Note
Before using this information and the product it supports, read the information in “Notices” on page 167.

This edition applies to Version 2 Release 3 of z/OS (5650-ZOS) and to all subsequent releases and modifications until otherwise indicated in new editions.

Last updated: 2019-02-16

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
List of Figures.................................................................................................................. xv
List of Tables.................................................................................................................... xvii
About this document....................................................................................................... xix
  How to use this document............................................................................................ xix
  Required product knowledge....................................................................................... xix
  Referenced documents............................................................................................... xx
How to send your comments to IBM...............................................................................xxi
z/OS Version 2 Release 3 summary of changes..............................................................xxii
  New information........................................................................................................... xxii
    Sort/Merge of Unicode data...................................................................................... xxii
    E15/E35 Block Support.............................................................................................. xxii
    Performance Improvements.................................................................................... xxii
    Message changes..................................................................................................... xxii
    New Reserved Words for Symbols.......................................................................... xxii
Summary of changes for SC23-6879-00 z/OS Version 2 Release 1..............................xxiv
  New information........................................................................................................... xxiv
    High Performance Ficon (zHPF) Exploitation......................................................... xxiv
    Date functions........................................................................................................... xxiv
    Message changes..................................................................................................... xxiv
Summary of changes for SC23-6879-01 z/OS Version 2 Release 2..............................xxv
  New information........................................................................................................... xxv
    64-bit support............................................................................................................. xxv
    Performance and resource usage improvements.................................................. xxv
    Extended address volumes..................................................................................... xxv
    Alphanumeric Comparison Tests.......................................................................... xxvii
    More PARSE Fields................................................................................................. xxvii
    Repeating PARSE Fields....................................................................................... xxvii
    Alphanumeric Tests for PARSE Fields................................................................. xxvii
    Insert a String at the End of Variable-Length Records....................................... xxvii
    Symbol Enhancements........................................................................................... xxviii
    Improved diagnostics............................................................................................ xxviii
  Operational changes that may require user action................................................... xxviii
    Central storage controls....................................................................................... xxviii
    Message changes.................................................................................................... xxviii
    New Reserved Words for Symbols........................................................................ xxviii
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............................................................................. 4
      Errors in record contents.................................................................................. 5
Part 2. Messages and codes ................................................................. 11

Chapter 2. Message considerations .................................................. 13
  Printing messages and control statements ........................................ 13
  Printing EFS messages and control statements .................................. 14
  Writing messages to the master console ........................................... 14
  Finding DFSORT control statement coding errors ............................ 15

Chapter 3. Informational and Error Messages ..................................... 17
  ICE000I ......................................................................................... 17
  ICE001A ...................................................................................... 17
  ICE002I ...................................................................................... 17
  ICE003A ...................................................................................... 18
  ICE005A ...................................................................................... 18
  ICE006A ...................................................................................... 18
  ICE007A ...................................................................................... 18
  ICE008A ...................................................................................... 19
  ICE010A ...................................................................................... 19
  ICE011A ...................................................................................... 19
  ICE012A ...................................................................................... 19
  ICE013A ...................................................................................... 19
  ICE014A ...................................................................................... 19
  ICE015A ...................................................................................... 20
  ICE016A ...................................................................................... 20
  ICE017A ...................................................................................... 20
  ICE018A ...................................................................................... 21
  ICE019A ...................................................................................... 22
  ICE020A ...................................................................................... 22
  ICE021A ...................................................................................... 22
  ICE022A ...................................................................................... 22
  ICE023A ...................................................................................... 23
  ICE024A ...................................................................................... 23
  ICE025A ...................................................................................... 24
  ICE026I ...................................................................................... 24
  ICE027A ...................................................................................... 24
  ICE028A ...................................................................................... 24
  ICE029A ...................................................................................... 24
  ICE030A ...................................................................................... 25
  ICE031A ...................................................................................... 25
  ICE032A ...................................................................................... 25
  ICE033A ...................................................................................... 25
  ICE034A ...................................................................................... 25
  ICE035A ...................................................................................... 26
  ICE036I ...................................................................................... 26
  ICE037I ...................................................................................... 26
  ICE038I ...................................................................................... 26
  ICE039A ...................................................................................... 26
  ICE040A ...................................................................................... 27
  ICE041A ...................................................................................... 27
  ICE042A ...................................................................................... 27
<table>
<thead>
<tr>
<th>Code</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICE043A</td>
<td>28</td>
</tr>
<tr>
<td>ICE044I</td>
<td>30</td>
</tr>
<tr>
<td>ICE045I</td>
<td>30</td>
</tr>
<tr>
<td>ICE046A</td>
<td>31</td>
</tr>
<tr>
<td>ICE047A</td>
<td>31</td>
</tr>
<tr>
<td>ICE048I</td>
<td>31</td>
</tr>
<tr>
<td>ICE049I</td>
<td>32</td>
</tr>
<tr>
<td>ICE050I</td>
<td>32</td>
</tr>
<tr>
<td>ICE051A</td>
<td>32</td>
</tr>
<tr>
<td>ICE052I</td>
<td>32</td>
</tr>
<tr>
<td>ICE053A</td>
<td>32</td>
</tr>
<tr>
<td>ICE054I</td>
<td>32</td>
</tr>
<tr>
<td>ICE055I</td>
<td>32</td>
</tr>
<tr>
<td>ICE056A</td>
<td>33</td>
</tr>
<tr>
<td>ICE057A</td>
<td>33</td>
</tr>
<tr>
<td>ICE058A</td>
<td>33</td>
</tr>
<tr>
<td>ICE059A</td>
<td>33</td>
</tr>
<tr>
<td>ICE060I</td>
<td>34</td>
</tr>
<tr>
<td>ICE061A</td>
<td>34</td>
</tr>
<tr>
<td>ICE062A</td>
<td>34</td>
</tr>
<tr>
<td>ICE063A</td>
<td>34</td>
</tr>
<tr>
<td>ICE064A</td>
<td>35</td>
</tr>
<tr>
<td>ICE065A</td>
<td>35</td>
</tr>
<tr>
<td>ICE066I</td>
<td>35</td>
</tr>
<tr>
<td>ICE067I</td>
<td>35</td>
</tr>
<tr>
<td>ICE068A</td>
<td>35</td>
</tr>
<tr>
<td>ICE069A</td>
<td>36</td>
</tr>
<tr>
<td>ICE070A</td>
<td>36</td>
</tr>
<tr>
<td>ICE071A</td>
<td>36</td>
</tr>
<tr>
<td>ICE072I</td>
<td>37</td>
</tr>
<tr>
<td>ICE073A</td>
<td>37</td>
</tr>
<tr>
<td>ICE074I</td>
<td>37</td>
</tr>
<tr>
<td>ICE075A</td>
<td>38</td>
</tr>
<tr>
<td>ICE076A</td>
<td>38</td>
</tr>
<tr>
<td>ICE077A</td>
<td>38</td>
</tr>
<tr>
<td>ICE078A</td>
<td>38</td>
</tr>
<tr>
<td>ICE079A</td>
<td>39</td>
</tr>
<tr>
<td>ICE080I</td>
<td>39</td>
</tr>
<tr>
<td>ICE081A</td>
<td>39</td>
</tr>
<tr>
<td>ICE082I</td>
<td>39</td>
</tr>
<tr>
<td>ICE083A</td>
<td>39</td>
</tr>
<tr>
<td>ICE084I</td>
<td>40</td>
</tr>
<tr>
<td>ICE085A</td>
<td>40</td>
</tr>
<tr>
<td>ICE088I</td>
<td>41</td>
</tr>
<tr>
<td>ICE089I</td>
<td>41</td>
</tr>
<tr>
<td>ICE090I</td>
<td>41</td>
</tr>
<tr>
<td>ICE091I</td>
<td>41</td>
</tr>
<tr>
<td>ICE092I</td>
<td>42</td>
</tr>
<tr>
<td>ICE093I</td>
<td>42</td>
</tr>
<tr>
<td>ICE094I</td>
<td>42</td>
</tr>
<tr>
<td>ICE095A</td>
<td>42</td>
</tr>
<tr>
<td>ICE096I</td>
<td>43</td>
</tr>
<tr>
<td>ICE097I</td>
<td>43</td>
</tr>
<tr>
<td>ICE098I</td>
<td>43</td>
</tr>
<tr>
<td>ICE099A</td>
<td>43</td>
</tr>
<tr>
<td>ICE101A</td>
<td>44</td>
</tr>
<tr>
<td>ICE102A</td>
<td>44</td>
</tr>
<tr>
<td>ICE103A</td>
<td>44</td>
</tr>
<tr>
<td>ICE162I</td>
<td>61</td>
</tr>
<tr>
<td>ICE163A</td>
<td>61</td>
</tr>
<tr>
<td>ICE164A</td>
<td>61</td>
</tr>
<tr>
<td>ICE165I</td>
<td>61</td>
</tr>
<tr>
<td>ICE168I</td>
<td>62</td>
</tr>
<tr>
<td>ICE170I</td>
<td>62</td>
</tr>
<tr>
<td>ICE171I</td>
<td>62</td>
</tr>
<tr>
<td>ICE172A</td>
<td>62</td>
</tr>
<tr>
<td>ICE173I</td>
<td>63</td>
</tr>
<tr>
<td>ICE174I</td>
<td>63</td>
</tr>
<tr>
<td>ICE175I</td>
<td>63</td>
</tr>
<tr>
<td>ICE176A</td>
<td>64</td>
</tr>
<tr>
<td>ICE177A</td>
<td>64</td>
</tr>
<tr>
<td>ICE178I</td>
<td>64</td>
</tr>
<tr>
<td>ICE179A</td>
<td>64</td>
</tr>
<tr>
<td>ICE180I</td>
<td>64</td>
</tr>
<tr>
<td>ICE181A</td>
<td>65</td>
</tr>
<tr>
<td>ICE182A</td>
<td>65</td>
</tr>
<tr>
<td>ICE183A</td>
<td>65</td>
</tr>
<tr>
<td>ICE184I</td>
<td>65</td>
</tr>
<tr>
<td>ICE185A</td>
<td>65</td>
</tr>
<tr>
<td>ICE186A</td>
<td>66</td>
</tr>
<tr>
<td>ICE187I</td>
<td>66</td>
</tr>
<tr>
<td>ICE188I</td>
<td>66</td>
</tr>
<tr>
<td>ICE189A</td>
<td>66</td>
</tr>
<tr>
<td>ICE190A</td>
<td>67</td>
</tr>
<tr>
<td>ICE191I</td>
<td>68</td>
</tr>
<tr>
<td>ICE192I</td>
<td>68</td>
</tr>
<tr>
<td>ICE193I</td>
<td>68</td>
</tr>
<tr>
<td>ICE195A</td>
<td>68</td>
</tr>
<tr>
<td>ICE196A</td>
<td>69</td>
</tr>
<tr>
<td>ICE197I</td>
<td>69</td>
</tr>
<tr>
<td>ICE198I</td>
<td>69</td>
</tr>
<tr>
<td>ICE199I</td>
<td>69</td>
</tr>
<tr>
<td>ICE200I</td>
<td>69</td>
</tr>
<tr>
<td>ICE201I</td>
<td>69</td>
</tr>
<tr>
<td>ICE202I</td>
<td>70</td>
</tr>
<tr>
<td>ICE203I</td>
<td>70</td>
</tr>
<tr>
<td>ICE204A</td>
<td>70</td>
</tr>
<tr>
<td>ICE205A</td>
<td>70</td>
</tr>
<tr>
<td>ICE206A</td>
<td>71</td>
</tr>
<tr>
<td>ICE209A</td>
<td>71</td>
</tr>
<tr>
<td>ICE210I</td>
<td>71</td>
</tr>
<tr>
<td>ICE211I</td>
<td>71</td>
</tr>
<tr>
<td>ICE212A</td>
<td>71</td>
</tr>
<tr>
<td>ICE213A</td>
<td>72</td>
</tr>
<tr>
<td>ICE214A</td>
<td>72</td>
</tr>
<tr>
<td>ICE215A</td>
<td>74</td>
</tr>
<tr>
<td>ICE216A</td>
<td>74</td>
</tr>
<tr>
<td>ICE217A</td>
<td>74</td>
</tr>
<tr>
<td>ICE218A</td>
<td>75</td>
</tr>
<tr>
<td>ICE219I</td>
<td>75</td>
</tr>
<tr>
<td>ICE220A</td>
<td>75</td>
</tr>
<tr>
<td>ICE221A</td>
<td>76</td>
</tr>
<tr>
<td>ICE222A</td>
<td>77</td>
</tr>
<tr>
<td>ICE223A</td>
<td>77</td>
</tr>
<tr>
<td>ICE224A</td>
<td>78</td>
</tr>
<tr>
<td>ICE225A</td>
<td>78</td>
</tr>
<tr>
<td>Module Code</td>
<td>Page</td>
</tr>
<tr>
<td>-------------</td>
<td>------</td>
</tr>
<tr>
<td>ICE622A</td>
<td>109</td>
</tr>
<tr>
<td>ICE623A</td>
<td>109</td>
</tr>
<tr>
<td>ICE624A</td>
<td>110</td>
</tr>
<tr>
<td>ICE625A</td>
<td>110</td>
</tr>
<tr>
<td>ICE626A</td>
<td>110</td>
</tr>
<tr>
<td>ICE627I</td>
<td>110</td>
</tr>
<tr>
<td>ICE628I</td>
<td>111</td>
</tr>
<tr>
<td>ICE629A</td>
<td>111</td>
</tr>
<tr>
<td>ICE630I</td>
<td>111</td>
</tr>
<tr>
<td>ICE631I</td>
<td>111</td>
</tr>
<tr>
<td>ICE632I</td>
<td>112</td>
</tr>
<tr>
<td>ICE633A</td>
<td>112</td>
</tr>
<tr>
<td>ICE634A</td>
<td>112</td>
</tr>
<tr>
<td>ICE635A</td>
<td>112</td>
</tr>
<tr>
<td>ICE636A</td>
<td>112</td>
</tr>
<tr>
<td>ICE637A</td>
<td>112</td>
</tr>
<tr>
<td>ICE638I</td>
<td>113</td>
</tr>
<tr>
<td>ICE639A</td>
<td>113</td>
</tr>
<tr>
<td>ICE640A</td>
<td>114</td>
</tr>
<tr>
<td>ICE641A</td>
<td>114</td>
</tr>
<tr>
<td>ICE642A</td>
<td>114</td>
</tr>
<tr>
<td>ICE643I</td>
<td>114</td>
</tr>
<tr>
<td>ICE644A</td>
<td>115</td>
</tr>
<tr>
<td>ICE645A</td>
<td>115</td>
</tr>
<tr>
<td>ICE646A</td>
<td>115</td>
</tr>
<tr>
<td>ICE647I</td>
<td>115</td>
</tr>
<tr>
<td>ICE648I</td>
<td>116</td>
</tr>
<tr>
<td>ICE649A</td>
<td>116</td>
</tr>
<tr>
<td>ICE650I</td>
<td>116</td>
</tr>
<tr>
<td>ICE651I</td>
<td>116</td>
</tr>
<tr>
<td>ICE652A</td>
<td>116</td>
</tr>
<tr>
<td>ICE653A</td>
<td>117</td>
</tr>
<tr>
<td>ICE654A</td>
<td>117</td>
</tr>
<tr>
<td>ICE655I</td>
<td>117</td>
</tr>
<tr>
<td>ICE656A</td>
<td>117</td>
</tr>
<tr>
<td>ICE657A</td>
<td>117</td>
</tr>
<tr>
<td>ICE658A</td>
<td>118</td>
</tr>
<tr>
<td>ICE659I</td>
<td>118</td>
</tr>
<tr>
<td>ICE660A</td>
<td>118</td>
</tr>
<tr>
<td>ICE661A</td>
<td>118</td>
</tr>
<tr>
<td>ICE700A</td>
<td>118</td>
</tr>
<tr>
<td>ICE701A</td>
<td>118</td>
</tr>
<tr>
<td>ICE702A</td>
<td>119</td>
</tr>
<tr>
<td>ICE703A</td>
<td>119</td>
</tr>
<tr>
<td>ICE704A</td>
<td>119</td>
</tr>
<tr>
<td>ICE705A</td>
<td>119</td>
</tr>
<tr>
<td>ICE706I</td>
<td>119</td>
</tr>
<tr>
<td>ICE707A</td>
<td>120</td>
</tr>
<tr>
<td>ICE708A</td>
<td>120</td>
</tr>
<tr>
<td>ICE709A</td>
<td>120</td>
</tr>
<tr>
<td>ICE710A</td>
<td>120</td>
</tr>
<tr>
<td>ICE711I</td>
<td>120</td>
</tr>
<tr>
<td>ICE712I</td>
<td>120</td>
</tr>
<tr>
<td>ICE713A</td>
<td>121</td>
</tr>
<tr>
<td>ICE714A</td>
<td>121</td>
</tr>
<tr>
<td>ICE715A</td>
<td>121</td>
</tr>
<tr>
<td>ICE716A</td>
<td>121</td>
</tr>
<tr>
<td>ICE717A</td>
<td>122</td>
</tr>
</tbody>
</table>
# List of Figures

1. Keyword String Flowchart (Part 1 of 2) .................................................................................................... 139  
2. Keyword String Flowchart (Part 2 of 2) .................................................................................................... 140  
3. Keyword String Syntax .............................................................................................................................. 141  
4. Finding the System Abend Code and the PSW in the SYSUDUMP ........................................................... 145  
5. Finding the Wait State Bit in the PSW ....................................................................................................... 145  
6. Finding the Abend Address ....................................................................................................................... 148  
7. Finding the Module Name ......................................................................................................................... 149  
8. Finding the Offset ...................................................................................................................................... 150
# List of Tables

1. Reference documentation........................................................................................................................... xx
2. Getting a Dump..............................................................................................................................................3
3. Common System Abends Related to Storage...............................................................................................6
4. Miscellaneous System Abends..................................................................................................................... 6
5. Program Failure Resolution...................................................................................................................... 137
6. Types of DFSORT Failures.........................................................................................................................144
7. Steps to Search the IBM Software Support Facility................................................................................. 153
8. User Exit Return Codes - ICE071A........................................................................................................... 161
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 11 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 135 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 163.

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 137, to describe the problem and to find out if corrections for it are already available.

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:

<table>
<thead>
<tr>
<th>Related Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>z/OS DFSORT Application Programming Guide</td>
</tr>
<tr>
<td>z/OS MVS System Messages, Vol 7 (IEB-IEE)</td>
</tr>
<tr>
<td>z/OS MVS System Messages, Vol 8 (IEF-IGD)</td>
</tr>
<tr>
<td>z/OS MVS System Codes</td>
</tr>
</tbody>
</table>

© Copyright IBM Corp. 1995, 2017
Referenced documents

This document refers to the following documents:

Table 1: Reference documentation

<table>
<thead>
<tr>
<th>Reference documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMP/E for z/OS Reference</td>
</tr>
<tr>
<td>z/OS DFSMSdfp Diagnosis</td>
</tr>
<tr>
<td>z/OS DFSMS Macro Instructions for Data Sets</td>
</tr>
<tr>
<td>z/OS DFSMS Using Data Sets</td>
</tr>
<tr>
<td>z/OS MVS JCL Reference</td>
</tr>
<tr>
<td>z/OS MVS Programming: Authorized Assembler Services Guide</td>
</tr>
<tr>
<td>z/OS MVS System Management Facilities (SMF)</td>
</tr>
<tr>
<td>z/OS TSO/E Command Reference</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Task</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Planning For and Customizing DFSORT</td>
<td>z/OS DFSORT Installation and Customization</td>
</tr>
<tr>
<td>Learning to Use DFSORT</td>
<td>z/OS DFSORT: Getting Started</td>
</tr>
<tr>
<td>Application Programming</td>
<td>z/OS DFSORT Application Programming Guide</td>
</tr>
<tr>
<td>Tuning DFSORT</td>
<td>z/OS DFSORT Tuning Guide</td>
</tr>
</tbody>
</table>
How to send your comments to IBM

We invite you to submit comments about the z/OS® product documentation. Your valuable feedback helps to ensure accurate and high-quality information.

**Important:** If your comment regards a technical question or problem, see instead “If you have a technical problem” on page xxi.

Submit your feedback by using the appropriate method for your type of comment or question:

**Feedback on z/OS function**
If your comment or question is about z/OS itself, submit a request through the IBM RFE Community (www.ibm.com/developerworks/rfe/).

**Feedback on IBM® Knowledge Center function**
If your comment or question is about the IBM Knowledge Center functionality, for example search capabilities or how to arrange the browser view, send a detailed email to IBM Knowledge Center Support at ibmkc@us.ibm.com.

**Feedback on the z/OS product documentation and content**
If your comment is about the information that is provided in the z/OS product documentation library, send a detailed email to mhvrcfs@us.ibm.com. We welcome any feedback that you have, including comments on the clarity, accuracy, or completeness of the information.

To help us better process your submission, include the following information:
- Your name, company/university/institution name, and email address
- The following deliverable title and order number: z/OS DFSORT Messages, Codes and Diagnosis Guide, SC23-6879-30
- The section title of the specific information to which your comment relates
- The text of your comment.

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

IBM or any other organizations use the personal information that you supply to contact you only about the issues that you submit.

**If you have a technical problem**
If you have a technical problem or question, do not use the feedback methods that are provided for sending documentation comments. Instead, take one or more of the following actions:
- Go to the IBM Support Portal (support.ibm.com).
- Contact your IBM service representative.
- Call IBM technical support.
This document contains information that was previously presented in z/OS DFSORT Messages, Codes and Diagnosis Guide, SC23-6879-02, which supported z/OS Version 2 Release 2.

The following sections summarize the changes to that information.

The message ICE290A will be issued for the following additional situation:

The E15 or E35 exit was specified with block support, but the required address of the user block list parameter area was not present in the 64-bit invocation parameter list or the block list type is invalid in the user block list parameter area.

Any automated actions based on the presence of this message should be evaluated.

New information

This edition includes the following new enhancements:

Sort/Merge of Unicode data

DFSORT now supports the Sorting and Merging of Unicode data. This support allows users to:

- SORT/MERGE Unicode Data with control field length of 1 to 450 Unicode characters for UTF-8 format data.
- SORT/MERGE Unicode Data with control field length of 1 to 450 Unicode characters for UTF-16 format data.
- SORT/MERGE Unicode Data with control field length of 1 to 450 Unicode characters for UTF-32 format data.

E15/E35 Block Support

E15/E35 exits have been enhanced to support the transfer of block of records between DFSORT and the E15/E35 exits.

Performance Improvements

DFSORT is enhanced to provide performance improvements for several DFSORT functions that generate code at run time. The enhancements will improve performance in both CPU and elapsed times. The design of performance improvements are based on the concept of storing data into the "256 byte area for the Instruction Cache". DFSORT’s design has the ability to write data directly into the instruction cache stream. Modern processors can also access a portion of this cache and may invalidate the instructions stored by DFSORT. In the modern processors like z13, the instruction cache stream is considered to be any location within the same 256-byte block of storage where an instruction resides, regardless of whether that location contains an actual instruction. This means that storing data into the same 256 byte block of an instruction can cause an impact on performance. The following functions will see improvements in both CPU and elapsed times.

- INCLUDE COND Processing
- OMIT COND processing
- SORT statement with more than 5 keys

Message changes

The following changes are made to z/OS Version 2 Release 3 (V2R3).

The following messages are new, changed, or no longer issued for z/OS DFSORT Messages, Codes and Diagnosis Guide in z/OS V2R3.
**New**
The following messages are new.

