

IMS  
Version 13

*Messages and Codes, Volume 3:  
IMS Abend Codes  
(November 5, 2018 edition)*

**IBM**



IMS  
Version 13

*Messages and Codes, Volume 3:  
IMS Abend Codes  
(November 5, 2018 edition)*



**Note**

Before you use this information and the product it supports, read the information in “Notices” on page 687.

November 5, 2018 edition.

This edition applies to IMS Version 13 (program number 5635-A04), IMS Database Value Unit Edition, V13.1 (program number 5655-DSM), IMS Transaction Manager Value Unit Edition, V13.1 (program number 5655-TM2), and to all subsequent releases and modifications until otherwise indicated in new editions.

© **Copyright IBM Corporation 1974, 2018.**

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

---

# Contents

<b>About this information</b> . . . . .	<b>v</b>	<b>Chapter 13. IMS abend codes 0401 - 0500</b> . . . . .	<b>203</b>
Prerequisite knowledge. . . . .	v	<b>Chapter 14. IMS abend codes 0501 - 0600</b> . . . . .	<b>229</b>
IMS function names used in this information . . . . .	vi	<b>Chapter 15. IMS abend codes 0601 - 0700</b> . . . . .	<b>281</b>
How new and changed information is identified . . . . .	vi	<b>Chapter 16. IMS abend codes 0701 - 0750</b> . . . . .	<b>307</b>
Accessibility features for IMS Version 13. . . . .	vi	<b>Chapter 17. IMS abend codes 0751 - 0800</b> . . . . .	<b>341</b>
How to send your comments . . . . .	vii	<b>Chapter 18. IMS abend codes 0801 - 0850</b> . . . . .	<b>401</b>
<b>Chapter 1. New, changed, and deleted messages and codes for IMS Version 13. 1</b>		<b>Chapter 19. IMS abend codes 0851 - 0900</b> . . . . .	<b>439</b>
New messages and codes for Version 13 . . . . .	1	<b>Chapter 20. IMS abend codes 0901 - 0950</b> . . . . .	<b>465</b>
Changed messages and codes for Version 13 . . . . .	7	<b>Chapter 21. IMS abend codes 0951 - 1000</b> . . . . .	<b>495</b>
Deleted messages and codes for Version 13. . . . .	11	<b>Chapter 22. IMS abend codes 1001 - 1050</b> . . . . .	<b>517</b>
<b>Chapter 2. IMS problem determination</b> . . . . .	<b>13</b>	<b>Chapter 23. IMS abend codes 1051 - 2000</b> . . . . .	<b>581</b>
<b>Chapter 3. IMS abend codes</b> . . . . .	<b>17</b>	<b>Chapter 24. IMS abend codes 2001 - 3000</b> . . . . .	<b>587</b>
IMS abends . . . . .	17	<b>Chapter 25. IMS abend codes 3001 - 3100</b> . . . . .	<b>599</b>
Standard abends. . . . .	18	<b>Chapter 26. IMS abend codes 3101 - 3300</b> . . . . .	<b>641</b>
Pseudoabends . . . . .	18	<b>Chapter 27. IMS abend codes 3301 - 4100</b> . . . . .	<b>663</b>
Multimodule abends . . . . .	19	<b>Notices</b> . . . . .	<b>687</b>
Getting information about IMS abends . . . . .	19	Trademarks . . . . .	689
Failure analysis tables . . . . .	20	Terms and conditions for product documentation . . . . .	689
Formatted dump . . . . .	20	IBM Online Privacy Statement. . . . .	690
IMS abend search and notification. . . . .	20		
<b>Chapter 4. System abend codes</b> . . . . .	<b>21</b>		
<b>Chapter 5. IMS abend codes 0000 - 0050</b> . . . . .	<b>23</b>		
<b>Chapter 6. IMS abend codes 0051 - 0100</b> . . . . .	<b>51</b>		
<b>Chapter 7. IMS abend codes 0101 - 0150</b> . . . . .	<b>85</b>		
<b>Chapter 8. IMS abend codes 0151 - 0200</b> . . . . .	<b>109</b>		
<b>Chapter 9. IMS abend codes 0201 - 0250</b> . . . . .	<b>143</b>		
<b>Chapter 10. IMS abend codes 0251 - 0300</b> . . . . .	<b>155</b>		
<b>Chapter 11. IMS abend codes 0301 - 0350</b> . . . . .	<b>167</b>		
<b>Chapter 12. IMS abend codes 0351 - 0400</b> . . . . .	<b>179</b>		

**Bibliography. . . . . 691**

---

## About this information

This information provides reference information for the abend codes, messages, status codes, and return codes that are produced by IMS.

This information provides reference information for the abend codes, messages, status codes, and return codes that are produced by IMS.

Volume 1 provides reference information for the IMS messages that have the DFS prefix, along with their associated return codes. It also provides diagnostic information that helps programmers, operators, and system-support personnel diagnose problems in IMS.

Volume 2 provides reference information for non-DFS prefixed IMS messages that are associated with IMS Base Primitive Environment (BPE); IMS Common Queue Server (CQS); IMS Common Service Layer (CSL); Database Recovery Control (DBRC) facility; IMS Connect; IBM® Resource Lock Manager (IRLM), and so on. It provides diagnostic reference information that helps programmers, operators, and system-support personnel diagnose problems in IMS.

Volume 3 provides reference information for all IMS abnormal termination (abend) codes, including analysis, explanation, possible causes, and APAR processing instructions.

Volume 4 provides return, reason, sense, function, and status codes for IMS Base Primitive Environment (BPE); IMS Common Queue Server (CQS); IMS Common Service Layer (CSL); Database Recovery Control (DBRC) facility; IMS Connect; IBM Resource Lock Manager (IRLM), and so on. It also provides diagnostic reference information that helps programmers, operators, and system-support personnel diagnose problems in IMS.

This information is available as part of IBM Knowledge Center at [www.ibm.com/support/knowledgecenter](http://www.ibm.com/support/knowledgecenter).

---

## Prerequisite knowledge

Before using this information, you should have knowledge of either IMS Database Manager (DB) or IMS Transaction Manager (TM). You should also understand basic z/OS® and IMS concepts, the IMS system for your installation, and have general knowledge of the tasks involved in project planning.

Before using this book, you should have knowledge of either IMS Database Manager (DB) or IMS Transaction Manager (TM). You should also understand basic z/OS and IMS concepts, the IMS system for your installation, and have general knowledge of the tasks involved in project planning.

You should be familiar with z/OS, its system generation, and telecommunications, and the access methods used by IMS. Related publications are listed in the Bibliography.

You can learn more about z/OS by visiting the “z/OS basic skills” topics in IBM Knowledge Center.

You can gain an understanding of basic IMS concepts by reading *An Introduction to IMS*, an IBM Press publication.

IBM offers a wide variety of classroom and self-study courses to help you learn IMS. For a complete list of courses available, go to the IBM Skills Gateway and search for IMS.

---

## IMS function names used in this information

In this information, the term HALDB Online Reorganization refers to the integrated HALDB Online Reorganization function that is part of IMS Version 13, unless otherwise indicated.

---

## How new and changed information is identified

New and changed information in most IMS library PDF publications is denoted by a character (revision marker) in the left margin. The first edition (-00) of *Release Planning*, as well as the *Program Directory* and *Licensed Program Specifications*, do not include revision markers.

