- *n* is the length of the variable-length record
- *m* is the maximum length specified or defaulted
- *ddname* indicates that the variable-length record, read from a SORTIN (sort or copy) or SORTINnn (merge) data set or to be written to the SORTOUT or OUTFIL data set, had a length greater than the LRECL specified or defaulted. For SORTINnn, the ddname is shown as 'SORTINNN'. For SORTIN, SORTOUT, or OUTFIL, the actual ddname is shown.
- E15 indicates that an E15 exit passed a variable-length record with a length greater than the L2 value specified or defaulted
- E32 indicates that an E32 exit passed a variable-length record with a length greater than the L1 value specified or defaulted
- E35 indicates that an E35 exit passed a variable-length record with a length greater than the L3 value specified or defaulted

System action: The program terminates.

Programmer response: If variable-length records are longer than the LRECL of the SORTOUT or OUTFIL data set and you want to truncate long records to the LRECL, use the VLLONG option. Keep in mind that the use of VLLONG causes data at the end of long records to be lost.

If variable-length output records are longer than 32757 bytes for the SORTOUT or OUTFIL data set, you can specify an LRECL up to 32767 bytes if you also specify RECFM=VBS.

If variable-length records are longer than the LRECL of the OUTFIL data set, you may be able to use the VLTRIM=byte parameter of the OUTFIL statement to shorten them. For example, VLTRIM=C' ' will remove trailing blanks from the records.

If variable-length output records are longer than the LRECL of the SORTIN data set, the first data set in the concatenation does not have the largest LRECL as required. Either specify the data set with the largest LRECL first in the SORTIN concatenation, or specify LRECL=n as a parameter on the SORTIN DD statement where n is equal to or greater than the length of the longest record.

If variable-length records from an E15, E32, or E35 exit are longer than the corresponding Ln value, correct the RDW or Ln value, as appropriate.

You can use the ICETOOL STATS operator with ON(VLEN) to show the length of the longest variable-length record. You can use the ICETOOL DISPLAY operator with ON(VLEN) to show the length of each variable-length record.

ICE218A *n* **BYTE VARIABLE RECORD IS SHORTER THAN** *m* **BYTE MINIMUM FOR [***ddname***] FIELDS**

Explanation: Critical. A variable-length record was shorter than 4 bytes, or was too short to contain all specified fields. The values shown in the message are as follows:

- *n* is the length of the variable-length record
- *m* is 4 or the minimum length needed for all fields specified
- if *ddname* is blank, the short record was detected during non-OUTFIL processing. Otherwise, the first data set in the OUTFIL group for which the short record was detected is indicated by ddname.

System action: The program terminates.

Programmer response: If a variable-length record was too short to contain all INREC, OUTREC, or OUTFIL fields, use an INREC or OUTREC statement with operands OVERLAY or IFTHEN as appropriate, or an OUTFIL statement with operands OVERLAY, IFTHEN, or BUILD and VLFILL=C 'x' or VLFILL=X 'yy' as appropriate, to prevent DFSORT from terminating.

If the variable-length record was too short to contain all SORT, MERGE, or SUM fields, use the VLSHRT option to prevent DFSORT from terminating.

If a variable-length record was too short to contain all INCLUDE or OMIT fields, use the VLSCMP or VLSHRT option to prevent DFSORT from terminating.

See the Programmer Response for message ICE015A for additional information.

ICE219I **DUPLICATE DDNAME** *ddname* **IGNORED**

Explanation: The indicated ddname was specified previously in an OUTFIL statement in this source. The OUTFIL specifications are not used for this ddname, although they may be used for other ddnames in this OUTFIL statement that have not been previously specified in this source.

This message is only printed for the first 128 duplicate ddnames found.

System action: Processing continues. The first OUTFIL ddname encountered in a source is used (unless overridden by the use of the same ddname in an OUTFIL statement in a higher source).

Programmer response: No action is necessary if the OUTFIL processing specified for the duplicate ddname is not needed. Otherwise, correct the ddname conflicts.

ICE220A *ddname* **COULD NOT BE USED - REASON CODE IS** *nn*

Explanation: Critical. The OUTFIL data set associated with *ddname* could not be processed. *nn* is the reason code value, as follows:

1 An attempt to OPEN the OUTFIL tape data set to retrieve its attributes failed.

2 The OUTFIL data set resided on an unsupported device.

3 An attempt to read the DSCB for the OUTFIL data set failed.

4 The OUTFIL data set was BDAM.

System action: The program terminates.

Programmer response: Correct the situation indicated by the reason code.

ICE221A **INVALID FIELD OR CONSTANT IN {*INCLUDE|*OMIT|*INREC|*OUTREC|***ddname***} IFTHEN** *n* **CONDITION** *m*

Explanation: Critical. An error was detected in a COND, INCLUDE, OMIT, or IFTHEN WHEN operand. The specific cause of the error is identified as follows:

- *INCLUDE indicates that the COND operand of the INCLUDE statement caused the error. *n* is 0.
- *OMIT indicates that the COND operand of the OMIT statement caused the error. *n* is 0.
- *ddname* and n=0 indicates that the INCLUDE or OMIT operand of an OUTFIL statement caused the error. *ddname* identifies the first data set in the associated OUTFIL group.
- *INREC indicates that an IFTHEN WHEN operand of the INREC statement caused the error. *n* identifies the number of the associated IFTHEN clause (starting at 1 for the first IFTHEN clause in the INREC statement).
- *OUTREC indicates that an IFTHEN WHEN operand of the OUTREC statement caused the error. *n* identifies the number of the associated IFTHEN clause (starting at 1 for the first IFTHEN clause in the OUTREC statement).
- *ddname* and n>0 indicates that an IFTHEN WHEN operand of an OUTFIL statement caused the error. *ddname* identifies the first data set in the associated OUTFIL group. *n* identifies the number of the associated IFTHEN clause (starting at 1 for the first IFTHEN clause in the OUTFIL statement).

One of the following errors was detected:

- the length for a field with a format other than SS was greater than 256
- the length for a PD field not used with NUM was 256
- the length for a PD0 field was less than 2 or greater than 8
- the length for a CSF or FS field not used with NUM was greater than 32
- the length for a UFF or SFF field was greater than 44
- the length for a CSL, CST, ASL, or AST field was 1
- the decimal constant for an FI field was greater than +9223372036854775807 or less than -9223372036854775808
- the decimal constant for a BI field was greater than 18446744073709551615 or less than +0
- the number of digits (including leading zeros) in the decimal constant for an FI or BI field was greater than 31
- the length for a Y2 field was not 2 for Y2C, Y2Z, Y2P or Y2S, or 1 for Y2D or Y2B, or 3–6 for Y2T or Y2W, or 2–3 for Y2U or Y2X, or 3–4 for Y2V or Y2Y
- a Y2 field was compared to another Y2 field with a different number of non-year digits
- a Y2 field was compared to a Y constant with a different number of non-year digits
- a Y2 field other than Y2S, Y2T or Y2W was compared to Y'LOW', Y'BLANKS' or Y'HIGH'
- a Y2 field was compared to a constant that was not a Y constant.

m indicates the number of the relational condition in which the error was found (starting at 1 for the first relational condition). For example, in

```
INCLUDE COND=(5,2,CH,EQ,8,2,CH,OR,
  11,257,BI,EQ,301,257,BI)
```

the second relational condition (after the OR) has the error described in the first bullet above, so *m* is 2.

System action: The program terminates.

Programmer response: Correct the field length or constant in error in relational condition m.

ICE222A *n* **BYTE FIXED RECORD LENGTH IS NOT EQUAL TO** *m* **BYTE LRECL FOR** *ddname*

Explanation: Critical. The LRECL specified or retrieved for the fixed-length OUTFIL data set was not equal to the computed length of the output records for that data set. You cannot use the LRECL value to pad the OUTFIL records or to truncate the records produced by OUTREC, BUILD, OVERLAY, IFTHEN BUILD, or IFTHEN OVERLAY operand processing. The values shown in the message are as follows:

- *n* is the computed length of the output records for the OUTFIL group
- *m* is the specified or retrieved LRECL of the OUTFIL data set
- *ddname* indicates the OUTFIL data set for which padding or truncation was required

System action: The program terminates.

Programmer response: Take one of these actions as appropriate:

- Do not set the LRECL explicitly. Instead, let DFSORT set the LRECL to the computed record length.
- If you are using IFTHEN operands, specify IFOUTLEN=m. (Remember to allow an extra byte for OUTFIL report data sets for the ANSI carriage control character unless you specify the REMOVECC operand.)
- If you are not using IFTHEN operands, ensure that the computed length for the BUILD or OVERLAY operand is equal to m. (Remember to allow an extra byte for OUTFIL report data sets for the ANSI carriage control character unless you specify the REMOVECC operand.)

ICE223A REPORT FIELD ERROR

Explanation: Critical. The LINES, HEADER1, TRAILER1, HEADER2, TRAILER2, or SECTIONS parameter of an OUTFIL statement contained an invalid column, report element, position, length, format, keyword, pattern, sign, or constant. Some common errors are :

- A 0 value was used.
- A null value was used where it was not permitted.
- A null string, pattern, or sign was used.
- A column was greater than 32752, preceded / or n/ (new line), or was followed by another column.
- A column overlapped the previous output field in the report record (a missing new line (/ or n/) to end the current report record and start the next one can cause this error).
- A position plus length was greater than 32753.
- DATE=(abcd) or DATENS(abc) was specified with a, b or c not M, D, Y or 4, with M, D, Y or 4 specified more than once, or with Y and 4 both specified.
- YDDD=(abc) or YDDDNS=(ab) was specified with a or b not D, Y or 4, with D, Y or 4 specified more than once, or with Y and 4 both specified.
- TIME=(abc) or TIMENS=(ab) was specified with ab not 12 or 24.
- The length for an input field or section break field was greater than 256 bytes.
- A repetition factor was greater than 4095 for a blank, character string, or hexadecimal string report element, or greater than 255 for a blank lines report element or a section skip line count.
- A character or hexadecimal constant was longer than 256 bytes.
- An invalid digit or an odd number of digits was specified for a hexadecimal string.
- The length for a statistics field was greater than 8 for BI or FI, 16 for PD, 31 for ZD, 32 for CSF/FS, or 44 for UFF or SFF.
- The length for a statistics field was not 4 or 8 for FL.
- More than 31 digits or 44 characters were specified in an edit pattern.
- SIGNz (where z is not S) was specified with Mn or without EDIT or EDxy.
- x, y, or z in EDxy or SIGNz were the same character.
- The value for LENGTH was greater than 44.
- The value for LINES was greater than 255.
- A section break field was not followed by SKIP, HEADER3, or TRAILER3.
- A statistics field was specified in HEADER1, HEADER2, or HEADER3.
- HEADER3, TRAILER3, SKIP, or PAGEHEAD was specified more than once after a section break field.

System action: The program terminates.

Programmer response: Correct the invalid value.

ICE224A *ddname* **CANNOT BE USED FOR A REPORT - RECFM WITHOUT 'A' SPECIFIED, OR VSAM DATA SET**

Explanation: Critical. The OUTFIL report specified for the OUTFIL data set associated with *ddname* cannot be produced because either:

- the OUTFIL data set is VSAM, or
- the record format specified or retrieved for the OUTFIL data set does not contain 'A' (ANSI carriage control characters).

System action: The program terminates.

Programmer response: If the OUTFIL data set is VSAM, it cannot be used for a report. If the OUTFIL data set is non-VSAM, take one of the following actions:

- Let DFSORT set the RECFM appropriately.
- Specify a RECFM that includes 'A'.
- Specify the REMOVECC parameter on the OUTFIL statement so 'A' is not needed in the RECFM.

ICE225A *ddname* **: LRECL =** *n* **IS INVALID**

Explanation: Critical. The specified or computed LRECL (*n*), for the OUTFIL data set associated with *ddname*, is invalid for one of the following reasons:

- LRECL=X was specified (*n* will be 32768)
- n is less than the minimum allowed: 1 (not a report) or 2 (report) for a fixed-length record data set or 5 (not a report) or 6 (report) for a variable-length record data set
- n is greater than the maximum allowed: 32760 for a fixed-length record data set, 32756 for a variable-length record data set, 32767 for a spanned record data set or 32752 for a VSAM variable-length record data set

System action: The program terminates.

Programmer response: If LRECL=X was specified, specify the actual maximum LRECL instead. If the LRECL is less than the minimum allowed, increase it. If the LRECL is greater than the maximum allowed, decrease it.

ICE226A BLKSIZE = *n* IS INVALID (LRECL = *m*)

Explanation: Critical. The specified or computed block size (*n*), for the OUTFIL data set associated with ddname, is invalid for one of the following reasons (*m* is the associated LRECL):
- *n* is less than the minimum allowed: 1 (not a report) or 2 (report) for a fixed-length record data set or 9 (not a report) or 10 (report) for a variable-length record data set
- *n* is less than *m* for a fixed-length record data set
- *n* is not a multiple of *m* for a fixed-length record data set
- *n* is not equal to *m* for an unblocked fixed-length record data set
- *n* is less than *m* + 4 for a variable-length record data set
- *n* is greater than the track capacity of the device on which the data set resides

System action: The program terminates.

Programmer response: Change the block size to a valid value.

ICE227I *ddname* : DELETED = *n*, REPORT = *n*, DATA = *n*

Explanation: For the OUTFIL data set associated with *ddname*, lists the number of OUTFIL input records deleted (that is, not written to this data set) and the number of report and data records written.

System action: None.

Programmer response: None.

ICE228I *ddname* : TOTAL IN = *n*, TOTAL OUT = *n*

Explanation: For the OUTFIL data set associated with *ddname*, lists the number of OUTFIL input records and the number of OUTFIL output records (report plus data) written.

System action: None.

Programmer response: None.

ICE229A *ddname* : LINES = *n* IN EFFECT, BUT *m* LINES REQUIRED

Explanation: Critical. For the OUTFIL group whose first data set is associated with *ddname*, the value specified or defaulted for LINES (*n*) is too low to contain the number of lines (*m*) required for the maximum of:
- The HEADER1 lines
- The TRAILER1 lines
- The sum of all lines for HEADER2, TRAILER2, HEADER3s, and TRAILER3s, and the data lines and blank lines produced from an input record.

System action: The program terminates.

Programmer response: Set LINES=*n* to a value equal to or greater than *m*, or reduce the number of lines required.

ICE230A *n* BYTE HEADER/TRAILER RECORD LENGTH EXCEEDS *m* BYTE LRECL FOR *ddname*

Explanation: Critical. The specified, retrieved, or computed LRECL for the OUTFIL data set was less than the computed length for the report records for that data set. The values shown in the message are as follows:
- *n* is the computed length of the report records for the OUTFIL group
- *m* is the specified, retrieved, or computed LRECL of the OUTFIL data set
- *ddname* indicated the OUTFIL data set for the report

System action: The program terminates.

Programmer response: Use the BUILD, OVERLAY, IFTHEN, or IFOUTLEN operand to force a length for the data records that is longer than any report record, and either let DFSORT compute and set the LRECL, or ensure that the computed LRECL is equal to the existing or specified LRECL. (Remember to allow an extra byte in the LRECL for the ANSI carriage control character, unless you specify the REMOVECC parameter.)

For example, if your data records are 40 bytes, but your longest report record is 60 bytes, you could use an OVERLAY parameter such as:

```
OUTFIL HEADER1=(51:DATE=(4MD/)),OVERLAY=(80:X)
```

DFSORT will then set the LRECL to 81 (1 byte for the ANSI carriage control character plus 80 bytes for the length of the data records), and pad the data records with blanks on the right.

ICE231I STORAGE USED FOR OUTFIL : BELOW 16M = *x*, ABOVE 16M = *y*

Explanation: Gives information related to the main storage used exclusively for OUTFIL processing. *x* is the storage used below 16 MB virtual and *y* is the storage used above 16 MB virtual.

System action: None.

Programmer response: None.

ICE232A *ddname*: SPLIT, SPLITBY, SPLIT1R OR REPEAT CANNOT BE USED FOR A REPORT

Explanation: Critical. For the OUTFIL group whose first data set is associated with *ddname*, a SPLIT, SPLITBY, SPLIT1R, or REPEAT parameter was specified along with one or more report parameters (LINES, HEADER1, TRAILER1, HEADER2, TRAILER2, SECTIONS or NODETAIL). The records of a report

cannot be repeated, or split among a group of OUTFIL data sets.

System action: The program terminates.

Programmer response: Remove either the SPLIT, SPLITBY, SPLIT1R, or REPEAT parameter or the report parameters.

ICE233A OUTFIL STATEMENTS CANNOT BE PASSED TO OR FROM AN EFS PROGRAM

Explanation: Critical. An OUTFIL statement was to be passed to an EFS program or an EFS program passed back an OUTFIL statement. DFSORT does not allow OUTFIL statements to be processed by an EFS program that requests them, and does not allow an EFS program to change other statements passed to it into OUTFIL statements.

System action: The program terminates.

Programmer response: Change the EFS program such that it does not request OUTFIL statements or change other statements it requests into OUTFIL statements.

ICE234A STARTREC, ENDREC, SAMPLE OR REPEAT VALUES ARE INCONSISTENT

Explanation: Critical. Specified values were inconsistent in one of the following ways:

- The n value of ENDREC=n is less than the n value of STARTREC=n. Example: STARTREC=10,ENDREC=9

  The n value of ENDREC=n must be equal to or greater than the n value of STARTREC=n. Example: STARTREC=10,ENDREC=10 to process record 10, or STARTREC=10,ENDREC=20 to process records 10-20.
- The n value of SAMPLE=n or SAMPLE=(n,m) is 1, which would result in processing every record instead of a sample of records. Example: SAMPLE=1.

  The n value of SAMPLE=n or SAMPLE=(n,m) must be equal to or greater than 2. Example: SAMPLE=2 to process records 1, 3, and so on, or SAMPLE=(10,2) to process records 1, 2, 11, 12, and so on.
- The n value of REPEAT=n is 1, which would result in no repetitions. Example: REPEAT=1.

  The n value of REPEAT=n must be equal to or greater than 2. Example: REPEAT=5 to repeat each output record five times.

System action: The program terminates.

Programmer response: Specify consistent values for STARTREC, ENDREC, SAMPLE, and REPEAT.

ICE235I OPTIONS: NULLOUT=s

Explanation: See ICE133I.

ICE241A {*INREC|*OUTREC|*ddname*} IFTHEN *n* OVERLAY COLUMN OVERLAPS RECORD DESCRIPTOR WORD

Explanation: Critical. For variable-length record processing, the OVERLAY or IFTHEN OVERLAY operand of an INREC, OUTREC or OUTFIL statement specified an overlay item that overlapped the record descriptor word (RDW). Only data bytes, which start at position 5 for variable-length records, can be overlaid. The specific cause of the error is identified as follows:

- *INREC and *n*=0 indicates that the OVERLAY operand of the INREC statement caused the error.
- *OUTREC and *n*=0 indicates that the OVERLAY operand of the OUTREC statement caused the error.
- *ddname* and *n*=0 indicates that the OVERLAY operand of an OUTFIL statement caused the error. *ddname* identifies the first data set in the associated OUTFIL group
- *INREC and *n*>0 indicates that an IFTHEN OVERLAY operand of the INREC statement caused the error. *n* identifies the number of the associated IFTHEN clause (starting at 1 for the first IFTHEN clause in the INREC statement).
- *OUTREC and *n*>0 indicates that an IFTHEN OVERLAY operand of the OUTREC statement caused the error. *n* identifies the number of the associated IFTHEN clause (starting at 1 for the first IFTHEN clause in the OUTREC statement).
- *ddname* and *n*>0 indicates that an IFTHEN OVERLAY operand of an OUTFIL statement caused the error. *ddname* identifies the first data set in the associated OUTFIL group. *n* identifies the number of the associated IFTHEN clause (starting at 1 for the first IFTHEN clause in the OUTFIL statement).

The error is one of the following:

- c: was not specified for the first OVERLAY item so the default of 1: was used for that item. Example:

  ```
  OVERLAY=(C'ABC')
  ```
- c: was specified for an overlay item with a value for c which was less than 5. Example:

  ```
  OVERLAY=(3:C'ABC')
  ```

System action: The program terminates.

Programmer response: Specify c: with a value of 5 or more for the first OVERLAY item. Ensure that c is 5 or more for any other c: values you specify. Example:

```
OVERLAY=(8:C'ABC',1,2,HEX,25:5C'*')
```

ICE242A Z/ARCHITECTURE MODE IS REQUIRED FOR THIS FUNCTION

Explanation: Critical. The indicated function cannot be used because it requires z/Architecture mode, but you are running in ESA/390 mode. z/Architecture mode is required for the following functions:
- FL (hexadecimal floating-point) conversion to integer with the INREC, OUTREC or OUTFIL statement.
- FL (hexadecimal floating-point) conversion to integer with the DISPLAY operator.

System action: The program terminates.

Programmer response: A $ marks the point at which the error was detected. Remove the function or switch to z/Architecture mode.