- ICE292A
- ICE500A
- ICE501A
- ICE502A
- ICE503A
- ICE504A
- ICE505A
- ICE506I

**Changed**
The following messages are changed.

- ICE150I
- ICE189A
- ICE290A

**Deleted**

z/OS DFSORT Messages, Codes and Diagnosis Guide no longer issues the following messages.

None

**New Reserved Words for Symbols**
The following are new DFSORT/ICETOOL reserved words which are no longer allowed as symbols: UTF8, UTF16, UTF32, and COLLKEY. If you used any of these words as a symbol previously, you must change them. For example, if you used UTF8, you can change it to utf8.
Summary of changes for SC23-6879-01 z/OS Version 2
Release 2

This document contains information that was previously presented in z/OS DFSORT Messages, Codes and Diagnosis Guide, SC23-6879-00, which supported z/OS Version 2 Release 1.

The following sections summarize the changes to that information.

ICE000I
ICE099A
ICE121A
ICE288I

The message ICE099A is enhanced to list the failing member name in case of an error. Any automated actions based on the presence of this message should be evaluated.

The text for existing DFSORT messages ICE000I and ICE288I has been changed. Any automated actions based on the presence of these messages should be evaluated.

New information

This edition includes the following new enhancements:

High Performance Ficon (zHPF) Exploitation

A program that invokes DFSORT, ICETOOL or ICEGENER will now be able to exploit the High Performance Ficon (zHPF) hardware. When zHPF is available, DFSORT SORT or MERGE applications will use BSAM for SORTIN, SORTOUT, and OUTFIL data sets.

Date functions

Two new date functions to calculate the week number of a given date and Age as date duration are added.

WEEKNUM function converts a given Julian/Gregorian date to number of week. There are 2 versions of this function, the standard USA format and the ISO format.

WEEKNUM=USA function returns an integer in the range of 1 to 54 that represents the week of the year. The week starts with Sunday, and January 1 is always in the first week.

WEEKNUM=ISO function returns an integer in the range of 1 to 53 that represents the week of the year. The week starts with Monday and includes 7 days. Week 1 is the first week of the year to contain a Thursday, which is equivalent to the first week containing January 4.

AGE function returns a date duration that specifies the number of years, months, and days between an input date and current date.

Age=YMD produces an 8-byte result which has duration in years (0-9999), months (00-12), and days (00-31).

Age=YM produces a 6-byte result which has duration in years (0-9999), months (00-12).

Age=YD produces a 7-byte result which has duration in years (0-9999), days (00-366).

Message changes

The text for existing DFSORT messages has been changed. Any automated actions based on the presence of these messages should be evaluated.

ICE000I
Summary of changes for SC23-6879-00 z/OS Version 2
Release 1

This document contains information that was previously presented in z/OS DFSORT Messages, Codes and Diagnosis Guide, SC26-7525-06, which supported z/OS Version 1.

The following sections summarize the changes to that information.

New information

This edition includes the following new enhancements:

64-bit support

Eligible user programs and exits can now be written to:

- Call DFSORT from a program in 64-bit addressing mode (AMODE 64) using a new 64-bit parameter list and the entry name ICEMAN64 or SORT64
- Use DFSORT E15, E35 and E32 exits running in 64-bit addressing mode (AMODE 64)
- Pass 64-bit addressed records to DFSORT using new 64-bit parameter lists for E15, E32 and E35 exits.

Performance and resource usage improvements

DFSORT has been enhanced to improve use of central storage in relation to system activity. The existing EXPMAX, EXPOLD and EXPRES installation options will now be evaluated in conjunction with available resources at run-time which can give you better control over how DFSORT uses available central storage resources. The likelihood of over committed central storage resources and excessive paging has been reduced, which can provide improved reliability and performance for all workloads including DFSORT.

The shipped default for EXPOLD is changed from MAX to 50%. The shipped default for EXPRES is changed from 0 to 10%.

DFSORT has been enhanced to allocate storage in smaller increments and then check resource availability before each additional increment is reserved. This can allow concurrent sort applications to better share available storage and allows sorts to stop allocating additional storage if resources are no longer available. Additionally, DFSORT’s dynamic allocation of work data sets will be adjusted to reduce the likelihood of failure when a sort application is unable to obtain all of the expected central storage and must then use more disk work space than expected.

A new TUNE installation default allows you to specify whether DFSORT should allocate storage in increments with additional disk work space to minimize the risk of failure, or to allocate all storage at initialization so disk work space allocation can be reduced.

Blockset merge applications can now use storage above 16 MB virtual with more functions (such as E61, INREC, OUTREC, INCLUDE, OMIT and SUM), providing improved performance and virtual storage constraint relief.

XTIOT, uncaptured UCBs and DSAB above 16 megabytes virtual

Work data sets dynamically allocated by DFSORT will use options for XTIOT, uncaptured UCBs, and DSAB above 16 megabyte virtual to the extent that z/OS supports them.

Extended address volumes

DFSORT support for EAS-eligible data set types on Extended Address Volumes has been enhanced to increase the maximum size of a DFSORT work data set. With full track blocking, the maximum number of
tracks that can be used for a single work data set has been increased from 1048576 to 16777215. In situations where DFSORT must use a reduced block size for the work data sets, less than 16777215 tracks can be used.

Alphanumeric Comparison Tests

New UC, LC, MC, UN, LN and MN keywords (similar to the previously available NUM keyword) now allow you to test a field for various combinations of alphanumeric characters or non-alphanumeric characters using binary (BI) format.

The new keywords allow you to select any of the following alphanumeric character sets:

- UC: Uppercase characters (A-Z)
- LC: Lowercase characters (a-z)
- MC: Mixed case characters (A-Z, a-z)
- UN: Uppercase and numeric characters (A-Z, 0-9)
- LN: Lowercase and numeric characters (a-z, 0-9)
- MN: Mixed case and numeric characters (A-Z, a-z, 0-9)

You can use these new keywords in the following comparison operands: COND, INCLUDE, OMIT, BEGIN, END, WHEN and TRLID.

More PARSE Fields

You can now use up to 1000 parsed fields (%0-%999) with the PARSE function; the previous limit was 100 parsed fields (%0-%99).

Repeating PARSE Fields

REPEAT=v is a new PARSE option that can be used to repeat a particular parse field definition multiple times.

REPEAT=v can be used with % to specify v identically defined consecutive parsed fields to be ignored.

REPEAT=v can be used with %n, %nn or %nnn to specify v identically defined consecutive parsed fields for which data is to be extracted. The parsed fields will start with the %n, %nn or %nnn field you select and be incremented by one for each repeated parsed field.

Alphanumeric Tests for PARSE Fields

STARTAFT=an, STARTAT=an, ENDBEFR=an and ENDAT=an can now be used with the PARSE function to start or end when a character from any of various alphanumeric character sets is found. New keywords for an allow you to select any of the following alphanumeric character sets:

- UC: Uppercase characters (A-Z)
- LC: Lowercase characters (a-z)
- MC: Mixed case characters (A-Z, a-z)
- UN: Uppercase and numeric characters (A-Z, 0-9)
- LN: Lowercase and numeric characters (a-z, 0-9)
- MN: Mixed case and numeric characters (A-Z, a-z, 0-9)
- NUM: Numeric characters (0-9)

Insert a String at the End of Variable-Length Records

VLTRAIL=string is a new OUTFIL option that allows you to insert a character string (C'string') or hexadecimal string (X'yy...yy') at the end of each variable-length OUTFIL output record.
Symbol Enhancements

DFSORT Symbols can now be used for many more operands. The following operands of the form KEYWORD=n can now be specified as KEYWORD=symbol where symbol represents an equivalent number (for example, if you have New_Length,25 in SYMNAMES, you can use LENGTH=New_Length wherever you can use LENGTH=25): ABSPOS, ACCEPT, ADDPOS, AVGRLEN, DO, ENDPOS, ENDREC, FIXLEN, ID, IFOUTLEN, INCR, LENGTH, LINES, MAXLEN, RECORDS, REPEAT, SAMPLE, SEQ, SKIPREC, SPLIT1R, SPLITBY, START, STARTPOS, STARTREC, STOPAFT and SUBPOS.

A symbol for a number can now be used for the length of the output field with the CHANGE operand, and for the length of the sequence number with the SEQNUM operand.

A symbol can be used for any of the new %000-%999 parsed fields.

A symbol can be used for string with the new VLTRAIL=string operand.

Improved diagnostics

DFSORT now provides specific reason codes and associated documentation to aid in diagnosing and correcting errors associated with messages ICE083A and ICE141A.

The text for message ICE118I has been changed to recommend the use of FILSZ=En.

The ICE236I options-in-effect message now includes values for TUNE, EXPMAX, EXPOLD and EXPRES.

The text for message ICE285A has been changed for clarification.

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:

Central storage controls

The IBM-supplied default for the existing EXPOLD installation option has been changed to EXPOLD=50%. If you want DFSORT to use EXPOLD, as in previous releases, you can set EXPOLD=MAX.

The IBM-supplied default for the existing EXPRES installation option has been changed to EXPRES=10%. If you want DFSORT to use EXPRES, as in previous releases, you can set EXPRES=0.

The IBM-supplied default for the new TUNE installation option is TUNE=STOR which specifies allocation of available central storage as needed in increments sized to balance resource usage for concurrent sorts. If you want DFSORT to allocate available central storage using fixed sized increments, as in previous releases, you can set TUNE=OLD.

Message changes

The text for existing DFSORT messages ICE000I, ICE083A, ICE118I, ICE141A, ICE236I and ICE285A has been changed. Any automated actions based on the presence of these messages should be evaluated.

New Reserved Words for Symbols

The following are new DFSORT/ICETOOL reserved words which are no longer allowed as symbols: LC, LN, MC, MN, UC and UN.

If you used any of these words as a symbol previously, you must change them. For example, if you used MC, you can change it to mc.
Part 1. Eliminating common sources of error
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 runtime). You can change the installation default value with the NOMSGDD installation option 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 on page 3. 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 installation option ESTAE=NO or run-time option NOESTAE to delete it.

<table>
<thead>
<tr>
<th>If you received</th>
<th>Then use</th>
</tr>
</thead>
<tbody>
<tr>
<td>A user or system abend code</td>
<td>//SYSUDUMP DD SYSOUT=A</td>
</tr>
<tr>
<td>A condition code 16</td>
<td>//SYSUDUMP DD SYSOUT=A//DFSPARM DD PARM=DFSPARM DEBUG ABEND</td>
</tr>
</tbody>
</table>
Table 2: Getting a Dump. (continued)

<table>
<thead>
<tr>
<th>If you received</th>
<th>Then use</th>
</tr>
</thead>
</table>

**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, ARESELL, RESINV, or AREINV 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 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 and z/OS DFSMS Using Data Sets for more details.

Correcting common errors

You can encounter abends or performance problems not caused by failures in the DFSORT 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 11 and z/OS MVS System Codes.

Common system abends related to storage

Table 3 on page 6 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

<table>
<thead>
<tr>
<th>Abend Code</th>
<th>Problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>106</td>
<td>There was not sufficient storage to load a module. Ensure that the values for REGION, MAINSIZE, SIZE, RESALL or RESINV are sufficient.</td>
</tr>
<tr>
<td>804</td>
<td>The amount of storage requested exceeded that available. Ensure that the values for REGION, MAINSIZE, SIZE, RESALL or RESINV are sufficient.</td>
</tr>
<tr>
<td>878</td>
<td>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.</td>
</tr>
</tbody>
</table>
| 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 on page 6 covers other common system abend codes. The items are listed by system abend code.

Table 4: Miscellaneous System Abends

<table>
<thead>
<tr>
<th>Abend Code</th>
<th>Problem</th>
</tr>
</thead>
</table>
| 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). |
Table 4: Miscellaneous System Abends (continued)

<table>
<thead>
<tr>
<th>Abend Code</th>
<th>Problem</th>
</tr>
</thead>
</table>
| 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. |
| 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. |
<table>
<thead>
<tr>
<th>Abend Code</th>
<th>Problem</th>
</tr>
</thead>
</table>
| 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). |
| 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.

Common problems reported by DFSORT users

The situations that follow 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

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 17).
- **Informational** messages, which provide information only and require no programmer action (explained in Chapter 3, “Informational and Error Messages,” on page 17).
- **Diagnostic** messages, which provide additional information useful in debugging and tuning (explained in Chapter 4, “Diagnostic messages,” on page 125).

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 the following information.

- Information on the installation options discussed in this section
- Descriptions of the following messages issued by the ICEOPT started task command: ICEP500I, ICEP501I, ICEP502I, ICEP503I, ICEP504I, ICEP505I, ICEP506I, ICEP507I, ICEP508I, ICEP509I, ICEP510I and ICEP511I
- Descriptions of the messages issued during the expansion of the ICEMAC macro and message modules.

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 installation 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 installation option DIAGSIM=YES is equivalent to specifying a SORTDIAG DD statement for every job.

Note: When an error message (ICExxxA) is issued and the message data set is available, DFSORT will display the diagnostic messages (ICE75xI, ICE8xxI and ICE9xxI) even if they are not requested (by DIAGSIM=YES or a SORTDIAG DD). This can improve first failure data capture and remove the need to rerun the application to display diagnostic messages.

**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.

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 the MSGCON installation option:

ALL
Specifies that all messages except option-in-effect messages (ICE127I through ICE133I, ICE235I and ICE236I) 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 13 and “Printing EFS messages and control statements” on page 14, 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.sssssss text

Where

- Represents
nnn
  Message number.

- Severity code. Values for s are:
  A
    Critical error message; requires programmer action.
  I
    Informational message; usually requires no programmer action.

- Diagnostic trace character. This character is usually not needed, but might be requested by your IBM representative for diagnostic purposes.

jjjjjjji
  Job name.

ssssssss
  Step name.

text
  Message text.

ICE000I - CONTROL STATEMENTS FOR 5650-ZOS, Z/OS DFSORT VvRrr - hh:mm ON day mmm dd, yyyy -

Explanation
Identifies the following for this DFSORT run:

- DFSORT version level (v) 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.

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 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  BLANK NEEDED IN COLUMN 1 OR OPERATION NOT DEFINED CORRECTLY

Explanation
Critical. One of the following errors was found:

- An operation definer, that is, ALTSEQ, DEBUG, END, INCLUDE, INREC, JOINKEYS, JOIN, MERGE, MODS, OMIT, OPTION, OUTFIL, OUTREC, RECORD, REFORMAT, SORT or SUM, started in column 1 instead of column 2. An operation definer must be preceded by at least one blank in column 1.
- A control statement did not contain one of the valid operation definers.
- A control statement did not contain an operation definer followed by a blank on the first line.
- A DFSPARM PARM parameter was continued using the continuation column.
- A continuation line followed a line with an error.

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 DFSFORM PARM options using the continuation column.
• Symbol used where it is not allowed
• Parsed field (%nnn, %nn or %n) 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 that 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:

- A record passed to DFSORT from user exit E15 or E35 contained a length in an RDW less than 4.
- An input data set contained a length in an RDW less than 4.
- An input data set contained a length in a BDW less than 8.
- 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 INVALID POSITION OR LENGTH IN CONTROL FIELD - REASON CODE IS rsn

Explanation
Critical. A position (p) or length (m) in a SORT or MERGE control field is invalid. rsn is the reason code associated with the error. The reason code helps you determine the specific cause of the error. Reason code values (rsn) are as follows:

1. A length (m) was less than the minimum allowed for the specified format (f). For example:
   
   ```
   SORT FIELDS=(11,1,P00,A)  
   * 1 IS LESS THAN THE MINIMUM LENGTH OF 2  
   * FOR A P00 FIELD
   ```

2. A length (m) was greater than the maximum allowed for the specified format (f). For example:
   
   ```
   MERGE FIELDS=(21,33,Fs,D)  
   * 33 IS GREATER THAN THE MAXIMUM LENGTH OF 32  
   * FOR AN FS FIELD
   ```
3. A bytes.bit position (p.x) or length (m.y) was specified but the format (f) was not BI. For example:

```
SORT FIELDS=(7.2,3,CH,A)
* BI MUST BE USED WITH A POSITION OF 7.2
```

4. A position (p) was 0, or a bytes.bit position (p.x) was 0.x. For example:

```
SORT FORMAT=CH,FIELDS=(21,3,D,0,5,A)
* 0 IS LESS THAN THE THE MINIMUM OF 1
* FOR A POSITION
```

5. A bit position (.x) or bit length (.y) was greater than 7. For example:

```
MERGE FIELDS=(11,5.8,BI,A)
* .8 IS GREATER THAN THE MAXIMUM OF .7
* FOR A BIT LENGTH
```

6. A length (m) was 0, or a bytes.bit length (m.y) was 0.0. For example:

```
SORT FIELDS=(11,0,0,BI,D)
* 0.0 IS LESS THAN THE MINIMUM OF 0.1
* FOR A BYTES.BIT LENGTH
```

7. A position (p) plus length (m) was greater than 32753. For example:

```
SORT FIELDS=(32751,3,CH,A)
* 32751 + 3 = 32754 IS GREATER THAN
* THE MAXIMUM OF 32753 FOR POSITION+LENGTH
```

8. VLSHRT was in effect and control fields were too complex.

9. The total length of the control fields was greater than the maximum allowed.

**System action**

The program terminates.

**Programmer response**

Take the action for the indicated reason code value (rsn) as follows:

1. Ensure that each length is valid for its specified format.
2. Ensure that each length is valid for its specified format.
3. Ensure that BI format is used with each bytes.bit position or length.
4. Ensure that each position is equal to or greater than 1.
5. Ensure that each bit position and bit length is between .0 and .7.
6. Ensure that each bytes.bit length is equal to or greater than 0.1.
7. Ensure that the position plus length is less than or equal to 32753.
8. Specify NOVLSHRT or reduce the complexity of the control fields.
9. Reduce the total length of the control fields.

**ICE018A INVALID OR MISSING FIELD FORMAT - REASON CODE IS rsn**

**Explanation**

Critical. A field in a SORT, MERGE, INCLUDE, OMIT, JOINKEYS, SUM, INREC, OUTFIL or OUTFIL statement has an invalid format or no format. rsn is the reason code associated with the error. The reason code helps you determine the specific cause of the error. Reason code values (rsn) are as follows:

1. A SORT, MERGE, SUM, INCLUDE, or OMIT statement has a field with an invalid format. For example:

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

2. An INCLUDE or OMIT operand of an OUTFIL statement or JOINKEYS statement, or a TRLID operand of an OUTFIL statement, or an IFTHEN WHEN, BEGIN or END operand of an INREC, OUTREC, or OUTFIL statement, has a field with an invalid format or has a field without a format. For example:

```
OUTFIL FNAMES=OUT1,
INREC IFTHEN=(WHEN=(21,2,D2,EQ,C'AB'),
OVERLAY=(21:C'OK'))
* FORMATS ARE MISSING IN WHEN CONDITION
```

3. A SORT, MERGE, SUM, INCLUDE, or OMIT statement without a FORMAT=f operand, or with an invalid FORMAT=f operand, has 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
```

4. D2 format was specified in the INCLUDE or OMIT operand of an OUTFIL statement or JOINKEYS statement, or in the TRLID operand of an OUTFIL statement, or in the IFTHEN WHEN, BEGIN or END operand of an INREC, OUTREC, or OUTFIL statement. For example:

```
INREC IFTHEN=((WHEN=(21,2,D2,EQ,C'AB'),
OVERLAY=(21:C'OK'))
```

```
INFO:

D2 IS INVALID FOR WHEN
OUTFIL FNames=OUT1,
   INCLUDE=(5,2,D2,EQ,'CAB')
D2 IS INVALID FOR OUTFIL INCLUDE

5. More than 112 SORT or MERGE fields were specified, and Blockset was not selected.
6. CSF, FS, UFF, SFF, Y2x or PD0 format was specified for SORT, MERGE or SUM and Blockset was not selected. For example:

   OPTION NOBLKSET
   SORT FORMAT=PD0,FIELDS=(5,3,ZD,A)
   * PD0 AND NOBLKSET SPECIFIED

**System action**
The program terminates.

**Programmer response**
Take the action for the indicated reason code value (rsn) as follows:

1. Ensure that the format for each field in the SORT, MERGE, SUM, INCLUDE, and OMIT statements is valid.
2. Specify p,m,f for each field in the INCLUDE, OMIT, TRLID, WHEN, BEGIN and END operands, and ensure that the format for each field is valid.
3. Ensure that each field in the SORT, MERGE, SUM, INCLUDE and OMIT statements has a format by specifying p,m,f or FORMAT=f with p,m, as appropriate.
4. Use the D2 format in an INCLUDE or OMIT statement rather than in an INCLUDE or OMIT operand, or replace the D2 format with a valid format in the INCLUDE, OMIT, TRLID, WHEN, BEGIN or END operand.
5. Rerun the job with a SORTDIAG DD statement to get message ICE8001, which indicates the reason Blockset could not be used. If possible, remove the condition preventing the use of Blockset. Alternatively, reduce the number of SORT or MERGE fields to 112 or less.
6. Rerun the job with a SORTDIAG DD statement to get message ICE8001, which indicates the reason Blockset could not be used. If possible, remove the condition preventing the use of Blockset. Alternatively, do not use a CSF, FS, UFF, SFF, Y2x or PD0 format.

**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 or JOINKEYS control statement contained an invalid value.

System action
The program terminates.

Programmer response
Check that the TYPE operand value on the RECORD statement or JOINKEYS 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 SORTINnnn DD statement
- A temporary empty input VSAM data set defined without a RECSZ value

ICE024A  MAXIMUM RECORD LENGTH EXCEEDED - REASON CODE IS rsn

Explanation
Critical. A specified or calculated record length was greater than the maximum length DFSORT could process. The record length can be one of the following:
- Input record length: Can be obtained from the L1 or L2 value of the RECORD LENGTH operand, from the RECORDSIZE value for a VSAM input data set, or from the LRECL of a non-VSAM input data set.
- Reformatted record length: Can be obtained from the INREC statement or OUTREC statement.
- Output record length: Can be obtained from the L3 value of the RECORD LENGTH operand, from the RECORDSIZE value for a VSAM output data set, or from the LRECL of a non-VSAM output data set.

Note that for VSAM data sets processed as variable-length, the record length is increased by 4 bytes since DFSORT prepends an RDW to each record.

rsn is the reason code associated with the error. The reason code helps you determine the specific cause of the error. Reason code values (rsn) are as follows:
1. An input record length was greater than 32767.
2. An input record length for fixed-length records was greater than 32760.
3. An input record length for variable-length unspanned records was greater than 32756.
4. An output record length was greater than 32767.
5. An output record length for fixed-length records was greater than 32760.
6. An output record length for variable-length unspanned records was greater than 32756.
7. A reformatted record length was greater than 32767.
8. A reformatted record length for fixed-length records was greater than 32760.
9. A reformatted record length for variable-length unspanned records was greater than 32756.
10. Blockset was not selected and the specified or calculated minimum record length (L4) was greater than the specified or calculated maximum record length (L2).
11. Blockset was not selected and the specified or calculated minimum record length (L4) was greater than the specified or calculated average record length (L5).
12. Blockset was not selected and the specified or calculated average record length (L5) was greater than the specified or calculated maximum record length (L2).

System action
The program terminates.

Programmer response
Ensure that specified and calculated record lengths are not greater than the maximum length DFSORT can process. 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.

**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.
• N64 was specified for the fourth parameter, but DFSORT was not invoked using a 64-bit invocation parameter list.

**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.

**ICE036I**  
**B = a**

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

**ICE037I**  
**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. 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 nK 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).

For a JOINKEYS application, you may need to use REGION=0M.

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**

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 - reason code is rsn**

**Explanation**

Critical. An error was detected for the data set associated with the indicated ddname. The following information is displayed in the message:

- ddname: The input, output or work ddname (for example, SORTIN, SORTOUT or SORTWK01).
- rsn: The reason code associated with the error. The reason code (listed later in this section) helps you determine the specific cause of the error.