Revision markers follow these general conventions:

- Only technical changes are marked; style and grammatical changes are not marked.
- If part of an element, such as a paragraph, syntax diagram, list item, task step, or figure is changed, the entire element is marked with revision markers, even though only part of the element might have changed.
- If a topic is changed by more than 50%, the entire topic is marked with revision markers (so it might seem to be a new topic, even though it is not).

Revision markers do not necessarily indicate all the changes made to the information because deleted text and graphics cannot be marked with revision markers.

---

## Accessibility features for IMS Version 13

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use information technology products successfully.

### Accessibility features

The following list includes the major accessibility features in z/OS products, including IMS Version 13. These features support:

- Keyboard-only operation.
- Interfaces that are commonly used by screen readers and screen magnifiers.
- Customization of display attributes such as color, contrast, and font size.

### Keyboard navigation

You can access IMS Version 13 ISPF panel functions by using a keyboard or keyboard shortcut keys.

For information about navigating the IMS Version 13 ISPF panels using TSO/E or ISPF, refer to the *z/OS TSO/E Primer*, the *z/OS TSO/E User's Guide*, and the *z/OS ISPF User's Guide Volume 1*. These guides describe how to navigate each interface,



including the use of keyboard shortcuts or function keys (PF keys). Each guide includes the default settings for the PF keys and explains how to modify their functions.

### **Related accessibility information**

Online documentation for IMS Version 13 is available in IBM Knowledge Center.

### **IBM and accessibility**

See the *IBM Human Ability and Accessibility Center* at [www.ibm.com/able](http://www.ibm.com/able) for more information about the commitment that IBM has to accessibility.

---

## **How to send your comments**

Your feedback is important in helping us provide the most accurate and highest quality information. If you have any comments about this or any other IMS information, you can take one of the following actions:

- Click the **Contact Us** tab at the bottom of any IBM Knowledge Center topic.
- Send an email to [imspubs@us.ibm.com](mailto:imspubs@us.ibm.com). Be sure to include the book title and the publication number.

To help us respond quickly and accurately, please include as much information as you can about the content you are commenting on, where we can find it, and what your suggestions for improvement might be.



---

# Chapter 1. New, changed, and deleted messages and codes for IMS Version 13

A number of messages and codes are new, changed, or deleted in IMS Version 13.

---

## New messages and codes for Version 13

IMS Version 13 includes new BPE, CQS, CSL, DFS, DSP, G, HWS, and XDFLD messages, as well as new abend, DL/I, AIB, and SQL codes.

### New BPE messages

The following messages are new:

BPE0048E  
BPE0049I  
BPE0050I  
BPE0051E  
BPE0052E  
BPE0053E  
BPE0054I

### New CQS messages

The following messages are new:

CQS0012I  
CQS0034A  
CQS0247W  
CQS0248I  
CQS0301I  
CSL2504I  
CSL3070I  
CSL4115E

### New CSL messages

The following messages are new:

CSL4113E  
CSL4114E

### New DFS messages

The following messages are new:

DFS0006E  
DFS0095I  
DFS0123E  
DFS0904I  
DFS1849E  
DFS1853I  
DFS1937I  
DFS3187I  
DFS3197I  
DFS3198I

| DFS3287E  
| DFS3298E  
| DFS3417E  
| DFS3428W  
| DFS3429E  
| DFS3436E  
| DFS3437E  
| DFS3523W  
| DFS3527E  
| DFS3529  
| DFS3531E  
| DFS3549E  
| DFS3553E  
| DFS3555E  
| DFS3785E  
| DFS3786E  
| DFS3787E  
| DFS3788I  
| DFS3789I  
| DFS3948E  
| DFS3950A  
| DFS4408I  
| DFS4409A  
| DFS4410I  
| DFS4411E  
| DFS4412I  
| DFS4413E  
| DFS4414I  
| DFS4432E  
| DFS4433E  
| DFS4434I  
| DFS4435I  
| DFS4436I  
| DFS4437I  
| DFS4438E  
| DFS4439E  
| DFS4440I  
| DFS4441I  
| DFS4442I  
| DFS4443I  
| DFS4500E  
| DFS4518I  
| DFS4575E  
| DFS4576W  
| DFS4585W  
| DFS4586E  
| DFS4600E  
| DFS4600I  
| DFS4601E  
| DFS4601I  
| DFS4602E  
| DFS4603I  
| DFS4604I  
| DFS4605I  
| DFS4606I  
| DFS4607I

DFS4608E  
DFS4610E  
DFS4611E  
DFS4612E  
DFS4613E  
DFS4614E  
DFS4616E  
DFS4617E  
DFS4618E  
DFS4619E  
DFS4620E  
DFS4621E  
DFS4622E  
DFS4623W  
DFS4624I  
DFS4625E  
DFS4626I  
DFS4627E  
DFS4628I  
DFS4629E  
DFS4630E  
DFS4634I  
DFS4635I  
DFS4636I  
| DFS4637E  
DFS4639E  
DFS4640E  
DFS4641E  
DFS4642E  
DFS4643E  
DFS4644E  
DFS4645W  
DFS4646I  
| DFS4648E  
| DFS4649A  
DFS4650I  
DFS4651I  
| DFS4652E  
| DFS4660E  
| DFS4661E  
| DFS4662E  
DFS4663E  
DFS4664I  
DFS4665I  
DFS4666E  
| DFS4667I  
| DFS4668I  
| DFS4669E  
DFS4671I  
| DFS4677E  
| DFS4678E  
| DFS4679E  
DFS4687E  
| DFS4704E  
| DFS4705E  
| DFS4706E

| DFS4707E  
| DFS4708E  
| DFS4709E  
| DFS4710E  
| DFS4711E  
| DFS4720I  
| DFS4721I  
| DFS4725I  
| DFS4831E  
| DFS4865E  
| DFS4866E  
| DFS4930I  
| DFS4965E  
| DFS5386I  
| DFSIX089

### **New DSP messages**

The following messages are new:

DSP0397E  
DSP0398E  
DSP0399E  
DSP1074E  
DSP1075E  
DSP1097E  
| DSP1206E  
DSP1249E  
DSP1250E  
DSP1251E  
| DSP1256E  
| DSP1262I  
| DSP1267E  
| DSP1268E

### **New G messages**

The following messages are new:

G115

### **New HWS messages**

The following messages are new:

HWSG4000I  
HWSG4005I  
HWSG4010I  
HWSG4015E  
HWSG4020W  
HWSG4025W  
HWSG4030W  
HWSG4040W  
HWSG4045W  
HWSG4060I  
HWSG4065E  
HWSG4080W  
HWSG4100E  
HWSG4105W

HWSG4230W  
HWSG4235W  
HWSG4255W  
HWSI1610W  
HWSS0776W  
HWSU3900E  
HWSV4400I  
HWSV4405I  
HWSV4410E  
HWSV4415E  
HWSV4425I  
HWSV4435W  
HWSV4440E  
HWSV4480W  
HWSV4485W  
HWSV4500E  
HWSV4600E  
HWSV4630W  
HWSV4800E  
HWSV4820E  
HWSV4830W  
HWSV4840W  
HWSV5000E  
HWSV5030W  
HWSI1610W  
HWSU3900E  
HWSV4400I  
HWSV4405I  
HWSV4410E  
HWSV4415E  
HWSV4420E  
HWSV4425I  
HWSV4435W  
HWSV4440E  
HWSV4480W  
HWSV4485W  
HWSV4500E  
HWSV4600E  
HWSV4630W  
HWSV4800E  
HWSV4820E  
HWSV4830W  
HWSV4840W  
HWSV5000E  
HWSV5030W