ICE243A PARSED FIELD DEFINITION ERROR

Explanation: Critical. The PARSE operand of an INREC, OUTREC or OUTFIL statement contained an invalid or incorrect definition for a parsed field. One of the following errors was found:
- More than 2 digits was specified for a parsed field definition (for example, %001 instead of %1 or %01). A parsed field must start with % and must have 1 digit, 2 digits or no digits.
- FIXLEN was not specified for %nn or %n.
- FIXLEN, ABSPOS, ADDPOS, SUBPOS or PAIR was specified more than once.
- ABSPOS and ADDPOS, ABSPOS and SUBPOS, or ADDPOS and SUBPOS were specified.
- FIXLEN, ABSPOS, ADDPOS or SUBPOS was specified with a value of 0 or a value greater than 32752.
- A null string was used.
- A character or hexadecimal constant was longer than 256 bytes.
- An invalid digit or an odd number of digits was specified for a hexadecimal string.
- An invalid keyword was specified.

System action: The program terminates.

Programmer response: Check the PARSE operand for the errors indicated in the explanation and correct the errors.

ICE244A DUPLICATE PARSED FIELD DEFINITION

Explanation: Critical. A %nn parsed field was defined previously (for example, %03 was defined twice), a %0n parsed field was defined previously as a %n parsed field (for example, both %3 and %03 were defined), or a %n parsed field was defined previously as a %0n parsed field (for example, both %03 and %3 were defined). Each %nn (including %n and %0n) parsed field must only be defined once in all PARSE operands of the INREC, OUTREC and OUTFIL statements.

System action: The program terminates.

Programmer response: A $ marks the duplicate parsed field. Use unique %nn parsed fields (%00-%99).

ICE245A PARSED FIELD NOT DEFINED FOR USE IN THIS OVERLAY, BUILD, FIELDS OR OUTREC OPERAND

Explanation: Critical. An OVERLAY, BUILD, FIELDS or OUTREC operand, or an IFTHEN OVERLAY or IFTHEN BUILD suboperand, of an INREC, OUTREC or OUTFIL statement specified a parsed field (%nn) that was not previously defined for use with that operand or suboperand. %nn parsed fields must be defined and used as follows:
- %nn used in INREC OVERLAY/BUILD/FIELDS: The %nn parsed field must be defined in the INREC statement in a PARSE operand that precedes the OVERLAY, BUILD or FIELDS operand.
- %nn used in OUTREC OVERLAY/BUILD/FIELDS: The %nn parsed field must be defined in the OUTREC statement in a PARSE operand that precedes the OVERLAY, BUILD or FIELDS operand.
- %nn used in OUTFIL OVERLAY/BUILD/OUTREC: The %nn parsed field must be defined in the same OUTFIL statement in a PARSE operand that precedes the OVERLAY, BUILD or OUTREC operand.
- %nn used in INREC IFTHEN OVERLAY/BUILD: The %nn parsed field must be defined in the INREC statement in an IFTHEN PARSE suboperand of a WHEN=INIT clause, or in an IFTHEN PARSE suboperand of the same clause, before the OVERLAY or BUILD suboperand.
- %nn used in OUTREC IFTHEN OVERLAY/BUILD: The %nn parsed field must be defined in the OUTREC statement in an IFTHEN PARSE suboperand of a WHEN=INIT clause, or in an IFTHEN PARSE suboperand of the same clause, before the OVERLAY or BUILD suboperand.
- %nn used in OUTFIL IFTHEN OVERLAY/BUILD: The %nn parsed field must be defined in the same OUTFIL statement in an IFTHEN PARSE suboperand of a WHEN=INIT clause, or in an IFTHEN PARSE suboperand of the same clause, before the OVERLAY or BUILD suboperand.

The following are some examples of using %nn parsed fields **correctly**:

```
* %00 defined in PARSE and used in BUILD.
  INREC PARSE=(%00=(FIXLEN=IO,ENDBEFR=C',')),
        BUILD=(%00)

* %01 defined in WHEN=INIT clause and used
* in WHEN=(logexp) clause.
* %02 defined in WHEN=(logexp) clause
* and used in that clause.
  OUTREC IFTHEN=(WHEN=INIT,
            PARSE=(%01=(ENDBEFR=C',',
              FIXLEN=8))),
          IFTHEN=(WHEN=(5,1,CH,EQ,C'A'),
            PARSE=(%02=(STARTAFT=C('',
```

```
|              FIXLEN=12)),
|            BUILD=(%01,%02))
|
| * %03 defined and used for OUTFIL OUT1.
| * %04 defined and used for OUTFIL OUT2.
|   OUTFIL FNAMES=OUT1,INCLUDE=(8,1,CH,EQ,C'A'),
|          PARSE=(%03=(FIXLEN=IO,ENDBEFR=C',')),
|          OVERLAY=(21:%03)
|   OUTFIL FNAMES=OUT2,SAVE,
|          PARSE=(%04=(FIXLEN=8,ENDBEFR=C':')),
|          OVERLAY=(31:%04)
```

| The following are some examples of using %nn parsed fields **incorrectly**:

```
| * %00 used in BUILD, but defined in PARSE
| * after BUILD.
|   INREC BUILD=(%00),
|         PARSE=(%00=(FIXLEN=IO,ENDBEFR=C','))
|
| * %02 used in WHEN=NONE clause but not defined
| * in that clause or in a WHEN=INIT clause
|   OUTREC IFTHEN=(WHEN=(5,1,CH,EQ,C'A'),
|             PARSE=(%02=(STARTAFT=C'(',
|               FIXLEN=12)),
|             BUILD=(%02)),
|           IFTHEN=(WHEN=NONE,
|             BUILD=(1,20,%02))
|
| * %03 used in OUTFIL for OUT2, but defined
| * in OUTFIL for OUT1.
|   OUTFIL FNAMES=OUT1,INCLUDE=(8,1,CH,EQ,C'A'),
|          PARSE=(%03=(FIXLEN=IO,ENDBEFR=C',')),
|          OVERLAY=(21:%03)
|   OUTFIL FNAMES=OUT2,SAVE,
|          OVERLAY=(21:%03)
```

| **System action:** The program terminates.

| **Programmer response:** A $ marks the incorrectly used | %nn parsed field. Define the %nn parsed field correctly.

ICE250I VISIT http://www.ibm.com/storage/dfsort FOR DFSORT PAPERS, EXAMPLES AND MORE

Explanation: The DFSORT website at http://www.ibm.com/storage/dfsort is an important source of information about DFSORT. It contains papers, examples, online documents, news, tips, techniques and more that can help you get the most out of DFSORT.

System action: None.

Programmer response: Optional. Visit the DFSORT website to see what it has to offer.

ICE270I PROCESSING SYMNAMES STATEMENTS

Explanation: A SYMNAMES data set was specified indicating that symbol processing is to be performed. DFSORT or ICETOOL will process the SYMNAMES statements and show any SYMNAMES statements in error after ICE270I.

System action: None.

Programmer response: None.

ICE271I SYMNAMES IS EMPTY - SYMBOL SUBSTITUTION WILL NOT BE PERFORMED

Explanation: A SYMNAMES data set was specified, but was empty (that is, it contained no statements). Symbol substitution will not be performed.

System action: Processing continues without symbol substitution.

Programmer response: If symbol substitution is needed, add appropriate SYMNAMES statements to the specified SYMNAMES data set or specify a non-empty SYMNAMES data set.

ICE272A SYMBOL, VALUE OR SYNTAX IS INVALID

Explanation: Critical. The SYMNAMES statement has one of the following errors:

- The symbol starts with a number (0-9) or a hyphen (-).
- | The value contains an invalid parsed field. A valid | parsed field must be %nn (nn is 00 to 99) or %n (n is | 0 to 9).
- The symbol or value contains an invalid character. The valid characters are uppercase letters (A-Z), lowercase letters (a-z), numbers (0-9), the number sign (#), the dollar sign ($), the commercial at sign (@), the underscore(_), and the hyphen (-).
- The symbol, keyword or value is null (for example, symbol,,5,CH).
- The symbol, keyword or value is followed by or contains an invalid delimiter.
- p or m in p,m or p,m,f is 0 or greater than 32752 or contains a non-numeric character.
- q in POSITION,q is 0 or greater than 32752 or contains a non-numeric character.
- symbol in POSITION,symbol references a symbol that was not previously defined or which was previously defined without a valid position (for example, a symbol for a constant).
- n in SKIP,n is 0 or greater than 32752 or contains a non-numeric character.
- The decimal constant contains a non-numeric character other than a leading plus sign (+) or minus sign (-).
- An equal sign (=) is specified for p, m or f, but the *previous position*, *previous length* or *previous format*, respectively, was not established.
- f in p,m,f is not a valid format. The valid formats are AC, AQ, ASL, AST, BI, CH, CLO, CSF, CSL, CST, CTO, DC1, DC2, DC3, DE1, DE2, DE3, DT1, DT2, DT3, D1, D2, FI, FL FS, LS, OL, OT, PD, PD0, SFF, SS,

TC1, TC2, TC3, TC4, TE1, TE2, TE3, TE4, TM1, TM2, TM3, TM4, TS, UFF, Y2B, Y2C, Y2D, Y2DP, Y2P, Y2PP, Y2S, Y2T, Y2TP, Y2U, Y2UP, Y2V, Y2VP, Y2W, Y2WP, Y2X, Y2XP, Y2Y, Y2YP, Y2Z and ZD, and lowercase or mixed case variations

- x in ALIGN,x is not a valid alignment. The valid alignments are H, F, D, h, f and d.
- The character constant, system symbol constant, hexadecimal constant, or bit constant does not have an ending apostrophe after the string.
- The hexadecimal constant is null (X'') or contains an odd number of digits (for example, X'123').
- The hexadecimal constant contains an invalid character. The valid characters are 0-9, A-F and a-f.
- The bit constant is null (B'') or contains a number of bits that is not a multiple of 8 (for example, B'1010').
- The bit constant contains an invalid character. The valid characters are . (period), 0 and 1.

System action: The program terminates.

Programmer response: Correct the symbol, value or syntax error.

ICE273A SYMBOL OR VALUE IS TOO LONG

Explanation: Critical. The SYMNAMES statement has one of the following errors:

- The symbol is longer than 50 characters.
- The value contains a number longer than 31 significant digits.
- The constant contains a string longer than 64 characters.

System action: The program terminates.

Programmer response: Specify a symbol, value or string that is less than or equal to the allowed limit.

ICE274A NEXT POSITION TO BE USED FOR * IS LARGER THAN 32752

Explanation: Critical. The SYMNAMES statement specifies an asterisk (*) for p requesting that the *next position* be used for p. However, the *next position* is greater than the limit for p of 32752.

Calculation of the *next position* will be discontinued once a SYMNAMES statement error has been detected. Therefore, ICE274A is only issued once and only if a previous SYMNAMES processing error was not detected.

System action: The program terminates.

Programmer response: Specify a value for p that is less than or equal to 32752. You can do this directly by specifying a number for p or indirectly by specifying * for p and ensuring that the *next position* is less than or equal to 32752. See *z/OS DFSORT Application Programming Guide* for an explanation of how the *next position* is set.

ICE275A DUPLICATE SYMBOL

Explanation: Critical. The symbol was used in a previous SYMNAMES statement. A symbol can only be used once in SYMNAMES.

System action: The program terminates.

Programmer response: Remove or rename one of the duplicate symbols.

ICE276A RESERVED WORD - NOT ALLOWED FOR SYMBOL

Explanation: Critical. The SYMNAMES statement specifies a DFSORT/ICETOOL reserved word for the symbol. Reserved words cannot be used for symbols. The reserved words are as follows (uppercase only as shown): A, AC, ADD, ALL, AND, AQ, ASL, AST, BI, CH, CLO, COPY, COUNT, COUNT15, CSF, CSL, CST, CTO, D, DATE, DATE1, DATE1..., DATE2, DATE2..., DATE3, DATE3..., DATE4, DC1, DC2, DC3, DE1, DE2, DE3, DIV, DT1, DT2, DT3, D1, D2, E, F, FI, FL, FS, H, HEX, LS, MAX, MIN, MOD, MUL, Mn, Mnn, NONE, NUM, OL, OR, OT, PAGE, PAGEHEAD, PD, PDC, PDF, PD0, SEQNUM, SFF, SS, SUB, SUBCOUNT, SUBCOUNT15, TC1, TC2, TC3, TC4, TE1, TE2, TE3, TE4, TIME, TIME1, TIME1P, TIME2, TIME2P, TIME3, TIME3P, TM1, TM2, TM3, TM4, TS, UFF, VALCNT, VLEN, X, Y2x, Y2xx, Z, ZD, ZDC, and ZDF, where n is 0-9 and x is any character.

System action: The program terminates.

Programmer response: Use a symbol that is not one of the reserved words, such as a lowercase or mixed case version of the word being used. For example, you could use Valcnt (which is not a reserved word) instead of VALCNT (which is).

ICE279A ONE OR MORE ERRORS ENCOUNTERED DURING SYMNAMES PROCESSING

Explanation: Critical. DFSORT or ICETOOL detected one or more errors while processing SYMNAMES statements.

System action: The program terminates.

Programmer response: Correct each error indicated by a previous error message (ICE272A through ICE276A).

ICE280I ORIGINAL STATEMENTS FROM source FOLLOW

Explanation: This message is issued for each source specified. It is followed by a listing of the original DFSORT control statements before they are transformed by symbol processing. Source can be one of the following:

- DFSPARM or the ddname specified in ICEMAC option PARMDDN

- SYSIN
- xxxxCNTL
- PARMLIST

System action: None.

Programmer response: None.

ICE281I ORIGINAL OPERATOR STATEMENTS FOLLOW

Explanation: This message is followed by a listing of the original ICETOOL operator statements before they are transformed by symbol processing. The operator statements listed are from TOOLIN or the calling program's parameter list as indicated by message ICE632I.

System action: None.

Programmer response: None.

ICE282I PERFORMING SYMBOL SUBSTITUTION AS NEEDED

Explanation: DFSORT or ICETOOL successfully processed the SYMNAMES statements. ICETOOL will transform operators as necessary using symbol substitution. DFSORT will transform control statements as necessary using symbol substitution. Any operators or control statements with substitution errors will be shown after ICE282I.

System action: None.

Programmer response: None.

ICE283A SYMBOL, SYNTAX OR DELIMITER ERROR

Explanation: Critical. During symbol substitution processing, one of the following errors was found in a control statement or operator:

- A specified symbol was not defined in SYMNAMES.

  Note: Symbols are treated as case-sensitive: Frank, FRANK, and frank are three **different** symbols. Be careful to use symbols in control statements and operators exactly as they are defined in SYMNAMES.
- A specified symbol was longer than 50 characters.
- A specified symbol started with an invalid character or with a delimiter. Examples:

```
SORT FIELDS=(.Sym1,A)
* .Sym1 is invalid.  FIELDS=(Sym1,A) is valid.
INREC FIELDS=((Sym1)
* ((Sym1) is invalid.  FIELDS=(Sym1) is valid.
```

- A specified symbol contained an invalid character. Examples:

```
SUM FIELDS=(New-Field)
* New*Field is invalid. New_Field is valid.
* New-Field is valid.
OUTREC FIELDS=(A%B)
* A%B is invalid.
```

  Note: A symbol with an invalid character cannot be defined in SYMNAMES.
- A specified symbol was followed by an invalid delimiter. Examples:

```
SUM FIELDS=(Sym1 )
* (Sym1 ) is invalid.  FIELDS=(Sym1) is valid.
OUTFIL OUTREC=(Sym1/)
* Sym1/ is invalid.  OUTREC=(Sym1,/) is valid.
```

- A delimiter was misplaced. Examples:

```
VERIFY ON(,Sym1)
* (,Sym1) is invalid.  ON(Sym1) is valid.
VERIFY ON((5,4,PD)
* ((5,4,PD) is invalid.  ON(5,4,PD) is valid.
VERIFY ON(5,4,PD))
* (5,4,PD)) is invalid.  ON(5,4,PD) is valid.
```

- A syntax error caused a value to be misinterpreted as an undefined symbol. Example:

```
OUTREC FIELDS=(1,4,ZD,EDIT(IIIT))
```

  EDIT is misinterpreted as an undefined symbol because an equal sign is missing. The following is valid:

```
OUTFIL OUTREC=(1,4,ZD,EDIT=(IIIT))
```

System action: The program terminates.

Programmer response: Correct the symbol, syntax or delimiter error.

ICE284A SYMBOL SUBSTITUTION CAUSES OPERAND TO OVERFLOW COLUMN 72

Explanation: Critical. The operand contained a symbol. Substitution of the value for the symbol caused the operand to exceed 72 characters if the line was not continued or 71 characters if the line was continued. ICETOOL operators must be contained in columns 1-72, so the transformed operand cannot be processed by ICETOOL.

System action: The program terminates.

Programmer response: Use a symbol or value that will not cause the operand to overflow column 72. If necessary, change the value in the SYMNAMES statement for the symbol.

ICE285A BLOCKSET IS REQUIRED FOR WORK DATA SETS ON VOLUMES WITH MORE THAN 17476 CYLINDERS

Explanation: Critical. One or more JCL or dynamically allocated work data sets was allocated on a device with more than 17476 cylinders. Blockset is the only DFSORT technique that supports the use of such

devices for work data sets, but Blockset was not selected.

System action: The program terminates.

Programmer response: Rerun the job with a SORTDIAG DD statement to get message ICE800I, which indicates the reason Blockset could not be used. If possible, remove the condition preventing the use of Blockset. Alternatively, ensure that work data sets are not allocated on devices with more than 17476 cylinders for this application.

ICE287A ONE OR MORE ERRORS ENCOUNTERED DURING SYMBOL SUBSTITUTION

Explanation: Critical. DFSORT or ICETOOL detected one or more errors during symbol substitution processing.

System action: The program terminates.

Programmer response: Correct each error indicated by a previous error message (ICE283A or ICE284A).

ICE600I DFSORT ICETOOL UTILITY RUN STARTED

Explanation: Indicates the start of the ICETOOL run.

System action: None.

Programmer response: None.

ICE601I DFSORT ICETOOL UTILITY RUN ENDED - RETURN CODE: *nn*

Explanation: Indicates the end of the ICETOOL run and the highest return code encountered. Message ICE602I gives the return code for each operation.

System action: None.

Programmer response: None if *nn* is 00. Otherwise, check the ICE602I messages for non-zero return codes and take the action indicated.

ICE602I OPERATION RETURN CODE: *nn*

Explanation: Indicates the return code for this operation.

System action: If *nn* is greater than 04, this operation was terminated due to an error.

Programmer response: None if *nn* is 00, or if *nn* is 04 and message ICE651I was issued. Otherwise:

- If *nn* is 04, match the identifier in message ICE606I or ICE627I for this operation to the same identifier in DFSORT message ICE200I (in the DFSMSG data set) to find the set of DFSORT messages for this operation. If appropriate, correct the condition that caused the return code of 4 indicated in the relevant DFSORT informational message.
- If *nn* is 12, check the error messages for this operation in the TOOLMSG data set and correct the indicated errors.
- If *nn* is 16, match the identifier in message ICE606I or ICE627I for this operation to the same identifier in DFSORT message ICE200I (in the DFSMSG data set) to find the set of DFSORT messages for this operation. Correct the error indicated in the DFSORT error message.

ICE603I INFORMATION PRINTED IN *ddname* **DATA SET**

Explanation: The information requested for this operator was printed in the list data set with the indicated *ddname*. If the operation was terminated, the list data set can be incomplete.

System action: None.

Programmer response: None.