**Note:** The ddname and reason code are not displayed for a tape work data set sort or conventional merge application. In this case, the message will appear as:

**ICE042A UNIT ASSIGNMENT ERROR**

These techniques should be avoided by not using tape work data sets for a sort or by removing the condition preventing the use of Blockset indicated by the reason code displayed in message ICE800I when a SORTDIAG DD statement is specified.

**Reason code values (rsn) are as follows:**

1. VSAM and non-VSAM input data sets were specified for a merge application. Example: SORTIN01 is a VSAM data set and SORTIN02 is a non-VSAM data set. SORTIN01 and SORTIN02 also are non-VSAM data sets and SORTIN03 is a VSAM data set.
2. Concatenated VSAM and non-VSAM input data sets were specified. Example: SORTIN has a non-VSAM data set followed by a VSAM data set. SORTIN has two VSAM data sets followed by a non-VSAM data set.
3. A DUMMY, NULLFILE, spool, pipe, or HFS file, an HFS file or an invalid device type was specified for a work data set, or both tape and disk work data sets were specified. Example: SORTWK01 is an HFS data set.
4. Blockset was not selected and a pipe data set was specified for input or output. Example: SORTIN is a pipe data set but Blockset could not be used.
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. Example: ICEGENER with //SYSIN DD *.

6. The QNAME parameter was specified on an output DD statement. Example: SORTOUT with QNAME=NAME1.

7. An HFS data set (not an HFS file) was specified for input or output. Example: SORTIN is an HFS data set.

**System action**

The program terminates.

**Programmer response**

For the data set associated with the indicated ddname, take the action for the indicated reason code value (rsn) as follows:

1. Use only VSAM data sets, or only non-VSAM data sets, but not both.
2. Use only a single VSAM data set, or concatenated non-VSAM data sets.
3. Remove the work data sets or use only valid work data sets.
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, do not use a pipe data set for input or output.
5. If you do not need to use any IEBGENER control statements, use DUMMY for SYSIN (or the alternate ddname for SYSIN). If you do need to use IEBGENER control statements, no action is necessary.
6. Do not use the QNAME parameter on the output DD statement.
7. Do not use an HFS data set for input or output; you can use an HFS file for input or output.

**ICE043A INVALID DATA SET ATTRIBUTES**

<table>
<thead>
<tr>
<th>ddname attribute</th>
<th>REASON CODE IS rsn</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Explanation</strong></td>
<td>Critical. An error associated with a record format, record length or block size was detected, or a conflict between these attributes was detected. The following information is displayed in the message:</td>
</tr>
</tbody>
</table>

- **ddname**: The input or output ddname for the associated data set (for example, SORTIN for a single or concatenated input data set used for a copy or sort application, SORTIN01 or SORTIN02 for an input data set used for a merge application, SORTOUT for an output data set, OUT1 for an OUTFIL output data set, and so on).
- **attribute=RECFM**: The error is associated with a record format. The record format can be set or derived from a RECFM value or a RECORD TYPE value, or defaulted based on other attributes. The record format can be fixed length (for example, RECFM=F, RECFM=FB or TYPE=F) or variable length (for example, RECFM=V, RECFM=VB, RECFM=VS, RECFM=VBS or TYPE=V).
- **attribute=LRECL**: The error is associated with a record length. The record length can be set or derived from an LRECL value, a RECORD LENGTH value, a VSAM RECORDSIZE value or the reformatted record length, or defaulted based on other attributes.
- **attribute=BLKSIZE**: The error is associated with a block size. The block size can be set or derived from a BLKSIZE value, or defaulted based on other attributes.
- **rsn**: The reason code associated with the error. The reason code (listed later in this section) helps you determine the specific cause of the error.

**Note**: The ddname, attribute, and reason code are not displayed for a tape work data set sort or conventional merge application. In this case, the message will appear as:

**ICE043A INVALID DATA SET ATTRIBUTES**

These techniques should be avoided by not using tape work data sets for a sort or by removing the condition preventing the use of Blockset indicated by the reason code displayed in message ICE800I when a SORTDIAG DD statement is specified.

**Reason code values (rsn) are as follows:**

1. A fixed length output data set has a block size smaller than the record length, or a variable length output data set has a block size smaller than the record length plus four. Example: The record length is 1000 and BLKSIZE=800 is specified on the SORTOUT DD statement.
2. The record length cannot be determined, LRECL=X was specified, or the record length is larger than the maximum allowed. Example: The SORTIN data set was created with LRECL=X.
3. Input data sets have mixed fixed length and variable length record formats, or mixed valid and invalid record formats. Examples: Concatenated SORTIN data sets have RECFM=FB and RECFM=V. The SORTIN01 data set has RECFM=V and the SORTIN02 data set has...
RECFM=FB. Concatenated SORTIN data sets have RECFM=VB and RECFM=U.

4. A variable length spanned input data set was found after a variable length unspanned input data set. Example: Concatenated SORTIN tape data sets have RECFM=VB and RECFM=VBS.

5. Fixed length input data sets have different record lengths or ICETPEX returned a record length that does not match the record length in the tape label. Examples: Concatenated SORTIN tape data sets have RECFM=FB with LRECL=100 and RECFM=FB with LRECL=80. The SORTIN01 data set has RECFM=FB with LRECL=100 and the SORTIN02 data set has RECFM=FB with LRECL=135. ICETPEX returned 255 as the record length, but the tape label has 360 as the record length.

6. The record format is fixed length and the block size is smaller than the record length or is not a multiple of the record length. Example: The reformatted fixed record length is 1200 from OUTREC BUILD=(11,1200) and the SORTOUT data set has BLKSIZE=25000.

7. An input tape data set with the largest block size was not the first data set, and the block size for that data set was not available in the DD statement, from DFMSrmm, or from a tape management system using ICETPEX. Example: Concatenated SORTIN tape data sets are unmanaged: the first data set was created with RECFM=FB, LRECL=80 and BLKSIZE=32720 and the second data set was created with RECFM=FB, LRECL=80 and BLKSIZE=262080.

8. Input and output data sets have mixed fixed length and variable length record formats, or mixed valid and invalid record formats. Examples: The SORTIN data set has RECFM=FB and the SORTOUT data set has RECFM=V. The SORTIN01 data set has RECFM=V and the SORTOUT data set has RECFM=F or RECFM=U.

9. The record format is fixed length and the output record length is larger than the input record length or reformatted record length. Examples: The SORTIN01 data set has RECFM=FB and LRECL=80 and the SORTOUT data set has LRECL=100. The SORTIN01 data set has RECFM=FB and LRECL=80, the reformatted record length is 60 from OUTREC BUILD=(1,60) and the SORTOUT data set has LRECL=80.

10. The record format cannot be determined or is not valid for DFSORT processing. Examples: The SORTIN data set has DUMMY or NULLFILE but does not have RECFM=value. The SORTOUT data set has RECFM=U.

11. The track capacity of the device for the output data set was smaller than the block size for the output data set. Example: The SORTOUT data set is allocated on a 3350 and has BLKSIZE=32000.

12. The block size cannot be determined or is not valid for DFSORT processing. Examples: The SORTIN DD statement has DUMMY or NULLFILE but does not have BLKSIZE=value. The block size of the first SORTIN data set cannot be determined.

13. The record format is variable length and the OUTFIL output data set has a fixed length record format, or OUTFIL parameter FTOV or VLFILL is specified and the OUTFIL output data set has a fixed length record format. Examples: SORTIN has RECFM=V, OUTFIL FNAMES=OUT1 is specified, and OUT1 has RECFM=FB. SORTIN has RECFM=FB, OUTFIL FNAMES=OUT2,FTOV is specified and OUT2 has RECFM=F. SORTIN has RECFM=V, OUTFIL FNAMES=OUT3, VLFILL=C'*',BUILD=(70) is specified and OUT3 has RECFM=FB.

14. The record format is fixed length and the OUTFIL output data set has a variable length record format, or OUTFIL parameter VTOF or CONVERT is specified and the OUTFIL output data set has a variable length record format. Examples: SORTIN has RECFM=FB, OUTFIL FNAMES=OUT1 is specified, and OUT1 has RECFM=VBS. SORTIN has RECFM=V, OUTFIL FNAMES=OUT2,VTOF,BUILD=(5,60) is specified and OUT2 has RECFM=VB.

15. The input data set with the largest block size was not the first input data set. Example: SORTIN01 has BLKSIZE=15000 and SORTIN02 has BLKSIZE=20000 and Blockset could not be used.

16. A data set with RECFM=F has a block size that is not equal to the record length, or the first input data set has RECFM=F and is followed by a data set with a different block size. Example: SORTOUT has RECFM=F, LRECL=80 and BLKSIZE=160.

System action
The program terminates.

Programmer response
For the data set associated with the indicated ddname, and the indicated attribute, take the action for the indicated reason code value (rsn) as follows:

1. Let DFSORT set the output data set block size, specify BLKSIZE=value with a value that is valid for the record length, or use an existing data set with a block size that is valid for the record length, as appropriate.

2. Do not use a data set with LRECL=X. Ensure that the record length is available, and is not larger than 32767 or the maximum allowed for the
record format. Specify LRECL=value on the DD statement, if appropriate, and especially for a DUMMY or NULLFILE data set.

3. Ensure that all of the input data sets have fixed length record formats (for example, RECFM=FB), or that all of the input data sets have variable length record formats (for example, RECFM=VB and RECFM=VBS). Do not use a data set with RECFM=U.

4. Ensure that a variable length spanned input data set (for example RECFM=VBS) is not preceded by a variable length unspanned input data set (for example, RECFM=VB).

5. Ensure that all of the fixed length input data sets have the same record length. If appropriate, use fixed input data sets padded to the largest record length. Ensure that ICETPEX returns the correct record length.

6. Let DFSORT set the output data set block size, specify BLKSIZE=value with a value equal to or larger than the record length and a multiple of the record length, or use an existing data set with a block size that is valid for the record length, as appropriate.

7. Use managed tapes, or specify BLKSIZE=value, with a value equal to the largest block size, in the first input DD statement.

8. Ensure that all of the input data and output data sets have fixed length record formats (for example, RECFM=FB), or that all of the input and output data sets have variable length record formats (for example, RECFM=VB and RECFM=VBS). If you want to convert from fixed length input records to variable length output records, use an OUTFIL statement with FTOV. If you want to convert from variable length input records to fixed length output records, use VTOF and BUILD.

9. Ensure that the SOLRF parameter is in effect. Use INREC or OUTREC to pad the output records to the needed length. Let DFSORT set the output record length, specify LRECL=value with a value equal to the reformatted record length, or use an existing data set with a record length equal to the reformatted record length, as appropriate.

10. Ensure that the record format is available and is valid for DFSORT processing (for example, RECFM=F, RECFM=FB, RECFM=VB, RECFM=VBS). Do not use RECFM=U. Specify RECFM=value on the DD statement, if appropriate, and especially for a DUMMY or NULLFILE data set. Specify RECORD TYPE=value, if appropriate, and especially for a VSAM input data set.

11. Ensure that the output data set block size is equal to or smaller than the track capacity for the device on which the output data set resides.

12. Ensure that the block size is available and is valid for DFSORT processing. Specify BLKSIZE=value on the DD statement, if appropriate, and especially for a DUMMY or NULLFILE data set.

13. If you want to use FTOV or VLFILL, ensure that the OUTFIL output data set has a variable length record format or no record format. If you want to convert from variable length input records to fixed length output records, use VTOF and BUILD and ensure that the OUTFIL output data set has a fixed length record format or no record format.

14. If you want to use VTOF or CONVERT, ensure that the OUTFIL output data set has a fixed length record format or no record format. If you want to convert from from fixed length input records to variable length output records, use FTOV and ensure that the OUTFIL output data set has a variable length record format or no record format.

15. 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, specify BLKSIZE=value, with a value equal to the largest block size, in the first input DD statement.

16. Ensure that data sets with RECFM=F have a block size equal to the record length, or specify RECFM=FB in the first input DD statement.

ICE044I EXIT Enn 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 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 159 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 dynamic allocation was used, refer to message ICE254I (if issued) for more information.
- If JCL work data sets were used, increase the amount of work space available to DFSORT.
- If message ICE118I was received, refer to message ICE253I for more information.
- If you are sorting variable length records, refer to message ICE098I for more information.
- 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 as this can degrade performance and increase intermediate storage requirements. In some cases, a small amount of main storage can force DFSORT to perform an intermediate merge as indicated by message ICE247I; refer to ICE247I for more information on the consequences of an intermediate merge and how to avoid it.
- 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.

**ICE047A**  
RECORD COUNT OFF, SPECIFIED n, RECEIVED n1

**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.
**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.

**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: \( n^2 \)

**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. For a JOINKEYS subtask, the OUT field is zero; the number of joined records is displayed in message ICE421I for the main task. 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 message.

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
- Merge applications without a specified OPTION MERGEIN ddname or with an OPTION MERGEIN ddname also used for SORTOUT.
- 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 or a missing OPTION SORTIN or OPTION SORTOUT ddname.
- CKPT in effect with no work data set specified
- Conventional merge applications without a SORTINO1 DD statement (Conventional merge ignores SORTINO1) or a SORTOUT DD statement.
- A DD statement for a specified or defaulted OUTFIL ddname was not present.
- A DD statement for a specified JOINKEYS F1 or F2 ddname was not present. The F1 ddname is SORTJNF1 (or ccccJNF1 if SORTDD=cccc is in effect) if FILE=F1 or FILES=F1 is specified. The F2 ddname is SORTJNF2 (or ccccJNF2 if SORTDD=cccc is in effect) if FILE=F2 or FILES=F2 is specified.

**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**  DELE T 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.

**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**

- 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.

**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 SORTINnn

**Explanation**

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

If Blockset was selected:

- If input was not supplied through exit E32, ddname is the ddname associated with the data set which was found to be out of sequence.
• If input was supplied through user exit E32, 
  SORTINnn identifies the file which was found to be 
  out of sequence. 00 signifies the first input file, 01 
  the second, and so on.

If Blockset was not selected:
• If input was not supplied through exit E32, 
  SORTINnn is the ddname associated with the data 
  set which was found to be out of sequence.
• If input was supplied through exit E32, SORTINnn 
  identifies the file which was found to be out of 
  sequence. 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 xxxxxxx 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 Enn [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 161.

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, installation 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.
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 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.

ICE076A   VSAM INPUT 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 GET 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.

ICE077A   VSAM OUTPUT ERROR i (nnn)

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.

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 (derr) (srsn)
Explanation
Critical. Dynamic allocation of sort work data sets could not be attempted or failed. derr indicates the error code associated with the dynamic allocation failure as set by DFSORT, or as returned in field S99ERROR by the system DYNALLOC service used by DFSORT.

One of the following derr values is set by DFSORT when dynamic allocation could not be attempted:
• I64K: The work space needed for each work data set exceeded 64K tracks for a single, physical data set, but DFSORT was unable to allocate the work data sets as large format.
• I999: The work space needed for each work data set exceeded an internal DFSORT limit.
• IDEV: Conflicting device types were received for the work data sets.

If the derr value does not start with I, DFSORT attempted to dynamically allocate work data sets using the DYNALLOC/DYNALLOC and DYNNSPC options in effect, but the attempt failed. derr is the error code returned by dynamic allocation.

**System action**
The program terminates.

**Programmer response**
- If derr is I64K:
  Use the DYNALLOC=(,n) parameter to increase the maximum number (n) of work data sets such that the work space needed for each work data set does not exceed 64K tracks. Alternatively, if the VIO=YES installation option is in effect, setting the VIO=NO installation option may allow DFSORT to allocate the work data sets as large format.
- If derr is I999:
  Use the DYNALLOC=(,n) parameter to increase the maximum number (n) of work data sets; this will reduce the work space needed for each work data set.
- If derr is IDEV:
  Ensure that each work data set has the same device type.
- If derr is 97xx:
  An SMS failure occurred. 97xx is the S99ERROR code (in hexadecimal) associated with the failure. If srsn is displayed in the message, it is the S99ERSN SMS reason code (in hexadecimal) associated with the failure. See z/OS MVS Programming: Authorized Assembler Services Guide for an explanation of the S99ERROR code (derr) and S99ERSN SMS reason code (srsn). Take the appropriate indicated action to correct the error.
- If derr is none of the previous values:
  derr is the S99ERROR code (in hexadecimal) associated with the failure. See z/OS MVS Programming: Authorized Assembler Services Guide for an explanation of the S99ERROR code (derr). Take the appropriate indicated action to correct the error.

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 DYNNSPC=n to decrease the primary space allocated.

For a JOINKEYS operation, you can supply any needed parameters as follows:
- In DFSPARM for the JOINKEYS main task.
- In JNF1CNTL for the JOINKEYS F1 subtask.
- In JNF2CNTL for the JOINKEYS F2 subtask.

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 nK 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  

**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  

**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 output records or V for EBCDIC or ASCII variable-length output records.

**System action**  
None.

**Programmer response**  
None.

ICE090I  

**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  

**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.
**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 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.

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 - PROCESSED: p, EXPECTED: e

Explanation

Indicates the processed (p) and expected (e) average record length in bytes for the variable-length records processed by sort, that is, the number of bytes in the variable-length records (including the record descriptor word) divided by the number of records. The number of records includes all records received, added and not deleted during DFSORT input processing.

p is based on the number of bytes and records actually processed. e is based on DFSORT’s determination of the average record length it expected taking into account various factors such as the AVGRLEN value, the estimated input file size, the FILSZ value, and so on.

If DFSORT terminated before completing the input phase, p is based on the number of bytes and records processed up to the point of failure.

System action

None.

Programmer response

Take appropriate action depending on whether or not DFSORT issued message ICE046A:

- ICE046A: If p is greater than e, specify AVGRLEN=n, where n is greater than or equal to p and is a reasonably accurate estimate of the average record length.
- No ICE046A: If p is significantly different from e, specifying AVGRLEN=n with a reasonably accurate estimate of the average record length can prevent DFSORT from overallocating or underallocating dynamic work space. p was the actual average record length for this run. If future runs will process approximately the same amount of data and number of records, consider specifying AVGRLEN=p.

See z/OS DFSORT Application Programming Guide for additional information about parameters you can use to provide accurate information to DFSORT that can prevent overallocation or underallocation of dynamic work space (FILSZ, AVGRLEN, DYNSPC, and so on). Also, see the following messages, if issued: ICE253I, ICE254I, and ICE258I.

For a JOINKEYS operation, you can supply any needed parameters as follows:

- In DFSPARM for the JOINKEYS main task.
- In JNF1CNTL for the JOINKEYS F1 subtask.
- In JNF2CNTL for the JOINKEYS F2 subtask.

**ICE099A**  BDL FAILED FOR ddnamedata SET, MEMBER member

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 BDL 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  xxxxxxxxxxxx STATEMENT NOT SUPPORTED FOR TECHNIQUE USED

Explanation
Critical. The control statement indicated by xxxxxxxxxxxx is not supported for the technique used (tape work data set sort or Conventional merge). xxxxxxxxxxxx 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:

- An operand, other than IFTHEN, was specified twice. Example:

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

- PARSE, FIELDS, BUILD, OVERLAY or FINDREP was specified with IFTHEN or IFOUTLEN. Example:

  OUTREC FINDREP=(IN=C'ONE',OUT=C'TWO'),
  IFTHEN=(WHEN=INIT,OVERLAY=(25:C'YES'))

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

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

- 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, OVERLAY, or FINDREP. Example:

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

- For an IFTHEN clause, WHEN=GROUP was specified without BEGIN, KEYBEGIN, END or RECORDS, or without PUSH. Example:

  INREC IFTHEN=(WHEN=GROUP,BEGIN=(9,2,CH,EQ,C 'NO'))

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

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

- For an IFTHEN clause, WHEN=INIT, WHEN=(logexp), WHEN=ANY, or WHEN=NONE was specified with

  BEGIN, KEYBEGIN, END, RECORDS or PUSH. Example:

  OUTREC IFTHEN=(WHEN=INIT,PUSH=(9:5,8))

- For an IFTHEN clause, WHEN=GROUP was specified with PARSE, BUILD, OVERLAY or FINDREP. Example:

  OUTREC IFTHEN=(WHEN=GROUP,BUILD=(9:5,8))

- 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=GROUP 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,OVERLAY=(25:C'YES'))

- 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,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  INVALID POSITION OR LENGTH IN SUM FIELD - REASON CODE IS rsn

Explanation
Critical. A position (p) or length (m) in a SUM field is invalid. rsn is the reason code associated with the error. The reason code helps you determine the specific cause of the error. Reason code values (rsn) are as follows:

1. The length (m) was invalid for the specified format (f). For example:
   ```plaintext
   SUM FIELDS=(21,3,FI)
   * A LENGTH OF 3 IS NOT ALLOWED FOR AN FI FIELD.
   * THE LENGTH MUST BE 2, 4 OR 8.
   ```

2. A position (p) plus length (m) was greater than 32753. For example:
   ```plaintext
   SUM FIELDS=(32751,3,ZD)
   * 32751 + 3 = 32754 IS GREATER THAN THE MAXIMUM OF 32753 FOR POSITION+LENGTH
   ```

3. A position (p) was 0 or a length (m) was 0. For example:
   ```plaintext
   SUM FORMAT=PD,FIELDS=(0,3)
   * 0 IS LESS THAN THE MINIMUM OF 1 FOR A POSITION
   ```

4. A position (p) plus length (m) was greater than 4093 and Blockset was not selected. For example:
   ```plaintext
   OPTION NOBLKSET
   SUM FIELDS=(4092,2,FI)
   * 4092+2 = 4094 IS GREATER THAN THE MAXIMUM
   * OF 4093 FOR POSITION+LENGTH WITH NOBLKSET
   ```

System action
The program terminates.

Programmer response
Take the action for the indicated reason code value (rsn) as follows:

1. Ensure that each length is valid for its specified format. 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.

2. Ensure that the position plus length is less than or equal to 32753.

3. Ensure that each position and each length is equal to or greater than 1.

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, ensure that the position plus length is less than or equal to 4093.

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 as follows (note that %nn is used to represent %nnn, %nn or %n):

- 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.

• DT=(abcd) or DTNS=(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.

• The length (m for p,m or FIXLEN=m for %nn) for a HEX or TRAN=HEX field was greater than 16376.

• The length (m for p,m or FIXLEN=m for %nn) for a TRAN=BIT field was greater than 4094 bytes.

• 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 a Y4 format field was not 7-8 for Y4T or Y4W, 4 for Y4U or Y4X, or 5 for Y4Y.

• 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 256.

• 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.

• The length for a KEYBEGIN field (m for p,m) 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.
INFO-343D 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 JOINKEYS statement, or the TRLID operand of an OUTFIL statement, or the IFTHEN WHEN, BEGIN or END 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 JOINKEYS statement, or the TRLID operand of an OUTFIL statement, or the IFTHEN WHEN, BEGIN or END 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, or used a field with UC, LC, MC, UN, LN or MN that was not BI.

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

**ICE118I**
UNKNOWN FILE SIZE - RECOMMEND SPECIFYING FILSZ=EN TO REDUCE RISK OF FAILURE OR DEGRADED 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.
For a JOINKEYS operation, you can supply any needed parameters as follows:

- In DFSPARM for the JOINKEYS main task.
- In JNF1CNTL for the JOINKEYS F1 subtask.
- In JNF2CNTL for the JOINKEYS F2 subtask.

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 Enn IGNORED

Explanation
User exit E14 or E2n was specified in the MODS statement, as identified in the message by Enn. 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. EQUALS in effect - 4294967295 records
2. VLSHRT in effect - 2147483647 records
3. Blockset technique not selected - 2147483647 records

System action
The program terminates.