### **New XDFLD messages**

The following messages are new:

XDFLD600  
XDFLD601  
XDFLD602  
XDFLD603  
XDFLD604

## New abend, DL/I, AIB, and utility codes

The following codes are new:

| 0055  
| 0841  
| 1002  
| 1143  
| 1155  
| BPE 3404  
| BPE 3405  
| AN  
| BH  
| SF  
| 0100/0110  
| OTMA 0036  
| Fast Path utility status codes

## New SQL codes

The following codes are new:

| 000  
| +098  
| +100  
| -010  
| -101  
| -104  
| -117  
| -122  
| -158  
| -198  
| -199  
| -203  
| -204  
| -205  
| -206  
| -208  
| -301  
| -302  
| -303  
| -305  
| -313  
| -350  
| -404  
| -407  
| -408  
| -413  
| -420  
| -490  
| -514  
| -516  
| -517  
| -518  
| -519  
| -530  
| -531  
| -840



| -904  
| -5001  
| -8001  
| -8002  
| -8003  
| -8004  
| -8005  
| -8006  
| -8007  
| -8008  
| -8009  
| -8010  
| -8011  
| -8013  
| -8014  
| -8015

---

## Changed messages and codes for Version 13

IMS Version 13 includes changed BPE, CQS, CSL, DBD, DFS, DSP, FLD, G, HWS, and SEG messages, as well as changed abend codes and component codes.

The following messages and codes are changed. An asterisk (\*) indicates that the message text is changed.

### Changed BPE messages

The following messages are changed:

BPE0001E  
BPE0006I  
BPE0010I  
BPE0014E  
BPE0017I  
BPE0028I  
BPE0029I  
BPE0037E

### Changed CQS messages

The following messages are changed:

| CQS0034A  
| CQS0300I  
| CQS0222E  
| CQS0350W  
| CQS0354E

### Changed CSL messages

The following messages are changed:

| CSL0001E  
| CSL0012I  
| CSL2501I

## Changed DBD messages

The following messages are changed:

DBD760

## Changed DFS messages

The following messages are changed:

DFS047A  
DFS0123E  
DFS0430I  
DFS0488I  
DFS0843I  
DFS0535I  
DFS550I  
DFS554A  
DFS686W  
DFS0730I  
DFS0793I  
DFS0832I  
DFS0842I  
DFS1190I  
DFS1269E  
DFS1913I  
DFS1919I  
DFS1994I  
DFS2166  
DFS2205I  
DFS2291I  
DFS2372I  
DFS2452I  
DFS2473I  
DFS2499W  
DFS2513I  
DFS2662A  
DFS2679A  
DFS2854A  
DFS2857E  
DFS2858E  
DFS0962I  
DFS2859I  
DFS2868W  
DFS2930I  
DFS2991E  
DFS3298E  
DFS3314W  
DFS3178E  
DFS3347A  
DFS3374W\*  
DFS3422X  
DFS3435A  
DFS3498W  
DFS3549E  
DFS3613I  
DFS3649A  
DFS3678E

| DFS3688I  
| DFS3702I  
| DFS3709A  
| DFS3176E  
| DFS4342E  
| DFS4388W  
| DFS4401E\*  
| DFS4419I\*  
| DFS4420E  
| DFS4421E  
| DFS4424E  
| DFS4426I  
| DFS4429E  
| DFS4440I  
| DFS4441I  
| DFS4447E  
| DFS4477E  
| DFS4478E  
| DFS4486E  
| DFS4494E  
| DFS4585W  
| DFS4600I  
| DFS4601I  
| DFS4637E  
| DFS4637I  
| DFS4662E  
| DFS4663E  
| DFS4810I

### **Changed DSP messages**

The following messages are changed:

| DSP0012I  
| DSP0141I  
| DSP0222I  
| DSP0224I  
| DSP1076I  
| DSP1095A

### **Changed FLD messages**

The following messages are changed:

| FLD180

### **Changed G messages**

The following messages are changed:

| G430  
| G449  
| G561

### **Changed HWS messages**

The following messages are changed:

| HWSC0001I  
| HWSC0114W

| HWSN1940W  
| HWSS0714E  
| HWSV4425I  
| HWSX0905E  
| HWSX0907E  
| HWSX0909E

### **Changed SEGM messages**

The following messages are changed:  
SEGM760

### **Changed abend codes**

The following abend codes are changed:

| 0021  
| 0029  
| 0071  
| 0073  
| 0088  
| 0107  
| 0113  
| 0119  
| 0166  
| 0370  
| 0587  
| 0602  
| 0630  
| 0717  
| 0718  
| 0757  
| 0780  
| 0800  
| 0840  
| 0979  
| 1002  
| 1009  
| 1011  
| 1021  
| 1026  
| 3044  
| 3045  
| 3048  
| 3049  
| 3053  
| 3055  
| 3303  
| 3305  
| 3422

### **Changed component codes**

The following component codes are changed:  
BPEATTCH codes  
BPELOAD codes  
BPERDPDS codes

| AIB 0108/0308  
| AIB 0110/0050  
| BPE 3400  
| BPE 3401  
| BPE 4095  
| ODBM 0010  
| OTMA 001A  
| Fast Path utility: UR  
| AC  
| BA  
| FM

---

## Deleted messages and codes for Version 13

A number of messages and codes were removed from IMS in Version 13.

### Deleted messages

The following messages were removed from IMS Version 13:

| CSL0010E  
| DFS3187W  
| DFS3253I  
| DFS4147E  
| DFS4149W

### Deleted codes

The following codes were removed from IMS Version 13:

| 0062



---

## Chapter 2. IMS problem determination

IMS messages and codes problem determination is accomplished by using procedures established by IBM.

Problem determination is accomplished by using procedures for identifying a failing module or program and determining whether the user or IBM is responsible for resolution of a given problem.

Problem determination information is included, as appropriate, for all messages and codes. In most instances, one or more numbers (for example, "4,6") appear after the heading **Problem Determination** for a given message or code. These numbers are cross-references to the numbered items that appear in the following list of standard problem determination actions to be taken by the user.