ICE604A ERROR IN KEYWORD, PARAMETER, OR DELIMITER

Explanation: Critical. The statement contained an error in an operand (keyword, parameter) or a delimiter was incorrect or missing. Some common errors are:

- A keyword or parameter was misspelled. Example: `ALLDUP` instead of `ALLDUPS`
- A keyword was used with an operator for which it is not valid. Example: NOSIGN was used with an operator other than VERIFY.
- A parameter or value was used with a keyword for which it is not valid. Example: VSAMTYPE(U) instead of VSAMTYPE(F) or VSAMTYPE(V).
- The cccc value for USING(cccc) was SYSc or was not four characters. Example: USING(ABC) instead of USING(ABCD) or USING(SYS1) instead of USING(SYX1).
- A left or right parenthesis was missing. Example: `FROM IN` instead of `FROM(IN)`
- A blank was used inside a parentheses. Example: `FROM( IN) instead of FROM(IN)`
- A continuation indicator (-) was used incorrectly. Example: `TO(OUT1,-` instead of `TO(OUT1,OUT2) -`
- Parameters were not separated by a comma or semicolon. Example: `ON(3:5:ZD)` instead of `ON(3,5,ZD)` or `ON(3;5;ZD)`
- A numeric value was specified incorrectly. Example: `ON(0,3,ZD)` instead of `ON(1,3,ZD)` or `LIMIT(+1)` instead of `LIMIT(1)`
- An operand that can be specified only once per operator was specified more than once. Example: `COPY TO(OUT1) TO(OUT2)` instead of `COPY TO(OUT1,OUT2)`
- Mutually exclusive operands were used. Example: `BLANK` and `PLUS` for `DISPLAY` or `WITHEACH` and `VLENOVLY` for `SPLICE`.
- A numeric value was too low or too high. Example: `LINES(9)` or `LINES(1000)`

- A string was not enclosed in apostrophes. Example: `HEADER(Revenue)`, `HEADER('Revenue)`, `HEADER('Revenue")`, or `HEADER("Revenue")`instead of `HEADER('Revenue')`
- A string exceeded the character limit allowed. Example:

```
TITLE('This string is longer
than the limit of 50 characters
for TITLE')
```

  Note: In your ICETOOL statement, the entire operand must be on *one* line.
- A parameter was specified incorrectly. Example: `DATE(DMY)` instead of `DATE(DMY.)`
- NOHEADER was used when HEADER(NONE) was required. Example: `HEADER('Name') NOHEADER` instead of `HEADER('Name') HEADER(NONE)`
- Mutually exclusive items were specified within an operand. Example: `DATE(YM4/)` instead of `DATE(YMD/)` or `DATE(DM4/)`
- A formatting item was specified for an operator other than DISPLAY or OCCUR. Example:

  `SELECT ON(1,5,ZD,A1)` instead of `SELECT ON(1,5,ZD)`
- A symbol was specified where it is not allowed. Example: `LINES(Max_Lines)` instead of `LINES(50)`
- Too many 9's were specified in the pattern for E'pattern'. Example:

  ON(VALCNT,E'999-999-999-999-999-9') instead of ON(VALCNT,E'99-999-999-999-999-9')
- Too many characters were specified in the pattern for E'pattern'. Example:

  ON(VALCNT,E'*999**999**999**999**999*') instead of ON(VALCNT,E'999**999**999**999**999*')
- A required parameter was missing. Example: ON(25,5) instead of ON(25,5,ZD)
- A parameter was specified where it isn't allowed. Example: WITH(25,5,ZD) instead of WITH(25,5)

System action: This operation is terminated.

Programmer response: A $ marks the point at which the error was detected. Correct the error.

ICE605A REQUIRED DD STATEMENT NOT FOUND: *ddname*

Explanation: Critical. A DD statement for the indicated *ddname* was not present. The ddname was required because:
- It was DFSMSG.
- It was TOOLIN and the ICETOOL Parameter List Interface was not used.
- It was specified in the FROM, TO, LIST, or DISCARD operand for this operator.
- USING(*xxxx*) was specified for this operator.

System action: If the ddname was TOOLIN or DFSMSG, ICETOOL terminates without performing any operations. Otherwise, this operation is terminated.

Programmer response: Supply a DD statement for the indicated ddname.

ICE606I DFSORT CALL *nnnn* **FOR {COPY|SORT} FROM** *ddname1* **TO {***ddname2***|E35 EXIT|OUTFIL} USING** *xxxx***CNTL {COMPLETED|TERMINATED}**

Explanation: Supplies information about a specific call to DFSORT for this operation as follows:

nnnn The call identifier for this call to DFSORT. You can match *nnnn* to the same *nnnn* in DFSORT message ICE200I in the DFSMSG data set to find the set of DFSORT messages for this call to DFSORT.

COPY The DFSORT copy function was used.

SORT The DFSORT sort function was used.

ddname1
The ddname of the data set that was copied or sorted.

ddname2
The ddname of the data set resulting from the copy or sort.

E35 EXIT
ICETOOL's E35 user exit was used.

OUTFIL
Either OUTFIL data sets were produced or an E35 user exit routine was used to dispose of all the output records

*xxxx*CNTL
The ddname of the DFSORT control data set used for the sort or copy.

COMPLETED
DFSORT did not detect any errors preventing completion of this ICETOOL operation.

TERMINATED
DFSORT detected an error which prevented completion of this ICETOOL operation.

System action: None.

Programmer response: You can use the information in this message to determine the result of this call to DFSORT. If DFSORT terminated, use the call identifier to find the set of DFSORT messages for this call (in the DFSMSG data set), and correct the error indicated in the DFSORT error message.

ICE607I STATISTICS FOR {(*p,m,f***)|(VLEN)}:**

Explanation: Indicates the field to which the statistics in subsequent messages ICE608I, ICE609I, and ICE648I apply. (*p,m,f*) or (VLEN) is a field you specified for this STATS operator.

System action: None.

Programmer response: None.

ICE608I **MINIMUM:** *snnnnnnnnnnnnnnn*, **MAXIMUM:** *snnnnnnnnnnnnnnn*

Explanation: Indicates the minimum and maximum for the field indicated in the ICE607I message preceding this message. Each value consists of a + or - sign and 15 decimal digits (padded with zeros on the left as needed). If the values could not be determined due to an error (as indicated in a previous message), asterisks were printed for the values.

Note: ICE648I messages are printed for the minimum and maximum values (instead of ICE608I) if either value contains more than 15 significant digits, or if LMSG is specified.

System action: None.

Programmer response: None unless asterisks were printed for the values, in which case you should correct the error indicated by the previous error message for this operation.

If you want all of the minimum and maximum values displayed with 31 digits, specify the LMSG keyword for this STATS operator.

ICE609I **AVERAGE:** *snnnnnnnnnnnnnnn*, **TOTAL:** *snnnnnnnnnnnnnnn*

Explanation: Indicates the average and total for the field indicated in the ICE607I message preceding this message. Each value consists of a + or - sign and 15 decimal digits (padded with zeros on the left as needed). If a value could not be determined due to an error (as indicated in a previous message), asterisks were printed for the value.

Note: ICE648I messages are printed for the average and total values (instead of ICE609I) if either value contains more than 15 significant digits, or if LMSG is specified.

System action: None.

Programmer response: None unless asterisks were printed for a value, in which case you should correct the error indicated by the previous error message for this operation.

If you want all of the average and total values displayed with 31 digits, specify the LMSG keyword for this STATS operator.

ICE610I **NUMBER OF UNIQUE VALUES FOR (*p,m,f*) | (VLEN):** *nnnnnnnnnnnnnnn*

Explanation: Indicates the count of unique values for this field, printed as 15 decimal digits (padded with zeros on the left as needed). (*p,m,f*) or (VLEN) is the field you specified for this UNIQUE operator.

System action: None.

Programmer response: None.

ICE611A **TOTAL FOR {(*p,m,f*) | (VLEN)} OVERFLOWED *nn* DECIMAL DIGITS**

Explanation: Critical. The total for the indicated ON(*p,m,f*) or ON(VLEN) field (STATS or DISPLAY) exceeded the number of digits (*nn*) allowed for it. *nn* is 31 for STATS, or 15 or 31 for DISPLAY, depending on the number of digits allowed for the total.

System action: This operation is terminated.

For a STATS operator, asterisks are printed in ICE648I messages for the average and total for this field.

For a DISPLAY operator, asterisks are printed for this field in any BAVERAGE, BTOTAL, AVERAGE, or TOTAL lines requested.

Programmer response: If *nn* is 15, specify an Ndd or Udd formatting item large enough to prevent overflow of the total (that is, use an appropriate Ndd or Udd value between 16 and 31).

If *nn* is 31 and you need the average or total for this field, use the STATS or DISPLAY operator for subsets of the data set that do not cause the total to overflow. Use the statistics for the subsets to determine the needed statistics for the field.

ICE612I **NO ERRORS FOUND IN STATEMENT**

Explanation: Printed in SCAN mode to indicate that no errors were found in the statement.

System action: DFSORT is not called in SCAN mode.

Programmer response: If SCAN mode was entered due to an error while in STOP mode, correct the error. If SCAN mode was entered due to a MODE SCAN statement, replace it with a MODE STOP or MODE CONTINUE statement.

ICE613A **REQUIRED KEYWORD MISSING:** *keyword*

Explanation: Critical. The indicated *keyword* was required for this operator, but was not specified. The required keywords and their operands for each operator are:

- COPY - FROM, and TO or USING
- COUNT - FROM, and EMPTY, NOTEMPTY, HIGHER, LOWER, EQUAL, or NOTEQUAL if RC4 is specified.
- DEFAULTS - LIST
- DISPLAY - FROM, ON, and LIST, and BREAK if BTITLE, BTOTAL, BMAXIMUM, BMINIMUM, or BAVERAGE is specified
- MODE - STOP, CONTINUE, or SCAN
- OCCUR - FROM, LIST, and ON(p,m,f), ON(p,m,HEX), or ON(VLEN)

- RANGE - FROM, ON, and HIGHER, LOWER, EQUAL, or NOTEQUAL
- SELECT- FROM, TO or DISCARD, ON, and ALLDUPS, NODUPS, HIGHER, LOWER, EQUAL, FIRST, LAST, FIRSTDUP, or LASTDUP
- SORT - FROM and USING
- SPLICE - FROM, TO, ON, and WITH
- STATS - FROM and ON
- UNIQUE - FROM and ON
- VERIFY - FROM and ON

System action: This operation is terminated.

Programmer response: Supply the indicated keyword or operand.

ICE614A INVALID OPERATOR

Explanation: Critical. The first keyword in the statement was not a valid operator. The valid operators are: COPY, COUNT, DEFAULTS, DISPLAY, MODE, OCCUR (or OCCURS), RANGE, SELECT, SORT, SPLICE, STATS, UNIQUE, and VERIFY.

A common cause of this error is a missing hyphen (-) on the previous line to indicate continuation.

System action: This operation is terminated.

Programmer response: A $ marks the point at which the error was detected. If an invalid operator was used, replace it with a valid operator. If this is a continuation line, use a hyphen after the last operand on the previous line.

ICE615A DDNAME USED MORE THAN ONCE: *ddname*

Explanation: Critical. This message was issued for one of the following reasons:
- TOOLIN, TOOLMSG, DFSMSG, or SSMSG was used as a *ddname* in a FROM, TO, LIST, or DISCARD operand for this operator.
- The indicated *ddname* was found more than once in a FROM, TO, LIST, or DISCARD operand for this operator.
- USING(*xxxx*) was specified and *xxxx*CNTL was used in a FROM or TO operand for this operator.

System action: This operation is terminated.

Programmer response: A $ marks the duplicate ddname. Replace it with a ddname that is unique for this operation and that is not TOOLIN, TOOLMSG, DFSMSG, or SSMSG.

ICE616A STATEMENT DOES NOT END AT OR BEFORE COLUMN 72

Explanation: Critical. The statement overran column 72, or a string contained unmatched quotes.

System action: This operation is terminated.

Programmer response: Recode the statement to end at or before column 72. Continuation can be indicated by a hyphen (-) after the operator or any operand. For example:

```
SORT FROM(INDD) -
     USING(ABCD) -
     TO(OUTPUT1,OUTPUT2,OUTPUT3)
```

Ensure that all quotes are matched as this message can also result from unmatched quotes in a string, for example, `TITLE('Status Report)`.

ICE617A RECORD COUNT OVERFLOWED 15 DECIMAL DIGITS

Explanation: Critical. The number of records processed by ICETOOL for this operation was greater than 999999999999999.

System action: This operation is terminated.

Programmer response: Perform the operation on subsets of the data set containing less than 999999999999999 records.

ICE618A INVALID (*p,m,f*) VALUE - RECORD: *nnnnnnnnnnnnnnn*

Explanation: Critical. ICE618A and ICE649A identify an invalid decimal value in a specified field. One of the following was found:
- An invalid digit (A-F) in a field specified for a DISPLAY, OCCUR, RANGE, STATS, or VERIFY operator.
- An invalid sign (0-9) in a field specified for a VERIFY operator (NOSIGN was not specified).

(*p,m,f*) is a field you specified for this operator.

The invalid value was found in record number *nnnnnnnnnnnnnnn* (prints as 15 decimal digits padded with zeros on the left as needed). For a DISPLAY, RANGE, STATS, or VERIFY operator, *nnnnnnnnnnnnnnn* is the input record number (that is, the relative record number). For an OCCUR operator, *nnnnnnnnnnnnnnn* is the sorted record number, and thus may not be useful.

h...h in the ICE649A message that follows this ICE618A message is the invalid value in hexadecimal.

System action:
- For a DISPLAY operator with an invalid BREAK value, this operation is terminated.
- For a DISPLAY operator with an invalid ON value, asterisks are printed for this value in the data line and in any statistics lines requested. If this incorrect value causes the limit for decimal values to be reached, this operation is terminated after the current record is printed in the list data set. Otherwise, processing continues.
- For an OCCUR or RANGE operator, this operation is terminated.

- For a STATS operator, asterisks were printed in messages ICE608I, ICE609I or ICE648I for minimum, maximum, average, and total for this field.
- For a VERIFY operator, if this bad value caused the limit for invalid decimal values to be reached, this operation is terminated. Otherwise, processing continues.

Programmer response: Correct the invalid digit or sign in the identified field. The VERIFY or DISPLAY operator can be used to print all the invalid values and their relative record numbers.

ICE619A INVALID LENGTH, FORMAT, OR COMBINATION FOR *operator* OPERATION

Explanation: Critical. One of the following conditions was detected in the parameters of an ON or BREAK operand:

- The format was invalid. Example: `ON(10,2,CST)` or `BREAK(10,2,FL)`
- The format was not allowed for this operator. Example: `VERIFY ON(10,2,BI)`
- The length was not within the range allowed for the format and operator. Example: `STATS ON(10,9,BI)`
- ON(VLEN) was specified for a VERIFY operator. Example: `VERIFY ON(VLEN)`
- ON(NUM) was specified for an operator other than DISPLAY. Example: `STATS ON(NUM)`
- ON(VALCNT) was specified for an operator other than OCCUR. Example: `DISPLAY ON(VALCNT)`
- ON(p,m,HEX) was specified for an operator other than DISPLAY or OCCUR. Example: `UNIQUE ON(5,4,HEX)`
- The length was not within the range allowed for ON(p,m,HEX). Example: `ON(5,1001,HEX)`
- BREAK(p,m,HEX) was specified. Example: `BREAK(5,4,HEX)`

System action: This operation is terminated.

Programmer response: A $ marks the point at which the error was detected. Correct the error.

ICE620A SUM OF POSITION AND LENGTH GREATER THAN *max*

Explanation: Critical. A specified field extended beyond the maximum position (max - 1) allowed for this operation.

System action: This operation is terminated.

Programmer response: A $ marks the point at which the error was detected. Change the position or length so that the field ends at or before position max - 1.

ICE621I DFSORT DETECTED AN ERROR AFTER E35 EXIT PROCESSING COMPLETED

Explanation: A call to DFSORT for this operation resulted in a return code of 16. However, because the error was detected by DFSORT after ICETOOL's E35 user exit completed its processing, the error did not prevent the completion of this ICETOOL operation.

System action: None.

Programmer response: None required, but you can examine the DFSORT messages in DFSMSG to determine the error detected by DFSORT and correct it if appropriate.

ICE622A EXPECTED CONTINUATION LINE NOT FOUND

Explanation: Critical. This message was issued for one of the following reasons:

- Continuation was indicated by a hyphen (-) after the last operand in the last line of the TOOLIN data set, but the continuation line was not found.
- A parameter list statement area did not contain an operator statement.
- Continuation was indicated by a hyphen (-) in the last line of a parameter list statement area, but the continuation line was not found.

System action: This operation is terminated.

Programmer response: Supply the continuation line.

ICE623A MAXIMUM NUMBER OF *keyword* KEYWORDS EXCEEDED

Explanation: Critical. Too many keywords of the indicated type were specified for this operator. The maximum number of HEADER fields is 20 for a DISPLAY operator or 10 for an OCCUR operator. The maximum number of WITH operands is 50 for a SPLICE operator. The maximum number of ON fields for each operator is:

- DISPLAY - 20
- OCCUR - 10
- RANGE - 1
- SELECT - 10
- SPLICE - 10
- STATS - 10
- UNIQUE - 1
- VERIFY - 10

System action: This operation is terminated.

Programmer response: A $ marks the point at which the error was detected. Reduce the number of indicated keywords for this operator to the maximum allowed. If necessary, use additional operators to handle all the required fields.

ICE624A MAXIMUM NUMBER OF TO DDNAMES EXCEEDED

Explanation: Critical. Too many TO ddnames were specified for this operator. The maximum number of TO ddnames for each operator is:
- COPY - 10
- SELECT - 1
- SORT - 10
- SPLICE - 1

System action: This operation is terminated.

Programmer response: A $ marks the point at which the error was detected. Reduce the number of TO ddnames for this operator to the maximum allowed. Use additional operators to handle all the data sets required.

ICE625A OPEN FAILED FOR *ddname* DATA SET

Explanation: Critical. The data set with the indicated *ddname* could not be opened. This occurred either for the data set specified for TOOLIN, SYMNAMES or SYMNOUT, or for the data set whose *ddname* was specified in the LIST operand for this operator.

System action: If the ddname is TOOLIN, SYMNAMES, or SYMNOUT, ICETOOL terminates without performing any operations. Otherwise, this operation is terminated.

Programmer response: Ensure that the attributes of the data set (record format, record length, block size) conform to the rules for that type of data set. See the ICETOOL chapter (TOOLIN or LIST operand data set) or Symbols chapter (SYMNAMES or SYMNOUT) in *z/OS DFSORT Application Programming Guide* for information on the rules.

ICE626A LIMIT FOR INVALID VALUES REACHED

Explanation: Critical. The limit for the number of invalid decimal values specified by the LIMIT operand (or defaulted to 200) was reached for this DISPLAY or VERIFY operator.

System action: This operation is terminated.

Programmer response: Correct the invalid decimal values indicated in the ICE618A and ICE649A messages, or set a higher LIMIT value.

ICE627I DFSORT CALL *nnnn* FOR {COPY|SORT} FROM *ddname1* TO {*ddname2*|E35 EXIT|OUTFIL} {COMPLETED|TERMINATED}

Explanation: Supplies information about a specific call to DFSORT for this operation as follows:

nnnn The call identifier for this call to DFSORT. You can match nnnn to the same nnnn in DFSORT message ICE200I in the DFSMSG data set to find the set of DFSORT messages for this call to DFSORT.

COPY The DFSORT copy function was used.

SORT The DFSORT sort function was used.

ddname1
The ddname of the data set that was copied or sorted.

ddname2
The ddname of the data set resulting from the copy or sort.

E35 EXIT
ICETOOL's E35 user exit was used.

OUTFIL
OUTFIL data sets were produced

COMPLETED
DFSORT did not detect any errors preventing completion of this ICETOOL operation.

TERMINATED
DFSORT detected an error which prevented completion of this ICETOOL operation.

System action: None.

Programmer response: You can use the information in this message to determine the result of this call to DFSORT. If DFSORT terminated, use the call identifier to find the set of DFSORT messages for this call (in the DFSMSG data set), and correct the error indicated in the DFSORT error message.

ICE628I RECORD COUNT: *nnnnnnnnnnnnnn*

Explanation: Indicates the number of records processed by ICETOOL (prints as 15 decimal digits padded with zeros on the left as needed).

If ICETOOL completed the operation successfully, this count reflects the number of records in the input data set or in the subset of the input data set selected by DFSORT statements (for example, INCLUDE).

If ICETOOL did not complete the operation successfully, this count reflects the number of records processed before an error was detected that caused ICETOOL to terminate processing of this operation.

System action: None.

Programmer response: None.

ICE629A HIGHER AND LOWER VALUES EXCLUDE ALL RECORDS

Explanation: Critical. The values specified for HIGHER and LOWER excluded all records from the range. For example, HIGHER(5) and LOWER(6) define a range of 5 < value < 6, which excludes every value.

System action: This operation is terminated.

Programmer response: A $ marks the point at which the error was detected. Specify values for HIGHER and LOWER that define a valid range.

ICE630I MODE IN EFFECT: *mode*

Explanation: Indicates the processing mode to be used for subsequent operators as follows:
- STOP mode - set at the beginning of the run and when a MODE STOP statement is processed. Stops subsequent operations if an error is detected (by entering SCAN mode).
- CONTINUE mode - set when a MODE CONTINUE statement is processed. Continues with subsequent operations if an error is detected.
- SCAN mode - set when an error is encountered in STOP mode and when a MODE SCAN statement is processed. ICETOOL statements are checked for errors, but DFSORT is not called to perform the operations.

System action: None.

Programmer response: If SCAN mode was set due to an error while in STOP mode, correct the error.

ICE631I NUMBER OF VALUES IN RANGE FOR {(*p,m,f*) | (VLEN)}: *nnnnnnnnnnnnnnn*

Explanation: Indicates the count of values within the specified range for this field (prints as 15 decimal digits padded with zeros on the left as needed). (*p,m,f*) or (VLEN) is the field you specified for this RANGE operator.

System action: None.

Programmer response: None.

ICE632I SOURCE FOR ICETOOL STATEMENTS: *source*

Explanation: Indicates whether ICETOOL statements were processed from TOOLIN or from the calling program's PARMLIST (parameter list). Statements are processed from TOOLIN unless the ICETOOL Parameter List Interface is used.

System action: None.

Programmer response: None.

ICE633A RETURN AREA IS *n* **BYTES, BUT** *m* **BYTES ARE REQUIRED**

Explanation: The length of the return area (in bytes) as indicated in the calling program's parameter list was too small to contain all of the information to be returned for this operation. A return area of m bytes is needed.

System action: This operation is terminated.

Programmer response: Provide a return area of m bytes (or more) for this operation and set the return area length appropriately.