Programmer response
For cases 1 and 2, 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 3, 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
Specify a numeric value for the indicated keyword that is less than the processing limit.

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

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 REFORMATTING FOR {*INREC|*OUTREC|ddname}: REASON CODE rsn, IFTHEN n**

**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 and variable-length output records.

The following information is displayed in the message:
- *INREC if the error is associated with an INREC statement. *OUTREC if the error is associated with an OUTREC statement. ddname if the error is associated with an OUTFIL statement.
- rsn: The reason code associated with the error. The reason code (listed later in this section) helps you determine the specific cause of the error.
- n: 0 if the error is not associated with an IFTHEN clause. Otherwise, n identifies the number of the IFTHEN clause associated with the error (starting at 1 for the first IFTHEN clause).

The statement and operand associated with the error are 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).

Reason code values (rsn) are as follows:

1. An IFOUTLEN=n operand was specified without an IFTHEN operand.
   
   Example:
   ```
   OUTREC IFOUTLEN=50
   ```

2. The last position was specified without a corresponding length for fixed-length input records.
   
   Examples (fixed-length input):
   ```
   INREC BUILD=(1,5,6X,8)  
   or
   OUTFIL FNAMES=O1,  
   OUTREC=(1,28,28),FTOV
   ```

3. 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):

Examples (variable-length input):

```plaintext
INREC FIELDS=(1,20,31)
OUTREC FIELDS=(1,25)
or
OUTREC BUILD=(1,20,31,10)
OUTFIL BUILD=(1,25,27)
```

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

Example (fixed-length input):

```plaintext
INREC IFTHEN=(WHEN=INIT,
BUILD=(81:SEQNUM,8,ZD)),
IFTHEN=(WHEN=(81,8,ZD,EQ,+1),
BUILD=(3,20,20:C'ABC'))
```

5. A VTOF or CONVERT operand was specified without an OUTREC or BUILD operand for variable-length input records in an OUTFIL statement.

Example (variable-length input):

```plaintext
OUTFIL VTOF
```

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

6. 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):

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

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

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

Example (fixed-length input):

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

**System action**

The program terminates.

**Programmer response**

Take the action for the indicated reason code value (rsn) as follows:

1. Specify an IFTHEN operand with an IOUTLEN=n operand.

Example:

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

2. Specify the last position with a corresponding length.

Example (fixed-length input):

```plaintext
INREC BUILD=(1,5,6X,8,25)
OUTFIL FNAMES=O1,
OUTREC=(1,20,28,40),FT0V
```

3. Specify the last position without a corresponding length for the INREC, OUTREC and OUTFIL statements, or specify the last position with a corresponding length for the INREC, OUTREC, and OUTFIL statements, as appropriate.

Examples (variable-length input):

```plaintext
INREC FIELDS=(1,20,31)
OUTREC FIELDS=(1,20,5X,21)
or
OUTREC BUILD=(1,20,31,10)
OUTFIL BUILD=(1,20,5X,21,10)
```

4. Ensure that columns do not overlap previous fields in the reformatted record.

Example (fixed-length input):

```plaintext
INREC IFTHEN=(WHEN=INIT,
BUILD=(81:SEQNUM,8,ZD)),
IFTHEN=(WHEN=(81,8,ZD,EQ,+1),
BUILD=(3,20,25:C'ABC'))
```

5. Specify an OUTREC or BUILD operand with the VTOF or CONVERT operand.

Example (variable-length input):

```plaintext
OUTFIL VTOF,BUILD=(5,60)
```

6. Specify an OUTREC or BUILD operand with the VLFILL=C'x' or VLFILL=X'yy' operand.

Example (variable-length input):

```plaintext
OUTFIL VLFILL=C'*',BUILD=(1,4,8,45)
```

7. Specify an input or separation field.

Example (fixed-length input):

```plaintext
OUTFIL OUTREC=(//,X)
```

**ICE127I**

**OPTIONS:**

```
OVFL0=s,PAD=s,TRUNC=s,
SPANINC=s,VLSCMP=x,SZERO=x,
RESET=x,VSAMEMT=x,DYNSPC=t
```
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 for ICE127I through ICE133I are as follows:

- **x** ‘Y’ for YES; ‘N’ for NO
- **n** A decimal value representing bytes
- **a** ‘RC16’ or ‘ABEND’
- **b** ‘SYSOUT’, or the alternate ddname for the message data set
- **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 for the number of data sets
- **e** ‘SHORT’, ‘FULL’, or ‘NO’
- **f** ‘COB1’ or ‘COB2’
- **g** A decimal value representing the SVC number, or ‘109A’ for SVC 109 using the alternate SVC module
- **h** EFS program name, or ‘NONE’
- **j** ‘MSG’, or a decimal value representing the ABCODE number

Messages ICE127I through ICE133I, ICE235I and ICE236I indicate the options in effect for a Blockset sort, copy, or merge application or a Peerage/Vale sort application.
k
'S' for STRONG; 'W' for WEAK

p
A decimal value representing megabytes, or 'OPTIMAL'

q
'DFSPARM', or the alternate ddname for the DFSPARM data set

r
A decimal value representing megabytes, or 'MAX'

s
'RC0', 'RC4', or 'RC16'

t
A decimal value representing megabytes

System action
None.

Programmer response
None.

Note:
1. Do not expect the values in ICE127I through ICE133I, ICE235I and ICE236I 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, ICE133I and ICE236I 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.


Explanation
Critical. The ddname specified for OPTION statement operand SORTIN or SORTOUT in DFSPARM or in the extended parameter list was ccccWKdd, ccccWKdd, ccccDKd, or ccccDKdd. 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:

```plaintext
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:

```plaintext
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
None.
Programmer response
None.

ICE141A SPANNED RECORD ON ddname COULD NOT BE ASSEMBLED – REASON CODE IS rsn

Explanation
Critical. A spanned record on the indicated data set could not be properly assembled.

rsn is the reason code associated with the error. The reason code helps you determine the specific cause of the error. Reason code values (rsn) are as follows:
1. A segment length was greater than the LRECL.
2. A segment length was less than 4 bytes.
3. The total length of segments was greater than the LRECL.
4. 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
Take the action for the indicated reason code (rsn) as follows:
1. Ensure all records in the indicated data set have segment lengths that are less than the LRECL.
2. Ensure all records in the indicated data set have segment lengths that are at least 4 bytes.
3. Ensure all records in the indicated data set have segment lengths that total less than the LRECL.
4. Ensure all records in the indicated VBS data set are blocked correctly.

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
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.

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.

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 installation defaults 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 xxxxCNTL - 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.

---

**ICE147A ddname CONCATENATION 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 OPTION STATEMENT OPERAND SORTIN OR SORTOUT 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 IFAPRDxx 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 IFAPRDxx member of parmlib. See z/OS Planning for Installation for information about adding the appropriate entry to IFAPRDxx.
After you have added the appropriate entry to IFAPRDxx, 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.
- Unicode data formats (UTF8/UTF16/UTF32) 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, TRLID, or IFTHEN WHEN, BEGIN or END 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, or the INCLUDE operand of a JOINKEYS statement, caused the error. n is 0.
- *OMIT indicates that the COND operand of the OMIT statement, or the OMIT operand of a JOINKEYS statement, caused the error. n is 0.
- nnname and n=0 indicates that the INCLUDE, OMIT or TRLID operand of an OUTFIL statement caused the error. nname identifies the first data set in the associated OUTFIL group.
- *INREC indicates that an IFTHEN WHEN, BEGIN or END 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, BEGIN or END 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).
- \( \text{ddname and } n > 0 \) indicates that an IFTHEN WHEN, BEGIN or END operand of an OUTFIL statement caused the error. \( n \) 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, TRLID, or IFTHEN WHEN, BEGIN or END 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" after the end of each record that satisfies a condition. Use an OUTFIL statement to keep (INCLUDE) or delete (OMIT) the records which have the flag on, and then remove the flag.