1. Save the console sheet from the z/OS console.
2. Save the input associated with the job.
3. Save the SYSOUT (system output) associated with the job.
4. Save the storage memory dump at time of failure of the IMS control region, and save the memory dump of all associated message regions. (If SYS1.DUMP is used, print, and save it.)
5. Save the IMS log printout.
6. Save the IMS master terminal printout.
7. Save the local or remote terminal printout.
8. Save the JCL listings.
9. Save the ACB generation output.
10. Save the IMSGEN listing.
11. Save the SYSABEND memory dump. (If SYS1.DUMP is used, print, and save it.)
12. Save the assembly listing of any user-modified IMS modules.
13. Save the bind map.
14. Save the LOG trace.
15. Save the IMS log printout starting with the records that were written just before abnormal termination.
16. Save the output listing from the MFS Language Utility execution.
17. Collect the following problem-related data:
  - a. System memory dump if running batch.
  - b. If running online, print log records type X'67', sub-record type X'FF' (Pseudoabend). Use the File Select and Formatting Print program to print these records. The OPTION PRINT control statement must specify EXITR=DFSERA30.
  - c. Run the DL/I test program (DFSDDLTO) with SNAP statements before and after the failing call.
  - d. Save the PSB and DBD listings or statements, or both, related to the failure.
  - e. Create a memory dump of log records type X'67', sub-record type X'FE' (program isolation trace). Use the File Select and Formatting print program to print the records. Program isolation trace must be enabled when the log is created. The OPTION PRINT control statement must specify EXITR=DFSERA40.

- f. If online, a snap from the log of the online region.
  - g. A history of the database backout, recovery, and reorganization activity.
18. Save the assembly listing for the associated DBDs.
  19. Save the assembly listing for the associated PSBs.
  20. Save a listing of the VSAM catalog entries for the associated data sets at the time of failure.
  21. Reserved for future use
  22. Reserved for future use
  23. Contact IBM for hardware support.
  24. Create a memory dump of the 3270 diagnostic error log records off the IMS log: Record type X'67': sub-record type X'03'.
  25. Submit a copy of the IMS log.
  26. Determine the CCB involved from the SPACCB field of the scratch pad area or from the CTBCCBPT field. Determine the associated (a) source and output CNT, and (b) the line and PTERM (CTB).  
Determine from log records 01, 02, 03, 07, 08, 11, 12 and 13 the status of the conversation:
    - a. The message was ready, but the application was not scheduled. A 01 destination is a conversational SMB.
    - b. The application was scheduled and a 08 record was produced.
    - c. The application inserted a segment to a conversational terminal. If the destination of the message is CNT, 03 and 13 records are produced.
    - d. The application inserted a segment to a conversational program. SPA is the first segment of a message in the 03 records.
    - e. The application finished normally or terminated abnormally. The 07 record contains the completion code of a conversational application. The DLRFLAG2 area (the last word of the 07 log record) contains synchronization point information.
    - f. A 12 record indicates that the conversation was terminated.
    - g. Determine if 02 records, produced for the following commands, affected the terminal in conversation: /EX1, /HOLD, /REL, or /START LINE (no PTERM).
    - h. Does restarting IMS affect the conversation? For details, see the input to and output logs from restart.
  27. Save the z/OS hardcopy message log output.
  28. Determine the status of z/OS and IRLM operating in the other system with which this system is connected during normal operations.
  29. Save the IRLM address space dump generated at the time of the failure.
  30. Save the memory dump of any IMS online or batch address space that failed near the time of the IRLM failure. Follow IMS problem determination procedures relative to the IMS memory dump. The IMS failure might be related to the IRLM failure.
  31. Save a listing of the IRLM start procedure JCL.
  32. Save a listing of the IMSGEN performed to install the IRLM.
  33. Save listings of the VTAM<sup>®</sup> definitions performed to install the IRLM.
  34. Save the GTF trace output, if any is available showing activity at the time of the failure.
  35. Refer to the following manuals for further assistance to resolve your problem:
    - *IMS Version 13 Messages and Codes, Volume 1: DFS Messages*



- *IMS Version 13 Messages and Codes, Volume 2: Non-DFS Messages*
- *IMS Version 13 Messages and Codes, Volume 3: IMS Abend Codes*
- *IMS Version 13 Messages and Codes, Volume 4: IMS Component Codes*
- *IMS Version 13 Diagnosis*

Check all input for accuracy and analyze any memory dumps.

If you are still unable to solve your problem, contact IBM Software Support for further assistance.

36. If you are unable to solve the problem, contact IBM Software Support for assistance.
37. Submit a copy of the type X'02' log records.
38. Save the CCTL SNAP or SDUMP data set.
39. Save a listing of the DBRC RECON.
40. Take an SVC memory dump and contact IBM Software Support for further assistance.



---

## Chapter 3. IMS abend codes

IMS abend codes are issued when IMS modules terminate abnormally. IMS abends are *user abends*.

When you encounter an IMS abend, use the codes to do the following:

- Define the documentation required to solve a problem
- Identify conditions—register contents, bit settings— that can help pinpoint the cause of the error
- Make it possible to standardize IMS problem definition statements

The IMS abend code descriptions help identify the failed subroutines from the functions that were active when the failure occurred.

You can use this information about problematic symptoms for the IBM Early Warning System (EWS) and RETAIN. You can use abend code information during failure diagnosis to quickly identify the failing unit of code. For example, the information in the tables can be used to uniquely define the abnormal termination (abend) by identifying the module and the label of the unit of code within the module that detected an error condition and issued the abend.

If troubleshooting does not resolve the problem or an IMS internal error occurs, please contact IBM Software Support.

---

### IMS abends

IMS modules that terminate abnormally issue an IMS abend code.

When IMS modules detect an error, they terminate in one of two ways:

- Standard abnormal terminations (abends)
- Pseudoabends

Each abend is identified by a 4-digit code, for example 0123. These abends are also known as *user abends*. They are often identified in the form of ABENDUxxxx in the service information and occasionally in a shortened form as Uxxxx., for example as ABENDU0123 and U0123.

Use these topics to diagnose IMS problems encountered in the DB/DC and DBCTL environments. You can also use the tables to effectively define problems encountered in batch processing of DL/I databases. The documentation you need to define IMS problems is:

- A system (SYSABEND) or user (SYSUDUMP) dump or a DFSERA30 Formatted Log Print.
- A listing of the assembled modules that detected the error condition and listings of the modules related to the module that detected the error.
- A listing of the data areas used by the module that detected the error. To obtain these data areas, first remove the PRINT NOGEN statement from the source that detected the error, then assemble the IDLI macro with the module that detected the error.

## Standard abends

A standard abend code is issued by a macro at the point of error detection or by branching to a subroutine that issues the code.

The program status word (PSW) at entry to abnormal termination, which is indicated in the dump, contains the address of the instruction in the module that issued the abend macro.

## Pseudoabends

When an IMS module detects a problem, it can issue a pseudoabend. A pseudoabend terminates the application in a dependent region, but the region does not terminate. When a pseudoabend occurs, the module that detects the error condition does not issue the abend macro.

Instead, it passes control back to the call analyzer module, DFSDLA00, which indicates a dependent-region abend. The call analyzer module calls DFSERA20 to determine whether to write the contents of important control blocks to the system log or to create a memory dump.

At abnormal termination, the program status word (PSW) contains the address of the request handler module, not the module that detected the error. For a batch region, the request handler module is DFSPR000. For an online region, the module is DFSPROX0 (a subroutine of DFSISI00). One of two types of dumps is produced, depending on whether an online or batch region is terminating.

When it is not necessary for the control region to terminate in a DB/DC or DBCTL environment, only the dependent region is dumped. When a dump is not issued for the pseudoabend, print the control blocks from the log. Pass the appropriate OPTION statement to the File Select and Formatting Print Utility (DFSERA10) to:

- Select the X'67FF' records from the appropriate system log data set (SLDS).
- Invoke the Record Format and Print Module (DFSERA30) exit routine to format and print the control blocks.