ICE634A VALUE FOR (*p,m,f*) EXCEEDS 31 DIGITS

Explanation: Critical. A value for the indicated FS or CSF field had 32 digits, or a value for the indicated UFF or SFF field had 32 or more digits, exceeding the limit of 31 digits.

System action: This operation is terminated.

Programmer response: Limit all FS, CSF, UFF, and SFF data values for this operation to 31 digits. The DISPLAY operator with CH format can be used to visually identify the values with 32 digits.

ICE635A NUMBER OF HEADER KEYWORDS DOES NOT MATCH NUMBER OF ON KEYWORDS

Explanation: Critical. There was not a one-to-one correspondence between the HEADER keywords and the ON keywords. For example, two HEADER keywords were specified with three ON keywords.

System action: This operation is terminated.

Programmer response: Specify one HEADER('string') or HEADER(NONE) keyword for each ON keyword.

ICE636A INVALID VALUE IN DFSORT INSTALLATION MODULE ICE*yyy*

Explanation: Critical. A value in the indicated DFSORT installation module (ICEyyy) could not be interpreted by ICETOOL because the module was not at the same level as the ICETOOL modules. This indicates a serious problem with ICEyyy and can be indicative of problems with the other installation modules as well. Any or all of the values printed can be incorrect.

Note: ICETOOL cannot detect all possible invalid values in the ICEyyy modules; it only prints this message for the first invalid value it detects.

System action: '> > INVALID < <' is printed in the list data set for the invalid value, and all subsequent processing for this operator is terminated.

Programmer response: Have your system programmer check that the ICEMAC macro and the DFSORT modules were installed correctly and that the installation modules (ICEAM1, ICEAM2, ICEAM3, ICEAM4, ICETD1, ICDTD2, ICETD3, and ICETD4) were generated correctly and are at the same level as the DFSORT modules.

ICE637A *ddname* **RECORD LENGTH OF** *n* **BYTES EXCEEDS MAXIMUM WIDTH OF** *m* **BYTES**

Explanation: Critical. The calculated record length for the indicated list data set was greater than 2048 or the maximum width specified. *n* is the total bytes required in the list data set record for the carriage control character, the title line (resulting from specified title elements), column widths (resulting from specified ON, HEADER, PLUS, BLANK, TOTAL, BREAK, BTITLE, and BTOTAL operands), and blanks before and between title elements and columns (resulting from specified INDENT, TBETWEEN, BETWEEN, and STATLEFT operands). *m* is the value specified for the WIDTH operand, or 2048 if WIDTH was not specified.

System action: This operation is terminated.

Programmer response: If *m* is less than 2048, either remove the WIDTH operand and let ICETOOL set the width, or if you need to set the WIDTH explicitly, increase its value to *n* or greater.

If *m* is 2048, take one or more of the following actions:
- Use formatting items or the PLUS or BLANK operand. For example, use ON(21,18,ZD,U19) instead of ON(21,18,ZD) with TOTAL to change the column width from 32 bytes to 20 bytes.
- Reduce the length of one or more HEADER strings.
- Reduce the length of one or more ON fields. For example, if an ON(1,8,PD) field always has zeros in bytes 1 through 3, use instead (1,8,PD,U09), or ON(4,5,PD) with BLANK, to reduce the column width from 16 bytes to 10 bytes.
- Reduce the number of ON fields, especially if the BTOTAL or TOTAL operand is used.
- Reduce BETWEEN(n).
- Reduce INDENT(n).
- Remove STATLEFT.
- Reduce TBETWEEN(n).

ICE638I **NUMBER OF RECORDS RESULTING FROM CRITERIA:** *nnnnnnnnnnnnnn*

Explanation: Indicates the count of records produced as a result of the specified criteria, for example, ALLDUPS (prints as 15 decimal digits padded with zeros on the left as needed).
- For an OCCUR operator, the count indicates the total number of records in the list data set.
- For a SELECT or SPLICE operator, the count indicates the total number of records in the outdd data set.

System action: None.

Programmer response: None.

ICE639A **INSUFFICIENT MAIN STORAGE - ADD AT LEAST** *n***K BYTES TO REGION**

Explanation: Critical. ICETOOL could not get the additional *n* KB of storage needed for this operation.

System action: This operation is terminated.

Programmer response: Increase the REGION by at least *n* KB. In some cases, more than *n* KB are required to meet all the storage requirements for ICETOOL and DFSORT for this operation.

ICE640A **INVALID FORMATTING ITEM**

Explanation: Critical. An ON(p,m,f,formatting), ON(VLEN,formatting) or ON(NUM,formatting) operand for this DISPLAY operator, or an ON(p,m,f,formatting), ON(VLEN,formatting) or ON(VALCNT,formatting) operand for this OCCUR operator, contained an invalid formatting item as follows:
- The formatting item was not /x (/x can be /D, /C, /K, /DK, /CK, /M, /G, /KB, /MB, or /GB), L'string', F'string', T'string', E'pattern', NOST, LZ, Ndd, Udd, or a valid mask (mask can be A0-A5, B1-B6, C1-C6, D1-D6, E1-E4, F1-F5, or G1-G6).
- /x, F'string', E'pattern', LZ, NOST, Ndd, Udd, or a mask was specified for a character field.
- /x or NOST was specified for ON(NUM,formatting) or for OCCUR.
- /x, NOST or Ndd was specified for BREAK(p,m,f,formatting).
- More than one /x was specified, more than one mask was specified, Ndd and Udd were both specified, or L'string', F'string', T'string', E'pattern', Udd, or Ndd was specified more than once.
- L'', F'', T'' or E'' was specified.
- F'string' or a mask was specified with E'pattern'.
- dd for Ndd or Udd was not two digits from 01 to 31.
- dd for Ndd or Udd was greater than 15 for ON(NUM,formatting) or ON(VALCNT,formatting).

System action: This operation is terminated.

Programmer response: A $ marks the point at which the error was detected. Correct the error.

ICE641A *service* **SERVICE ERROR (***x***-***y***)**

Explanation: Critical. The indicated system service failed when ICETOOL called it. Message values are as follows:

x The return code from the service, in decimal.

y The reason code from the service, in decimal, or zero if unavailable.

System action: This operation is terminated.

Programmer response: Contact IBM for programming support.

ICE642A DFSORT RELEASE AND ICE*yyy* RELEASE DO NOT MATCH

Explanation: Critical. The release level of the indicated DFSORT installation module (ICE*yyy*) does not match the release level of DFSORT (as shown in the DEFAULTS heading line). This indicates that the ICEMAC macro used to compile ICE*yyy* was not at the same release level as the DFSORT modules being used for the run.

Note: This message is only printed for the first ICE*yyy* with a release level that does not match the DFSORT release level.

System action: '>> release <<' is printed in the list data set for the invalid release level, and all subsequent processing for this operator is terminated.

Programmer response: If one or more installation modules (ICEAM1, ICEAM2, ICEAM3, ICEAM4, ICETD1, ICETD2, ICETD3 and ICETD4) were compiled with the wrong level of the ICEMAC macro, recompile them with the correct level of the ICEMAC macro. If the wrong level of DFSORT modules were used for the run, rerun with the correct level of DFSORT modules.

ICE643I WIDTH OF REPORT IS *n* BYTES

Explanation: *n* indicates the line length and the LRECL for this DISPLAY or OCCUR list data set, determined as follows:
- *n* if WIDTH(*n*) was specified, or
- 121 if WIDTH(*n*) was not specified and the calculated line length was less than or equal to 121, or
- the calculated line length if WIDTH(*n*) was not specified and the calculated line length was greater than 121.

System action: None.

Programmer response: None.

ICE644A {VALUE|TOTAL} FOR {(*p,m,f*)|(VLEN)} OVERFLOWED *n* DECIMAL DIGITS

Explanation: Critical. A value for the indicated ON(p,m,f) or ON(VLEN) field (DISPLAY or OCCUR) or BREAK(p,m,f) field (DISPLAY), or the total for the indicated ON(p,m,f) field, exceeded the number of digits (*n*) you allowed for it. *n* is the number of digits allowed for the value or total as determined from the length and format of the field, or the Ndd or Udd formatting item you specified.

System action:
- For a DISPLAY operator with an overflowing BREAK value, this operation is terminated.
- For a DISPLAY operator with an overflowing ON value, asterisks are printed for this value in the data line and in any statistics lines requested. Processing continues.
- For a DISPLAY operator with an overflowing total, asterisks are printed for the total. Processing continues.
- For an OCCUR operator with an overflowing ON value, this operation is terminated.

Programmer response: Specify an Ndd or Udd formatting item large enough to prevent overflow of the value or total (that is, use an appropriate Ndd or Udd value between n+1 and 31).

ICE645A {(NUM)|(VALCNT)} OVERFLOWED *n* DECIMAL DIGITS

Explanation: Critical. The number of digits (*n*) allowed from the Ndd or Udd formatting item you specified was too small, as follows:
- (NUM) indicates a record number for this DISPLAY operator exceeded the number of digits you allowed for it.
- (VALCNT) indicates a value count for this OCCUR operator exceeded the number of digits you allowed for it.

System action: This operation is terminated.

Programmer response: Specify an Ndd or Udd formatting item large enough to prevent overflow of the record number or value count (that is, use an appropriate Ndd or Udd value between n+1 and 15).

ICE646A RECORD COUNT MEETS CRITERIA - RC=12 SET

Explanation: Critical. EMPTY, NOTEMPTY, HIGHER(n), LOWER(n), EQUAL(n), or NOTEQUAL(n) was specified for this COUNT operator. Because the record count met the specified criteria, RC=12 was set for this COUNT operation.

System action: The operation is terminated.

Programmer response: None.

ICE647I RECORD COUNT DOES NOT MEET CRITERIA - RC=0 SET

Explanation: EMPTY, NOTEMPTY, HIGHER(n), LOWER(n), EQUAL(n), or NOTEQUAL(n) was specified for this COUNT operator. Because the record count did not meet the specified criteria, RC=0 was set for this COUNT operation.

System action: None.

Programmer response: None.

ICE648I *statistic*:
snnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnn

Explanation: Indicates the minimum, maximum, average or total for the field indicated in the ICE607I message preceding this message. *statistic* is MINIMUM, MAXIMUM, AVERAGE or TOTAL. Each value consists

of a + or - sign and 31 decimal digits (padded with zeros on the left as needed).

Note: An ICE608I message is printed for the minimum and maximum values (instead of ICE648I messages) if both values contain less than 16 significant digits, unless LMSG is specified. An ICE609I message is printed for the average and total values (instead of ICE648I messages) if both values contain less than 16 significant digits, unless LMSG is specified.

System action: None.

Programmer response: None unless asterisks were printed for the AVERAGE and TOTAL values, in which case you should correct the error indicated by the previous error message for this operation.

If you want all of the minimum, maximum, average and total values displayed with 31 digits, specify the LMSG keyword for this STATS operator.

ICE649A HEX VALUE: *h...h*

Explanation: Critical. Displays the invalid value associated with preceding message ICE618A. *h...h* is the invalid value in hexadecimal. See the Explanation of message ICE618A for details

System action: See the System Action of message ICE618A for details.

Programmer response: See the Programmer Response of message ICE618A for details.

ICE650I VISIT http://www.ibm.com/storage/dfsort FOR ICETOOL PAPERS, EXAMPLES AND MORE

Explanation: The DFSORT website at http://www.ibm.com/storage/dfsort is an important source of information about DFSORT's ICETOOL utility. It contains papers, examples, online documents, news, tips, techniques and more that can help you get the most out of ICETOOL.

System action: None.

Programmer response: Optional. Visit the DFSORT website to see what it has to offer.

ICE651I RECORD COUNT MEETS CRITERIA - RC=4 SET

Explanation: RC4 and EMPTY, NOTEMPTY, HIGHER(n), LOWER(n), EQUAL(n), or NOTEQUAL(n) was specified for this COUNT operator. Because the record count met the specified criteria, RC=4 was set for this COUNT operation.

System action: Processing continues and ICETOOL passes back a return code of 4 unless it sets a higher return code for some other reason.

Programmer response: None.

ICE652A OUTREC STATEMENT FOUND BUT NOT ALLOWED - USE OUTFIL STATEMENT INSTEAD

Explanation: Critical. A DFSORT OUTREC statement was specified for this SELECT or SPLICE operator, but you cannot use an OUTREC statement with SELECT or SPLICE.

System action: The operation is terminated.

Programmer response: If you want to reformat the output records produced by this SELECT or SPLICE operator, use one or more OUTFIL statements as explained in *z/OS DFSORT Application Programming Guide*, instead of an OUTREC statement.

ICE700A EFS PROCESSING IS NOT SUPPORTED FOR THE TECHNIQUE SELECTED

Explanation: Critical. EFS processing is not supported for the technique selected (tape work data set sort or Conventional merge).

System action: The program terminates.

Programmer response: Rerun the job with SORTDIAG DD statement to get the reason for the revert from message ICE800I. Remove the cause of the revert or do not use EFS processing by specifying EFS=NONE.

ICE701A EFS PROCESSING WAS TERMINATED BY *pppppppp* **AT MAJOR CALL** *n*

Explanation: Critical. Termination of the DFSORT run was requested by EFS program *pppppppp* at Major Call 1, 2, 3, 4, or 5.

System action: If the termination was requested at Major Call 1, the program terminates. If the termination was requested at Major Call 2 through 5, DFSORT scans all remaining control statements and then the program terminates.

Programmer response: Refer to the message issued by EFS program *pppppppp* for the error description.

ICE702A INVALID RETURN CODE FROM EFS PROGRAM *pppppppp* **AT MAJOR CALL** *n*

Explanation: Critical. EFS program *pppppppp* returned an invalid return code to DFSORT at Major Call 1, 2, 3, 4, or 5.

System action: If the invalid return code was returned at Major Call 1, the program terminates. If the invalid return code was returned at Major Call 2 through 5, DFSORT scans all remaining control statements and then the program terminates.

Programmer response: Check the EFS program thoroughly and ensure that the return code is either 0 or 16.

ICE703A INVALID RETURN CODE FROM *ppppp* OF EFS PROGRAM *pppppppp*

Explanation: Critical. The *ppppp*-generated user exit routine, EFS01 or EFS02, of EFS program *pppppppp* returned an invalid return code to DFSORT.

System action: The program terminates.

Programmer response: Check the EFS-generated user exit routines thoroughly and ensure that the return code returned by EFS01 is either 0 or 16 and that the return code returned by EFS02 is 0, 4, or 16.

ICE704A LOAD FAILED FOR EFS PROGRAM *pppppppp*

Explanation: Critical. EFS program *pppppppp* could not be loaded.

System action: The program terminates.

Programmer response: Check that EFS program *pppppppp* is installed.

ICE705A EFS PROCESSING WAS TERMINATED BY *ppppp* OF EFS PROGRAM *pppppppp*

Explanation: Critical. Termination of the DFSORT run was requested by a user exit routine generated by EFS program *pppppppp*.

ppppp is one of the following names:

EFS01 The SORT/MERGE user exit routine.

EFS02 The INCLUDE/OMIT user exit routine.

System action: The program terminates.

Programmer response: Refer to the message issued by the user exit routine generated by EFS program *pppppppp*.

ICE706I END OF SUPPLIED CONTROL STATEMENTS - EFS RETURNED CONTROL STATEMENTS FOLLOW

Explanation: When control statements are specified in the invoking parameter list or control statements are specified with the DD statement for SYSIN or *xxxx*CNTL, this message separates the listing of those control statements from the listing of the control statements returned by an EFS program.

A control statement returned by an EFS program completely overrides the corresponding control statement in the parameter list, or in SYSIN or *xxxx*CNTL. (OPTION control statement operands EFS, LIST, NOLIST, LISTX, NOLISTX, MSGDDN, MSGPRT, SORTDD, SORTIN, SORTOUT, and USEWKDD are ignored if returned by an EFS program.)

For complete details on the order of override, see *z/OS DFSORT Application Programming Guide*.

System action: None.

Programmer response: None.

ICE707A EXIT E61 IS NOT SUPPORTED FOR EFS FIELDS

Explanation: Critical. The SORT or MERGE control statement defined a control field to be processed both by an EFS program and an E61 user exit routine.

System action: The program terminates.

Programmer response: Check the SORT or MERGE control statement for errors resulting in the specification of a D1 field format type with an E field sequence type.

ICE708A AN EFS PROGRAM IS REQUIRED

Explanation: Critical. A SORT/MERGE control field or an INCLUDE/OMIT compare field was to be processed by an EFS program, but the EFS program was not activated by the EFS option, nor was a verb request list returned at Major Call 1.

System action: The program terminates.

Programmer response: Check the SORT/MERGE or INCLUDE/OMIT control statements for errors resulting in the specification of a D1 or D2 field format type. Check that the EFS option has been specified.

ICE709A *ppppp* EXIT ROUTINE IS REQUIRED

Explanation: Critical. A SORT/MERGE control field or an INCLUDE/OMIT compare field was to be processed by an EFS program, but the EFS program did not return an address to EFS01 or EFS02.

ppppp is one of the following names:

EFS01 The SORT/MERGE user exit routine.

EFS02 The INCLUDE/OMIT user exit routine.

System action: The program terminates.

Programmer response: Check the SORT/MERGE and INCLUDE/OMIT control statements for errors resulting in the specification of a D1 or D2 field format type. Check the EFS program and ensure that it returns an address to EFS01 or EFS02.

ICE710A THE *aaaaaaaa* CONTROL STATEMENT WAS NOT SENT TO EFS PROGRAM *pppppppp*

Explanation: Critical. The control statement *aaaaaaaa* requested by EFS program *pppppppp* contained a syntax error. DFSORT could not determine the true length of

the requested control statement or option.

System action: The program terminates.

Programmer response: Respond as indicated in the accompanying message ICE007A.

ICE711I END OF EFS RETURNED CONTROL STATEMENTS FROM *xxxx*CNTL - PARAMETER LIST CONTROL STATEMENTS FOLLOW

Explanation: When returned control statements are specified in the invoking parameter list and with the DD statement for *xxxx*CNTL, this message separates the listing of requested control statements from both sources.

If all returned control statements came from either SORTCNTL or the parameter list, message ICE711I was not printed; only ICE706I was printed.

For complete details on the order of override, see *z/OS DFSORT Application Programming Guide*.

System action: None.

Programmer response: None.

ICE712I END OF EFS RETURNED PARAMETERS FROM *ddname* - SYSIN OR *xxxx*CNTL/PARAMETER LIST CONTROL STATEMENTS FOLLOW

Explanation: When returned parameters are specified with:
- The SYSIN DD statement
- The SORTCNTL DD statement, or
- The invoking parameter list

and are specified with the DD statement indicated by *ddname*, this message separates the listing of the requested data set's parameters from the former sources.

See *z/OS DFSORT Application Programming Guide* for details on the order of override.

System action: None.

Programmer response: None.

ICE713A LANGUAGE ENVIRONMENT INITIALIZATION FAILED

Explanation: Critical. Language Environment® initialization failed.

System action: The program terminates.

Programmer response: Contact IBM for programming support.

ICE714A LOAD FAILED FOR ACTIVE LOCALE *name*

Explanation: Critical. Indicates that the designated active locale could not be loaded and gives its name.

System action: The program terminates.

Programmer response: If the indicated locale *name* is that of an IBM supplied locale, DFSORT must have access to the run-time library containing the dynamically loadable routines. For example, the data set containing them might be called SYS1.SCEERUN. If you are unsure of the name of the data set where the compiled locale load modules are installed at your location, contact your system administrator.

If the indicated name is that of a user defined locale, DFSORT must have access to the load library containing it.

ICE715A LOCALE PROCESSING CONFLICTS WITH *keyword*

Explanation: Critical. Locale processing cannot be used with the function or feature indicated by the keyword shown. See *z/OS DFSORT Application Programming Guide* for details on locale processing and the function or feature indicated.

System action: The program terminates.

Programmer response: If D1, D2, or EFS is the indicated keyword, take one of the following actions:
- Specify run-time option EFS=NONE to use locale processing without EFS processing. (Locale processing might eliminate the need for an EFS program.)
- Specify run-time option LOCALE=NONE to use EFS processing without locale processing.

If CHALT is the indicated keyword, take one of the following actions:
- Specify run-time option NOCHALT to use locale processing without alternate sequence processing for CH fields. (Locale processing might eliminate the need for alternate sequence processing.) If alternate sequence processing is needed for a particular field, specify AQ for that field instead of CH.
- Specify run-time option LOCALE=NONE to use alternate sequence processing for CH fields without locale processing.

If INREC is the indicated keyword, take one of the following actions:
- Use the OUTREC statement or the OUTREC operand of the OUTFIL statement instead of the INREC statement to allow locale processing.
- Specify run-time option LOCALE=NONE to use the INREC statement without locale processing.

If E61 is the indicated keyword, take one of the following actions:

- Eliminate the E61 exit, and change E to A or D for each field, to use locale processing without E61 processing. (Locale processing might eliminate the need for an EFS program.)
- Specify run-time option LOCALE=NONE to use E61 processing without locale processing.