In the following example, the input data set has RECFM=F8 and LRECL=80. a1 to an and b1 to bn represent relational conditions. We use an INREC statement to set a 'Y' flag in position 81 after the end of each record for which any specified condition is true. We use an OUTFIL statement to INCLUDE the records with a 'Y' in position 81, and then we remove the flag so the output records have the same length as the input records.

```c
//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 OVFL0=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 occur; the original records are left unchanged. 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 OVFL0=RC0, OVFL0=RC4, or OVFL0=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)
**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 installation 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 rsn

**Explanation**
Critical. The COPY function was specified but could not be processed. rsn is the reason code associated with the error. The reason code helps you determine the specific cause of the error. Reason code values (rsn) 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.
The SORTIN data set had keyed records (ICEGENER).

User labels were present (ICEGENER).

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 (rsn).

**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** ICEEXIT CHANGED ONE OR MORE OPTIONS IN EFFECT

**Explanation**
The initialization exit (ICEEXIT) was used to change one or more of the options that apply to your application. DFSORT used the changed options where appropriate. Messages ICE127I through ICE133I, ICE235I and ICE236I indicate which options DFSORT used.

**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)
```

```
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 shown previously 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/SORTINnn 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 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 NULLOUT=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.

ICE175I ATTEMPT TO REALLOCATE VIO WORK DATA SETS TO REAL RESULTED IN VIO

Explanation

Since VIO=NO was in effect, DFSORT attempted to reallocate work data sets to real (non-VIO) devices, but could not. Since the reallocated data sets were directed to VIO again, DFSORT continued processing with the VIO work data sets, and issued this warning message. The use of VIO for work data sets is not recommended because:

- It prevents the use of large format data sets
- Performance might be degraded since DFSORT's Memory Object, Hiperspace, and Dataspace functions already exploit processor storage to avoid DASD I/O to work datasets.

System action

Processing continues using the VIO work data sets. Large format data sets are not used.

Programmer response

No action is necessary. If appropriate, verify that the DFSMS Automatic Class Selection (ACS) routines do not force DFSORT work data sets to VIO.
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
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 = nK BYTES
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) [yyyyyyy]

Explanation
Critical. The HPSERV SREAD service request for a Hiperspace read failed. xx is the HPSERV return code (in decimal). yyyyyyy is the reason code (set only when the return code is 8).

System action
The program terminates.

Programmer response
Refer to the HPSERV service documentation in the z/OS MVS Programming: Authorized Assembler Services Reference EDT-IXG for a description of HPSERV 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) [yyyyyyy]

Explanation
Critical. The HPSERV SWRITE service request for a Hiperspace write failed. xx is the HPSERV return code (in decimal). yyyyyyy is the reason code (set only when the return code is 8).

System action
The program terminates.

Programmer response
Refer to the HPSERV service documentation in the z/OS MVS Programming: Authorized Assembler Services Reference EDT-IXG for a description of HPSERV SWRITE return codes and reason codes. If possible, take appropriate action.

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 its 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:

- The abend type, either S for system or U for user
- 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.

- 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.
- 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 Unnnn 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.

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 = nK 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 rsn
**Explanation**

Critical. DFSORT's primary technique, Blockset, could not be used due to the reason indicated by reason code *rsn*, the meaning of which is documented in message ICE800I.

Blockset was required to handle one of the following situations which cannot be handled by DFSORT's secondary techniques:

- **locale** processing
- Unicode data formats (UTF8/UTF16/UTF32) processing
- **OUTFIL** processing
- **JOINKEYS** processing
- **MERGEIN** processing
- **Y2x, Y2xx, Y4x, PD0, FS, CSF, UFF, or SFF format** processing
- **PARSE, OVERLAY, FINDREP, IFTHEN, or IFOUTLEN** processing
- **3Pn"string" in EXEC PARM (where n is 0-9)**
- **INREC or OUTREC** processing with one of the following:
  - p,m,HEX
  - p,HEX
  - p,m,TRAN=keyword
  - p,TRAN=keyword
  - p,m,f
  - p,m,lookup
  - p,m,JFY=(...)
  - p,m,SQZ=(...)
  - %nnn, %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, DATE5, 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 DATASORT, SUBSET, or Resize processing**
- **ICETOOL called DFSORT for an operation involving SORTOUT, and NULLOUT=RC16 is in effect**
- **Db2 Utilities called DFSORT.**
- **INCLUDE or OMIT processing with NUM, UC, LC, MC, UN, LN or MN**
- 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
- A program dynamically allocated DFSORT input, output or work data sets using the options for XTIOT, uncaptured UCBs, or DSAB above 16 megabyte virtual.
- DFSORT was called with the 64-bit invocation parameter list.

**System action**

The program terminates.

**Programmer response**

Correct the situation indicated by the reason code (*rsn*) 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...
Informational and Error Messages

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 Enn WAS ACCEPTED

**Explanation**
User exit Enn (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**
None.

**Programmer response**
None.

**ICE193I**  ICEAMx INVOCATION ENVIRONMENT IN EFFECT - ICEyyy ENVIRONMENT SELECTED

**Explanation**
ICEAMx indicates the invocation installation 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

ICEyyy indicates the time-of-day installation environment (ICETDx) or invocation installation environment (ICEAMx) selected for this run. The set of installation defaults for the selected installation environment are used for the run.

An ICETDx installation environment can be selected only if:

- It is specified by the ENABLE parameter of the ICEAMx installation environment 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 ICETDx installation environment is not selected, the ICEAMx installation environment in effect for the run-time environment is selected.

**Note:** You can use ICETOOL's DEFAULTS operator to list merged PARMLIB/ICEMAC defaults in effect for each installation environment.

**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 SOROUT 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 SOROUT LRECL (x) was larger than the SORTIN/SORTINnn LRECL (y)
- TRUNC=RC16 was in effect and the SOROUT LRECL (x) was smaller than the SORTIN/SORTINnn LRECL (y).

System action
The program terminates.

Programmer response
Change the LRECL of SOROUT or SORTIN/SORTINnn 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 of 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 AS MAIN STORAGE = nM BYTES

Explanation
n is the number of megabytes of memory object storage DFSORT used as main storage during this sort. If n is zero, DFSORT did not use memory object storage as main storage 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 SYSTU2 LRECL OF $x$ IS DIFFERENT FROM SYSTU1 LRECL OF $y$ - RC=n

Explanation
The ICEGENER application was successfully processed, but the SYSTU2 LRECL ($x$) and SYSTU1 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:

- Installation option GNPAD=RC4 was specified and $x$ was larger than $y$ (LRECL padding), or
- Installation 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 SYSTU1 or SYSTU2 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, VLTRAIL, REPEAT, SPLIT, SPLITBY, SPLIT1R, NULLOFL, LINES, HEADER1, TRAILER1, HEADER2, TRAILER2, SECTIONS, NODETAIL, BLKCCH1, BLKCCH2, BLKCCT1, REMOVECC, ACCEPT or IFTRAIL. 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 shown in the Explanation 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 REPEAT=5,REPEAT=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, OUTREC, BUILD, OVERLAY or FINDREP was specified with IFTHEN or IFOUTLEN.
  
  **Example:**
  
  ```
  OUTFIL FINDREP=(IN=C'ONE',OUT=C'TWO'),
  IFTHEN=(WHEN=INIT,OVERLAY=(25:C'YES'))
  ```

- OUTFIL and BUILD, OUTFIL and OVERLAY, OUTFIL and FINDREP, BUILD and OVERLAY, BUILD and FINDREP, or OVERLAY and FINDREP were specified.
• IFTRAIL was specified with HEADER1, TRAILER1, HEADER2, TRAILER2, SECTIONS, NODETAIL, LINES, SPLIT, SPLITBY, SPLIT1R, REPEAT, FTOV, VTOF, VLTRIM or VLFILL.

Example:
```
OUTFILE REPEAT=100,
    IFTRAIL=(TRLID=(1,1,CH,EQ,C'T'),
              TRLUPD=(21:COUNT=(M11,LENGTH=5)))
```

• For an IFTHEN clause, WHEN was not specified.

Example:
```
OUTFILE IFTHEN=(OVERLAY=(10:C'A'))
```

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

Example:
```
OUTFILE IFTHEN=(WHEN=(5,1,CH,EQ,C'1'),HIT=NEXT)
```

• For an IFTHEN clause, WHEN=GROUP was specified without BEGIN, KEYBEGIN, END or RECORDS, or without PUSH.

Example:
```
OUTFILE IFTHEN=(WHEN=GROUP,
                BEGIN=(9,2,CH,EQ,C'NO'))
```

• For an IFTHEN clause, WHEN=(logexp), WHEN=ANY, or WHEN=NONE was specified without PARSE, but without BUILD, OVERLAY, or FINDREP.

Example:
```
OUTFILE IFTHEN=(WHEN=None,
                PARSE=(%01=(FIXLEN=5,ENDBEFR=BLANKS)))
```

• For an IFTHEN clause, WHEN=INIT, WHEN=(logexp), WHEN=ANY, or WHEN=NONE was specified with BEGIN, KEYBEGIN, END, RECORDS or PUSH.

Example:
```
OUTFILE IFTHEN=(WHEN=INIT,PUSH=(9:5,8))
```

• For an IFTHEN clause, WHEN=GROUP was specified with PARSE, BUILD, OVERLAY or FINDREP.

Example:
```
OUTFILE IFTHEN=(WHEN=GROUP,BUILD=(9:5,8))
```

• For an IFTHEN clause, WHEN=INIT and BUILD with / were specified

Example:
```
OUTFILE IFTHEN=(WHEN=INIT,BUILD=(1,25,/,26,25))
```

• For an IFTHEN clause, BUILD with / and HIT=NEXT were specified.

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

• An IFTHEN clause with WHEN=GROUP was preceded by an IFTHEN clause with WHEN=(logexp), WHEN=ANY or WHEN=NONE.

Example:
```
OUTFILE IFTHEN=(WHEN=(5,2,CH,EQ,C'AA'),
                OVERLAY=(10:C'A'),
                IFTHEN=(WHEN=GROUP,RECORDS=3,
                        PUSH=(8:SEQ=2))
```

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

Example:
```
OUTFILE 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:
```
OUTFILE 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:
```
OUTFILE 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 VL_FILL=byte operand
- A VL_TRIM=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 VL_FILL=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 VL_TRIM=byte operand was specified, use the same operand and BUILD or IFTHEN BUILD in an OUTFIL statement instead. For example:

```
OUTFIL VL_TRIM='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**
If variable-length records longer than the LRECL of the SORTOUT or OUTFIL data set are acceptable (for example, if you added bytes at the end with VLTRAIL=string), increase the LRECL of the SORTOUT or OUTFIL data set so it is equal to or greater than the largest record to be written.

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.

This message can be issued in the JOINKEYS F1 or F2 subtask for an INCLUDE or OMIT statement specified in JNFnCNTL or for an INCLUDE or OMIT operand specified on a JOINKEYS statement. An INCLUDE or OMIT statement specified in JNFnCNTL will be displayed in the subtask. An INCLUDE or OMIT operand specified on a JOINKEYS statement will be displayed in the main task. If you specify an INCLUDE or OMIT statement in JNFnCNTL, you can also specify VLSHRT in JNFnCNTL if appropriate. If you specify an INCLUDE or OMIT operand on the JOINKEYS statement, you can specify VLSHRT in DFSPARM if appropriate.

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 rsn

**Explanation**

Critical. The OUTFIL data set associated with the indicated ddname could not be processed. rsn is the reason code associated with the error. The reason code helps you determine the specific cause of the error. Reason code values (rsn) are as follows:
An attempt to OPEN the OUTFIL tape data set to retrieve its attributes failed.

The OUTFIL data set resided on an unsupported device.

An attempt to read the DSCB for the OUTFIL data set failed.

The OUTFIL data set was BDAM.

System action
The program terminates.

Programmer response
Correct the situation indicated by the reason code (rsn).

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, TRLID, or IFTHEN WHEN, BEGIN or END operand. The specific cause of the error is identified as follows:

- *INCLUDE indicates that the COND operand of the INCLUDE statement, or an INCLUDE operand of a JOINKEYS statement, caused the error. n is 0.
- *OMIT indicates that the COND operand of the OMIT statement, or an OMIT operand of a JOINKEYS statement, caused the error. n is 0.
- ddname and n=0 indicates that the INCLUDE, OMIT, or TRLID 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, BEGIN or END 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 OUTFIL statement).
- *OUTREC indicates that an IFTHEN WHEN, BEGIN or END 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, BEGIN or END 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 BUILD, OUTREC, OVERLAY, FINDREP, IFTHEN BUILD, IFTHEN OVERLAY, IFTHEN FINDREP or IFTHEN PUSH 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, OVERLAY, or FINDREP operand, or the specified MAXLEN length for the FINDREP 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 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=(ab) 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 was greater than 32752 bytes.
• The length for a 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: 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**  
**ddname** : DELETED = **n**, REPORT = **n**, DATA = **n

**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** : TOTAL IN = **n**, TOTAL OUT = **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.
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.

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.

Example:

\begin{verbatim}
OUTFIL FNAMEs=out,
   HEADER2=('Page ',PAGE),REPEAT=5
\end{verbatim}

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:
  
  OUTFIL STARTREC=10, ENDREC=9

  The n value of ENDREC=n must be equal to or greater than the n value of STARTREC=n. Examples:
  
  OUTFIL STARTREC=10, ENDREC=10

  to process record 10, or:
  
  OUTFIL 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:
  
  OUTFIL SAMPLE=1

  The n value of SAMPLE=n or SAMPLE=(n,m) must be equal to or greater than 2. Examples:
  
  OUTFIL SAMPLE=2

to process records 1,3, and so on, or:

  OUTFIL 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:

  OUTFIL REPEAT=1

  The n value of REPEAT=n must be equal to or greater than 2. Example:

  OUTFIL 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=a

Explanation
Issued in conjunction with messages ICE127I through ICE133I and ICE236I to indicate the options in effect.

a 'RC0', 'RC4', or 'RC16'

See ICE133I for more information.

ICE236I OPTIONS:

DYNAPCT=a, MOWRK=b, TUNE=c, EXPMAX=d, EXPOLD=d, EXPRES=d

Explanation
Issued in conjunction with messages ICE127I through ICE133I and ICE235I to indicate the options in effect.

a A decimal value representing a percentage, or 'OLD'

b 'Y' for YES; 'N' for NO

c 'STOR', 'DISK', 'DDYN', or 'OLD'

d A decimal value representing megabytes, a percentage, or 'MAX'

See ICE133I for more information.
ICE241A  {*INREC|*OUTREC|ddname}  
IFTHEN n COLUMN OVERLAPS RECORD DESCRIPTOR WORD

Explanation
Critical. For variable-length record processing, the OVERLAY, IFTHEN OVERLAY, or IFTHEN PUSH operand of an INREC, OUTREC or OUTFIL statement specified an 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 or IFTHEN PUSH 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 or IFTHEN PUSH 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 or IFTHEN PUSH 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, IFTHEN OVERLAY, or IFTHEN PUSH item so the default of 1: was used for that item. Example:
  \[
  \text{OVERLAY} = \left( \text{C} ^{ \text{abc} } \right)
  \]
- c: was specified for an OVERLAY, IFTHEN OVERLAY, or IFTHEN PUSH item with a value for c which was less than 5. Example:
  \[
  \text{PUSH} = \left( 3 : \text{SEQ}=5 \right)
  \]

System action
The program terminates.

Programmer response
Specify c: with a value of 5 or more for the first OVERLAY, IFTHEN OVERLAY, or IFTHEN PUSH item. Ensure that c is 5 or more for any other c: values you specify. Example:

\[
\text{OVERLAY} = \left( 8 : \text{C}\text{'}\text{abc}', 1, 2, \text{HEX}, 25 : 5\text{C}\text{'\top'})
\]

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 3 digits was specified for a parsed field definition (for example, %0001 instead of %1, %01 or %001). A parsed field must start with % and must have 1 digit, 2 digits, 3 digits or no digits.
- FIXLEN was not specified for %nnn, %nn or %n.
- FIXLEN, ABSPOS, ADDPOS, SUBPOS, REPEAT 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.
- REPEAT was specified with a value less than 2 or greater than 1000.

System action
The program terminates.
• The definition for %nnn included a REPEAT value which would result in a parsed field higher than %999 (for example, %998=(REPEAT=3,...)).
• 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 %nnn, %nn or %n parsed field was defined previously (for example, %03 was defined twice, %6 and %06 were both defined, or %15 and %015 were both defined). Each value for a parsed field from 0-999 must only be defined once in all PARSE operands of the INREC, OUTFREC and OUTFIL statements. %n, %0n and %00n (for example, %2, %02 and %002) are treated as duplicates. %nn and %0nn (for example, %12 and %012) are treated as duplicates.

%nnn parsed fields resulting from the use of REPEAT=v must not be duplicates of %nnn parsed fields defined previously or subsequently. For example, %11=(REPEAT=5,...),%15=(...) would be duplicate definitions of %15, and %21=(...),%20=(REPEAT=2,...) would be duplicate definitions of %21.

System action
The program terminates.

Programmer response
A $ marks the duplicate parsed field. Use unique values (%nnn, %nn or %n) for parsed fields (%0-%999). Ensure that %nnn values defined using REPEAT=v are not duplicates of previous or subsequent values.

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, OUTFREC or OUTFIL statement specified a parsed field (%nnn, %nn or %n) that was not previously defined for use with that operand or suboperand. %6nn, %nn and %n parsed fields (referred to as %nn below for simplicity) 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 OUTFREC OVERLAY/BUILD/FIELDS: The %nn parsed field must be defined in the OUTFREC 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 OUTFREC 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 OUTFREC IFTHEN OVERLAY/BUILD: The %nn parsed field must be defined in the OUTFREC 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=10,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.
  OUTFIL 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))
* %103 defined and used for OUTFIL OUT1.
* %104 defined and used for OUTFIL OUT2.
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=10,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=10,ENDBEFR=C',')),
    OVERLAY=(21:%03)
    OUTFIL FNAMES=OUT2,SAVE,
    PARSE=(%03=(FIXLEN=8,ENDBEFR=C':')),
    OVERLAY=(31:%03)

### System action
The program terminates.

### Programmer response
A $ marks the incorrectly used %nnn, %nn or %n parsed field. Define the %nnn, %nn or %n parsed field correctly.

ICE246A SORT CAPACITY EXCEEDED - RECORD COUNT: n

### Explanation
Critical. Blockset was not selected, and DFSORT was not able to complete processing with the intermediate storage available (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 on any allocated work data set.

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

The amount of intermediate storage 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 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 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.

ICE247I INTERMEDIATE MERGE ENTERED - PERFORMANCE MAY BE DEGRADED

**Explanation**
Because the amount of virtual storage available to DFSORT was a small fraction of the amount of data to be sorted, DFSORT required additional intermediate merge passes over the data to perform the sort. Such intermediate merge passes often result in significant performance degradation with a consequent increase in intermediate storage requirements.

### System action
Processing continues using intermediate merge passes as required.

### Programmer response
Increase the amount of virtual storage available for the sort. For information and guidelines for optimizing virtual storage, refer to “Run-Time Considerations” in z/OS DFSORT Tuning Guide.

Informational and Error Messages 83
ICE248I  DFSORT ATTEMPTED TO ALLOCATE xMB OF DISK WORK SPACE ON y WORK DATA SETS

Explanation
Provides information on the total disk work space DFSORT attempted to allocate dynamically when it was unable to allocate all of the work space it requested. x indicates the total megabytes of disk work space DFSORT attempted to allocate. y indicates the total number of work data sets used for the allocations. x/y is the amount of free space that must be available on at least y volumes for DFSORT to allocate all of the work space it requested.

System action
None.

Programmer response
Optional. Visit the DFSORT website to see what it has to offer.

ICE249I  DFSORT SUCCESSFULLY ALLOCATED xMB OF DISK WORK SPACE ON y WORK DATA SETS

Explanation
Provides information on the total disk work space DFSORT successfully allocated dynamically when it was unable to allocate all of the work space it requested. x indicates the total megabytes of disk work space DFSORT successfully allocated. y indicates the total number of work data sets used for dynamic allocation.

System action
None.

Programmer response
Optional. Visit the DFSORT website to see what it has to offer.

ICE251A  MISSING RDW OR DATA FOR {*INREC|*OUTREC|ddname}: REASON CODE rsn, IFTHEN n

Explanation
Critical. For variable-length records, 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, did not start with positions 1 to 4 for the RDW (record descriptor word) or did not contain any data bytes after the RDW. The operand must start with 1,4 or 1:1,4 or 1,n or 1:1,n (with n greater than 4) and must be followed by an input field (for example, 1,8,11,5) or a separation field (for example, 1:1,8,12:C'A')

Remember that the first data byte starts in position 5 after the RDW in positions 1-4.

The following information is displayed in the message:
- *INREC if the error is associated with an INREC statement. *OUTREC if the error is associated with an OUTREC statement. ddname if the error is associated with an OUTFIL statement.
- rsn: The reason code associated with the error. The reason code (listed later in this section) helps you determine the specific cause of the error.
- n: 0 if the error is not associated with an IFTHEN clause. Otherwise, n identifies the number of the IFTHEN clause associated with the error (starting at 1 for the first IFTHEN clause).

The statement and operand associated with the error are 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).

Reason code values (rsn) are as follows:

1. The first field was a separation field, or the first field was preceded by a column other than 1.
   Examples:
   - INREC FIELDS=(4X)
   - OUTREC BUILD=(8:1,20)
   - OUTFIL BUILD=(1:C'ABCD')

2. The first field was a position without a length.
   Examples:
   - INREC BUILD=(1)
   - OUTREC BUILD=(5)

3. The first field did not start with 1. Example:
   - OUTREC IFTHEN=(WHEN=INIT, BUILD=(2,8,3X,21,5,HEX))

4. The first field has a length less than 4.
   Example:
   - INREC BUILD=(1,3,10:X)

5. The first field is modified or edited.
   Examples:
   - INREC BUILD=(1,4,HEX)
   - OUTREC BUILD=(1,9,PD,M11)
   - OUTFIL PARSE=(%01=(ENDBEFR=C',',FIXLEN=5)), BUILD=(1,4,%01)

6. 1,4 is the only field. Example:
   - OUTREC BUILD=(1,4)

System action
The program terminates.

Programmer response
Take the action for the indicated reason code value (rsn) as follows:

1. Specify 1,4 before the separation field. Examples:
   - INREC FIELDS=(1,4,4X)
   - OUTREC BUILD=(1,4,8:5C'*')
   - OUTFIL BUILD=(1,4,C'ABCD')

2. Specify 1,n with n equal to or greater than 4 as the first field. Examples:
   - INREC BUILD=(1,4,5)
   - OUTREC BUILD=(1,8,9,10)

3. Specify 1,n with n equal to or greater than 4 as the first field. Example:
   - OUTREC IFTHEN=(WHEN=INIT, BUILD=(1,8,3X,21,5,HEX))

4. Specify 1,n with n equal to or greater than 4 as the first field. Example:
   - INREC BUILD=(1,4,10:X)

5. Specify 1,4 before the modified or edited field.
   Examples:
   - INREC BUILD=(1,4,1,4,HEX)
   - OUTREC BUILD=(1,4,5,9,PD,M11)
   - OUTFIL PARSE=(%01=(ENDBEFR=C',',FIXLEN=5)), BUILD=(1,4,%01)

6. Specify at least one input field or separation field after 1,4. Examples:
   - INREC BUILD=(1,4,X)
   - OUTFREC FIELDS=(1,5)

ICE252I PARMLIB OPTIONS WERE MERGED WITH INSTALLATION MODULE DEFAULTS

Explanation
For this run, installation options from one or more ICEPRMxx members in PARMLIB were merged with the defaults for the selected installation module indicated in message ICE191I. DFSORT used the merged installation defaults where appropriate.
**System action**

None.

**Programmer response**

No action is necessary. You can use the DEFAULTS operator of ICETOOL to display the merged installation defaults, the ICEPRMxx PARMLIB options, and the ICEMAC defaults.

ICE253I  RECORDS SORTED - PROCESSED:  p, EXPECTED:  e

**Explanation**

Indicates the processed (p) and expected (e) records sorted. The number of records includes all records received, added and not deleted during DFSORT input processing.

p is based on the number of records actually processed. e is based on DFSORT’s determination of the total number of records it expected taking into account various factors such as the estimated input file size, the FILSZ value, and so on.

If DFSORT terminated before completing the input phase, p is based on the number of records processed up to the point of failure.

**System action**

None.

**Programmer response**

Take appropriate action depending on whether or not DFSORT issued message ICE046A:

- ICE046A: If p is greater then e, specify FILSZ=En, where n is greater than or equal to p and is a reasonably accurate estimate of the number of records to be processed.
- No ICE046A: If p is significantly different from e, specifying FILSZ=En with a reasonably accurate estimate of the number of records to be sorted can prevent DFSORT from overallocating or underallocating dynamic work space. p was the actual number of records processed for this run. If future runs will process approximately the same amount of data, consider specifying FILSZ=Ep. See z/OS DFSORT Application Programming Guide for information about the FILSZ parameter and factors that can cause a difference between p and e (E15, INCLUDE/OMIT, unmanaged tape, and so on).

See z/OS DFSORT Application Programming Guide for additional information about parameters you can use to provide accurate information to DFSORT that can prevent overallocation or underallocation of dynamic work space (FILSZ, AVGRLEN, DYNSPC, and so on). Also, see the following messages, if issued: ICE098I, ICE254I and ICE258I.

For a JOINKEYS operation, you can supply any needed parameters as follows:

- In DFSPARM for the JOINKEYS main task.
- In JNF1CNTL for the JOINKEYS F1 subtask.
- In JNF2CNTL for the JOINKEYS F2 subtask.

ICE254I  DFSORT FAILED TO DYNAMICALLY ALLOCATE THE REQUIRED WORK DATA SET SPACE

**Explanation**

The work data set space DFSORT was able to dynamically allocate for this sort application was not sufficient for the application to complete. DFSORT's determination of the required space takes into account various factors such as the estimated input file size, the average record length, the FILSZ value, the AVGRLEN value, and so on.

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

**System action**

None.

**Programmer response**

You should investigate why DFSORT could not dynamically allocate all of the space it required. You may need to increase the number of work data sets or the available space in the volume pool for the work data sets. You should ensure that DFSORT is provided accurate file size (FILSZ=En) and average record length (AVGRLEN=n) information.

See z/OS DFSORT Application Programming Guide for additional information about parameters you can use to provide accurate information to DFSORT that can prevent overallocation or underallocation of dynamic work space (FILSZ, AVGRLEN, DYNSPC, and so on). Also, see the following messages, if issued: ICE098I and ICE253I.

For a JOINKEYS operation, you can supply any needed parameters as follows:

- In DFSPARM for the JOINKEYS main task.
- In JNF1CNTL for the JOINKEYS F1 subtask.
- In JNF2CNTL for the JOINKEYS F2 subtask.

ICE255I  ddname STATISTICS MAY BE INCORRECT - FILSZ=EN MAY IMPROVE RESOURCE USAGE AND PERFORMANCE
**Explanation**

DFSORT could not be certain that the data set information provided by Catalog was correct for this sort application. The catalog entry for this data set indicated that the component was not closed properly and the statistics for the data set are invalid.

Generally, DFSORT can automatically determine the input file size. However, in a few cases, such as when the catalog statistics are invalid, DFSORT may not be able to determine an accurate file size.

**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.

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.

For more information regarding the cause of the invalid statistics, or the steps necessary to correct the statistics, see the "STA: Statistics Group" section of z/OS DFSMS Access Method Services Commands.

**ICE256I**  DFSORT CODE IS ELIGIBLE TO USE ZIIP FOR THIS DB2 UTILITY RUN

**Explanation**

For this Db2 Utility run, DFSORT will execute code that is eligible to use the IBM System z Integrated Information Processor (zIIP) subject to its availability.

**System action**

None.

**Programmer response**

None.

**ICE257A**  CRITICAL ERROR RETURNED FROM SRB - a b c d

**Explanation**

Critical. A system error occurred or an abend was requested while DFSORT was executing a Service Request Block (SRB) eligible to use the IBM System z Integrated Information Processor (zIIP). a, b, c and d provides information to help IBM diagnose the failure.

**ICE258I**  DFSORT COULD NOT DYNAMICALLY ALLOCATE THE OPTIMAL WORK DATA SET SPACE

**Explanation**

The work data set space DFSORT was able to dynamically allocate for this sort application was not the optimal amount. DFSORT’s determination of the optimal space takes into account various factors such as the estimated input file size, the average record length, the FILSZ value, the AVGRLEN value, and so on.

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

**System action**

None.

**Programmer response**

You should consider investigating why DFSORT could not dynamically allocate all of the space it requested. Although DFSORT can often complete successfully with less than the optimal work data set space, a failure is more likely if the work space requirements are greater than expected. You may want to increase the number of work data sets or the available space in the volume pool for the work data sets. You should ensure that DFSORT is provided accurate file size (FILSZ=En) and average record length (AVGRLEN=n) information.

See z/OS DFSORT Application Programming Guide for additional information about parameters you can use to provide accurate information to DFSORT that can prevent overallocation or underallocation of dynamic work space (FILSZ, AVGRLEN, DYNSPC, and so on). Also, see the following messages, if issued: ICE098I and ICE253I.

For a JOINKEYS operation, you can supply any needed parameters as follows:

- In DFSPARM for the JOINKEYS main task.
- In JNF1CNTL for the JOINKEYS F1 subtask.
- In JNF2CNTL for the JOINKEYS F2 subtask.

**ICE259A**  PUSH FIELD ERROR
**Explanation**
Critical. The PUSH operand of an INREC, OUTREC or OUTFIL statement contained an invalid column, position or length, as follows:

- A 0 value was used.
- A column was greater than 32752, or was followed by another column.
- An input position plus length was greater than 32753.
- The length for a SEQ or ID field was greater than 15.
- An output field was beyond position 32767.

**System action**
The program terminates.

**Programmer response**
Correct the invalid column, position or length.

**ICE260A FIND/REPLACE FIELD ERROR**

**Explanation**
Critical. One of the following errors was found in the FINDREP operand of an INREC, OUTREC or OUTFIL statement:

- A 0 value was used.
- A null value was used where it was not permitted.
- IN or INOUT was not specified.
- INOUT, IN, OUT, STARTPOS, ENDPOS, DO, MAXLEN, OVERRUN or SHIFT was specified more than once.
- INOUT was specified with IN or OUT.
- IN was specified without OUT, or OUT was specified without IN.
- An input constant was specified without a matching output constant in INOUT.
- A repetition factor was 0 or greater than 256 for a character string or hexadecimal string.
- The total length of a single or repeated character or hexadecimal constant was greater than 256 bytes.
- An invalid digit or an odd number of digits was specified for a hexadecimal string.
- The value for STARTPOS, ENDPOS or MAXLEN was greater than 32752.
- The value for DO was greater than 1000.

**System action**
The program terminates.

**Programmer response**
Check the FINDREP operand for the errors indicated in the explanation and correct the errors.

**ICE261A FIND AND REPLACE CAUSED OVERRUN OF n BYTE (*INREC/ *OUTREC/ddname) OUTPUT RECORD END**

**Explanation**
Critical. OVERRUN=ERROR was specified or defaulted for the FINDREP operand of an INREC, OUTREC or OUTFIL statement, and processing of the find/replace constants caused nonblank characters to overrun the end of the maximum of n bytes allowed for the output record. This can occur when an output constant is longer than an input constant (for example, INOUT=(C'A',C'XYZ')) or when the length of the record to be created by FINDREP is less than the length of the original record (for example, when the input record is 80 bytes and MAXLEN=50 is used to decrease the length of the record).

The specific source of the error is identified as follows:

- *INREC indicates that a FINDREP operand in the INREC statement caused the error.
- *OUTREC indicates that a FINDREP operand in the OUTREC statement caused the error.
- ddname indicates that a FINDREP operand in an OUTFIL statement caused the error; ddname identifies the first data set in the associated OUTFIL group.

**System action**
The program terminates.

**Programmer response**
If you want to truncate nonblank characters that overrun the end of the record, specify OVERRUN=TRUNC. If you do not want to truncate nonblank characters that overrun the end of the record, increase the length of the record using MAXLEN=m with a value for m equal to or greater than the maximum length record to be created by FINDREP.

**ICE264I ddname STATISTICS MAY be INCORRECT**

**Explanation**
DFSORT could not be certain that the data set information provided by Catalog was correct for this copy application. The catalog entry for this data set indicated that the component was not closed properly and the statistics for the data set are invalid.
System action
None.

Programmer response
None required. For more information regarding the cause of the invalid statistics, or the steps necessary to correct the statistics, see the "STA: Statistics Group" section of z/OS DFSMS Access Method Services Commands.

ICE265A  TRLUPD FIELD ERROR

Explanation
Critical. The TRLUPD parameter of an OUTFIL statement contained an invalid column, keyword, position, length, format, pattern or sign. Some common errors are:

- A 0 value was used.
- A column was greater than 32752 or was followed by another column.
- A column overlapped the previous output field in the record.
- A keyword other than COUNT, COUNT+n, COUNT-n, TOTAL or TOT was specified.
- A position plus length for a TOTAL field was greater than 32753.
- The length for a TOTAL field was greater than 8 for BI or FI, 16 for PD, 32 for ZD, 32 for CSF/FS, or 44 for UFF or SFF.
- 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.

System action
The program terminates.

Programmer response
Correct the invalid value.

ICE266A  ddbname TRLUPD COLUMN OVERLAPS RECORD DESCRIPTOR WORD

Explanation
Critical. For variable-length record processing, the TRLUPD parameter of the OUTFIL data set associated with ddbname specified an item that overlapped the record descriptor word (RDW). Only data bytes, which start at position 5 for variable-length records, can be overlaid.

The error is one of the following:
- c: was not specified for the first item so the default of 1: was used for that item. Example:

  TRLUPD=(COUNT=(M10,LENGTH=5))

- c: was specified for an item with a value for c which was less than 5. Example:

  TRLUPD=(3:COUNT=(M10,LENGTH=5))

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 %nnn (nn is 000 to 999), %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, B1, 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, Y4T, Y4U, Y4V, Y4W, Y4X, Y4Y 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, ADDDAYS, ADDMONS, ADDYEARS, ALL, AND, AQ, ASL, AST, BI, CH, CLO, COPY, COUNT, COUNT15, CSF, CSL, CST, CTO, D, DATE, DATEDIFF, DATE1, DATE1..., DATE2, DATE2..., DATE3, DATE3..., DATE4, DATE5, DC1, DC2, DC3, DE1, DE2, DE3, DIV, DT1, DT2, DT3, D1, D2, E, F, FI, FL, FS, H, HEX, LASTDAYM, LASTDAYQ, LASTDAYW, LASTDAYY, LC, LN, LS, MAX, MC, MIN, MN, MOD, MUL, Mn, Mnn, NEXTDday, NONE, NUM, OL, OR, OT, PAGE, PAGEHEAD, PD, PDC, PDF, PD0, PREVDday, SEQNUM, SFF, SS, SUB, SUBCOUNT, SUBCOUNT15, SUBDAYS, SUBMONS, SUBYEARS, TC1, TC2, TC3, TC4, TE1, TE2, TE3, TE4, TIME, TIME1, TIME1P, TIME2, TIME2P, TIME3, TIME3P, TM1, TM2, TM3, TM4, TS, UC, UFF, UN, VALCNT, VLEN, X, Y2x, Y2xx, Y4x, Z, ZD, ZDC, and ZDF, where n is 0-9, day is SUN, MON, TUE, WED, THU, FRI or SAT, 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).

ICE278I x WORK DATA SETS WERE INSUFFICIENT TO COMPLETE THIS SORT SO y ADDITIONAL WORK DATA SETS WERE USED

Explanation
DFSORT initially intended to use only x work data sets for this sort application. However, the work space requirements for the sort were unexpectedly larger than anticipated, so DFSORT used y additional work data sets.

System action
None.

Programmer response
Verify that DFSORT has the correct file size information to allow it to accurately calculate the initial amount of work space needed. See messages ICE098I and ICE253I for additional information.

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 by the PARMDDN installation option
- 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:

  • SUM FIELDS=(New-Field)
    * New*Field is invalid. New_Field is valid.
    * New-Field is valid.
    OUTREC FIELDS=(AMB)
    * AMB 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
  PARSE=(%01=(STARTAFT=MQ,FIXLEN=10)),BUILD=(%01)
  If MQ is not defined as a symbol, STARTAFT=MQ is misinterpreted as an undefined symbol because MQ is an invalid keyword for STARTAFT. The following is valid:

  OUTREC
  PARSE=(%01=(STARTAFT=MC,FIXLEN=10)),BUILD=(%01)

System action
The program terminates.

Programmer response
Correct the symbol, syntax or delimiter error.
**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**  
DEVICE WITH CAPACITY OVER 17476 CYLINDERS REQUIRES BLOCKSET FOR AlLOcATED WORK DATA SETS

**Explanation**

Critical. One or more JCL or dynamically allocated work data sets was allocated on a type of device with a capacity of 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 for this application on types of devices with more than 17476 cylinders.

**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.

**ICE289A**  
UNSUPPORTED EXTENDED ATTRIBUTES FOR **ddname**

**Explanation**

Critical. Extended attributes for the data set associated with the indicated ddname are not supported by the system on which DFSORT is running. A data set with extended attributes is described with format 8 and 9 DSCBs (Data Set Control Blocks) in the VTOC and may contain extents that are allocated in the...
Extended Addressing Space (EAS) where tracks are described by 28-bit cylinder numbers.

**System action**  
The program terminates.

**Programmer response**  
Use this data set with DFSORT on a system that supports the data set's extended attributes.

**ICE290A**  
**INVALID 64-BIT INVOCATION PARAMETER LIST - REASON CODE IS rsn**

**Explanation**  
Critical. The invocation parameter list passed to DFSORT using ATTACH, LINK or XCTL contained the eyecatcher C'PL64SORT', identifying it as a 64-bit invocation parameter list. However, an error was found in this parameter list.

rsn is the reason code associated with the error. The reason code helps you determine the specific cause of the error. Reason code values (rsn) are as follows:

1. More than one AMODE flag was set for a single exit in the 64-bit invocation parameter list. For example, the AMODE 64 and AMODE 31 flags were set for an E35 exit.
2. The AMODE flag set for an exit in the 64-bit invocation parameter list does not correspond to the actual AMODE used by that exit. For example, the AMODE 64 flag was set for an E35 exit, but the E35 exit uses AMODE 31.
3. Zeros were not present in the 64-bit invocation parameter list where required. For example, a reserved field was not set to zeros, or the high order word of an address was not set to zeros.
4. The block support flag for an E15/E35 was set, but the required address of the user block list parameter area was not present in the 64-bit invocation parameter list.

**System action**  
The program terminates.

**Programmer response**  
Take the action for the indicated reason code value (rsn) as follows:

1. Ensure that only one AMODE flag is set for each exit in the 64-bit invocation parameter list.

**ICE291I**  
**AMODE FLAG FOR EXIT WITHOUT CORRESPONDING EXIT ADDRESS IGNORED**

**Explanation**  
The invocation parameter list passed to DFSORT using ATTACH, LINK or XCTL contained the eyecatcher C'PL64SORT', identifying it as a 64-bit invocation parameter list. However, an AMODE flag was set for an exit for which a corresponding exit address was not present in the parameter list. For example, the AMODE 64 flag was set for the E15 exit, but the E15 exit address was zeros.

**System action**  
Processing continues. Each AMODE flag for an exit without a corresponding exit address is ignored.

**Programmer response**  
No action is necessary. If you want to eliminate this warning message, turn off the unneeded AMODE flag or specify a corresponding exit address.

**ICE292A**  
**CRITICAL ERROR WHEN USING BLOCK EXITS: Enn - RSN=rsn**

**Explanation**  
Critical. The invocation parameter list passed to DFSORT using ATTACH, LINK or XCTL contained the eyecatcher C'PL64SORT' and the block support flag is set to ON, identifying it as a block support parameter list. However, an error was found in this parameter list.

Enn is either an E15 or an E35 exit.

rsn is the reason code associated with the error. The reason code helps you determine the specific cause of the error. Reason code values (rsn) are as follows:

1. Block list address is zero.
2. Block list type is invalid.
3. The address of first or last active entry in block list is zero.
4. The address of block records (block list type 2) or record (block list type 1) in active entry is zero or
the length of block of records is invalid (zero or less than record size).

**System action**
The program terminates.

**Programmer response**
Take the action for the indicated reason code value (rsn) as follows:

1. Ensure that a valid block list address is passed.
2. Ensure that a valid value for block list type is passed. Valid values are 1 or 2.
3. Ensure that the first or last active entry has a valid block list address is passed.
4. Ensure the address of the block list or the length of the blocks is valid.

### ICE298I AVERAGE RECORD LENGTH = n BYTES

**Explanation**
Indicates the average record length (n) in bytes for the variable-length records processed by sort when Blockset was not selected, that is, the number of bytes in the variable-length records (including the record descriptor word) divided by the number of records. The number of records includes all records received, added and not deleted during DFSORT input processing.

n is based on the number of bytes and records actually processed.

If DFSORT terminated before completing the input phase, p is based on the number of bytes and records processed up to the point of failure.

**System action**
None.

**Programmer response**
None. However, specifying AVGRLEN=n with a reasonably accurate estimate of the average record length can prevent DFSORT from overallocating or underallocating dynamic work space.

For a JOINKEYS operation, you can supply AVGRLEN=n, if needed, as follows:

- In DFSPARM for the JOINKEYS main task.
- In JNF1CNTL for the JOINKEYS F1 subtask.
- In JNF2CNTL for the JOINKEYS F2 subtask.

---

### ICE299I MEMORY OBJECT USED AS WORK STORAGE = nM BYTES

**Explanation**

n is the number of megabytes of memory object storage DFSORT used as intermediate work space during this sort. If n is zero, DFSORT did not use memory object storage as intermediate work space during this sort.

**System action**
None.

**Programmer response**
None.

### ICE400A INVALID JOINKEYS, JOIN OR REFORMAT STATEMENT OPERAND

**Explanation**
Critical. An invalid keyword operand was detected on a JOINKEYS, JOIN or REFORMAT control statement.

**System action**
The program terminates.

**Programmer response**
Make sure the JOINKEYS, JOIN or REFORMAT control statement contains only valid keyword operands.

### ICE401A DUPLICATE JOINKEYS, JOIN OR REFORMAT STATEMENT OPERAND

**Explanation**
Critical. One of the following errors was found:

- On a JOINKEYS, JOIN or REFORMAT statement, a keyword was specified twice.
- On a JOINKEYS statement, more than one of FILE=F1, FILES=F1, F1=ddname, FILE=F2, FILES=F2 or F2=ddname was specified.
- On a JOINKEYS statement, INCLUDE and OMIT were both specified.

**System action**
The program terminates.

**Programmer response**
Check the JOINKEYS, JOIN or REFORMAT control statement for the errors indicated in the explanation and correct the error.
ICE402A  JOINKEYS STATEMENT FOR F1 WAS REQUIRED, BUT NOT FOUND

Explanation
Critical. A JOINKEYS application was requested by a JOINKEYS, JOIN or REFORMAT statement, or a JKFROM operand (ICETOOL COPY or SORT), but a JOINKEYS statement for file F1 or file F2, as indicated, was not found.

System action
The program terminates.

Programmer response
Supply two JOINKEYS statements; one for F1 (with FILE=F1, FILES=F1 or F1=ddname) and another for F2 (with FILE=F2, FILES=F2 or F2=ddname).

ICE403A  OPERAND keyword WAS REQUIRED FOR verb STATEMENT, BUT NOT FOUND

Explanation
Critical. The indicated required keyword was missing for the indicated control statement as follows:
• For a JOINKEYS statement, FILE=F1, FILES=F1, FILE=F2, FILES=F2, F1=ddname or F2=ddname must be specified.
• For a JOINKEYS statement, FIELDS must be specified.
• For a JOIN statement, UNPAIRED must be specified.
• For a REFORMAT statement, FIELDS must be specified.

System action
The program terminates.

Programmer response
Specify the indicated keyword for the indicated control statement.

ICE404A  REFORMAT STATEMENT WAS REQUIRED, BUT NOT FOUND

Explanation
Critical. A JOIN statement with an ONLY operand was not specified, so a REFORMAT statement is required. However, a REFORMAT statement was not found.

System action
The program terminates.

ICE405A  JOINKEYS STATEMENTS HAD MISMATCH IN NUMBER, LENGTH OR ORDER OF KEYS

Explanation
Critical. The keys (p,m,s) specified in the FIELDS operands of the JOINKEYS statements for F1 and F2 did not match in one or more of the following ways:
• The two FIELDS operands have different numbers of keys. For example, FIELDS for F1 has two keys and FIELDS for F2 has three keys.
• Corresponding keys in the two FIELDS operands have different lengths. For example, the second key for F1 has a length of 5 and the second key for F2 has a length of 6.
• Corresponding keys in the two FIELDS operands have different orders. For example, the third key for F1 has ascending order (A) and the third key for F2 has descending order (D).

System action
The program terminates.

Programmer response
Ensure that the FIELDS operands of the JOINKEYS statements for F1 and F2 have the same number of keys, and that corresponding keys have the same length and order.

ICE406A  JOINKEYS STATEMENT FIELD ENDS AFTER POSITION 32752

Explanation
Critical. The last byte of the key (p,m,s) in a JOINKEYS FIELDS operand ended beyond position 32752. Each key must end at or before position 32752 (position plus length must not be greater than 32753). For example, 32752,1,A is valid because it ends at position 32752, but 32752,2,A is invalid because it ends at position 32753.

System action
The program terminates.

Programmer response
Ensure that each key in the JOINKEYS FIELDS operand ends at or before position 32752.
ICE407A  JOINKEYS STATEMENT HAD TOTAL KEY LENGTH GREATER THAN 4080 BYTES

Explanation
Critical. The total length of all of the keys (p,m,s) in a JOINKEYS FIELDS operand exceeded the limit of 4080 bytes.

System action
The program terminates.

Programmer response
Ensure that the total length of all of the keys in the JOINKEYS FIELDS operand is less than or equal to 4080 bytes.

ICE408A  MERGE FUNCTION CANNOT BE USED WITH JOINKEYS MAIN TASK

Explanation
Critical. A MERGE FIELDS=(p,m,s,...) statement was found for the main task of a JOINKEYS application in SYSIN, SORTCNTL, DFSPARM or a parameter list. A MERGE function cannot be used with JOINKEYS.

System action
The program terminates.

Programmer response
Replace the MERGE FIELDS=(p,m,s,...) statement with a MERGE FIELDS=COPY, SORT FIELDS=COPY, OPTION COPY or SORT FIELDS=(p,m,s,...) statement, as appropriate.

ICE409A  INSUFFICIENT STORAGE FOR JOINKEYS APPLICATION - ADD AT LEAST nMB

Explanation
Critical. DFSORT could not get the additional nMB of storage needed for this JOINKEYS application.

System action
The program terminates.

Programmer response
Add at least nMB to the storage available to DFSORT (for example, increase the REGION size).

ICE410A  JOINKEYS APPLICATION TERMINATED - SEE ddname MESSAGES

Explanation
Critical. This message is issued by subtask1 (for file F1) or subtask2 (for file F2) of a JOINKEYS application to indicate that the main task terminated. ddname is the ddname associated with the message data set for the main task.

System action
The program terminates.

Programmer response
See the messages in the indicated ddname data set for information about the main task. Correct the error that caused the main task to terminate.

ICE411I  THIS IS THE JOINKEYS MAIN TASK FOR JOINING F1 AND F2

Explanation
Indicates this is the main task for a JOINKEYS application. The main task processes the joined records from input files F1 and F2 and writes the output.

System action
None.

Programmer response
None.

ICE412A  REFORMAT REQUIRES RDW IN FIRST FIELD

Explanation
The FIELDS operand of the REFORMAT statement has a position without a length (p without m) as its last field, but the first field does not include the RDW (1,n with n equal to or greater than 4). For example, the REFORMAT statement is:

REFORMAT FIELDS=(F1:5,8,F2:1,20,F1:15)

instead of:

REFORMAT FIELDS=(F1:1,8,F2:1,20,F1:15)

or the REFORMAT statement is:

REFORMAT FIELDS=(?,F1:1,4,5)
instead of:

```plaintext
REFORMAT FIELDS=(F1:1,4,?,F1:5)
```

**System action**
The program terminates.

**Programmer response**
Include the RDW (1,n with n equal to or greater than 4) in the first field, or do not use a position without a length, as appropriate.

**ICE413A**

**REFORMAT REQUIRES TYPE=V FILE FOR RDW AND VARIABLE FIELDS**

**Explanation**
Critical. The FIELDS operand of the REFORMAT statement has a position without a length (p without m) as its last field, but has one of the following errors:

- F1: is used for the first field (which includes the RDW), but F1 does not refer to a TYPE=V file.
- F2: is used for the first field (which includes the RDW), but F2 does not refer to a TYPE=V file.
- F1: is used for a position without length field, but F1 does not refer to a TYPE=V file.
- F2: is used for a position without length field, but F2 does not refer to a TYPE=V file.

For example, SORTJNF1 for F1 has RECFM=FB and SORTJNF2 for F2 has RECFM=VB and the REFORMAT statement is:

```plaintext
REFORMAT FIELDS=(F1:1,20,F2:5,6,15)
```

instead of:

```plaintext
REFORMAT FIELDS=(F2:1,4,F1:1,20,F2:5,6,15)
```

**System action**
The program terminates.

**Programmer response**
Ensure that the file (F1 or F2) used for the first field, and for each position without length field, is a TYPE=V file.

**ICE414A**

**ddname (Fn) type FIELD END AT p IS BEYOND LENGTH OF n**

**Explanation**
Critical. A JOINKEYS or REFORMAT statement specifies a field which ends beyond the maximum record length. The information displayed in the message is as follows:

- the ddname of the input file with the field in error.
- the file (F1 or F2) with the field in error.
- the type of field in error as follows:
  - KEY to indicate a field in the FIELDS operand of a JOINKEYS statement.
  - INCLUDE to indicate a field in the INCLUDE operand of a JOINKEYS statement.
  - OMIT to indicate a field in the OMIT operand of a JOINKEYS statement.
  - REFORMAT to indicate a field in the FIELDS operand of a REFORMAT statement.
- the position (p) at which the field ends.
- the maximum record length (n) which the ending position exceeded

**System action**
The program terminates.

**Programmer response**
Ensure that all fields specified in JOINKEYS and REFORMAT statements are contained within the maximum record length indicated.

**ICE415A**

**TYPE=x JOINED RECORD LENGTH OF n EXCEEDS MAXIMUM OF m**

**Explanation**
Critical. The maximum length (n) of the joined records for a JOINKEYS application, as defined by the REFORMAT statement or by default if a REFORMAT statement was not specified, exceeds the maximum length (m) of 32760 for TYPE=F records or 32767 for TYPE=V records.

**System action**
The program terminates.

**Programmer response**
Reduce the maximum length of the joined records so it does not exceed the maximum length of 32760 for TYPE=F records or 32767 for TYPE=V records.

Note that the maximum LRECL for RECFM=VB records is 32756 and the maximum LRECL for RECFM=VBS records is 32767; if you want RECFM=VBS records, ensure that RECFM is specified appropriately for the output data set.
ICE416I
JOINKEYS IS USING THE Fn SUBTASK FOR ddname1 - SEE ddname2 MESSAGES

Explanation
Indicates DFSORT is using a subtask to process the Fn (F1 or F2) file for a JOINKEYS application. ddname1 is the ddname associated with the input file for the subtask. ddname2 is the ddname associated with the message data set for the subtask.

System action
None.

Programmer response
See the messages in the indicated ddname2 data set for information about the subtask.

ICE417I
THIS IS THE JOINKEYS Fn SUBTASK FOR ddname

Explanation
Indicates this is a subtask used to process the Fn (F1 or F2) file for a JOINKEYS application. The subtask passes the needed Fn fields to the main task. ddname is the ddname for the input file associated with the subtask.

System action
None.

Programmer response
None.

ICE418A
JOINKEYS Fn SUBTASK FOR ddname1 TERMINATED - SEE ddname2 MESSAGES

Explanation
Critical. This message is issued by the main task of a JOINKEYS application to indicate that the subtask used to process file Fn (F1 or F2) terminated. ddname1 is the ddname associated with the input file for the subtask. ddname2 is the ddname associated with the message data set for the subtask.

System action
The program terminates.

Programmer response
Supply a message data set using a DD statement for the indicated ddname.

ICE419I
JOINED RECORDS: TYPE=x, LENGTH=n

Explanation
Indicates the record type (F or V) and maximum record length (n) of the joined records passed as input to the main task of a JOINKEYS application.

System action
None.

Programmer response
None.

ICE420A
COULD NOT ALLOCATE ddname FOR Fn MESSAGES - SUPPLY DD STATEMENT

Explanation
Critical. For a JOINKEYS application, a required message data set was not found and could not be dynamically allocated. A DD statement for the indicated ddname was required for the messages associated with the subtask for the F1 or F2 file. A DD statement for that ddname was not found, so DFSORT attempted to dynamically allocate a SYSOUT=* message data set. However, the message data set could not be allocated.

System action
The program terminates.

Programmer response
Supply a message data set using a DD statement for the indicated ddname.

ICE421I
JOINED RECORDS: COUNT=n

Explanation
Indicates the number (n) of joined records passed as input to the main task of a JOINKEYS application.

System action
None.

Programmer response
None.
ICE422I  JOINKEYS STATEMENT FOR Fn FOUND PREVIOUSLY - THIS STATEMENT IGNORED

Explanation
A JOINKEYS statement for the indicated file (F1 or F2) was found previously in this source (for example, SYSIN) or in a higher source (for example, DFSPARM is higher than SYSIN). The second and subsequent JOINKEYS statement for F1 or F2 in the same source or in a lower source is ignored.

System action
None.

Programmer response
Correct the conflicting JOINKEYS statements, if appropriate.

ICE423A  REFORMAT STATEMENT FIELD ENDS AFTER POSITION 32767

Explanation
Critical. The last byte of a field (p,m) in a REFORMAT FIELDS operand ended beyond position 32767. Each field must end at or before position 32767 (position plus length must not be greater than 32768). For example, 32767,1 is valid because it ends at position 32767, but 32767,2 is invalid because it ends at position 32768.

System action
The program terminates.

Programmer response
Ensure that each field in the REFORMAT FIELDS operand ends at or before position 32767.

ICE424A  ddname (Fn) KEY IS OUT OF SEQUENCE

Explanation
Critical. The SORTED operand was specified without the NOSEQCK operand on the JOINKEYS statement associated with the indicated ddname for file F1 or F2. DFSORT checked the records of the indicated file and found a record out of sequence for the keys specified in the FIELDS operand of the JOINKEYS statement.

System action
The program terminates.

Programmer response
Remove the SORTED operand from the JOINKEYS statement for the indicated file (F1 or F2) to force DFSORT to sort that file by the specified keys.

ICE425A  ddname CANNOT BE USED AS DDNAME FOR BOTH F1 AND F2

Explanation
Critical. The indicated ddname was used on the JOINKEYS statements for both file F1 and F2 (for example, one JOINKEYS statement has F1=IN1 and the other JOINKEYS statement has F2=IN1). Different ddnames must be used for the two files.

System action
The program terminates.

Programmer response
Change the F1, F2 or FILE operand in one or both of the JOINKEYS statements to use different ddnames for the two files.

ICE426A  cccc CANNOT BE USED AS PREFIX FOR JOINKEYS MAIN TASK AND SUBTASK

Explanation
Critical. For a JOINKEYS application, the SORTDD value for the main task and the TASKID value for a subtask resulted in the same four character prefix indicated by cccc (for example, SORTDD=MYF1 and TASKID=MY both result in a prefix of MYF1). Different prefixes must be used for the main task and the subtasks.

System action
The program terminates.

Programmer response
Change the SORTDD operand for the main task, or the TASKID operand for the subtask, to use different prefixes.

ICE427A  verb STATEMENT CANNOT BE USED WITH JOINKEYS SUBTASK

Explanation
Critical. For a JOINKEYS application, a JOINKEYS, JOIN, MERGE, OUTFIL, OUTFILE, OUTREC, REFORMAT or SORT statement was found in JNF1CNTL for subtask1 or in JNF2CNTL for subtask2. These statements cannot be used for a JOINKEYS subtask.

Programmer response
Change the SORTDD operand for the main task, or the TASKID operand for the subtask, to use different prefixes.
**System action**
The program terminates.

**Programmer response**
Remove all statements that cannot be used for a JOINKEYS subtask from JNF1CNTL or JNF2CNTL.

**ICE428A** TOO MANY DUPLICATES OF ONE KEY IN ddname (F2)

**Explanation**
Critical. The F2 file associated with the indicated ddname contained more duplicates for a single key than DFSORT could process with the storage available.

**System action**
The program terminates.

**Programmer response**
If possible, increase the storage available to DFSORT (for example, try specifying REGION=0M). Alternatively, if the maximum duplicates for a single key in file F1 is less than the maximum duplicates for a single key in file F2, reverse F1 and F2. For example, if you received this error with these control statements:

```
JOINKEYS F1=IN1,FIELDS=(11,4,A)
JOINKEYS F2=IN2,FIELDS=(21,4,A)
REFORMAT FIELDS=(F1:1,20,F2:5,30)
```

and the maximum duplicates for a single F1 key is less than the maximum duplicates for a single F2 key, use these control statements instead:

```
JOINKEYS F1=IN2,FIELDS=(21,4,A)
JOINKEYS F2=IN1,FIELDS=(11,4,A)
REFORMAT FIELDS=(F2:1,20,F1:5,30)
```

**ICE429A** JOINKEYS APPLICATION IS ONLY ALLOWED WITH SORT OR COPY OPERATOR

**Explanation**
Critical. A JOINKEYS, JOIN or REFORMAT statement was specified for an ICETOOL operator other than SORT or COPY. A JOINKEYS application is only allowed for a SORT or COPY operator, not for any of the other operators.

**System action**
The program terminates.

**ICE500A** UNICODE CALLABLE SERVICE UNAVAILABLE

**Explanation**
Critical. A SORT or MERGE control statement with Unicode data format(UTF8/UTF16/UTF32) control fields was used, but Unicode callable service is not available or not active at run-time.

**System action**
The program terminates.

**Programmer response**
Contact your system programmer to ensure that Unicode callable service has been installed or activated.

**ICE501A** FUNCTION CAN NOT BE USED WITH UNICODE PROCESSING

**Explanation**
Critical. Unicode processing cannot be used with the function or feature indicated by the ffffffff shown. See SORT/MERGE statement notes in z/OS DFSORT Application Programming Guide for restrictions on functions or operands.

**System action**
The program terminates.

**Programmer response**
Redesign the application to not use the function or operand that conflicts with the usage of Unicode data formats (UTF8/UTF16/UTF32).

**ICE502A** INSUFFICIENT MAIN STORAGE FOR UNICODE PROCESSING - ADD AT LEAST nK BYTES

**Explanation**
Critical. There was not enough main storage available for DFSORT to run with Unicode system services, 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.

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.

ICE503A  INVALID UNICODE CHARACTER.  RC =nn, RSN =rsn, CONTROL STATEMENT =ffffffff, CONTROL FIELD POSITION =ppppp

Explanation

Critical. DFSORT or Unicode collation service have found an invalid Unicode characters in the specified SORT/MERGE fields.

nn is the return code associated with the error. The return code helps you determine the specific cause of the error. Return code values (rtn) are as follows:

4 – Invalid Unicode characters

For additional return codes See Appendix H. Unicode return and reason codes for Explanation: of the return code and the reason code in z/OS Unicode Services User's Guide and Reference.

rsn is the reason code associated with the error. The reason code helps you determine the specific cause of the error. Reason code values (rsn) are as follows:

1. Invalid first subfield (UCA version) in COLLKEY value. UCA version can be as follows: UCA600, UCA410, or UCA400R1.

2. Some invalid subfields in COLLKEY value.

3. Invalid Locale/Region subfields in COLLKEY value.

4. Subfields of COLLKEY value are overlaid or appears more than once.

For additional return codes See Appendix H. Unicode return and reason codes for Explanation: of the return code and the reason code in z/OS Unicode Services User's Guide and Reference.

vvvv Specified COLLKEY

System action

The program terminates.

Programmer response

Ensure that the specified SORT/MERGE fields contains valid Unicode characters.

ICE504A  INVALID COLLKEY VALUES.  RC =nn, RSN =rsn, COLLKEY=vvvv

Explanation

Critical. DFSORT or Unicode collation service found an invalid COLLKEY value.

nn is the return code associated with the error. The return code helps you determine the specific cause of the error. Return code values (rtn) are as follows:

8 – Invalid COLLKEY value

For additional return codes See Appendix H. Unicode return and reason codes for Explanation: of the return code and the reason code in z/OS Unicode Services User's Guide and Reference.

vvvv Specified COLLKEY

System action

The program terminates.

Programmer response

Specify a valid COLLKEY value. See Appendix E. Locales for collation support for supported locales in z/OS Unicode Services User's Guide and Reference.

ICE505A  UNICODE SERVICES ERROR.  RC =nn, RSN =rsn

Explanation

Critical. Unicode collation services returned an error for a service call used by DFSORT.

nn

Return code from Unicode service call
Return code from Unicode service call

System action
The program terminates.

Programmer response
See Appendix H, Unicode return and reason codes for Explanation: of the return code and the reason code in z/OS Unicode Services User’s Guide and Reference

ICE506I  ACTIVE COLLKEY value WAS USED FOR UNICODE PROCESSING

Explanation
Indicates that DFSORT used Unicode processing and gives the name of the active collation version used.

System action
None

Programmer response
None

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.

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(SYSX1).
- 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 VLENONLY 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') instead of
  ON(VALCNT,E'999-999-999-999-999-9')
- 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**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|MERGE} FROM {ddname1|JKFROM|MERGEIN} TO {ddname2|E35 EXIT|OUTFIL} USING xxxxCNTL {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.

- **MERGE**
  - The DFSORT merge function was used.

- **ddname1**
  - The ddname of the data set that was copied or sorted.

- **JKFROM**
  - A JOINKEYS application was used.

- **MERGEIN**
  - A MERGE operator was used.

- **ddname2**
  - The ddname of the data set resulting from the copy, sort or merge.

- **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.

- **xxxxCNTL**
  - The ddname of the DFSORT control data set used for the copy, sort or merge.

- **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):**

nnnnnnnnnnnnnn

**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.

**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, or JKFROM and USING
  - TO or USING
- COUNT
  - FROM
  - EMPTY, NOTEMPTY, HIGHER, LOWER, EQUAL, or NOTEQUAL if RC4, RC8 or RC12 is specified
  - WRITE if TEXT, DIGITS, EDCOUNT, or WIDTH is specified
- DATASORT
  - FROM, TO, and USING
  - HEADER or TRAILER
- DEFAULTS
  - LIST
- DISPLAY
  - FROM, ON, and LIST
  - BREAK if BTITLE, BTOTAL, BMAXIMUM, BMINIMUM, BAVERAGE, or BCOUNT is specified
  - COUNT if EDCOUNT is specified
  - BCOUNT if EDBCOUNT is specified
- MERGE
  - FROM and USING
- MODE
  - STOP, CONTINUE, or SCAN
- OCCUR
  - FROM and LIST
  - ON(p,m,f), ON(p,m,HEX), or ON(VLEN)
- RANGE
  - FROM and ON
  - HIGHER, LOWER, EQUAL, or NOTEQUAL
- RESIZE
  - FROM, TO and TOLEN
- SELECT
  - FROM and ON
  - TO or DISCARD
  - ALLDUPS, NODUPS, HIGHER, LOWER, EQUAL, FIRST, LAST, FIRSTDUP or LASTDUP
- SORT
  - FROM or JKFROM
  - USING
- SPLICE
  - FROM, TO, ON, and WITH
- STATS
  - FROM and ON
- SUBSET
  - FROM
  - TO or DISCARD
  - KEEP or REMOVE
  - INPUT or OUTPUT
- 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, DATASORT, DEFAULTS, DISPLAY, MERGE, MODE, OCCUR (or OCCURS), RANGE, RESIZE, SELECT, SORT, SPLICE, STATS, SUBSET, 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 xxxxCNTL 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:
  nnnnnnnnnnnnnnnn

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 nnnnnnnnnnnnnnnn (prints as 15 decimal digits padded with zeros on the left as needed). For a DISPLAY, RANGE, STATS, or VERIFY operator, nnnnnnnnnnnnnnnn is the input record number (that is, the relative record number). For an OCCUR operator, nnnnnnnnnnnnnnnn 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(5,2001,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 50 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 TITLE operands is 3 for a DISPLAY or OCCUR operator.

The maximum number of RRN operands is 300 for a SUBSET operator.

The maximum number of FROM operands is 10 for a MERGE operator.

The maximum number of ON fields for each operator is:
• DISPLAY - 50
• OCCUR - 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
- DATASORT - 1
- MERGE - 10
- RESIZE - 1
- SELECT - 1
- SORT - 10
- SPLICE - 1
- SUBSET - 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.
The ddname of the data set that was copied or sorted.

The ddname of the data set resulting from the copy or sort.

ICETOOL's E35 user exit was used.

OUTFIL data sets were produced

DFSORT did not detect any errors preventing completion of this ICETOOL operation.

DFSORT detected an error which prevented completion of this ICETOOL operation.

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.

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 an ADD operand was specified for a COUNT operator, the count will reflect addition of the specified value. If a SUB operand was specified for a COUNT operator, the count will reflect subtraction of the specified value, but will not be reduced below 0.

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.

None.

A $ marks the point at which the error was detected. Specify values for HIGHER and LOWER that define a valid range.

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.

None.

If SCAN mode was set due to an error while in STOP mode, correct the error.

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).
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.
Explanation
Critical.
• For a DISPLAY or OCCUR operator without NOCC: The calculated record length for the indicated list data set was greater than 8192 with LONGLINE, greater than 2048 without LONGLINE, or greater than the maximum width specified by WIDTH(m). n is the total bytes required in the list data set record for the carriage control character, the title lines (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 the maximum allowed value if WIDTH was not specified.
• For a DISPLAY or OCCUR operator with NOCC:
  – WIDTH(8192) was specified. This exceeds the maximum width of 8191 allowed with NOCC. n is 8192 and m is 8191.
  – The calculated record length for the indicated list data set was greater than 8191 with LONGLINE, greater than 2047 without LONGLINE, or greater than the maximum width specified by WIDTH(m). n is the total bytes required in the list data set record for the title lines (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 the maximum allowed value if WIDTH was not specified.
• For a COUNT operator with WRITE: The calculated record length for the indicated output data set was greater than the maximum width specified by WIDTH(m). n is the total bytes required in the output data set record for the count line (resulting from specified TEXT, DIGITS and EDCOUNT operands). m is the value specified for the WIDTH operand.

System action
This operation is terminated.

Programmer response
• For a DISPLAY or OCCUR operator:
  If WIDTH is specified, remove the WIDTH operand and let ICETOOL set the width automatically, or if you need to set the WIDTH explicitly, increase its value to n or greater.

ICE638I NUMBER OF RECORDS RESULTING
FROM CRITERIA:

nnnnnnnnnnnnnnn

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.
**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), ON(NUM,formatting), EDCOUNT(formatting), or EDBCOUNT(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, or an EDCOUNT(formatting) operand for this COUNT 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), EDCOUNT(formatting), or EDBCOUNT(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).
- dd for Udd was greater than 15 for EDCOUNT(formatting) or EDBCOUNT(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 xxxx AND module RELEASE yyyy DO NOT MATCH**

**Explanation**
Critical. The release level (xxxx) of DFSORT does not match the release level (yyyy) of the indicated module. If the module is 'ICEPRML', that module, which is used to process ICEPRMxx members in PARMLIB was not at the same release level as the DFSORT modules being used for the run. If the module is 'ICEAM1', 'ICEAM2', 'ICEAM3', 'ICEAM4', 'ICETD1', 'ICETD2', 'ICETD3' or 'ICETD4', the ICEMAC macro used to compile that installation module was not at the same release level as the DFSORT modules being used for the run.

**Note:** This message is only printed for ICEPRML or for the first installation module with a release level that does not match the DFSORT release level.

**System action**
This operation is terminated.

**Programmer response**
Have your System Programmer ensure that the release level of the DFSORT modules matches the release level of the indicated ICEPRML or installation module.

---

**ICE643I WIDTH OF REPORT IS n BYTES**

**System action**
This operation is terminated.
**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
- 121 if NOCC and WIDTH(n) were not specified and the calculated line length was less than or equal to 121
- the calculated line length if NOCC and WIDTH(n) were not specified and the calculated line length was greater than 121
- 120 if NOCC was specified, WIDTH(n) was not specified, and the calculated line length was less than or equal to 120
- the calculated line length if NOCC was specified, WIDTH(n) was not specified, and the calculated line length was greater than 120.

**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)|COUNT} OVERFLOWED n DECIMAL DIGITS

**Explanation**
Critical. The number of digits (n) allowed from the Ndd, Udd, or DIGITS(n) 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.
- COUNT indicates a count for this COUNT operator, or an overall count or break count for this DISPLAY 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, or a DIGITS(n) operand, large enough to prevent overflow of the record number, value count, overall count or break count (that is, use an appropriate Ndd, Udd, or DIGITS value between n+1 and 15).

ICE646A RECORD COUNT MEETS CRITERIA - RC=nn 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=08 (if RC8 was specified) or RC=12 (if RC12 was specified or defaulted) 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:
          snnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnn
          nnnn

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
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 DATASORT, RESIZE, SELECT, SPLICE, or
SUBSET operator, but you cannot use an OUTREC statement with DATASORT, RESIZE, SELECT, SPLICE, or SUBSET.

System action
The operation is terminated.

Programmer response
If you want to reformat the output records produced by this DATASORT, RESIZE, SELECT, SPLICE, or SUBSET operator, use one or more OUTFIL statements as explained in z/OS DFSORT Application Programming Guide, instead of an OUTREC statement.

ICE653A STOPAFT, SKIPREC OR COND (SUBSET) OR STOPAFT (DATASORT) NOT ALLOWED

Explanation
Critical. Either:
• a SUBSET operator was used and a DFSORT SKIPREC, STOPAFT or COND operand was specified, or
• a DATASORT operator was used and a DFSORT STOPAFT operand was specified.

You cannot use STOPAFT with SUBSET or DATASORT. You cannot use SKIPREC or COND with SUBSET.

System action
The operation is terminated.

Programmer response
If appropriate, use an OUTFIL statement with INCLUDE, OMIT, STARTREC, ENDREC, ACCEPT or other operands to remove unwanted records after they are processed by SUBSET or DATASORT.

ICE654A SORT FUNCTION IS REQUIRED FOR DATASORT OPERATION

Explanation
Critical. A DFSORT SORT statement was not found for this DATASORT operator, but you must supply a SORT statement with DATASORT.

System action
The operation is terminated.

Programmer response
Specify a SORT statement in the xxxxCNTL data set corresponding to the USING(xxxx) operand for this DATASORT operator. Ensure that the SORT statement is not overridden by an OPTION COPY statement.

ICE655I COUNT RECORD WRITTEN IN ddname DATA SET - LENGTH IS n BYTES

Explanation
The record containing the count was written in the output data set with the indicated ddname. n indicates the record length and the LRECL for the output data set, determined as follows:
• n if WIDTH(n) was specified
• the calculated length of the count record if WIDTH(n) was not specified.

System action
None.

Programmer response
None.

ICE656A DISCARD AND INPUT CANNOT BE USED WITH SORT FUNCTION FOR SUBSET OPERATION

Explanation
Critical. A DFSORT SORT statement was found for a SUBSET operator with DISCARD(savedd) and INPUT operands, but you cannot use a SORT statement with these SUBSET operands.

System action
This operation is terminated.

Programmer response
Remove the SORT statement or DISCARD(savedd) operand, or change the INPUT operand to an OUTPUT operand, as appropriate. Alternatively, use two SUBSET operators, one with REMOVE and INPUT operands, a SORT statement, and no DISCARD(savedd) operand, and the other with KEEP and INPUT operands, a SORT statement, and no DISCARD(savedd) operand.

ICE657A TOO MANY FROM DDNAMES FOR MERGE

Explanation
Critical. The maximum of 50 FROM ddnames was exceeded for this MERGE operation.
System action
This operation is terminated.

Programmer response
A $ marks the point at which the error was detected. Reduce the number of FROM ddnames for this MERGE operator to 50 or less. Use additional MERGE operators to handle all of the data sets required.

ICE658A MERGE FUNCTION IS REQUIRED FOR MERGE OPERATION

Explanation
Critical. A DFSORT MERGE statement was not found for this MERGE operator, but you must supply a MERGE statement with a MERGE operator

System action
This operation is terminated.

Programmer response
Specify a MERGE statement in the xxxxCNTL data set corresponding to the USING(xxxx) operand for this MERGE operator. Ensure that the MERGE statement is not overridden by a SORT statement, an OPTION COPY statement, or a MERGE FIELDS=COPY statement.

ICE659I RECORDS RESIZED FROM n BYTES TO m BYTES

Explanation
For this RESIZE operator, n indicates the original size of the input records, and m indicates the resulting size of the output records.

System action
None.

Programmer response
None.

ICE660A FROM DATA SET MUST HAVE FIXED LENGTH RECORDS

Explanation
Critical. For this RESIZE operator, the input data set associated with the FROM ddname has variable-length records rather than fixed-length records.

System action
This operation is terminated.

ICE661A TOLEN MUST NOT BE THE SAME AS FROM DATA SET RECORD LENGTH

Explanation
Critical. For this RESIZE operator, the output length specified by TOLEN is the same as:

- The reformatted record length if INREC is used.
- The record length of the input data set associated with the FROM ddname if INREC is not used.

A RESIZE operation requires a FROM record length different from TOLEN.

System action
This operation is terminated.

Programmer response
Specify a TOLEN value that is larger or smaller than the FROM record length.

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.
Informational and Error Messages

**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 pppp OF EFS PROGRAM pppppppp**

**Explanation**
Critical. The pppp-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.

**ICE705A**
**EFS PROCESSING WAS TERMINATED BY pppp OF EFS PROGRAM pppppppp**

**Explanation**
Critical. Termination of the DFSORT run was requested by a user exit routine generated by EFS program pppppppp.

pppp 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 xxxxCNTL, 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 xxxxCNTL. (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 and 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 aaaaaaaaa CONTROL STATEMENT WAS NOT SENT TO EFS PROGRAM ppppppppp

**Explanation**
Critical. The control statement aaaaaaaaa requested by EFS program ppppppppp 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 xxxxCNTL - PARAMETER LIST CONTROL STATEMENTS FOLLOW

**Explanation**
When returned control statements are specified in the invoking parameter list and with the DD statement for xxxxCNTL, 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 ddbname - SYSIN OR xxxxCNTL/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 
modesules 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.

**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-bbbbbbb ...

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.

ICE754I a b

Explanation
DFSORT diagnostic message for use by the IBM Support Center.

ICE755I a b c

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 rsn

**Explanation**
DFSORT's most efficient technique, Blockset, could not be used for this application. rsn is the reason code associated with the error. The reason code helps you determine the specific cause of the error. Reason code values (rsn) are as follows:

1. A critical message was issued.
2. BDAM was indicated in the SORTIN DD statement.
3. BDAM was indicated in the SORTIN data set label.
4. BDAM was indicated in the SORTOUT DD statement.
5. A DUMMY or spool SORTWKdd data set was specified.
6. An attempt to OPEN a data set caused a system error.
7. The SORTIN or SORTOUT data set resided on an unsupported device.
8. A SORTWKdd data set resided on an unsupported device (such as tape).
9. 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.
10. An attempt to read the DSCB for the SORTIN data set caused an error.
11. An attempt to read the DSCB for the SORTOUT data set caused an error.
12. An E18 or E38 user exit was specified with a tape SORTIN(nn) data set.
13. The NOBLKSET option was specified.
14. The CKPT run-time option and the IGNCKPT=NO installation option were specified.
15. Dynamic tape SORTWKdd data sets were requested.
16. Too many control fields were specified or control fields were too large.
17. Too many control fields were specified or control fields were too large.
18. An E61 user exit was specified and VLSHRT was in effect.
19. 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:** 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 (rsn) so Blockset can be used.

ICE8011 a: b c

**Explanation**
DFSORT diagnostic message for use by the IBM Support Center.
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.

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 Snnn 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.
ICE821I  BN=a X=b TO=c SN=d G=e

ICE822I  BN=a X=b G=c PN=d BT=e TO=f

ICE823I  BN=a X=b G=c TO=d BT=e

ICE824I  PE=a RP=b CX=c CO=d C0=e CR=f G=g WB=h

ICE825I  GP=a SA=b X=c

ICE826I  BN=a X=b RM=c

ICE827I  RD=a BL=b BY=c

ICE828I  a b c

ICE851I  a, LA=b, DA=c, LB=d, DB=e, MB=f, CB=g, UB=h

ICE852I  LA=a, DA=b, AA=c, BA=d, CP=e, TA=f

ICE853I  LB=a, DB=b, AB=c, BB=d, CP=e, RS=f, TB=g

ICE854I  F=a, P=b, M=c, B=d

ICE855I  QP=a QA=b HI=c LI=d MI=e TZ=f N1=g N2=h SZ=i HN=j

ICE856I  EQ=a DX=b D2=c D3=d D4=e AS=f SA=g SB=h SC=i HN=j

ICE857I  IX=a, IFC=b, INC=c, OX=d, OFC=e, OMC=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, CCW=hijk

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, i CZ, j DZ, k PLE

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, ENQT=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, ENQT=c, CMAX=d, HU=e, BUN=f, MD=g, h, DU=i, DR=j, HN=k

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
<table>
<thead>
<tr>
<th>Code</th>
<th>Message</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICE901I</td>
<td>a b . . .</td>
<td>DFSORT diagnostic message for use by the IBM Support Center.</td>
</tr>
<tr>
<td>ICE902I</td>
<td>0 a I b c . . .</td>
<td>DFSORT diagnostic message for use by the IBM Support Center.</td>
</tr>
<tr>
<td>ICE905I</td>
<td>a : RF=b, LR=c, BLK=d, BCT=e</td>
<td>DFSORT diagnostic message for use by the IBM Support Center.</td>
</tr>
<tr>
<td>ICE915I</td>
<td>MOFSZ=a, MOSZ=b, MOSYS=c(d), MOSTG=e, MEML=f(g)</td>
<td>DFSORT diagnostic message for use by the IBM Support Center.</td>
</tr>
<tr>
<td>ICE916I</td>
<td>MOFR=a, MOVR=b</td>
<td>DFSORT diagnostic message for use by the IBM Support Center.</td>
</tr>
<tr>
<td>ICE920I</td>
<td>Z a S1 b S2 c EN d ZI e</td>
<td>DFSORT diagnostic message for use by the IBM Support Center.</td>
</tr>
<tr>
<td>ICE921I</td>
<td>WLM aaaaaaaaa RC=b, RSN=cccccccc</td>
<td>DFSORT diagnostic message for use by the IBM Support Center.</td>
</tr>
<tr>
<td>ICE992I</td>
<td>RA a WR b TR c</td>
<td>DFSORT diagnostic message for use by the IBM Support Center.</td>
</tr>
<tr>
<td>ICE995I</td>
<td>PN=a, HN=b</td>
<td>DFSORT diagnostic message for use by the IBM Support Center.</td>
</tr>
<tr>
<td>ICE996I</td>
<td>ESM=a, ESO=b, ESR=c, ESP=d, ESS=e, CES=f, HSZ=g</td>
<td>DFSORT diagnostic message for use by the IBM Support Center.</td>
</tr>
<tr>
<td>ICE997I</td>
<td>HWSP=a, HMAX=b, MOMAX=c, ASV=d, EQ=e, HN=f</td>
<td>DFSORT diagnostic message for use by the IBM Support Center.</td>
</tr>
<tr>
<td>ICE999I</td>
<td>PWK=a PSP=b SWK=c SSP=d TWK=e TSP=f RWK=g RSP=h AWK=i AWP=j</td>
<td>DFSORT diagnostic message for use by the IBM Support Center.</td>
</tr>
</tbody>
</table>
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, 20 or 28 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 ABCODE installation option (either the error message number or a number between 1 and 99).

For additional information on DFSORT User Abends, see “User abends at run time” on page 8.

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. Installation 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. This operating system is not supported by this release of DFSORT.

28  Wrong entry name. DFSORT detected one of the following errors related to the program entry name that prevented it from completing successfully:
   • DFSORT was invoked directly with PGM=ICEMAN64 or PGM=SORT64 (instead of with PGM=ICEMAN or PGM=SORT).
   • DFSORT was invoked from a program using entry name ICEMAN64 or SORT64, and an extended invocation parameter list, or a 24-bit invocation parameter list (instead of a 64-bit invocation parameter list).
   • DFSORT was invoked from a program using an entry name other than ICEMAN64 or SORT64 (for example, ICEMAN or SORT) and a 64-bit invocation parameter list.
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.

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 8, 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 131 for details.
8  Unsuccessful completion. RC8 was specified for an ICETOOL COUNT operator and the record count met the specified criteria.
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. 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 ABCODE installation option (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:
   • Installation option GNPAD=RC4 was specified and the SYSUT2 LRECL was larger than the SYSUT1 LRECL (LRECL padding), or
   • Installation option GNTRUNC=RC4 was specified 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. This operating system is not supported by this release of DFSORT.
Part 3. Diagnosis guide
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 5 on page 137 to resolve a program failure.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>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.</td>
</tr>
<tr>
<td>2</td>
<td>Follow the instructions in “Developing your keyword string” on page 137. 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.</td>
</tr>
<tr>
<td>3</td>
<td>Follow the instructions in Chapter 7, “Searching the problem reporting databases,” on page 153. 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.</td>
</tr>
<tr>
<td>4</td>
<td>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 8, “Fixing or bypassing the problem,” on page 155.</td>
</tr>
<tr>
<td>5</td>
<td>If the problem has not been reported, follow the instructions in Chapter 9, “Reporting a problem,” on page 157, to report the problem.</td>
</tr>
</tbody>
</table>

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 7, “Searching the problem reporting databases,” on page 153. 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
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 139 and Figure 2 on page 140 show 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 140 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)
**Constructing the Keyword String Correctly**

![Flowchart](image)

**Constructing keyword strings**

Use Figure 3 on page 141 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.

<table>
<thead>
<tr>
<th>Wait/Loop</th>
<th>0 5740SM105 Rxxx</th>
<th>BLOCKSET</th>
<th>PERVAIE</th>
<th>CONVENTIONAL</th>
<th>SORT</th>
<th>MERGE</th>
<th>COPY</th>
<th>WAIT</th>
<th>LOOP</th>
<th>ICEname</th>
<th>OFFSETn</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Abend</td>
<td>0 5740SM105 Rxxx</td>
<td>BLOCKSET</td>
<td>PERVAIE</td>
<td>CONVENTIONAL</td>
<td>SORT</td>
<td>MERGE</td>
<td>COPY</td>
<td>ABENDnnn</td>
<td>ICEname</td>
<td>OFFSETn</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>User Abend</td>
<td>0 5740SM105 Rxxx</td>
<td>BLOCKSET</td>
<td>PERVAIE</td>
<td>CONVENTIONAL</td>
<td>SORT</td>
<td>MERGE</td>
<td>COPY</td>
<td>ABENDnnn</td>
<td>ICEname</td>
<td>OFFSETn</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Message (All Messages)</td>
<td>0 5740SM105 Rxxx</td>
<td>BLOCKSET</td>
<td>PERVAIE</td>
<td>CONVENTIONAL</td>
<td>SORT</td>
<td>MERGE</td>
<td>COPY</td>
<td>MSGICEEnnnA</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Message (ICE061A or ICE117A)</td>
<td>A</td>
<td>ddname</td>
<td>ECBnn</td>
<td>CSWnnnn</td>
<td>SENSEEnnn</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Incorrect Output</td>
<td>0 5740SM105 Rxxx</td>
<td>BLOCKSET</td>
<td>PERVAIE</td>
<td>CONVENTIONAL</td>
<td>SORT</td>
<td>MERGE</td>
<td>COPY</td>
<td>INCORROUT</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Performance</td>
<td>0 5740SM105 Rxxx</td>
<td>BLOCKSET</td>
<td>PERVAIE</td>
<td>CONVENTIONAL</td>
<td>SORT</td>
<td>MERGE</td>
<td>COPY</td>
<td>PERFM</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Documentation</td>
<td>0 5740SM105 Rxxx</td>
<td>DOC</td>
<td>order number</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Figure 3: Keyword String Syntax*
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:

<table>
<thead>
<tr>
<th>Keyword string</th>
</tr>
</thead>
<tbody>
<tr>
<td>5740SM105</td>
</tr>
</tbody>
</table>

3. Go to “FMID keyword procedure” on page 142, 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 V2R3 is HSM1N00.

2. Record the FMID keyword in the following format:

   ```
   Rccc
   ```

   where `ccc` is the last three characters of the FMID.

   For V2R3 problems, you now have the following keyword string:

<table>
<thead>
<tr>
<th>Keyword string</th>
</tr>
</thead>
<tbody>
<tr>
<td>5740SM105 RN00</td>
</tr>
</tbody>
</table>

N00 identifies the DFSORT base FMID HSM1N00.

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 147.

   If the problem is not in a DFSORT publication, go to “Type-of-technique keyword procedure” on page 142.

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:

```
ICEnnn c text
```

Where

- `nnn` Represents 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:

<table>
<thead>
<tr>
<th>Trace character</th>
<th>Keyword</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 through 9 or A through I</td>
<td>BLOCKSET</td>
</tr>
<tr>
<td>J through Z</td>
<td>PEERVALE</td>
</tr>
<tr>
<td>– (hyphen)</td>
<td>CONVENTIONAL</td>
</tr>
</tbody>
</table>

For example, if you received message ICE085A 1, the 1 means the Blockset technique. The keyword string so far would be:

```
5740SM105 RN00 BLOCKSET
```

2. Go to “Type-of-application keyword procedure” on page 143.

**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 RN00 BLOCKSET SORT
```

2. Go to “Type-of-failure keyword procedures” on page 144.
Type-of-failure keyword procedures

It is very important that you determine the type-of-failure keyword correctly. Use Table 6 on page 144 to find the procedure that best describes your problem.