However, in a batch environment, the batch region is terminated and a SYSABEND dump is printed. The program status word (PSW) at entry to abend indicates the program request handler module, DFSPR000. Use the information in these topics to understand the procedure for finding the contents of the program registers at the time of the failure. Finding the contents of the program registers is necessary to further investigate the cause of the failure.

**Attention:** To adequately diagnose the abend codes that result from errors in DL/I, it is necessary for the DL/I call trace to be active.

### Pseudoabends that can cause the input message to be rescheduled

The following abends might cause the input message to be queued for scheduling as part of normal abend processing.

- 0128
- 0129
- 0777
- 0778
- 0804
- 0830
- 2478

- 2479
- 3303
- 3048
- 3049
- 3051
- 3053
- 3057

To override the normal abend processing, use the Non-Discardable Messages exit routine (DFSNDMX0).

**Related concepts:**

[DL/I trace \(Diagnosis\)](#)

**Related reference:**

[Non-Discardable Messages user exit \(NDMX\) \(Exit Routines\)](#)

## Multimodule abends

*Multimodule abends* are IMS abends that are issued by more than one module.

Multinode abends use a different format than those abends issued by one module. The “Analysis” section of each abend designates the issuing modules and specifies the labels within the failing modules.

---

## Getting information about IMS abends

This information provides information explaining what caused an abnormal termination, how to determine the cause, and the actions to take to correct the problem.

The information for many of the abend codes includes a failure analysis table with details to help diagnoses the specific problem

The following is a sample of the information for an IMS abend code:

---

### 9999 (sample abend code)

**Explanation:** This section contains an explanation of the event that caused the abnormal termination.

**Analysis:** The Analysis section contains information to diagnose the cause of the problem. The information typically includes general register usage in the failing module, the general control flow before failure, and the names of pertinent control blocks. The information in this section can be divided by the issuing module, to help identify specific problems.

This information can include lists of return or reason codes and failure analysis tables. The following is an example of a failure analysis table:

*Table 1. Sample failure analysis table*

Key	Label	Description
Reg15=X'10'	ABNDXXX	<i>Information to help identify the specific problem</i>

**Possible cause:** Situations that can cause this type of failure.

**For APAR processing:** Additional documentation that is required to complete an APAR for this specific abend.

**System action:** The System action section describes the status of the system as the result of the abend.

**Operator response:** The Operator response section contains the actions a system operator should take in response to the abend.

**Programmer response:** The Programmer response section contains the actions a programmer or system programmer should take in response to the abend.

**Problem determination:** Problem Determination section provides information on the steps to take to identify and document the problem. The numbers in this section refer to a list of problem determination actions that are specified in Chapter 2, "IMS problem determination," on page 13.

**Module:** This section lists the modules that issue the abend

## Failure analysis tables

The information on IMS abend codes includes a table to help you diagnose the cause of the failure. These are called failure analysis tables.

Each of the tables consists of 3 columns: Key, Label, and Description, which contain the following information:

**Key** Key conditions that can be used to isolate the error to a single cause.

This column describes the contents of significant registers at the entry to abnormal termination. In some cases, no specific information is available (for example, register contents were not saved or were overlaid). In these cases, the Key portion of the table is blank.

**Label** Label of the routine that detected the error. This routine is not necessarily the routine that issued the abend.

**Description**

A synopsis of the function of this routine and an indication of the cause of failure.

## Formatted dump

To format and print an entire IMS dump, or only those control blocks and data areas that you need to analyze the problem, use the IMS Dump Formatter and the IMS Offline Dump Formatter.

You can format dumps online, although the Interactive and Offline formatters are the recommended approach. You can find out more about the IMS Dump Formatter and the IMS Offline Dump Formatter in the following information:

- *IMS Version 13 System Definition*
- *IMS Version 13 Diagnosis*

## IMS abend search and notification

You can use the IMS abend search and notification function to send a message to an e-mail address providing notification of the occurrence of an abend and additional information about the abend.

You can get more information about abends from IMS product documentation, technical notes in an IBM technical support database, and information in the preventive service planning (PSP) database. For information on setting up and using the IMS abend search and notification function, see IMS diagnosis information.

---

## Chapter 4. System abend codes

This topic describes the following system abend codes:

---

### AC6

**Explanation:** IMS uses this z/OS abend code when forcibly terminating one z/OS task from another z/OS task. There are three situations where this occurs:

- An ESTAE routine gained control because of an abend in the system. This terminating task attempts to purge the log buffers after it signals the physical log task (the owner of the buffers) of its intention. If the terminating task does not receive a response within one second, the physical log task is terminated with this abend. In either case, log buffers are purged.
- The DBRC instance required for this instance of the isolated log sender (ILS) terminated. All remaining ILS instances dependent upon the abended DBRC wait for that DBRC until they are terminated with this abend and reason code X'01'
- An SLDS read might forcibly terminate its OPEN/CLOSE subtask with this abend code and reason code X'02' when one of the following situations occurs:
  - A /STOP SLDSREAD command is entered
  - A conflict for the SYSZTIOT resource is detected
  - An auxiliary storage shortage is detected

**Source:** IMS system abend

**Module:** DFSFDLF0, DFSFDLG0, DFSIL220

---

### Fnn

**Explanation:** z/OS detected an incorrect installation of the IMS™ Type 2 SVC or the DBRC Type 4 SVC. The last 2 digits of this abend code, nn, are the IMS SVC number in hexadecimal. For example, abend code SFE1 means that the error occurred while the system was processing SVC X'E1', that is, SVC 225.

**Programmer response:** Correct the installation errors. For more information about installing the IMS™ Type 2 SVC or the DBRC Type 4 SVC, see z/OS interface considerations for IMS (System Administration).

**Source:** Supervisor control





---

## Chapter 5. IMS abend codes 0000 - 0050

This topic identifies and explains the following abend codes issued by IMS. Most codes are 4 decimal digits.

---

### 0002

**Explanation:** When the IMS control region abnormally terminates, it forces this abend for active dependent regions. This abend can also occur after normal shutdown of the control region if a new dependent region attempts to start.

**Analysis:** This is a pseudoabend scheduled by module DFSV4100 to abend the dependent regions.

**System action:** The dependent regions are abnormally terminated.

**Operator response:** Follow IMS restart procedures.

**Problem determination:** None.

**Source:** IMS abends

**Module:** DFSV4100

---

### 0005

**Explanation:** IMS restart was unable to close (terminate) the input online log data set (OLDS). Refer to messages DFS0738I and DFS0738X for further explanation.

**Analysis:** Abend 0005 is a standard abend that can be detected by DFSFDLW0 and issued by DFSFDLY0. DFSFDLW0 rebuilds blocks of log data not written to the online data sets (OLDS) at the time of the last IMS failure. The program status word (PSW) at entry-to-abend points to the instruction from which the abend (SVC 13) is issued.

Register 9 in the abend supervisor request block (SVRB) register contains the address of the work area used by DFSFDLW0. Register 11 and register 10 contain the addresses of the system contents directory (SCD) and the log control directory (LCD) respectively. Register 15 contains the return code.