ICE716A A CONTROL FIELD CONTAINS AN INVALID CHARACTER

Explanation: Critical. Locale processing of a control field detected that a character in the control field is not defined in the active locale.

System action: The program terminates.

Programmer response: Ensure that the correct locale is the active locale in effect or correct the invalid character.

ICE717A CONTROL FIELDS FOR LOCALE PROCESSING ARE TOO COMPLEX

Explanation: Critical. For a SORT or MERGE statement with control fields that require locale processing, the complexity of the application caused dynamic areas to exceed the storage allowed for them.

System action: The program terminates.

Programmer response: Reduce the number or length of SORT or MERGE control fields that require locale processing.

ICE721A A COMPARE FIELD CONTAINS AN INVALID CHARACTER

Explanation: Critical. Locale processing of a compare field detected that a character in the compare field is not defined in the active locale.

System action: The program terminates.

Programmer response: Ensure that the correct locale is the active locale in effect or correct the invalid character.

ICE722I ACTIVE LOCALE *name* WAS USED FOR LOCALE PROCESSING

Explanation: Indicates that DFSORT used locale processing and gives the name of the active locale. Locale processing is used for:
- SORT or MERGE character (CH) control fields
- INCLUDE or OMIT character (CH) compare fields
- INCLUDE or OMIT character or hexadecimal constant to character (CH) compare field comparisons.

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

System action: None

Programmer response: None

ICE725A LANGUAGE ENVIRONMENT SERVICE ERROR (*nnnn*)

Explanation: Critical. Language Environment services returned an unexpected feedback code for a service call used by DFSORT. *nnnn* is the error message number of the feedback code.

System action: The program terminates.

Programmer response: Take one or more of the following actions, as appropriate:
- Check that all SORT, MERGE, INCLUDE, or OMIT CH fields contain valid character data values.
- Contact IBM for programming support. The error message number can be used to reference Language Environment Run-Time Messages in *Locale Processing*.

ICE726A LOAD FAILED FOR LOCALE PROCESSING

Explanation: Critical. DFSORT's locale processing routine could not be loaded.

System action: The program terminates.

Programmer response: Contact your system programmer to ensure that DFSORT's locale processing support has been installed.

ICE727A EFS PROCESSING CANNOT BE USED WITH SYMBOL PROCESSING

Explanation: Critical. A SYMNAMES data set was specified indicating that symbol processing is to be performed. However, an EFS program was also specified. EFS processing and symbol processing cannot both be used.

System action: The program terminates.

Programmer response: Use EFS processing or symbol processing, but not both.

ICE750I DC a TC b C c DSVd KSZ e VSZ f

Explanation: DFSORT diagnostic message for use by the IBM Support Center.

ICE751I aa-bbbbbb ...

Explanation: DFSORT diagnostic message for use by the IBM Support Center.

ICE752I FSZ= a b IGN = c d AVG = e f WSP = g h DYN = i

Explanation: DFSORT diagnostic message for use by the IBM Support Center.

ICE753I FWK=(a,b) SWK=(c,d) TWK=(e,f) RWK=(g,h) TOTAL=(i,j) BLK=k

Explanation: DFSORT diagnostic message for use by the IBM Support Center.

Chapter 4. Diagnostic Messages

The following diagnostic messages can appear when a SORTDIAG DD statement is present, or when installation option DIAGSIM=YES is in effect.

ICE800I BLOCKSET TECHNIQUE COULD NOT BE USED - REASON CODE IS *nn*

Explanation: DFSORT's most efficient technique, Blockset, could not be used for this application. *nn* is the revert reason code. See the following list to determine the reason associated with this code.

1 A critical message was issued.

5 BDAM was indicated in the SORTIN DD statement.

6 BDAM was indicated in the SORTIN data set label.

7 BDAM was indicated in the SORTOUT DD statement.

10 A DUMMY or spool SORTWK*dd* data set was specified.

12 An attempt to OPEN a data set caused a system error.

16 The SORTIN or SORTOUT data set resided on an unsupported device.

17 A SORTWK*dd* data set resided on an unsupported device (such as tape).

22 ASCII tapes had the following parameters:

(LABEL=AL or OPTCD=Q) and RECFM=D and BUFOFF[=L

or

(LABEL=AL or OPTCD=Q) and RECFM[=D and BUFOFF[=0.

23 An attempt to read the DSCB for the SORTIN data set caused an error.

24 An attempt to read the DSCB for the SORTOUT data set caused an error.

28 An E18 or E38 user exit was specified with a tape SORTIN(*nn*) data set.

47 The NOBLKSET option was specified.

48 The CKPT run-time option and the IGNCKPT=NO installation option were specified.

58 Dynamic tape SORTWK*dd* data sets were requested.

63 Too many control fields were specified or control fields were too large.

64 Too many control fields were specified or control fields were too large.

65 A SORTWK*dd* data set was not present.

66 An INREC statement was specified and EFS01 was in effect.

74 An E61 user exit was specified and VLSHRT was in effect.

75 A MODS statement E*xx* operand had SYSIN in the third parameter or T or S in the fourth parameter (that is, dynamic link-editing was requested.).

Note: Blockset cannot be used to process BDAM data sets.

System action: None.

Programmer response: None. However, for improved performance, correct the situation indicated by the reason code so Blockset can be used.

ICE802I *t* TECHNIQUE IN CONTROL

Explanation: *t* indicates the DFSORT technique that was in control:

BLOCKSET Disk work data set sort, standard merge, or copy

PEER/VALE Disk work data set sort

CONVENTIONAL Nonstandard merge or tape work data set sort.

The indicated technique was not necessarily the technique ultimately selected (see message ICE143I).

System action: None.

Programmer response: None.

ICE804I *ddname* EXCP COUNT: *n*

Explanation: If the ddname is OUTFIL, lists the total number of EXCPs performed, using the EXCP access method, to all OUTFIL data sets. Otherwise, lists the number of EXCPs performed, using the EXCP access method, to the data set associated with the indicated *ddname*.

System action: None.

Programmer response: None.

© Copyright IBM Corp. 1995, 2006

ICE805I **JOBNAME:** *jobname* **, STEPNAME:** *stepname*

Explanation: Indicates the job name and step name for the DFSORT run.

System action: None.

Programmer response: None.

ICE816I **DFSORT SVC CALL RESULTED IN S***nnn* **ABEND**

Explanation: A call to the DFSORT SVC resulted in a system abend. *nnn* is the abend code. It occurred when the SVC was not available to DFSORT because of improper installation or because the wrong version of the SVC was installed. Because DFSORT continued the run as if the abend had not occurred, this message was issued to assist system programmers in determining the reason why the SVC was not available. DFSORT runs more efficiently with cache devices if the SVC is available. See *z/OS DFSORT Installation and Customization* for further information.

System action: None.

Programmer response: Ask your system programmer to check that the DFSORT SVC is installed properly for this level of DFSORT and that the correct routing code for the SVC is selected.

ICE818I **LOCATE CALL FAILED**

Explanation: A call to the LOCATE SVC failed while DFSORT was attempting to obtain information about an input or output data set.

System action: Processing continues without the information from LOCATE. This can result in one of the following situations:

- DFSORT cannot accurately estimate the input file size (message ICE118I indicates this situation)..
- DFSORT cannot obtain the input record length.
- An OPEN error may occur if a VSAM output data set is defined without REUSE, and RESET is in effect.

Programmer response: None required. If DFSORT cannot accurately estimate the input file size, refer to the Programmer Response for message ICE118I. If DFSORT cannot obtain the input record length, specify an L1 value in the LENGTH parameter of the RECORD statement. If a VSAM OPEN error occurs, specifying NORESET may eliminate the OPEN error.

ICE8201 **RL=a B=b IL=c IS=d IB=e RM=f EM=g BA=h IX=i OX=j**

Explanation: DFSORT diagnostic message for use by the IBM Support Center.

ICE821I **BN=a X=b TO=c SN=d G=e**

Explanation: DFSORT diagnostic message for use by the IBM Support Center.

ICE822I **BN=a X=b G=c PN=d BT=e TO=f**

Explanation: DFSORT diagnostic message for use by the IBM Support Center.

ICE823I **BN=a X=b G=c TO=d BT=e**

Explanation: DFSORT diagnostic message for use by the IBM Support Center.

ICE824I **PE=a RP=b CX=c CO=d C0=e CR=f G=g WB=h**

Explanation: DFSORT diagnostic message for use by the IBM Support Center.

ICE825I **GP=a SA=b X=c**

Explanation: DFSORT diagnostic message for use by the IBM Support Center.

ICE826I **BN=a X=b RM=c**

Explanation: DFSORT diagnostic message for use by the IBM Support Center.

ICE827I **RD=a BL=b BY=c**

Explanation: DFSORT diagnostic message for use by the IBM Support Center.

ICE855I **a : TX=b, R=c, L=d, B=e, BL=f, BR=g, DCT=h, VS=i, RU=j, SB=k**

Explanation: DFSORT diagnostic message for use by the IBM Support Center.

ICE857I **C=a, LA=b, DA=c, LB=d, DB=e, MB=f, CB=g, UB=h**

Explanation: DFSORT diagnostic message for use by the IBM Support Center.

ICE858I **LA=a, DA=b, AA=c, BA=d, TA=e**

Explanation: DFSORT diagnostic message for use by the IBM Support Center.

ICE859I **LB=a, DB=b, AB=c, BB=d, CP=e, RS=f, TB=g**

Explanation: DFSORT diagnostic message for use by the IBM Support Center.

ICE860I F=a, P=b, M=c, B=d

Explanation: DFSORT diagnostic message for use by the IBM Support Center.

ICE880I QP=a QA=b HI=c LI=d MI=e TZ=f N1=g N2=h SZ=i HN=j

Explanation: DFSORT diagnostic message for use by the IBM Support Center.

ICE881I EQ=a DX=b D2=c D3=d D4=e AS=f SA=g SB=h SC=i HN=j

Explanation: DFSORT diagnostic message for use by the IBM Support Center.

ICE882I a PLE b

Explanation: DFSORT diagnostic message for use by the IBM Support Center.

ICE883I IX=a, IFC=b, INC=c, OX=d, OFC=e, ONC=f, CRT=g, NCR=h

Explanation: DFSORT diagnostic message for use by the IBM Support Center.

ICE884I a b AVK c AVR d

Explanation: DFSORT diagnostic message for use by the IBM Support Center.

ICE885I DAT a DSR b BINS c BSZ d RCP e AJC f RLC g DUNIT h

Explanation: DFSORT diagnostic message for use by the IBM Support Center.

ICE886I SYS a TSTG b FS c INIT d MAX e LEN f

Explanation: DFSORT diagnostic message for use by the IBM Support Center.

ICE887I CSES a,b,c ES d,e,f

Explanation: DFSORT diagnostic message for use by the IBM Support Center.

ICE888I INI01 a AXI01 b W1 c INI03 d AXI03 e BRUNS f VDg

Explanation: DFSORT diagnostic message for use by the IBM Support Center.

ICE889I CT=a, SB=b, L=c, D=defg

Explanation: DFSORT diagnostic message for use by the IBM Support Center.

ICE891I a WMAIN, b CMAIN, c CALLOC, d SCN, e BA, f AZ, g BZ, h QC

Explanation: DFSORT diagnostic message for use by the IBM Support Center.

ICE892I a RIN b BLI c BLO d RUN e BUN f CPU g CVC

Explanation: DFSORT diagnostic message for use by the IBM Support Center.

ICE893I a XIN b WIN c GIN NDEd PFPe Bf CMg CIh UPTi LMj VSk RU1

Explanation: DFSORT diagnostic message for use by the IBM Support Center.

ICE894I a STR b MOR c IPB d OPB e CYL f MN

Explanation: DFSORT diagnostic message for use by the IBM Support Center.

ICE895I a MUNIT b SUNIT c OUNIT

Explanation: DFSORT diagnostic message for use by the IBM Support Center.

ICE896I a SET b DEXTOT c BLK d CSZ e WE

Explanation: DFSORT diagnostic message for use by the IBM Support Center.

ICE897I AVAL=a, BVAL=b, CVAL=c, ASV=d, HSZ=e, HM=f, HN=g

Explanation: DFSORT diagnostic message for use by the IBM Support Center.

ICE898I OMAX=a, NMAX=b, TMAX=c, CMAX=d, HU=e, BUN=f, MD=g, DU=h, DR=i, HN=j

Explanation: DFSORT diagnostic message for use by the IBM Support Center.

ICE899I HSR=a, HSW=b, HRE=c, HWE=d, HRP=e, HWP=f, HWM=g, HNM=h

Explanation: DFSORT diagnostic message for use by the IBM Support Center.

ICE900I **CON=a, MUV=b, VOL=c, ENU=d, SBK=e, SRC=f, VEM=g**

Explanation: DFSORT diagnostic message for use by the IBM Support Center.

ICE901I **a b . . .**

Explanation: DFSORT diagnostic message for use by the IBM Support Center.

ICE902I **0 a I b c . . .**

Explanation: DFSORT diagnostic message for use by the IBM Support Center.

ICE905I **a : RF=b, LR=c, BLK=d, BCT=e**

Explanation: DFSORT diagnostic message for use by the IBM Support Center.

ICE915I

MOFSZ=a,MOSZ=b,MOSYS=c(d), MOSTG=e,MEML=f(g)

Explanation: DFSORT diagnostic message for use by the IBM Support Center.

ICE916I **MOFR=a,MOVR=b**

Explanation: DFSORT diagnostic message for use by the IBM Support Center.

ICE917I

MOQP=a,MOQA=b,MOHI=c,MOLI=d, MOMI=e,MOTZ=f

Explanation: DFSORT diagnostic message for use by the IBM Support Center.

ICE995I **PN=a, HN=b**

Explanation: DFSORT diagnostic message for use by the IBM Support Center.

ICE996I **ESM=a, ESO=b, ESR=c, ESP=d, ESS=e, CES=f, HSZ=g**

Explanation: DFSORT diagnostic message for use by the IBM Support Center.

ICE997I **HWSP=a, HMAX=b, HES=c, ASV=d, EQ=e, HN=f**

Explanation: DFSORT diagnostic message for use by the IBM Support Center.

ICE999I **PWK=a PSP=b SWK=c SSP=d TWK=e TSP=f RWK=g RSP=h AWK=i AWP=j**

Explanation: DFSORT diagnostic message for use by the IBM Support Center.

Chapter 5. Return Codes and User Abends

DFSORT Return Codes

For successful completion, DFSORT passes back a return code of 0 or 4 to the operating system or the invoking program.

For unsuccessful completion due to an unsupported operating system, DFSORT passes back a return code of 24 to the operating system or the invoking program.

For unsuccessful completion with NOABEND in effect, DFSORT passes back a return code of 16 or 20 to the operating system or the invoking program.

For unsuccessful completion with ABEND in effect, DFSORT issues a user abend with the appropriate code as specified by ICEMAC option ABCODE (either the error message number or a number between 1 and 99).

The meanings of the return codes that DFSORT passes back (in register 15) are:

0 **Successful completion**. DFSORT completed successfully.

4 **Successful completion**. DFSORT completed successfully, and:

- OVFLO=RC4 was in effect and summary fields overflowed, or
- PAD=RC4 was in effect and the SORTOUT LRECL was larger than the SORTIN/SORTINnn LRECL (LRECL padding), or
- TRUNC=RC4 was in effect and the SORTOUT LRECL was smaller than the SORTIN/SORTINnn LRECL (LRECL truncation), or
- SPANINC=RC4 was in effect and one or more incomplete spanned records was detected, or
- NULLOUT=RC4 was in effect and there were no records for the SORTOUT data set, or
- NULLOFL=RC4 was in effect and there were no data records for an OUTFIL data set

16 **Unsuccessful completion**. DFSORT detected an error that prevented it from completing successfully.

20 **Message data set missing**. ICEMAC option NOMSGDD=QUIT was in effect and neither a message data set DD statement nor a SYSOUT DD statement was provided.

24 **Unsupported operating system**. Unsupported operating system. This operating system is not supported by this release DFSORT.

ICETOOL Return Codes

ICETOOL sets a return code for each operation it performs in STOP or CONTINUE mode and passes back the highest return code it encounters to the operating system or the invoking program.

For successful completion of all operations, ICETOOL passes back a return code of 0 or 4 to the operating system or the invoking program.

© Copyright IBM Corp. 1995, 2006

For unsuccessful completion due to an unsupported operating system, ICETOOL passes back a return code of 24 to the operating system or invoking program.

For unsuccessful completion of one or more operations, ICETOOL passes back a return code of 12, 16, or 20 to the operating system or the invoking program.

The meanings of the return codes that ICETOOL passes back (in register 15) are:

0 **Successful completion**. All operations completed successfully.

4 **Successful completion**. All operations completed successfully. Either:
- RC4 was specified for an ICETOOL COUNT operator and the record count met the specified criteria, or
- DFSORT passed back a return code of 4 for one or more operations. See "DFSORT Return Codes" on page 87 for details.

12 **Unsuccessful completion**. ICETOOL detected one or more errors that prevented it from completing successfully. Messages for these errors were printed in the TOOLMSG data set.

16 **Unsuccessful completion**. DFSORT detected one or more errors that prevented ICETOOL from completing successfully. Messages for these errors were printed in the DFSMSG data set.

20 **Message data set error**. The TOOLMSG DD statement was not present or the TOOLMSG data set was not opened.

24 **Unsupported operating system**. Unsupported operating system. This operating system is not supported by this release of DFSORT.

ICEGENER Return Codes

ICEGENER can use either IEBGENER or the DFSORT copy function. However, for unsuccessful completion due to an unsupported operating system, ICEGENER passes back a return code of 24 to the operating system or the invoking program, without using either IEBGENER or DFSORT copy.

If ICEGENER transfers control to IEBGENER, IEBGENER passes back its return code to the operating system or the invoking program.

If ICEGENER uses the DFSORT copy function:
- For successful completion, ICEGENER passes back a return code of 0 or 4 to the operating system or the invoking program.
- For unsuccessful completion with NOABEND in effect, ICEGENER passes back a return code of 12 (changed from 16) to the operating system or the invoking program.
- For unsuccessful completion with ABEND in effect, DFSORT issues a user abend with the appropriate code as specified by the ICEMAC option ABCODE (either the error message number or a number between 1 and 99).

The meanings of the return codes that ICEGENER passes back (in register 15) are:

0 **Successful completion**. ICEGENER completed successfully.

4 **Successful completion**. ICEGENER completed successfully, and:
- ICEMAC option GNPAD=RC4 was specified and the SYSUT2 LRECL was larger than the SYSUT1 LRECL (LRECL padding), or

- ICEMAC option GNTRUNC=RC4 was specifed and the SYSUT2 LRECL was smaller than the SYSUT1 LRECL (LRECL truncation), or
- SPANINC=RC4 was in effect and one or more incomplete spanned records was detected, or
- NULLOUT=RC4 was in effect and there were no records for the SORTOUT data set, or
- NULLOFL=RC4 was in effect and there were no data records for an OUTFIL data set

12 **Unsuccessful completion**. DFSORT detected an error that prevented ICEGENER from completing successfully.

24 **Unsupported operating system**. Unsupported operating system. This operating system is not supported by this release of DFSORT.

Part 3. Diagnosis Guide

Chapter 6. Resolving Failures in the DFSORT Program 93
Developing Your Keyword String 94
Constructing Keyword Strings 96
Component Identification Keyword Procedure. . . 97
FMID Keyword Procedure 98
Type-of-Technique Keyword Procedure 98
Type-of-Application Keyword Procedure 99
Type-of-Failure Keyword Procedures 99
Wait and Loop Keywords Procedure 100
System Abend Keyword Procedure 101
User Abend Keyword Procedure 101
Message Keyword Procedure 102
Incorrect-Output Keyword Procedure 102
Performance Keyword Procedure. 103
Document Keyword Procedure 103
Module Keyword Procedure 103
Offset Keyword Procedure 105
Message Variable Keywords 106
ICE061A Variable Keywords 106
ICE117A Variable Keywords 107

Chapter 7. Resolving Problems When Using DFSORT Panels 109
Obtaining Diagnostic Information 109
Determining Where the Failure Occurred 110
Resolving Failures in DFSORT Panels 110
DFSORT Panels Abend Panel 110
TSO Abend Message 111
Resolving Failures in ISPF or ISMF 112
Resolving Failures in the DFSORT Program . . . 113
Foreground Execution Failures 113
Background Execution Failures 114
Recording Information 114
Inter-Module Traces 115
ISMF Logging with SORTREC. 115
Using the Profile Application 115
Directing Abend Processing 116
Recovering from Abends 116
Passing Control to ISPF or TSO 116
Obtaining Diagnostic Messages 116

Chapter 8. Searching the Problem Reporting Databases 119

Chapter 9. Fixing or Bypassing the Problem 121
Fixing the Problem 121
Bypassing the Problem 121

Chapter 10. Reporting a Problem 123

© Copyright IBM Corp. 1995, 2006

Chapter 6. Resolving Failures in the DFSORT Program

To resolve a program failure, you describe the problem with a set of keywords, called a *keyword string*. The IBM Support Center (or you, if you have a tool such as Info Access) uses the keyword string to search an IBM software support database, such as the Software Support Facility or the Early Warning System, to determine whether an authorized program analysis report (APAR) has already been recorded.