<table>
<thead>
<tr>
<th>Type of Failure</th>
<th>Description</th>
<th>Messages and Codes</th>
<th>Procedure Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wait</td>
<td>Unexpected program suspension.</td>
<td>System completion code is n22 (where n is any hexadecimal digit).</td>
<td>&quot;Wait and loop keywords procedure&quot; on page 144.</td>
</tr>
<tr>
<td>Loop</td>
<td>Uncontrolled program repetition.</td>
<td>System completion code is n22 (where n is any hexadecimal digit).</td>
<td>&quot;Wait and loop keywords procedure&quot; on page 144.</td>
</tr>
<tr>
<td>System Abend</td>
<td>Abnormal termination of DFSORT.</td>
<td>System completion code is not n22 (where n is any hexadecimal digit), or zero.</td>
<td>&quot;System abend keyword procedure&quot; on page 145.</td>
</tr>
<tr>
<td>User Abend</td>
<td>Abnormal termination of DFSORT.</td>
<td>User completion code is not zero.</td>
<td>&quot;User abend keyword procedure&quot; on page 146.</td>
</tr>
<tr>
<td>Message</td>
<td>Errors identified by or associated with DFSORT error messages.</td>
<td>Error message is in the form ICEnnnA.</td>
<td>&quot;Message keyword procedure&quot; on page 146.</td>
</tr>
<tr>
<td>Incorrect Output</td>
<td>Unexpected, missing, or incorrect output not associated with a message.</td>
<td>Not applicable</td>
<td>&quot;Incorrect-output keyword procedure&quot; on page 147.</td>
</tr>
<tr>
<td>Performance</td>
<td>Performance degradation. (Use this keyword only when you are sure that this is not a wait or loop problem.)</td>
<td>Not applicable</td>
<td>&quot;Performance keyword procedure&quot; on page 147.</td>
</tr>
<tr>
<td>Documentation</td>
<td>Severe incorrect or missing information in DFSORT documentation.</td>
<td>Not applicable</td>
<td>&quot;Document keyword procedure&quot; on page 147.</td>
</tr>
</tbody>
</table>

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 4 on page 145.
Figure 4: 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 5 on page 145).

Figure 5: 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 5 on page 145, you would have the following keyword string:

Keyword string

```
5740SM105 RN00 BLOCKSET SORT LOOP
```

5. The next keyword is the module’s name. Go to “Module keyword procedure” on page 148.

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 144.

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:

<table>
<thead>
<tr>
<th>Keyword string</th>
</tr>
</thead>
<tbody>
<tr>
<td>5740SM105 RN00 BLOCKSET SORT ABEND0C1</td>
</tr>
</tbody>
</table>

2. The next keyword is the module’s name. Go to “Module keyword procedure” on page 148.

**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 146.
   - 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:

<table>
<thead>
<tr>
<th>Keyword string</th>
</tr>
</thead>
<tbody>
<tr>
<td>5740SM105 RN00 BLOCKSET SORT ABEND1004</td>
</tr>
</tbody>
</table>

3. The next keyword is the module’s name. Go to “Module keyword procedure” on page 148.

**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:

<table>
<thead>
<tr>
<th>ICEnnnA c text</th>
</tr>
</thead>
</table>

   **Where**
   - Represents
   - nnnn: 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:

```
Keyword string
5740SM105 RN00 BLOCKSET SORT MSGICE061A
```

3. If you received message ICE061A or ICE117A, go to “Message variable keywords” on page 150.

If you did not receive either message ICE061A or ICE117A, your message keyword string is now complete. Go to Chapter 7, “Searching the problem reporting databases,” on page 153.

**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:

```
Keyword string
5740SM105 RN00 BLOCKSET SORT INCORROUT
```

2. The incorrect-output keyword string is now complete. Go to Chapter 7, “Searching the problem reporting databases,” on page 153.

**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:

```
Keyword string
5740SM105 RN00 BLOCKSET SORT PERFM
```

3. The performance keyword string is now complete. Go to Chapter 7, “Searching the problem reporting databases,” on page 153.

**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:

<table>
<thead>
<tr>
<th>Keyword string</th>
</tr>
</thead>
<tbody>
<tr>
<td>5740SM105 RN00 DOC</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Keyword string</th>
</tr>
</thead>
<tbody>
<tr>
<td>5740SM105 RN00 DOC SC23687902</td>
</tr>
</tbody>
</table>

3. The Document keyword string is now complete. Go to Chapter 7, “Searching the problem reporting databases,” on page 153.

**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 6 on page 148 for an example. The last 31 bits of the PSW (that is, 22D417F2) is the instruction address.

<table>
<thead>
<tr>
<th>JOB SORT1</th>
<th>STEP S1</th>
<th>TIME 150902</th>
<th>DATE 07085</th>
<th>ID = 000</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMPLETION CODE</td>
<td>SYSTEM = 0C1</td>
<td>REASON CODE = 00000001</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PSW AT ENTRY TO ABEND</td>
<td>078D2000</td>
<td>A2D417F2</td>
<td>ILC 02</td>
<td>INTC 0001</td>
</tr>
<tr>
<td>↑</td>
<td>↑</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>instruction length code</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>instruction address</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

   Figure 6: Finding the Abend Address

2. Obtain the abend address by subtracting the instruction length code (ILC) from the instruction address.

   \[ \text{abend address} = \text{instruction address} - \text{ILC} \]

   \[ \begin{align*}
   22D417F2 & \quad \text{instruction address} \\
   - 2 & \quad \text{instruction length code} \\
   \hline
   22D417F0 & \quad \text{abend address}
   \end{align*} \]

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 Z/OSv.rr
where ICEname is the module name and Z/OSv.rr is the version and release.

In Figure 7 on page 149, ICEIPFT Z/OS2.03 is the closest module name, version, and release.

Figure 7: Finding the Module Name

If you cannot find the module name, the problem might not be with DFSORT. In this case, call your IBM Support Center.

4. Use the module name as the keyword. For this example, the keyword string would be:

<table>
<thead>
<tr>
<th>Keyword string</th>
</tr>
</thead>
<tbody>
<tr>
<td>5740SM105 RN00 BLOCKSET SORT ABEND0C1 ICEIPFT</td>
</tr>
</tbody>
</table>

Check to see that the version and release that appears after the module name is current. If the version and release is not current, ensure DFSORT has been installed correctly.

5. The next keyword shows the offset. See Figure 8 on page 150.

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 148). If 'LICENSED' does not appear, the first byte is the beginning of the module name.

<table>
<thead>
<tr>
<th>22D417F0</th>
<th>abend address</th>
</tr>
</thead>
<tbody>
<tr>
<td>22D41700</td>
<td>module address</td>
</tr>
<tr>
<td>F0</td>
<td>offset</td>
</tr>
</tbody>
</table>

In the example in Figure 8 on page 150, the address of the module ICEIPFT is 22D41700, and the abend address previously found is 22D417F0. Therefore, the offset is F0.
module address is 22D41700

abend address is 22D417F0 - offset is F0

Figure 8: 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:

```
Keyword string
```

```
5740SM105 RN00 BLOCKSET SORT ABEND0C1 ICEIPFT OFFSET F0
```

3. The keyword string is now complete. Go to Chapter 7, “Searching the problem reporting databases,” on page 153.

Message variable keywords

If your problem is for message ICE061A or ICE117A, you need to add a message variable keyword to your keyword string. Following 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. ECBnn (where nn is the completion code)
   c. CSWnnnn (where nnnn are the status bytes)
   d. SENSEnnnn (where nnnn are the sense bytes)

For example, if you received the message

```
ICE061A 01/0 ERROR, DD SORTWK04, DEV 6C0, ECB 41, CSW 0020, SENSE 0000
```

the keyword string would be:

```
Keyword string
```

```
5740SM105 RN00 BLOCKSET SORT MSGICE061A SORTWK ECB41 CSW0020 SENSE0000
```
2. The message keyword string is now complete. Go to Chapter 7, “Searching the problem reporting databases,” on page 153.

**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. OPnn (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, GE OP, UNKNOWN COND., 00000007000100, EXCP
```

the keyword string would be:

```
Keyword string
5740SM105 RN00 PEERVALE SORT MSGICE117A SORTOUT OP0E UNKNOWN COND
```

Because ICE117A is issued only for the Peerage/Vale technique, the type-of-technique keyword in this example is now PEERVALE.

2. The message keyword string is now complete. Go to Chapter 7, “Searching the problem reporting databases,” on page 153.
Chapter 7. 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>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>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.</td>
</tr>
<tr>
<td>2</td>
<td>If you find a set of APAR closing descriptions, compare the APAR closing descriptions in the IBM software support data base with your problem.</td>
</tr>
<tr>
<td>3</td>
<td>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 8, “Fixing or bypassing the problem,” on page 155.</td>
</tr>
</tbody>
</table>
| 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 9, “Reporting a problem,” on page 157. |
Chapter 8. Fixing or bypassing the problem

Fixing the problem
Apply the appropriate PTF for the 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 for z/OS 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 142.

Note: OPTION NOBLKSET should NOT be used as a permanent solution, because it can significantly degrade performance.
Chapter 9. Reporting a problem

Follow these instructions to report a problem to the IBM Support Center.

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Gather the information.</td>
</tr>
<tr>
<td></td>
<td>1. Have the following information ready when you report a problem:</td>
</tr>
<tr>
<td></td>
<td>• Customer number</td>
</tr>
<tr>
<td></td>
<td>• Current service level (PTF list and list of APAR fixes applied)</td>
</tr>
<tr>
<td></td>
<td>• The keyword string developed to search the IBM software support database</td>
</tr>
<tr>
<td></td>
<td>• Processor number (serial model)</td>
</tr>
<tr>
<td></td>
<td>2. Also have the following documents ready to assist in analyzing the problem:</td>
</tr>
<tr>
<td></td>
<td>• Source code (assembler or compiler listings) and load module for any user routines or invoking program.</td>
</tr>
<tr>
<td></td>
<td>• 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.</td>
</tr>
<tr>
<td></td>
<td>• A listing of the symbols information if a SYMNAMES DD statement was specified. You can produce this listing by specifying a SYMNOUT DD statement.</td>
</tr>
<tr>
<td></td>
<td>• A copy of the input data set in machine-readable form.</td>
</tr>
<tr>
<td></td>
<td>• 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.</td>
</tr>
<tr>
<td></td>
<td>• A dump from the failing run of DFSORT.</td>
</tr>
<tr>
<td>2</td>
<td>Contact the IBM Support Center.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td>3</td>
<td>Submit APAR documentation.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>Each tape must have the following information attached and visible:</td>
</tr>
<tr>
<td></td>
<td>• The APAR number assigned by IBM</td>
</tr>
<tr>
<td></td>
<td>• A list of data sets on the tape (for example, application source program, JCL, data, and load modules)</td>
</tr>
<tr>
<td></td>
<td>• Data on how the tape was created:</td>
</tr>
<tr>
<td></td>
<td>– The exact JCL listing or the list of commands used</td>
</tr>
<tr>
<td></td>
<td>– The record format and block size used for each data set</td>
</tr>
<tr>
<td></td>
<td>– Tape labeling</td>
</tr>
<tr>
<td></td>
<td>– The recording mode and density</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Option</th>
<th>E18</th>
<th>E19</th>
<th>E38</th>
<th>E39</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYNAD</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>EXLST</td>
<td>x¹</td>
<td></td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>EROPT</td>
<td></td>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EODAD</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BSAM EXLST</td>
<td>x</td>
<td></td>
<td>x²</td>
<td>x</td>
</tr>
<tr>
<td>VSAM PASSWORD</td>
<td>x</td>
<td></td>
<td>x²</td>
<td>x</td>
</tr>
</tbody>
</table>

Notes:
1. Cannot be used if input is concatenated on unlike devices.
2. For merge applications.
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>
<thead>
<tr>
<th>Enn</th>
<th>Case</th>
<th>EXITCK</th>
<th>Situation</th>
<th>Valid Return Codes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>E15</td>
<td>S1</td>
<td>STRONG</td>
<td>Without SORTIN</td>
<td>X</td>
</tr>
<tr>
<td>E15</td>
<td>W1</td>
<td>WEAK</td>
<td>Without SORTIN</td>
<td>X</td>
</tr>
<tr>
<td>E15</td>
<td>S2</td>
<td>STRONG</td>
<td>With SORTIN before EOI</td>
<td>X</td>
</tr>
<tr>
<td>E15</td>
<td>W2</td>
<td>WEAK</td>
<td>With SORTIN before EOI</td>
<td>X</td>
</tr>
<tr>
<td>E15</td>
<td>S3</td>
<td>STRONG</td>
<td>With SORTIN after EOI</td>
<td>X</td>
</tr>
<tr>
<td>E15</td>
<td>W3</td>
<td>WEAK</td>
<td>With SORTIN after EOI</td>
<td>X</td>
</tr>
<tr>
<td>E16</td>
<td>16</td>
<td>N/A</td>
<td>N/A</td>
<td>X</td>
</tr>
<tr>
<td>E32</td>
<td>32</td>
<td>N/A</td>
<td>N/A</td>
<td>X</td>
</tr>
<tr>
<td>E35</td>
<td>S0</td>
<td>STRONG</td>
<td>Without SORTOUT or OUTFIL before EOI</td>
<td>X</td>
</tr>
<tr>
<td>E35</td>
<td>W0</td>
<td>WEAK</td>
<td>Without SORTOUT or OUTFIL before EOI</td>
<td>X</td>
</tr>
<tr>
<td>E35</td>
<td>S1</td>
<td>STRONG</td>
<td>Without SORTOUT or OUTFIL after EOI</td>
<td>X</td>
</tr>
<tr>
<td>E35</td>
<td>W1</td>
<td>WEAK</td>
<td>Without SORTOUT or OUTFIL after EOI</td>
<td>X</td>
</tr>
<tr>
<td>E35</td>
<td>S2</td>
<td>STRONG</td>
<td>With SORTOUT or OUTFIL before EOI</td>
<td>X</td>
</tr>
<tr>
<td>E35</td>
<td>W2</td>
<td>WEAK</td>
<td>With SORTOUT or OUTFIL before EOI</td>
<td>X</td>
</tr>
<tr>
<td>E35</td>
<td>S3</td>
<td>STRONG</td>
<td>With SORTOUT or OUTFIL after EOI</td>
<td>X</td>
</tr>
<tr>
<td>E35</td>
<td>W3</td>
<td>WEAK</td>
<td>With SORTOUT or OUTFIL after EOI</td>
<td>X</td>
</tr>
</tbody>
</table>

Note: * Only applies to COBOL E15 and COBOL E35
Appendix C. Accessibility

Accessible publications for this product are offered through IBM Knowledge Center (www.ibm.com/support/knowledgecenter/SSLTBW/welcome).

If you experience difficulty with the accessibility of any z/OS information, send a detailed email message to mhvrdfs@us.ibm.com.

Accessibility features

Accessibility features help users who have physical disabilities such as restricted mobility or limited vision use software products successfully. The accessibility features in z/OS can help users do the following tasks:

• Run assistive technology such as screen readers and screen magnifier software.
• Operate specific or equivalent features by using the keyboard.
• Customize display attributes such as color, contrast, and font size.

Consult assistive technologies

Assistive technology products such as screen readers function with the user interfaces found in z/OS. Consult the product information for the specific assistive technology product that is used to access z/OS interfaces.

Keyboard navigation of the user interface

You can access z/OS user interfaces with TSO/E or ISPF. The following information describes how to use TSO/E and ISPF, including the use of keyboard shortcuts and function keys (PF keys). Each guide includes the default settings for the PF keys.

• z/OS TSO/E Primer
• z/OS TSO/E User’s Guide
• z/OS ISPF User’s Guide Vol I

Dotted decimal syntax diagrams

Syntax diagrams are provided in dotted decimal format for users who access IBM Knowledge Center with a screen reader. In dotted decimal format, each syntax element is written on a separate line. If two or more syntax elements are always present together (or always absent together), they can appear on the same line because they are considered a single compound syntax element.

Each line starts with a dotted decimal number; for example, 3 or 3.1 or 3.1.1. To hear these numbers correctly, make sure that the screen reader is set to read out punctuation. All the syntax elements that have the same dotted decimal number (for example, all the syntax elements that have the number 3.1) are mutually exclusive alternatives. If you hear the lines 3.1 USERID and 3.1 SYSTEMID, your syntax can include either USERID or SYSTEMID, but not both.

The dotted decimal numbering level denotes the level of nesting. For example, if a syntax element with dotted decimal number 3 is followed by a series of syntax elements with dotted decimal number 3.1, all the syntax elements numbered 3.1 are subordinate to the syntax element numbered 3.
Certain words and symbols are used next to the dotted decimal numbers to add information about the syntax elements. Occasionally, these words and symbols might occur at the beginning of the element itself. For ease of identification, if the word or symbol is a part of the syntax element, it is preceded by the backslash (\) character. The * symbol is placed next to a dotted decimal number to indicate that the syntax element repeats. For example, syntax element *FILE with dotted decimal number 3 is given the format 3 \* FILE. Format 3* FILE indicates that syntax element FILE repeats. Format 3* \* FILE indicates that syntax element * FILE repeats.

Characters such as commas, which are used to separate a string of syntax elements, are shown in the syntax just before the items they separate. These characters can appear on the same line as each item, or on a separate line with the same dotted decimal number as the relevant items. The line can also show another symbol to provide information about the syntax elements. For example, the lines 5.1*, 5.1 LASTRUN, and 5.1 DELETE mean that if you use more than one of the LASTRUN and DELETE syntax elements, the elements must be separated by a comma. If no separator is given, assume that you use a blank to separate each syntax element.

If a syntax element is preceded by the % symbol, it indicates a reference that is defined elsewhere. The string that follows the % symbol is the name of a syntax fragment rather than a literal. For example, the line 2.1 %OP1 means that you must refer to separate syntax fragment OP1.

The following symbols are used next to the dotted decimal numbers.

? Indicates an optional syntax element
The question mark (?) symbol indicates an optional syntax element. A dotted decimal number followed by the question mark symbol (?) indicates that all the syntax elements with a corresponding dotted decimal number, and any subordinate syntax elements, are optional. If there is only one syntax element with a dotted decimal number, the ? symbol is displayed on the same line as the syntax element, (for example 5? NOTIFY). If there is more than one syntax element with a dotted decimal number, the ? symbol is displayed on a line by itself, followed by the syntax elements that are optional. For example, if you hear the lines 5 ?, 5 NOTIFY, and 5 UPDATE, you know that the syntax elements NOTIFY and UPDATE are optional. That is, you can choose one or none of them. The ? symbol is equivalent to a bypass line in a railroad diagram.

! Indicates a default syntax element
The exclamation mark (!) symbol indicates a default syntax element. A dotted decimal number followed by the ! symbol and a syntax element indicate that the syntax element is the default option for all syntax elements that share the same dotted decimal number. Only one of the syntax elements that share the dotted decimal number can specify the ! symbol. For example, if you hear the lines 2? FILE, 2.1! (KEEP), and 2.1 (DELETE), you know that (KEEP) is the default option for the FILE keyword. In the example, if you include the FILE keyword, but do not specify an option, the default option KEEP is applied. A default option also applies to the next higher dotted decimal number. In this example, if the FILE keyword is omitted, the default FILE(KEEP) is used. However, if you hear the lines 2? FILE, 2.1, 2.1.1! (KEEP), and 2.1.1 (DELETE), the default option KEEP applies only to the next higher dotted decimal number, 2.1 (which does not have an associated keyword), and does not apply to 2? FILE. Nothing is used if the keyword FILE is omitted.

* Indicates an optional syntax element that is repeatable
The asterisk or glyph (*) symbol indicates a syntax element that can be repeated zero or more times. A dotted decimal number followed by the * symbol indicates that this syntax element can be used zero or more times; that is, it is optional and can be repeated. For example, if you hear the line 5.1* data area, you know that you can include one data area, more than one data area, or no data area. If you hear the lines 3* , 3 HOST, 3 STATE, you know that you can include HOST, STATE, both together, or nothing.

Notes:
1. If a dotted decimal number has an asterisk (*) next to it and there is only one item with that dotted decimal number, you can repeat that same item more than once.
2. If a dotted decimal number has an asterisk next to it and several items have that dotted decimal number, you can use more than one item from the list, but you cannot use the items more than once each. In the previous example, you can write HOST STATE, but you cannot write HOST HOST.
3. The * symbol is equivalent to a loopback line in a railroad syntax diagram.

**+ indicates a syntax element that must be included**

The plus (+) symbol indicates a syntax element that must be included at least once. A dotted decimal number followed by the + symbol indicates that the syntax element must be included one or more times. That is, it must be included at least once and can be repeated. For example, if you hear the line 6.1+ data area, you must include at least one data area. If you hear the lines 2+, 2 HOST, and 2 STATE, you know that you must include HOST, STATE, or both. Similar to the * symbol, the + symbol can repeat a particular item if it is the only item with that dotted decimal number. The + symbol, like the * symbol, is equivalent to a loopback line in a railroad syntax diagram.
This information was developed for products and services that are offered in the USA or elsewhere.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user’s responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
United States of America

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

This information could include missing, incorrect, or broken hyperlinks. Hyperlinks are maintained in only the HTML plug-in output for the Knowledge Centers. Use of hyperlinks in other output formats of this information is at your own risk.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Site Counsel
2455 South Road
Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions.

Applicability

These terms and conditions are in addition to any terms of use for the IBM website.

Personal use

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or
reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights
Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

IBM Online Privacy Statement
IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user, or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering’s use of cookies is set forth below.

Depending upon the configurations deployed, this Software Offering may use session cookies that collect each user’s name, email address, phone number, or other personally identifiable information for purposes of enhanced user usability and single sign-on configuration. These cookies can be disabled, but disabling them will also eliminate the functionality they enable.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM’s Privacy Policy at ibm.com/privacy and IBM’s Online Privacy Statement at ibm.com/privacy/details in the section entitled “Cookies, Web Beacons and Other Technologies,” and the “IBM Software Products and Software-as-a-Service Privacy Statement” at ibm.com/software/info/product-privacy.

Policy for unsupported hardware
Various z/OS elements, such as DFSMS, JES2, JES3, and MVS™, contain code that supports specific hardware servers or devices. In some cases, this device-related element support remains in the product even after the hardware devices pass their announced End of Service date. z/OS may continue to service element code; however, it will not provide service related to unsupported hardware devices. Software problems related to these devices will not be accepted for service, and current service activity will cease if a problem is determined to be associated with out-of-support devices. In such cases, fixes will not be issued.
Minimum supported hardware

The minimum supported hardware for z/OS releases identified in z/OS announcements can subsequently change when service for particular servers or devices is withdrawn. Likewise, the levels of other software products supported on a particular release of z/OS are subject to the service support lifecycle of those products. Therefore, z/OS and its product publications (for example, panels, samples, messages, and product documentation) can include references to hardware and software that is no longer supported.

- For information about software support lifecycle, see: IBM Lifecycle Support for z/OS (www.ibm.com/software/support/systemsz/lifecycle)
- For information about currently-supported IBM hardware, contact your IBM representative.

Programming interface information

This publication primarily documents information that is NOT intended to be used as Programming Interfaces of DFSORT.

This publication also documents intended Programming Interfaces that allow the customer to write programs to obtain the services of DFSORT. This information is identified where it occurs, either by an introductory statement to a chapter or section or by the following marking:

Trademarks

IBM, the IBM logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at Copyright and Trademark information (www.ibm.com/legal/copytrade.shtml).

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.