Key	Description
Reg15=X'01' RLWABC=X'01' PDCB, SDCB	OPEN of a write-ahead data set (WADS) failed. PDCB is the primary WADS DCB. When dual WADS are being used, SDCB is the secondary WADS DCB.
Reg15=X'02' RLWABC=X'02'	The WADS was not the one used when the OLDS was used.
Reg15=X'03' RLWABC=X'03'	An I/O error was encountered reading from the WADS.
Reg15=X'04' RLWABC=X'04'	A log sequence error was found during WADS record concatenation.
Reg15=X'06' RLWABC=X'06'	Missing WADS record problem encountered. An earlier record indicates more records should exist, but the end-of-file was encountered.
Reg15=X'07' RLWABC=X'07'	Either the WADS does not consist of enough tracks for a track group or there was a TRKCALC failure.
Reg15=X'08' RLWABC=X'08'	A length error was encountered. <ul style="list-style-type: none"><li>• Zero was found where a record length was expected.</li><li>• There was not enough room for a type 48 record in the last block.</li></ul>
Reg15=X'09' RLWABC=X'09' RLWP.PABLKCNT	There are log sequence numbers missing between rebuilt blocks. WADS data had larger block sequence numbers than OLDS data (RLWP.PABLKCNT), but the sequence numbers were not contiguous or the clock values were ordered differently from the sequence numbers.

Key	Description
Reg15=X'B' RLWABC=X'B'	A track group contained segments from more than one non-residual block.
Reg15=X'C' RLWABC=X'C' RLWPBA	RLWPBA points to the rebuilt block that contains the block descriptor containing a block size that does not match the OLDS block size.
Reg15=X'D' RLWABC=X'D' RLWPBA	An invalid record length was found in the rebuilt block pointed to by RLWPBA.
Reg15=X'E' RLWABC=X'E' RLWPBA	A negative record length was found. A negative record length indicates the end of the data in an incomplete block pointed to by RLWPBA. The incomplete block in which the negative record length is found is not the last block.
Reg15=X'F' RLWABC=X'F' RLWPBA	A log record sequence error was found in the rebuilt block pointed to by RLWPBA.

**System action:** IMS terminates abnormally.

**Programmer response:** Use the Log Recovery utility to close (terminate) the log.

**Problem determination:** 1, 5, 8, 11, 35, 40

Save a copy of the OLDSs and WADSs data sets.

**Source:** IMS abends

**Module:** DFSFDLY0

**Related information:**

 DFS0738I (Messages and Codes)

 DFS0738X (Messages and Codes)

Chapter 2, "IMS problem determination," on page 13

## 006

**Explanation:** The License Manager did not find a certificate for either the DB or the TM feature. See message DFS2930I to determine which (or both features) do not have a certificate.

**System action:** The region that issued the abend terminates.

**Programmer response:** Add the necessary certificates to LM or change the LM mode so that it grants requests even when certificates do not exist.

**Source:** IMS abends

**Module:** DFSXSTM0

## 0008

**Explanation:** The amount of storage specified for the region is insufficient to build a table to contain PROCLIB member data. This prevents IMS from acquiring the information needed for initialization. Message DFS0610W, issued prior to this abend, identifies the TABLE name and provides additional information.

**Analysis:** This standard abend is issued by DFSXRPS0. The module attempted to obtain storage by issuing the IMODULE GETMAIN macro. A nonzero return code was returned by IMODULE GETMAIN.

Message **DFS0610W - GETMAIN FAILED FOR TABLE = DFSXRPST** is issued because the IMODULE GETMAIN routine that builds the header table received a nonzero return code from IMS system service.

Message **DFS0610W - GETMAIN FAILED FOR TABLE = PROCLIB** is issued because the IMODULE GETMAIN routine that builds the ddname table received a nonzero return code from IMS system service.

**System action:** IMS terminates abnormally.

**Operator response:** Increase the size of the region, and retry IMS.

**Programmer response:** Increase the size of the region.

**Problem determination:** 1, 3, 6, 8

**Source:** IMS abends

**Module:** DFSXRPS0, DFSPMBR0, DFSRED20

**Related information:**

Chapter 2, "IMS problem determination," on page 13

## 0009

**Explanation:** The OPEN failed for a data set. This failure prevents IMS from acquiring the information required for initialization. Message DFS0597W, issued before this abend, identifies the data set name with DDNAME=xxxxxxx and provides additional information.

**Analysis:** This abend is a standard abend issued by DFSXRPS0, DFSIILD0, and DFSRED20. These modules attempted to open the data set, but the z/OS OPEN command failed. The most common problem is a missing DD statement. For DFSXRPS0, DFSIILD0, and DFSRED20, register 6 contains the DCB address DCBOFLGS.

Module	Key	Label	Description
DFSXRPS0, DFSIILD0, DFSRED20	Reg6=DCB	OPENCHK	The routine that opens the data set did not receive any indication that the DCB was opened.

**System action:** IMS terminates abnormally.

**Programmer response:** Refer to DFS0597W. Check the DDNAME xxxxxxxx data set.

**Problem determination:** 1, 3, 6, 8

**Source:** IMS abends

**Module:** DFSXRPS0, DFSIILD0, DFSRED20

**Related information:**

 DFS0597W (Messages and Codes)

Chapter 2, "IMS problem determination," on page 13

## 0010

**Explanation:** The data set cannot be processed because the data set record format is not valid. This situation prevents IMS from acquiring information required for initialization. The abend occurred for one of the following reasons:

- The record format is not valid for the data set. Message DFS0604W, issued before this abend, identifies the data set by DDNAME xxxxxxxx and provides additional information.
- The system returned message DFS0604W to indicate that the data set format is invalid. The record format must be fixed or fixed block. If this message was received during initialization, the system might not have received needed information. If this message was received from the MFS DCT utility, the descriptor members were unavailable and the utility terminated with return code 4.

This abend can occur for non-Java-dependent regions as well (MPP, BMP, IFP regions).

**Analysis:** This abend is a standard abend issued by DFSXRPS0, DFSIILD0, and DFSRED20. These modules check for a RECFM=F or RECFM=FB parameter for the data set. Check the DCBRECFM field to determine the record format. Be sure the JCL points to the correct PROCLIB data set. For DFSXRPS0, DFSIILD0, and DFSRED20, register 6 contains the DCB address DCBRECFM.

## 0011

Module	Key	Label	Description
DFSXRPS0	Reg6=DCB	XRCFMERR	The routine used to validate the DCB record format expects that DCBRECFM = X'80'.
DFSIILD0	Reg6=DCB	XRCFMERR	The routine used to validate the DCB record format expects that DCBRECFM = X'80'.
DFSRED20	Reg6=DCB	RECFMERR	The routine used to validate the DCB record format expects that DCBRECFM = X'80'.

### For DFSRRC10, DFSRRC40:

This abend is a standard abend issued by DFSRRC10, DFSRRC40

Key	Label	Description
Reg15=X'4'		Required JVMOPMAS= <i>member</i> not specified.
Reg15=X'6' Reg3=BPX1SDD return code Reg4=BPX1SDD reason code		BPX1SDD failed.

### For DFSPCC20:

This abend is a standard abend issued by DFSPCC20.

Key	Label	Description
Reg15=X'1' Reg3=CEEPIPI return code		Unable to create an Language Environment <sup>®</sup> enclave (CEEPIPI error)
Reg15=X'2' Reg3=possible CEEPIPI return code		Unable to create a JVM.
Reg15=X'3' Reg3=possible CEEPIPI return code		JVM unable to start the application (CEEPIPI error).
Reg15=X'5' Reg3=possible CEEPIPI return code		Unable to add an entry (CEEPIPI error).