An APAR contains information explaining how to fix or bypass the error. This might be a program temporary fix (PTF), an APAR fix, or a bypass.

If an APAR has not been recorded, the IBM Support Center personnel might ask you to obtain messages or a dump to provide information beyond the keyword string you developed.

Use Table 6 to resolve a program failure.

Table 6. Program Failure Resolution

Step	Action
1	Read and follow the directions in Chapter 1, "Eliminating Common Sources of Error," on page 3, to first eliminate your own user exit routines, calling programs (if you have any), and other factors as a source of error. If the source of error is not your own routines or calling programs, go to step 2.
2	Follow the instructions in "Developing Your Keyword String" on page 94. Each of the keywords in the string describes one aspect of a program failure and makes the search argument more specific. The more precise your keyword string, the more selective the search, and the fewer problem descriptions there are to evaluate. You need messages or a dump to develop most keyword strings. If you did not receive messages or a dump, read "How to Get Messages" on page 3, and Table 2 on page 3. When you complete your keyword string, go to step 3.
3	Follow the instructions in Chapter 8, "Searching the Problem Reporting Databases," on page 119. The search determines whether the problem has been reported previously. If the problem has already been reported, go to step 4. If the problem has not been reported, go to step 5.
4	If the problem has already been reported, you will be instructed to apply a PTF or APAR fix or to bypass the problem until a PTF or APAR fix is available. See Chapter 9, "Fixing or Bypassing the Problem," on page 121.
5	If the problem has not been reported, follow the instructions in Chapter 10, "Reporting a Problem," on page 123, to report the problem.

© Copyright IBM Corp. 1995, 2006

Developing Your Keyword String

This section helps you to develop a keyword string to describe your program failure. The keyword string is used to search an IBM software support database, explained in Chapter 8, "Searching the Problem Reporting Databases," on page 119. The number of keywords within a keyword string depends on the type of failure being described. The types of failures include:

System abend
User abend
Wait
Loop
Message
Incorrect output
Performance
Document

Before contacting the IBM Support Center for assistance, ensure that your keyword string contains all the required keywords for your particular type of failure.

Figure 1 on page 95 shows the steps you need to follow, depending on the type of failure you describe. The following are the keywords you might need to use:

Keyword	When required
Component Identification	Always required
FMID	Always required
Type-of-Technique	Required for all except document errors
Type-of-Application	Required for all except document errors
Type-of-Failure	Always required
Module	Required for system abend, user abend, wait, and loop
Offset	Required for system abend, user abend, wait, and loop
Message Variable	Required for messages ICE061A and ICE117A
Order Number	Required for document error

Begin with "Constructing Keyword Strings" on page 96 and follow the procedures until the keyword string is developed and you are instructed to use it as a search argument.

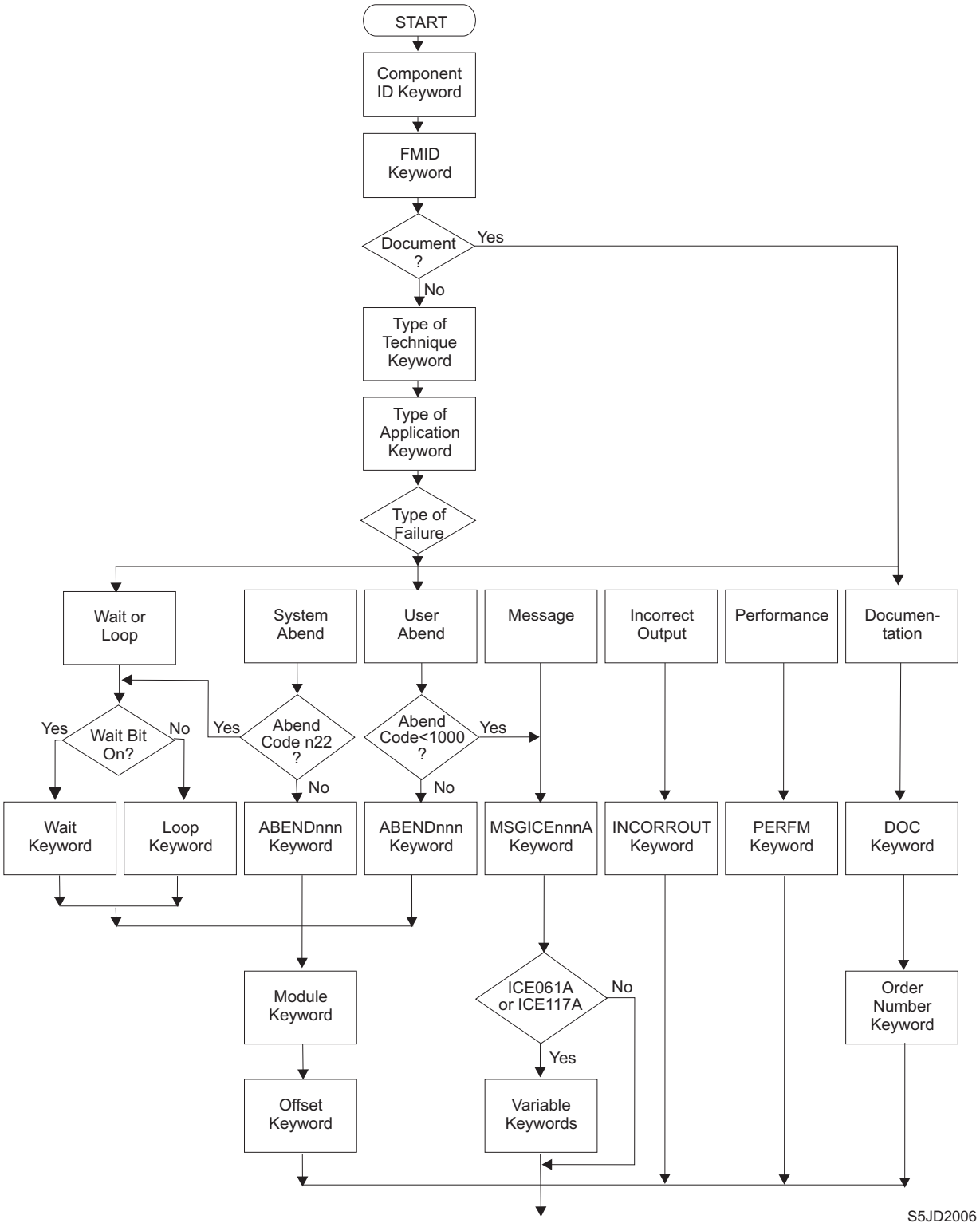


Figure 1. Keyword String Flowchart (Part 1 of 2)

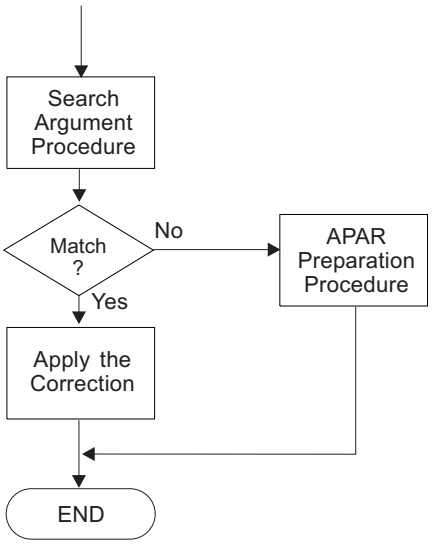


Figure 1. Keyword String Flowchart (Part 2 of 2)

Constructing Keyword Strings

Use Figure 2 on page 97 to help you construct the keyword string to describe DFSORT program failures.

To use the diagram:

1. Choose the type of keyword.
2. Start from the left.
3. Continue to work from left to right.
4. When you encounter parallel elements in one line, make a choice.
5. Type uppercase letters and special characters exactly as given.
6. Replace lowercase letters with the appropriate data.
7. Your statement is complete when you reach zero, on the right side.

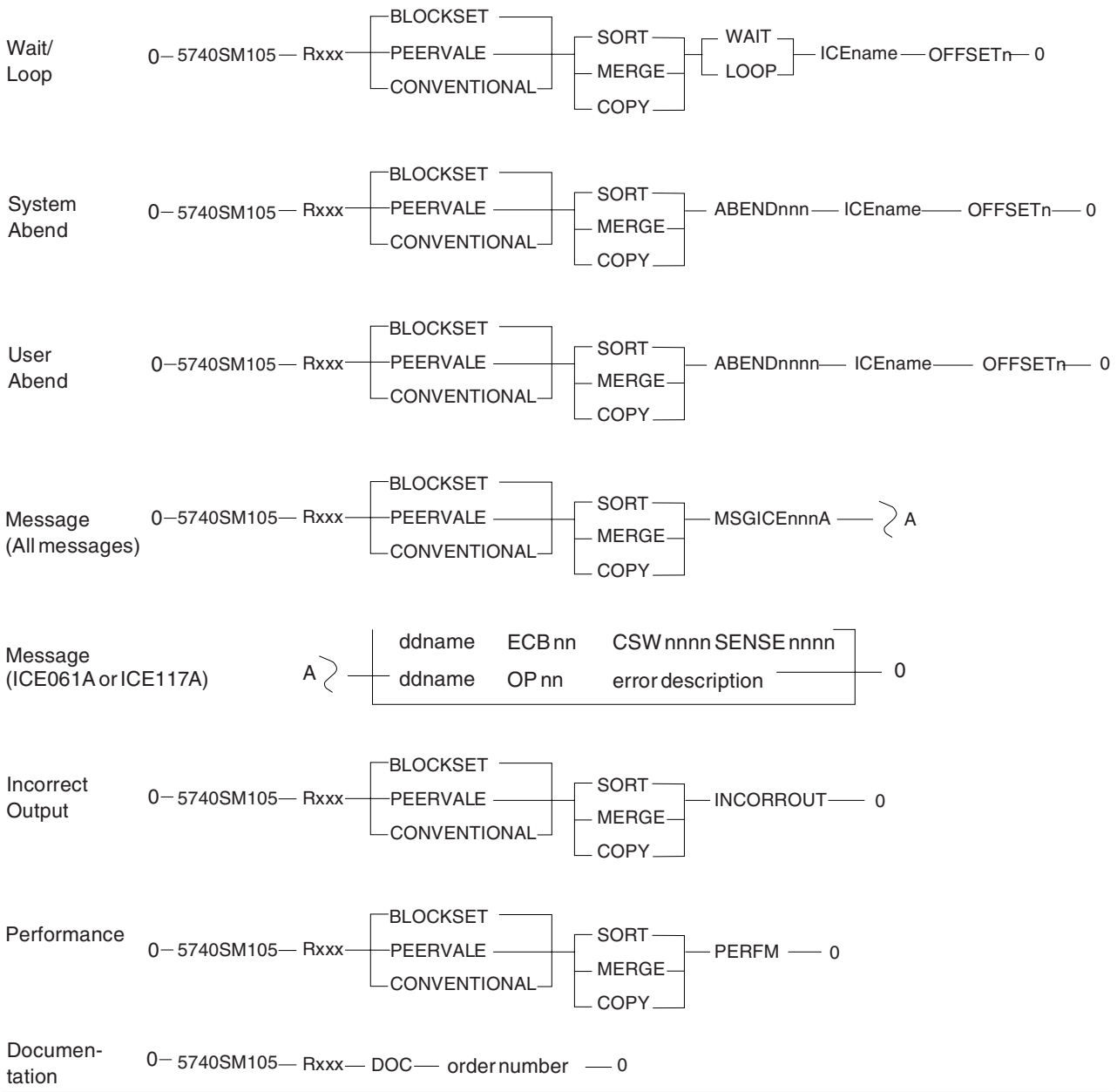


Figure 2. Keyword String Syntax. If you receive message ICE061A or ICE117A, fill in the second part of the message string. Otherwise, stop the keyword string at the message number.

Component Identification Keyword Procedure

The DFSORT component identifier is the first keyword in the search argument. This number identifies the product within the IBM software support database.

1. The component identification keyword is nine characters. For DFSORT, this keyword is **5740SM105**.
2. The keyword string so far is:

```
5740SM105
```

3. Go to "FMID Keyword Procedure" on page 98, to determine the current release level keyword for DFSORT.

FMID Keyword Procedure

The function modification identifier (FMID) keyword used with the component identification keyword narrows the search. The FMID shows the release of DFSORT.

1. Get the FMID from the control data set (CDS) list. To locate the FMID, find the module called ICEMAN in the list of module names. The FMID is on the same line as the ICEMAN module.

   The FMID format is:

   `HSM1ccc`

   For example, the FMID for z/OS DFSORT V1R5 is HSM1H00.
2. Record the FMID keyword in the following format:

   R*ccc*

   where *ccc* is the last three characters of the FMID.

   For V1R5 problems, you now have the following keyword string:

```
5740SM105 RH00
```

   H00 identifies the DFSORT base FMID HSM1H00.

   To identify DFSORT feature-related FMID information, see *z/OS DFSORT Installation and Customization*.
3. If the problem seems to result from incorrect or missing information in a DFSORT publication, go to "Document Keyword Procedure" on page 103.

   If the problem is not in a DFSORT publication, go to "Type-of-Technique Keyword Procedure."

Type-of-Technique Keyword Procedure

The next keyword in the keyword string specifies the sort, merge, or copy technique being used when the failure occurred.

You need a message for this procedure. If you did not receive any messages, see "How to Get Messages" on page 3.

1. Find out which technique was used by looking at the diagnostic trace character in the *last* message you received from DFSORT.

   There is a diagnostic trace character in all DFSORT messages. The format of a DFSORT message is:

   ICEnnns c text

Where	Represents
nnn	The error message number
s	The severity code (which can be either A for error messages or I for information messages)
c	The diagnostic trace character
text	An explanation of the message

   Each technique is associated with a set of diagnostic trace characters:

Trace character	Keyword
0 through 9 or A through I	BLOCKSET
J through Z	PEERVALE

Trace character	Keyword
— *(hyphen)*	CONVENTIONAL

For example, if you received message ICE085A 1, the 1 means the Blockset technique. The keyword string so far would be:

```
5740SM105 RH00 BLOCKSET
```

2. Go to "Type-of-Application Keyword Procedure."

Type-of-Application Keyword Procedure

The next keyword shows whether you were doing a sort, merge, or copy when the error occurred.

1. Record what you were doing when the error occurred.

   If you were using DFSORT to:
   - **Sort** data, add **SORT** to the keyword string.
   - **Merge** data, add **MERGE** to the keyword string.
   - **Copy** data, add **COPY** to the keyword string.

   For example, if you were sorting data, the keyword string so far would be:

```
5740SM105 RH00 BLOCKSET SORT
```

2. Go to "Type-of-Failure Keyword Procedures."

Type-of-Failure Keyword Procedures

It is very important that you determine the type-of-failure keyword correctly. Use Table 7 to find the procedure that best describes your problem.

Table 7. Types of DFSORT Failures

Type of Failure	Description	Messages and Codes	Procedure Name
Wait	Unexpected program suspension.	System completion code is n22 (where n is any hexadecimal digit).	"Wait and Loop Keywords Procedure" on page 100.
Loop	Uncontrolled program repetition.	System completion code is n22 (where n is any hexadecimal digit).	"Wait and Loop Keywords Procedure" on page 100.
System Abend	Abnormal termination of DFSORT.	System completion code is **not** n22 (where n is any hexadecimal digit), or zero.	"System Abend Keyword Procedure" on page 101.
User Abend	Abnormal termination of DFSORT.	User completion code is not zero.	"User Abend Keyword Procedure" on page 101.
Message	Errors identified by or associated with DFSORT error messages.	Error message is in the form ICEnnnA.	"Message Keyword Procedure" on page 102.
Incorrect Output	Unexpected, missing, or incorrect output not associated with a message. (Use this keyword only when no other keywords seem appropriate.)	Not applicable	"Incorrect-Output Keyword Procedure" on page 102.

Table 7. Types of DFSORT Failures (continued)

Type of Failure	Description	Messages and Codes	Procedure Name
Performance	Performance degradation. (Use this keyword only when you are sure that this is not a wait or loop problem.)	Not applicable	"Performance Keyword Procedure" on page 103.
Documentation	Severe incorrect or missing information in DFSORT documentation.	Not applicable	"Document Keyword Procedure" on page 103.

Wait and Loop Keywords Procedure

Because the wait and loop symptoms are often indistinguishable, the WAIT and LOOP keywords are included in the same type-of-failure procedure.

If your job enters a wait or loop, the system terminates the job and gives you a system abend code n22, where n is any hexadecimal digit. Common examples are 122, 222, and 322.

You need a dump for this procedure. If you did not receive a dump, see Table 2 on page 3.

1. Find the program status word (PSW) in the dump. See Figure 3.

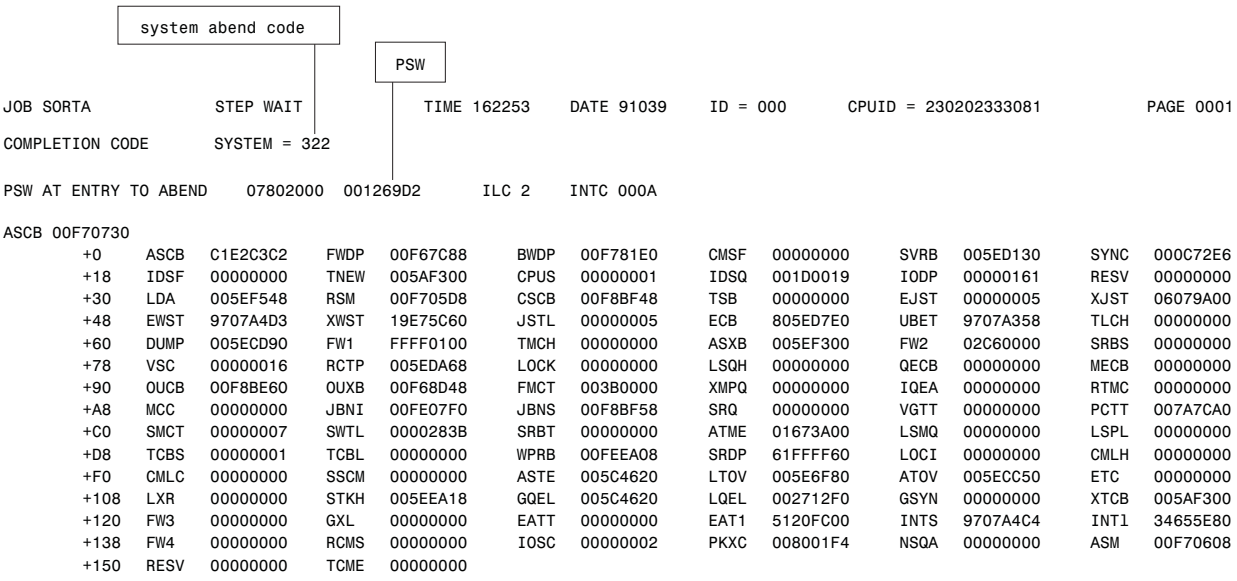


Figure 3. Finding the System Abend Code and the PSW in the SYSUDUMP

2. Check whether the wait state bit (bit 14) of the PSW is on or off (see Figure 4 on page 101).

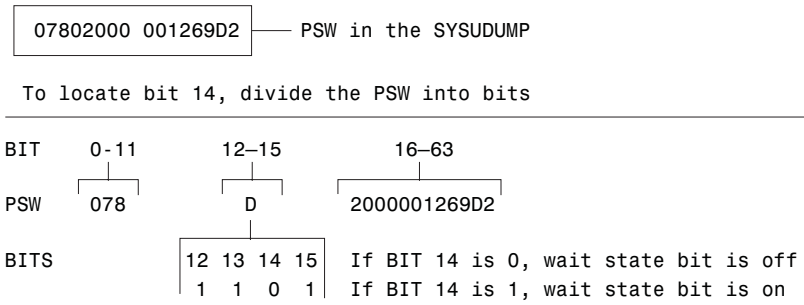


Figure 4. Finding the Wait State Bit in the PSW

3. If the wait state bit is on, use the keyword **WAIT**. If the wait state bit is off, use the keyword **LOOP**.
4. For the example in Figure 4, you would have the following keyword string:

```
5740SM105 RH00 BLOCKSET SORT LOOP
```

5. The next keyword is the module's name. Go to "Module Keyword Procedure" on page 103.

System Abend Keyword Procedure

Use this procedure when DFSORT terminates abnormally and when you receive a nonzero (except n22) *system* completion code. If the completion code is n22 (where n is any hexadecimal digit), go to "Wait and Loop Keywords Procedure" on page 100.

You need a message for this procedure. If you did not receive any messages, refer to "How to Get Messages" on page 3.

1. Record the system abend code in the following format:

   ABENDnnn

   where nnn is the system abend code.

   For example, if your system message contained ABEND=S0C1 U0000, you would have the following keyword string:

```
5740SM105 RH00 BLOCKSET SORT ABEND0C1
```

2. The next keyword is the module's name. Go to "Module Keyword Procedure" on page 103.

User Abend Keyword Procedure

Use this procedure when DFSORT terminates abnormally and when you receive a nonzero *user* completion code.

You need a message for this procedure. If you did not receive any messages, refer to "How to Get Messages" on page 3.

1. Look at the user completion code.
   - If the code is less than 1000, it corresponds to a DFSORT error message. Go to "Message Keyword Procedure" on page 102.
   - If the code is equal to or greater than 1000, continue with step 2.
2. Record the user abend keyword in the following format:

   ABENDnnnn

where nnnn is the user abend code.

For example, if your system completion message contained ABEND S000 U1004, your keyword string so far would be:

```
5740SM105 RH00 BLOCKSET SORT ABEND1004
```

3. The next keyword is the module's name. Go to "Module Keyword Procedure" on page 103.

Message Keyword Procedure

Use message keywords when an error is described by, or associated with, a DFSORT error message.

1. Locate the DFSORT error message in the message data set.

   The error message format is:

   ```
   ICEnnnA c text
   ```

Where	Represents
nnn	The error message number
c	The diagnostic trace character
text	An explanation of the error

2. Using the ICEnnnA part of the message, record the message keyword in the following format:

   ```
   MSGICEnnnA
   ```

   where nnn is the error message number.

   For example, if you received the error message ICE061A, the keyword string would be:

   ```
   5740SM105 RH00 BLOCKSET SORT MSGICE061A
   ```

3. If you received message ICE061A or ICE117A, go to "Message Variable Keywords" on page 106.

   If you did not receive either message ICE061A or ICE117A, your message keyword string is now complete. Go to Chapter 8, "Searching the Problem Reporting Databases," on page 119.

Incorrect-Output Keyword Procedure

Use the incorrect-output procedure when DFSORT:

- Generates unexpected output
- Fails to generate expected output
- Generates incorrect output

Do not use this procedure if the problem is associated with an error message. Use the message keyword procedure instead.

1. Use **INCORROUT** as your type-of-failure keyword.

   You now have the following keyword string:

   ```
   5740SM105 RH00 BLOCKSET SORT INCORROUT
   ```

2. The incorrect-output keyword string is now complete. Go to Chapter 8, "Searching the Problem Reporting Databases," on page 119.

Performance Keyword Procedure

Performance problems might be related to system tuning. Use this keyword when the performance problem cannot be corrected by system tuning.

1. Record the actual performance and the expected performance.
2. Use **PERFM** as your type-of-failure keyword.

   You now have the following keyword string:

```
5740SM105 RH00 BLOCKSET SORT PERFM
```

3. The performance keyword string is now complete. Go to Chapter 8, "Searching the Problem Reporting Databases," on page 119.

Document Keyword Procedure

Use this keyword when a problem seems to be caused by incorrect or missing information in a DFSORT publication.

If the documentation problem is not severe, use the reader's comment form (attached to the back of the manual that is causing the problem) to suggest corrections or improvements to the publication.

If you consider the problem a severe one, build the keyword string as follows:

1. Use **DOC** as your type-of-failure keyword.

   You now have the following keyword string:

```
5740SM105 RH00 DOC
```

2. Use the order number on the cover of the document as your next keyword. The order number is of the form *ccnn-nnnn-dd*. Include *ccnn*, *nnnn*, and *dd*, but omit the hyphens. *dd* is the edition number.

   For example, if you found a documentation error in *z/OS DFSORT Application Programming Guide*, your keyword string would be:

```
5740SM105 RH00 DOC SC26752300
```

3. The Document keyword string is now complete. Go to Chapter 8, "Searching the Problem Reporting Databases," on page 119.

Module Keyword Procedure

If your program terminated abnormally or entered a wait or loop, your next keyword in the keyword string is the name of the module in control when the failure occurred.

You need a dump for this procedure. If you did not receive a dump, see Table 2 on page 3.

If you invoke DFSORT from a COBOL program, you might receive both a COBOL dump and a DFSORT dump. For this procedure, use only the DFSORT dump.

If you are using PL/I, you must specify the NOSPIE and NOSTAE macro at run time to ensure that you receive a DFSORT dump. If you do not, you might receive only a PL/I dump.

You need the abend address to obtain the module's name:

1. Find the program status word (PSW) in the dump, and note the instruction address, which is the last 31 bits of the PSW.

   See Figure 5 for an example. The last 31 bits of the PSW (that is, 0720CF52) is the instruction address.

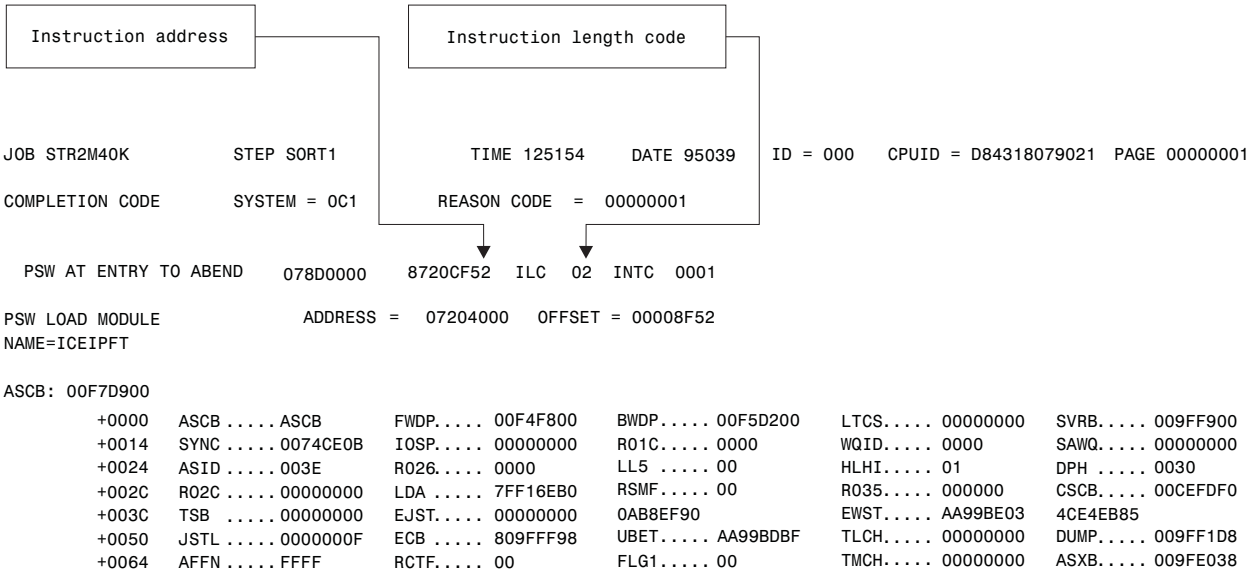


Figure 5. Finding the Abend Address

2. Obtain the abend address by subtracting the instruction length code (ILC) from the instruction address.

```
 0720CF52     instruction address
-       2     instruction length code
```

3. Go to the abend address and scan backward until you find, on the right side of the dump, the closest name of the form:

   ```
   ICEname REL nn.n
   ```

   where ICEname is the module name.

   In Figure 6 on page 105, ICEIPFT is the closest module name.

   If you cannot find the module name, the problem might not be with DFSORT. In this case, call your IBM Support Center.

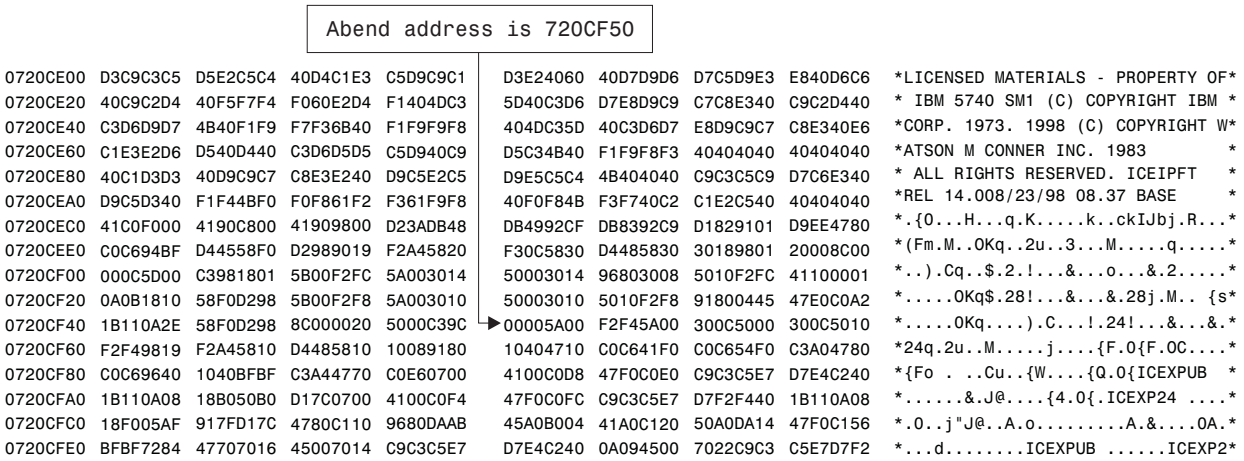


Figure 6. Finding the Module Name

4. Use the module name as the keyword. For this example, the keyword string would be:

```
5740SM105 RH00 BLOCKSET SORT ABEND0C1 ICEIPFT
```

   Check to see that the release level that appears after the module name is current. If the release level is not current, ensure DFSORT has been installed correctly.
5. The next keyword shows the offset. Go to Figure 7 on page 106.

Offset Keyword Procedure

If your program terminated abnormally or entered a wait or loop, your next keyword in the keyword string after the name of the module is the offset keyword.

To determine the offset:

1. Subtract the address of the first byte of the module from the abend address. The first byte of the module is the byte where the characters 'LICENSED' begin, just before the module name you found earlier by scanning the right side of the dump (as described in "Module Keyword Procedure" on page 103). If 'LICENSED' does not appear, the first byte is the beginning of the module name.

```
 0720CF50    abend address
-0720CE00 module's address
```

   In the example in Figure 7 on page 106, the address of the module ICEIPFT is 0720CE00, and the abend address previously found is 0720CF50. Therefore, the offset is 150.

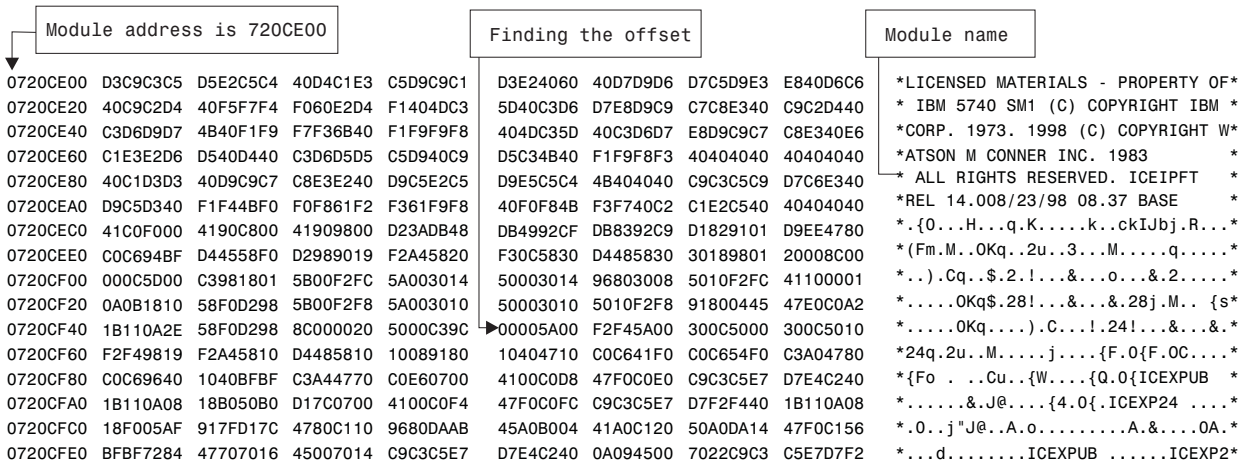


Figure 7. Finding the Offset

2. Record the offset keyword using the following format:
   OFFSET n

   where n is 1 to 4 digits and contains no leading zeros.

   For example, the keyword string for a system abend 0C1 would look like this:

```
5740SM105 RH00 BLOCKSET SORT ABEND0C1 ICEIPFT OFFSET 150
```

3. The keyword string is now complete. Go to Chapter 8, "Searching the Problem Reporting Databases," on page 119.

Message Variable Keywords

If your problem is for message ICE061A or ICE117A, you need to add a message variable keyword to your keyword string. Below are two sets of variable keywords. Add to your keyword string the set of variables that correspond to your message number.

ICE061A Variable Keywords

1. If you received message ICE061A, add to your keyword string the following four keywords from the message:
   a. The ddname as follows:
      - For an input data set, use "SORTIN"
      - For a work data set, use "SORTWK"
      - For an output data set, use "SORTOUT"
   b. **ECB**nn (where nn is the completion code)
   c. **CSW**nnnn (where nnnn are the status bytes)
   d. **SENSE**nnnn (where nnnn are the sense bytes)

   For example, if you received the message

   ```
   ICE061A 0 I/O ERROR, DD SORTWK04, DEV 6C0, ECB 41, CSW 0020, SENSE 0000
   ```

   the keyword string would be:

```
5740SM105 RH00 BLOCKSET SORT MSGICE061A SORTWK ECB41 CSW0020
SENSE0000
```

2. The message keyword string is now complete. Go to Chapter 8, "Searching the Problem Reporting Databases," on page 119.

ICE117A Variable Keywords

1. If you received message ICE117A, add to your keyword string the following three keywords from the message:
   a. The ddname as follows:
      - For an input data set, use "SORTIN"
      - For a work data set, use "SORTWK"
      - For an output data set, use "SORTOUT"
   b. **OP**nn (where nn is the operation attempted)
   c. The error description (In the message, the error description appears between the operation attempted and the last seek address or block count.)

   For example, if you received the message

   ```
   ICE117A J I/O ERROR NEW117, SORT1, 354, DA,
   SORTOUT, 0E OP, UNKNOWN COND., 00000007000100, EXCP
   ```

   the keyword string would be:

   5740SM105 RH00 PEERVALE SORT MSGICE117A SORTOUT OP0E UNKNOWN COND

   Because ICE117A is issued only for the Peerage/Vale technique, the type-of-technique keyword in the above example is now PEERVALE.
2. The message keyword string is now complete. Go to Chapter 8, "Searching the Problem Reporting Databases," on page 119.

Chapter 7. Resolving Problems When Using DFSORT Panels

DFSORT Panels is an interface with which users can create and submit DFSORT jobs using interactive panels. DFSORT Panels is an entirely separate program from the DFSORT program that sorts, copies, and merges records.

- In background execution, DFSORT Panels invokes the DFSORT program by submitting a job stream a user generates within DFSORT Panels.
- In foreground execution, DFSORT Panels invokes the DFSORT program through the TSO CALL command, or a similar command.

DFSORT Panels is supported under ISPF, but you can also install it as a selection on an ISMF menu. The SORTREC line operator uses DFSORT processing run from the ISMF Data Set Application.

Failures you encounter when using DFSORT Panels can be caused by problems in the DFSORT Panels or the DFSORT program.

- Diagnose failures in the DFSORT Panels by following the steps covered in this section and *z/OS DFSMSdfp Diagnosis* (for ISMF) and in *ISPF and ISPF/PDF Diagnosis*.
- Diagnose failures in the DFSORT program by using the numbered steps listed in Chapter 6, "Resolving Failures in the DFSORT Program," on page 93, to describe the problem and to find out if corrections for it are already available. Also see "Resolving Failures in DFSORT Panels" on page 110.

Obtaining Diagnostic Information

When you use DFSORT Panels, you can choose several options that supply dumps, messages, and diagnostic information in the event of an abend:

- Run with WTP message turned on in your TSO profile. This gives you the message identifier and an online symptom dump for error messages issued by components other than DFSORT Panels. See *z/OS TSO/E Command Reference* for information on setting up your TSO profile.
- Before invoking DFSORT Panels, allocate a SYSUDUMP, SYSABEND, or SYSMDUMP data set on TSO, so that you can have a full dump for an abend if you need one. For example, to dump to a new data set called UDUMP, use

  ```
  ALLOC DD(SYSUDUMP) DS(UDUMP) MOD SPACE(5,5) CYLINDERS CATALOG
  ```

  To dump to a system output data set, use

  ```
  ALLOC DD(SYSUDUMP) SYSOUT(Z) HOLD
  ```

  See *z/OS MVS JCL Reference* for more information on allocating a dump data set for background jobs.
- Run with the ISPF log turned on.
- Set the options in the DFSORT Panels user profile application to issue diagnostic information in the event of an abend. See "Using the Profile Application" on page 115 for details.

© Copyright IBM Corp. 1995, 2006

Determining Where the Failure Occurred

This section explains how to determine whether the error is in the DFSORT Panels or the DFSORT program, and explains how to obtain additional information to help diagnose the failure.

Answer the questions in Table 8 to locate the general cause of the abend.

Table 8. Abend Cause Location Procedure

Step	Action
1	Did your terminal display a DFSORT abend panel (Figure 8 on page 111) or a TSO message indicating failure (Figure 9 on page 111)? • If so, the abend is within the DFSORT Panels. Go to Chapter 7, "Resolving Problems When Using DFSORT Panels," on page 109. • If not, go to step 2.
2	Did your terminal display an ISPF or ISMF abend panel? See Figure 11 and Figure 12 on page 113 for examples. • If so, the abend is within ISPF or ISMF. Go to "Resolving Failures in ISPF or ISMF" on page 112. • If not, the abend is within the DFSORT program. Go to step 3.
3	Did you submit your job for foreground execution, and did you receive DFSORT error message ICE185A and a TSO abend symptom dump? • If so, go to "Foreground Execution Failures" on page 113. • If not, go to step 4.
4	Did you submit your job for background execution and receive the standard DFSORT abend and error messages? • Go to "Background Execution Failures" on page 114.

Resolving Failures in DFSORT Panels

DFSORT Panels is supported under ISPF, but you can also install it as a selection on an ISMF menu. The ISMF line operator SORTREC uses the DFSORT program to process records within the ISMF environment. Procedures for diagnosing failures in ISPF and ISMF are explained in *ISPF and ISPF/PDF Diagnosis* and *z/OS DFSMSdfp Diagnosis*. These documents explain how to construct a keyword string to search an IBM database for information about abends in ISPF and ISMF panel products.

If you are re-running a job to obtain diagnostic messages or a dump, read the information in "Obtaining Diagnostic Information" on page 109 to make sure that you have made any necessary changes to your TSO profile and have allocated any necessary data sets.

DFSORT Panels Abend Panel

If DFSORT Panels terminates abnormally, a DFSORT Panels abend panel, shown on Figure 8 on page 111, informs you of the error and identifies the appropriate abend code. Use this code to begin building the keyword string, as explained in *z/OS DFSMSdfp Diagnosis*.

```
                          DFSORT PANELS ABEND PANEL
COMMAND ===>



********************************************************************************
********************************************************************************
**                                                                            **
**                                                                            **
**             AN ABEND OCCURRED WHILE EXECUTING DFSORT PANELS                **
**                                                                            **
**                          SYSTEM ABEND CODE: 0C4                            **
**                                                                            **
**                      DFSORT PANELS CANNOT CONTINUE                         **
**                                                                            **
**         PRESS THE ENTER KEY OR USE END TO TERMINATE DFSORT PANELS          **
**             USE HELP TO DISPLAY A LIST OF COMMON ABEND CODES               **
**                                                                            **
********************************************************************************
********************************************************************************
```

Figure 8. DFSORT Panels Abend Panels Indicate the Problem Is in DFSORT Panels

TSO Abend Message

When DFSORT Panels terminates abnormally and ISPF is in Test/Trace mode, a TSO message indicates the failure. See Figure 9.

Test/Trace mode is set automatically any time you invoke the ISPF/PDF Dialog Test facility; it remains set until you end ISPF.

```
* SPF SUBTASK ABEND *
PDF      ENDED DUE TO ERROR+
READY
```

Figure 9. TSO Messages Also Indicate That the Problem Is in DFSORT Panels

The plus sign (+) at the end of the message indicates that additional information is available.

1. To see the TSO message that describes the error further, enter a question mark (**?**).
2. When your terminal displays the second TSO error message, press Enter to obtain a dump of the abend to help diagnose the error.