**System action:** IMS terminates abnormally.

**Programmer response:** See message DFS0604W. Be sure the data set identified by DDNAME *xxxxxxx* has RECFM=F or RECFM=FB.

**Problem determination:** 1, 3, 6, 8

**Source:** IMS abends

**Module:** DFSXRPS0, DFSIILD0, DFSRED20, DFSRRC10, DFSRRC40

**Related information:**

 DFS0604W (Messages and Codes)

Chapter 2, "IMS problem determination," on page 13

---

## 0011

**Explanation:** An attempt was made to enter the system console device-dependent module entry point to handle a WRITE interrupt. The system console device module, DFSDNSC0, was called by module DFSICIO0 to handle a WRITE interrupt. This entry point does not exist.

**Analysis:** This is a standard abend issued by the communication device module, DFSDNSC0, for the system console. The program status word (PSW) at entry-to-abend points to the instruction within the label ABEND from which the abend (SVC 13) is issued.

Register 12 in the abend SVRB registers is the base register. Register 14 at the time of abend contains the invalid entry vector value from the communication analyzer (DFSICIO0). This code is used as an index to a branch table to handle the condition.

The communication analyzer, DFSICIO0, calls device-dependent module entry 2 (WRITE INTERRUPT) at label OUTINT. This routine puts an entry vector value in register 14. The only *valid* contents of register 14 that DFSDNSC0 accepts are:

#### Codes

##### Code (hex)

X'00'

For a WRITE SETUP

X'08'

For a READ SETUP

X'0C'

For a READ INTERRUPT

An X'04' or X'10' causes a branch to abend.

Key	Label	Description
Reg6=address of WTOR dsect Reg14=X'04' or X'10'Reg1=Abend completion code, X'800000B'	DENTRY	This routine detected an invalid entry vector value in register 14 (either a X'04' or a X'10'), and branches to label ABEND to terminate.

**System action:** IMS terminates abnormally.

**Programmer response:** Make certain that CSECT DFSICIO0 has not been user-modified, nor any other user modification branches to CSECT DFSDNSC0.

**Problem determination:** 1, 2, 3, 4, 5, 11, 12, 35

**Source:** IMS abends

**Module:** DFSDNSC0

## 0012

**Explanation:** The data set cannot be processed because the block size is invalid. This prevents IMS from acquiring information required for initialization. The system terminated abnormally for one of the following reasons:

- The block size is not valid for the data set. Message DFS0605W, issued prior to this abend, identifies the data set by DDNAME xxxxxxxx and provides additional information.
- Message DFS0605W was returned, to indicate that the block size of the DDNAME data set was not valid. If this message was returned during initialization, the system might not have received needed information. If this message was received from the MFS DCT Utility, the utility terminated with return code 4.

**Analysis:** This is a standard abend issued by DFSXRPS0, DFSIILD0, and DFSRED20. These modules check that the DCBBLKSI field is a multiple of 80. Check the DCBBLKSI field to determine the block size. Ensure that the JCL points to the correct PROCLIB data set. For DFSXRPS0, DFSIILD0 and DFSRED20, register 6 contains the DCB address. For DFSXRPS0, register 9 has the DCBBLKSI value and register 8 has the remainder. For DFSIILD0 and DFSRED20, register 5 has the DCBBLKSI value and register 4 has the remainder.

Module	Key	Label	Description
DFSXRPS0	Reg6=DCB address Reg9=DCBBLKSI Reg8=remainder	XBLKERR	The routine that validates DCB block size expects the block size to be a multiple of 80.

Module	Key	Label	Description
DFSILDO0	Reg6=DCB address Reg5=DCBBLKSI Reg4=remainder	BLKSZERR	The routine that validates DCB block size expects the block size to be a multiple of 80.
DFSRED20	Reg6=DCB address Reg5=DCBBLKSI Reg4=remainder	XBLKERR	The routine that validates DCB block size expects the block size to be a multiple of 80.

**System action:** IMS terminates abnormally.

**Programmer response:** Refer to message DFS0605W. Be sure the data set identified by DDNAME *xxxxxxx* has a block size that is a multiple of 80.

**Problem determination:** 1, 3, 6, 8

**Source:** IMS abends

**Module:** DFSXRPS0, DFSILDO0, DFSRED20

**Related information:**

Chapter 2, "IMS problem determination," on page 13

## 0013

**Explanation:** IMS terminated abnormally for one of these reasons:

- IMS could not open the system log for IMS batch. Reg 15 = 1.
- The output system log data set (SLDS) IEFORDER/2 is allocated to an extended format data set. Reg 15 = 2.
- The macro ISITMGD returned a non-zero return code. Reg 15 = 3.

**Analysis:** 0013 is a standard abend that can be issued by DFSFDLS0. The program status word (PSW) at entry-to-abend points to the instruction from which the abend (SVC 13) is issued.

Register 12 in the abend SVRB registers is the base register for this entry (DFSFDLS1) of the module. Register 11 and register 10 contain the addresses of the system contents directory (SCD) and log control directory (LCD). Register 14 contains the address from which it was found that the primary log data set could not be opened.

Key	Label	Description
Reg5=A (DCB) DCBOFLGS ^= X'10'	OPNB0100	An OPEN (SVC 19) is issued for the primary DCB. The DCBOFLGS field of that DCB is tested to determine if the OPEN was successful. If it was not, this abend is issued.

**System action:** IMS terminates abnormally.

**Programmer response:** The response to this problem depends on the value of register 15 as follows:

- If Reg 15 = 1, make sure that you have included the DD statement for the IEFORDER and specified it correctly. Then resubmit the job.
- If Reg 15 = 2, reallocate the output SLDS (IEFORDER/2) to a non-SMS striped data set.
- If Reg 15 = 3, check the ISITMGD return code in register 15. For details on ISITMGD return codes, see *z/OS DFSMS Macro Instructions for Data Sets*.

**Source:** IMS abends

**Module:** DFSFDLS0

**Related reference:**

 z/OS: ISITMGD completion codes

## 0014

**Explanation:** The IMS system logger received an invalid log call. Register 15 contains a return code.

**Analysis:** For DFSFLLG0:

This is a standard abend that can be issued by the logical log writer, DFSFLLG0. When issued, the program status

word (PSW) at entry-to-abend points to the instruction within label LOGABEND from which the abend (SVC 13) is issued.

Register 12 in the abend SVRB registers is the base register for this module. Register 11, register 10, and register 9 contain the addresses of the system contents directory (SCD), log control directory (LCD), and DECB, respectively. Register 15 contains the reason code for this abend.

#### Code (hex)

##### Explanation

X'04'

Record length invalid

X'08'

Invalid parameter

X'0C'

Invalid request

X'10'

Latch requested when the latch was already held

X'14'

Latch release requested when the latch was not owned

X'20'

Incorrect post code

X'24'

No more save areas available

In addition, the save area trace shows the module that called the IMS logger.