If WTP is on in your TSO profile (see "Obtaining Diagnostic Information" on page 109), symptom dumps for the ISPF subtask (DFSORT Panels) and the main ISPF task are generated. See Figure 10 for an example.

```
IEA995I SYMPTOM DUMP OUTPUT
 ABEND CODE SYSTEM=0C4  TIME=11.33.40 SEQ=15485 CPU=0000 ASID=009F
 PSW AT TIME OF ERROR  078D0000   822E9074  ILC 2  INTC 0D
   ACTIVE LOAD MODULE=ICEPEP01  ADDRESS=022E8AF0  OFFSET=00000584
   DATA AT PSW  022E906E - 8910000C  0A0D5880  A00CD507
   GPR  0-3  822F9CA0  000C4000  022E28C6  00000011
   GPR  4-7  00000001  022E2758  822FD928  822FD390
   GPR  8-11 00000008  0001FE78  000218D0  022DA898
   GPR 12-15 822E8B60  022DA898  00021620  00000008
 END OF SYMPTOM DUMP
** LOGICAL SCREEN REQUEST FAILED - ABEND 0C4000 **
** CONTACT YOUR SYSTEMS PROGRAMMER OR DIALOG DEVELOPER **
*** SPF MAIN TASK ABEND ***
IEA995I SYMPTOM DUMP OUTPUT
 ABEND CODE SYSTEM=0C4  TIME=11.33.48 SEQ=15486 CPU=0000 ASID=009F
 PSW AT TIME OF ERROR  078D1000   8003BA6C  ILC 2  INTC 0D
   ACTIVE LOAD MODULE=ISPMAIN   ADDRESS=00039868  OFFSET=00002204
   DATA AT PSW  0003BA66 - 00181610  0A0D98EC  D09007FE
   GPR  0-3  80000000  800C4000  0001B5D9  0003C05C
   GPR  4-7  0003C03C  00000000  00000000  00000000
   GPR  8-11 000158A8  00016620  0003B868  0003A868
   GPR 12-15 80039868  0001B2A0  8003BA40  00000000
 END OF SYMPTOM DUMP
READY
```

Figure 10. Two Symptom Dumps from TSO Appear on Your Screen

If a SYSUDUMP, SYSMDUMP, or SYSABEND data set was allocated before the error (see "Obtaining Diagnostic Information" on page 109), full dumps for the ISPF subtask and the main task are generated. If DISP=MOD was specified, the dump data set contains both dumps; otherwise, it just contains the second dump.

Use the first full dump, from the abending ISPF subtask (DFSORT Panels), to locate the name of the failing load module. The offsets identify the specific failing module within the load module.

If you did not have a dump data set allocated before the abend occurred, follow these steps:

1. Allocate a dump data set using the TSO ALLOC command (see "Obtaining Diagnostic Information" on page 109).
2. Invoke ISPF with the TEST parameter by using the ISPF TEST (or PDF TEST) command.
3. Specify N for RECOVER FROM ABENDS in your DFSORT Panels user profile.
4. Attempt to recreate the problem by going through your DFSORT Panels session again.

Resolving Failures in ISPF or ISMF

If ISPF or ISMF terminates abnormally, a panel informs you of the error and identifies the appropriate abend code. Use this code to begin building the keyword string, as explained in *z/OS DFSMSdfp Diagnosis*. See Figure 11 on page 113 and Figure 12 on page 113 for examples.

```
----------------------------  ERROR RECOVERY  --------------------------------
COMMAND ===>


       * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
       * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
       * *            ISPF PROCESSOR ENDED ABNORMALLY                 * *
       * *                                                            * *
       * *                 Task ABEND code 0C4                        * *
       * *                                                            * *
       * *                                                            * *
       * *                                                            * *
       * *  Press ENTER key to display primary option menu.           * *
       * *  Enter HELP command for list of common ABEND codes.        * *
       * *                                                            * *
       * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
       * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
```

Figure 11. ISPF Abend Panels Indicate the Problem Is in ISPF

```
                           ISMF ABEND PANEL
COMMAND ===>



********************************************************************************
********************************************************************************
**                                                                            **
**                                                                            **
**                  AN ABEND OCCURRED WHILE EXECUTING ISMF                    **
**                                                                            **
**                        SYSTEM ABEND CODE: 0C4                              **
**                                                                            **
**                         ISMF CANNOT CONTINUE                               **
**                                                                            **
**             PRESS THE ENTER KEY OR USE END TO TERMINATE ISMF               **
**             USE HELP TO DISPLAY A LIST OF COMMON ABEND CODES               **
**                                                                            **
********************************************************************************
********************************************************************************
```

Figure 12. ISMF Abend Panels Indicate the Problem Is in ISMF

Resolving Failures in the DFSORT Program

The DFSORT program can fail during foreground or background execution.

Foreground Execution Failures

If a foreground job fails because of a DFSORT program abend, you receive a TSO abend symptom dump.

More diagnostic information is available in the DFSORT program abend dump and from the TSO message output.

- To obtain a DFSORT program abend dump, follow these steps:

1. Return to TSO and allocate a SYSUDUMP, SYSABEND, or SYSMDUMP data set to capture the abend dump. See "Obtaining Diagnostic Information" on page 109.
2. Invoke DFSORT Panels again and re-run the job. The DFSORT program abend dump appears in the SYSUDUMP, SYSABEND, or SYSMDUMP data set, and you can begin the diagnosis sequence starting in "How to use this document" on page ix.

- To obtain the TSO message output, change the ALLOC FILE(SYSOUT) statement in the ICEQFG01 CLIST (for SORT), ICEQFG02 CLIST (for COPY), or ICEQFG03 CLIST (for MERGE) to direct the SYSOUT output to a user data set or a system output data set. For example:

  ```
  ALLOC FILE(SYSOUT) DS(DFSORT.OUTMSG) MOD SPACE (1,1) TRACKS CATALOG
  ```

  All DFSORT Panels CLISTs are stored in a data set allocated to SYSPROC. To change the required CLIST, make a private copy with the modifications and place it in a private data set allocated to SYSPROC (with higher priority than the CLIST system data sets).

Be certain that the abend is not caused either by modifications made at your site to the CLISTs (ICEQFG01, ICEQFG02, ICEQFG03, ICEQFG04) invoked by DFSORT Panels or by any other modifications not supported by DFSORT or by DFSORT Panels.

Background Execution Failures

With background jobs, you can diagnose failures not caused by DFSORT Panels using the diagnosis procedures in Chapter 6, "Resolving Failures in the DFSORT Program," on page 93.

If a dump data set is not present in the job stream, then specify a SYSUDUMP, SYSABEND, or SYSMDUMP DD statement on the DFSORT Execute Statement Entry Panel and resubmit the job to obtain the dump. See Table 2 on page 3 for more information.

The job can also be saved in a data set for forwarding to IBM personnel.

Recording Information

DFSORT Panels uses the ISPF log to record information about the foreground and background execution of DFSORT program jobs and error conditions detected during processing.

To ensure that log entries are made, you must specify a primary allocation greater than zero in the ISPF log data set default options on the ISMF Log and List Defaults menu. If you specify zero as a primary allocation, no information about your session appears in the ISPF log. For more information about setting up ISPF log parameters, see*ISPF/PDF Guide*.

The ISPF log routinely records:
- Successful foreground execution of a DFSORT job
- Successful submission of a JCL job stream for background execution
- Successful saving of JCL job stream in a data set

The log also contains entries for error conditions:
- Foreground execution failures
- Failures to submit a job stream because of SUBMIT errors

- Failures to save a JCL job stream in a data set because of DYNALLOC, OPEN, CLOSE, or DCB address errors
- Data set verification failures because of LOCATE, DYNALLOC, OPEN, BLDL, or CLOSE errors
- User-Defined Panels Interface (UDPI) errors
- ISPF service failures

Inter-Module Traces

To receive a log record for every module that executes during a DFSORT Panels session, enter a **Y** in the LOG INTER-MODULE TRACE on the Logging/Abend Control Entry Panel in the Profile Application. This information can help diagnose loops or determine module flow when there is more than one possible path to an error.

Normally, set this option to N, because logging degrades performance.

ISMF Logging with SORTREC

Table 9 shows DFSORT reason codes can appear in ISMF error messages in the ISPF log if the DFSORT SORTREC line operator processor detects an error. For more information on ISMF's use of the ISPF log, see *z/OS DFSMSdfp Diagnosis*.

Table 9. DFSORT Reason Codes

Reason Code	Description
RSN08000	OBJECT NAME (DATA SET NAME IN DATA SET APPLICATION, VOLUME SERIAL NUMBER IN VOLUME APPLICATION) IS BLANK
RSN08001	INPUT DATA SET NAME WAS NOT FOUND IN ANY CATALOG IN THE SPECIFIED SEARCH ORDER
RSN08002	DATA SET NAME IS GDG BASE NAME
RSN08003	DATA SET HAS UNEXPECTED CATALOG ENTRY TYPE
RSN08004	DYNAMIC ALLOCATION FAILED FOR INPUT DATA SET DURING SORTREC PROCESSING
RSN08005	AN INVALID LOGGING TYPE WAS PASSED TO THE ISMF LOG UTILITY
RSN08006	LOG UTILITY—A REQUEST WAS MADE TO LOG ABEND INFORMATION BUT THE POINTER TO THE RETRY WORK AREA (RWA) IS INVALID
RSN08007	LOG: MESSAGE NOT FOUND, RC=12
RSN08008	LOG: SEVERE ERROR, RC=20
RSN08009	LOG: INVALID RETURN CODE
RSN08010	A REQUEST WAS MADE TO LOG STATUS INFORMATION BUT NO ADDITIONAL MESSAGE IDS WERE SPECIFIED. AT LEAST ONE MESSAGE ID MUST BE SPECIFIED.

Using the Profile Application

DFSORT Panels responds to abends based on:

- The value (**Y** or **N**) you specify in the RECOVER FROM ABENDS field of the Logging/Abend Control Entry Panel of your profile
- The way you invoke ISPF (in Normal or Test/Trace mode)

- The value (**Y** or **N**) you specify in the ISSUE DFSORT DIAGNOSTIC MESSAGES field of the DFSORT Control Entry Panel of your profile.

These options are summarized in the following sections. See *z/OS DFSORT Application Programming Guide* for complete details.

Directing Abend Processing

Use the RECOVER FROM ABENDS field of the Logging/Abend Control Entry Panel to direct the DFSORT Panels to attempt to recover from abends or to transfer control to ISPF or TSO.

Recovering from Abends

If you specify **Y** in the RECOVER FROM ABENDS field of the Logging/Abend Control Entry Panel, DFSORT Panels attempts to recover from the abend and retain control. When an abend occurs, a DFSORT Panels symptom dump from the time of the failure is placed in the ISPF log. In some cases the symptom dump is sufficient to diagnose the problem. It contains information about:

- The application involved in the failure
- The function involved in the failure
- The current release level and function modification identifier (FMID) for DFSORT Panels
- The abend code
- The name of the module involved in the failure (normally ICEPSA01)
- The entry point address (EPA) of the failing module
- The program status word (PSW)
- The contents of the registers when the abend occurred

Passing Control to ISPF or TSO

If you specify **N** in the RECOVER FROM ABENDS field, DFSORT does not attempt to retain control when an abend occurs. The type of dump you receive depends upon whether the ISPF mode is set to Normal or Test/Trace:

- Normal mode

  DFSORT Panels passes control to ISPF. The ISPF error recovery panel appears and the log receives an ISPF symptom dump, consisting of:
  - The level of ISPF
  - The abend code
  - The PSW
  - The name of the module involved in the failure, if available
  - The EPA of the module involved in the failure
  - The EPA of the modules ISPUBS and ISTBLS
  - The contents of the registers when the abend occurred
- Test/Trace mode

  ISPF passes control to TSO. TSO displays an error message. If a SYSUDUMP, SYSABEND, or SYSMDUMP data set was allocated on TSO, you receive a dump. This method gives you the most accurate information on the abend.

  Test/Trace mode is set automatically any time you invoke the ISPF/PDF Dialog Test facility; it remains set until you end ISPF.

Obtaining Diagnostic Messages

If you specify **Y** in the ISSUE DFSORT DIAGNOSTIC MESSAGES field of the DFSORT Control Entry Panel of your profile, DFSORT issues diagnostic messages to provide information about EXCP counts, buffer allocation, and processing techniques.

For background jobs, DFSORT generates a SORTDIAG DD statement to be inserted into your completed job stream. For foreground jobs, DFSORT generates the corresponding ALLOCATE command to achieve the same result under TSO.

Specify **N** in the field to direct DFSORT not to issue diagnostic messages.

Chapter 8. Searching the Problem Reporting Databases

After you complete your keyword string, you are ready to search the IBM Software Support Facility to determine if an APAR has already been recorded.

Table 10. Steps to Search the IBM Software Support Facility

Step	Action
1	Search the IBM software support data base for your keyword string by using a search tool, such as Info Access, or by calling the IBM Support Center for assistance. If the problem has been reported before, a correction might already be available. This correction could be a PTF, an APAR fix, or a bypass.
2	If you find a set of APAR closing descriptions, compare the APAR closing descriptions in the IBM software support data base with your problem.
3	If you find an APAR that matches your problem, correct your problem by applying the PTF, APAR fix, or bypass described in the APAR. See Chapter 9, "Fixing or Bypassing the Problem," on page 121.
4	If you cannot find an appropriate APAR, try varying your search argument: 1. Drop one keyword at a time starting from the right until you get to the type-of-application keyword. 2. If your type-of-failure keyword is **LOOP**, **WAIT**, or **PERFM**, try searching with one or both of the other two. Sometimes what seems like a WAIT state might actually be a LOOP or a PERFM problem. If you cannot find an appropriate APAR using either technique, go to Chapter 10, "Reporting a Problem," on page 123.

© Copyright IBM Corp. 1995, 2006

Searching the IBM Software Support Data Base

Chapter 9. Fixing or Bypassing the Problem

Fixing the Problem

Apply the appropriate PTF or APAR fix using SMP/E. If you are running DFSORT resident, you must re-IPL with CLPA to enable the fix, unless you are using Dynamic LPA.

For more information, see *SMP/E User's Guide*.

Bypassing the Problem

This list describes some actions you can take that might allow you to bypass a problem:

- **If the problem is related to I/O for SORTIN, SORTOUT, or OUTFIL,** try using the BSAM run-time option.
- **If the problem is related to storage**, try adding 32 KB by changing the REGION parameter or the MAINSIZE or SIZE run-time option. System 0C4 and DFSORT user 1nnn abends are sometimes related to storage problems and can be bypassed in this way.
- **If the problem is with the Blockset technique,** try using the NOBLKSET run-time option TEMPORARILY to force DFSORT to use a different technique. For more information, see "Type-of-Technique Keyword Procedure" on page 98.

Note: OPTION NOBLKSET should NOT be used as a permanent solution, because it can significantly degrade performance.

© Copyright IBM Corp. 1995, 2006

Chapter 10. Reporting a Problem

Follow these instructions to report a problem to the IBM Support Center.

Step	Action
1	Gather the information. 1. Have the following information ready when you report a problem: • Customer number • Current service level (PTF list and list of APAR fixes applied) • The keyword string developed to search the IBM software support database • Processor number (serial model) 2. Also have the following documents ready to assist in analyzing the problem: • Source code (assembler or compiler listings) and load module for any user routines or invoking program. • A listing of the job control language (JCL), DFSORT control statements, and all messages. You can use a SORTDIAG DD statement to obtain all messages, including diagnostic messages. • A listing of the symbols information if a SYMNAMES DD statement was specified. You can produce this listing by specifying a SYMNOUT DD statement. • A copy of the input data set in machine-readable form. • A list of the installation defaults selected at your site. You can produce this list by using the DEFAULTS operator of ICETOOL with the same step, job, or link library used for the job associated with this APAR. See *z/OS DFSORT Application Programming Guide* for complete details of the DEFAULTS operator of ICETOOL. • A dump from the failing run of DFSORT.
2	Contact the IBM Support Center. After gathering the information, contact the IBM Support Center for assistance in resolving the problem. If the difficulty is an IBM problem, the representative will assist you in preparing an APAR.
3	Submit APAR documentation. When submitting material for an APAR to IBM, carefully pack and clearly identify any tapes that contain application source programs, job stream data, data sets, or libraries. Each tape must have the following information attached and visible: • The APAR number assigned by IBM • A list of data sets on the tape (for example, application source program, JCL, data, and load modules) • Data on how the tape was created: – The exact JCL listing or the list of commands used – The record format and block size used for each data set – Tape labeling – The recording mode and density

© Copyright IBM Corp. 1995, 2006

Part 4. Appendixes

© Copyright IBM Corp. 1995, 2006

Appendix A. Valid Input/Output Options at User Exits—ICE044I

An *x* in the table indicates an option that you can use with the user exit in question.

Option	E18	E19	E38	E39
SYNAD	x	x	x	x
EXLST	x[1]	x	x	x
EROPT	x		x	
EODAD	x			
BSAM EXLST	x		x[2]	x
VSAM PASSWORD	x		x[2]	x

Notes:

1 Cannot be used if input is concatenated on unlike devices.

2 For merge applications.

© Copyright IBM Corp. 1995, 2006

Appendix B. User Exit Return Codes—ICE071A

An x in the table indicates a valid return code that your user exit can pass back for the given situation. All other values (those not marked with an x and those not in the table) are invalid for the given situation.

End of input (EOI) is indicated by a zero in the first word of the parameter list passed to your exit.

Table 11. User Exit Return Codes - ICE071A

				Valid Return Codes					
Enn	**Case**	**EXITCK**	**Situation**	**0**	**4**	**8**	**12**	**16**	**20***
E15	S1	STRONG	Without SORTIN			X	X	X	
E15	W1	WEAK	Without SORTIN	X	X	X	X	X	X
E15	S2	STRONG	With SORTIN before EOI	X	X	X	X	X	X
E15	W2	WEAK	With SORTIN before EOI	X	X	X	X	X	X
E15	S3	STRONG	With SORTIN after EOI			X	X	X	
E15	W3	WEAK	With SORTIN after EOI	X	X	X	X	X	X
E16	16	N/A	N/A	X	X	X			
E32	32	N/A	N/A			X	X	X	
E35	S0	STRONG	Without SORTOUT or OUTFIL before EOI		X	X		X	
E35	W0	WEAK	Without SORTOUT or OUTFIL before EOI	X	X	X	X	X	X
E35	S1	STRONG	Without SORTOUT or OUTFIL after EOI			X		X	
E35	W1	WEAK	Without SORTOUT or OUTFIL after EOI	X	X	X	X	X	X
E35	S2	STRONG	With SORTOUT or OUTFIL before EOI	X	X	X	X	X	X
E35	W2	WEAK	With SORTOUT or OUTFIL before EOI	X	X	X	X	X	X
E35	S3	STRONG	With SORTOUT or OUTFIL after EOI			X	X	X	
E35	W3	WEAK	With SORTOUT or OUTFIL after EOI	X	X	X	X	X	X
Note: * Only applies to COBOL E15 and COBOL E35									

© Copyright IBM Corp. 1995, 2006

Appendix C. Accessibility

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use software products successfully. The major accessibility features in z/OS enable users to:

- Use assistive technologies such as screen readers and screen magnifier software
- Operate specific or equivalent features using only the keyboard
- Customize display attributes such as color, contrast, and font size

Using assistive technologies

Assistive technology products, such as screen readers, function with the user interfaces found in z/OS. Consult the assistive technology documentation for specific information when using such products to access z/OS interfaces.

Keyboard navigation of the user interface

Users can access z/OS user interfaces using TSO/E or ISPF. Refer to *z/OS TSO/E Primer*, *z/OS TSO/E User's Guide*, and *z/OS ISPF User's Guide Vol I*for information about accessing TSO/E and ISPF interfaces. These guides describe how to use TSO/E and ISPF, including the use of keyboard shortcuts or function keys (PF keys). Each guide includes the default settings for the PF keys and explains how to modify their functions.

z/OS information

z/OS information is accessible using screen readers with the BookServer/Library Server versions of z/OS books in the Internet library at:

`www.ibm.com/servers/eserver/zseries/zos/bkserv/`

© Copyright IBM Corp. 1995, 2006

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:

© Copyright IBM Corp. 1995, 2006

IBM Corporation
Mail Station P300
2455 South Road
Poughkeepsie, NY 12601-5400
USA

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this 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 publication primarily documents information that is NOT intended to be used as Programming Interfaces of DFSORT.

This publication also documents intended Programming Interfaces that allow the customer to write programs to obtain the services of DFSORT. This information is identified where it occurs, either by an introductory statement to a chapter or section or by the following marking:

Programming Interface information

End of Programming Interface information

Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States or other countries or both:

DFSMSrmm
DFSORT
Hiperspace
IBM
IBMLink
Language Environment
Resource Link
z/OS
z/VM

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, and service names may be trademarks or service marks of others.

Readers' Comments — We'd Like to Hear from You

z/OS
DFSORT Messages, Codes and Diagnosis Guide

Publication No. SC26-7525-02

We appreciate your comments about this publication. Please comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of this book. The comments you send should pertain to only the information in this manual or product and the way in which the information is presented.

For technical questions and information about products and prices, please contact your IBM branch office, your IBM business partner, or your authorized remarketer.

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. IBM or any other organizations will only use the personal information that you supply to contact you about the issues that you state on this form.

Comments:

Thank you for your support.

Submit your comments using one of these channels:

- Send your comments to the address on the reverse side of this form.

If you would like a response from IBM, please fill in the following information:

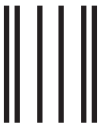
Name	Address
Company or Organization	
Phone No.	E-mail address

Readers' Comments — We'd Like to Hear from You
SC26-7525-02



Cut or Fold
Along Line

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
Department 55JA, Mail Station P384
2455 South Road
Poughkeepsie, NY 12601-5400



Fold and Tape | **Please do not staple** | Fold and Tape

Cut or Fold
Along Line



Program Number: 5694-A01

Printed in USA

SC26-7525-02