Key	Label	Description
Reg15=X'04' Reg5=length or Reg6=A(DFSPRMLL)	LLRET4	In the routine to transfer a log record to a buffer, the record length is less than 5 bytes or greater than log BLKSIZE. When the log parameter list is passed (DFSPRMLL), the record length is not positive and the caller did not specify an exit routine.
Reg15=X'08' Reg9=A(DECB) Reg6=(DFSPRMLL)	LLRET8	The parameter passed to the DECB was invalid. One of the following conditions exists: function code is invalid, no record pointer, no AWE pointer in DFSPRMLL, invalid combination of request flags in DFSPRMLL, invalid return code from an exit routine, or top segment length was less than 4.
Reg15=X'0C' Reg9=A(DECB) Reg10=A(LCD)	LLRET12	At the entry point in this module, an undefined function was requested, or the log was not opened. During restart, the required setup was not done prior to this call. In the routine of the release log latch, the caller was not the owner of the latch.
Reg15=X'20'	LLRET32C	The logical logger was waiting for a post from the physical logger. The event control block (ECB) has been posted with an invalid post code value.
Reg15=X'24'	LETERM	Another save area is needed to continue. The forward save area pointer is 0, which indicates that no more save areas are available.

#### For DFSCMS00:

The Multiple Systems Coupling (MSC) analyzer issued a call to the logger to log a X'64F2' log record. The log record size is larger than the input buffer.

Key	Label	Description
Reg15=X'04' Reg9=A(DECB)		The log record length is invalid.

## 0015

**System action:** IMS terminates abnormally.

**Operator response:** Save the log data set. Contact your IMS system programmer to restart.

**Programmer response:** The user can emergency restart IMS. The save area trace shows who called the logger with the invalid request.

*For APAR processing:* Include a copy of the log and system dump with the save area trace.

**Problem determination:** 1, 4, 5, 35

**Source:** IMS abends

**Module:** DFSFLLG0, DFSCMS00

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

## 0015

**Explanation:** The data set does not contain the required member. This prevents IMS from acquiring information required for initialization. Message DFS0579W, DFS0596W, DFS3652X, or DFS3659X identifies the name of the member not found and provides additional information.

The reasons for termination are as follows:

### DFS3652X

System initialization cannot find the required dynamic terminal or dynamic user descriptors from the library with the ddname PROCLIB. System initialization requires at least one valid logon descriptor and one valid user descriptor when DYNT=YES is requested. Neither DFSDSCT $y$  (where  $y$  is specified by the DSCT=EXEC parameter) nor DFSDSCM $y$  (where  $y$  is the nucleus suffix specified by the SUF=EXEC parameter) was found in IMS.PROCLIB, or did not contain at least one valid logon descriptor and one valid user descriptor.

### DFS3659X

System initialization received an I/O error while reading the descriptor records for the descriptor name from the IMSVS.PROCLIB member.

**Analysis:** This is a standard abend issued by DFSXRPS0, DFSIILD0, and DFSRED20. These modules check for members that are required by IMS. Be sure the JCL points to the correct PROCLIB data set. For DFSXRPS0, DFSIILD0, and DFSRED20, register 6 contains the DCB address.

Key	Label	Description
Reg15=X'08' Reg6=DCB address Reg2=member address	XOPVSMNM	The routine that validates members was unable to find the required member.

**System action:** IMS terminates abnormally.

**Programmer response:** Refer to messages DFS0579W, DFS0596W, DFS0671W, DFS3652X, and DFS3659X.

**Problem determination:** 1, 3, 6, 8

**Source:** IMS abends

**Module:** DFSXRPS0, DFSIILD0, DFSRED20

**Related information:**

 DFS0579W (Messages and Codes)

 DFS0596W (Messages and Codes)

 DFS0671W (Messages and Codes)

 DFS3652X (Messages and Codes)

 DFS3659X (Messages and Codes)

Chapter 2, "IMS problem determination," on page 13



## 0016

**Explanation:** The IMS system terminated abnormally for one of the following reasons:

- The request for storage (in subpool 0 of the IMS control region extended private area) for a hash table for the VTAM control blocks failed. Issued by DFSIINV0 after sending message DFS1996. R3 = length of storage request. R5 = IMODULE return code.
- The VTAM control blocks (DFSCLV $yx$ ) could not be initialized ( $y$  is a value from 0 to 9 or A to F;  $x$  is the IMS nucleus suffix). The abend is issued by DFSIINV0 after sending message DFS1998.
- One of the following occurred:
  - The VTAM control block modules could not be loaded. The abend is issued by DFSIINV0 after sending message DFS1999. R5 = IMODULE return code.
  - All of the VTAM control block modules could not be loaded in the IMS control region extended private area. Issued by DFSIINV0 after sending message DFS1999 RC=08. R5=8.
- The request for storage in subpool 0 of the IMS control region extended private area for a hash table for the CNT/LNB/RCNT, SPQB or CCB control blocks was unsuccessful. Issued by DFSIINU0 after sending message DFS1992. R3 = length of storage request. R5 = IMODULE return code.
- The request for storage in subpool 0 of the IMS control region extended the private area for a CCB control block bit map was unsuccessful. The abend is issued by DFSIINU0 after sending message DFS1992, with R3 = length of storage request and R5 = IMODULE return code.
- The LQB/RCNT control block modules (DFSCLC $ms$ , DFSCLS $ms$ , DFSCLR $ms$ , DFSCLI $Ds$ , where  $m$  is a value from 0 to 9 or A to F;  $s$  is the IMS nucleus suffix) could not be initialized. The abend is issued by DFSIINV0 after sending message DFS1990.
- The LQB/RCNT control block modules (DFSCLC $ms$ , DFSCLS $ms$ , DFSCLR $ms$ , DFSCLI $Ds$ , where  $m$  is a value from 0 to 9 or A to F;  $s$  is the IMS nucleus suffix) could not be loaded. The abend is issued by DFSIINV0 after sending message DFS1991, with R5 = IMODULE return code.
- The request for storage in subpool 0 of the IMS control region extended private area) for a work area or for system logon and user descriptors for creation of temporary structures was unsuccessful. The abend is issued by DFSIILD0 after sending message DFS1993, with R3 = length of storage request and R5 = IMODULE return code.
- The request for storage in subpool 0 of the IMS control region extended private area) for hash tables for logon and user descriptors was unsuccessful. The abend is issued by DFSIILD0 after sending message DFS1993, with R4 = length of storage request and R5 = IMODULE return code.
- The request for storage in subpool 214 of the IMS control region extended private area) for the VTAM control blocks failed. The abend is issued by DFSIINV0 after sending message DFS1996, with R4 = length of storage request and R5 = IMODULE return code.

**Analysis:** This abend is preceded by one of seven error messages: DFS1990I, DFS1991I, DFS1992I, DFS1993I, DFS1996I, DFS1998I or DFS1999I. Refer to the appropriate message to determine the required action.

**System action:** IMS terminates abnormally.

**Programmer response:** See message DFS1990I, DFS1991I, DFS1992I, DFS1993X, DFS1996I, DFS1998I, or DFS1999I to determine the appropriate action.

**Problem determination:** 4, 6, 8, 10

**Source:** IMS abends

**Module:** DFSIINV0, DFSIINU0, DFSIILD0

**Related information:**

 DFS1990I (Messages and Codes)

 DFS1991I (Messages and Codes)

 DFS1992I (Messages and Codes)

 DFS1993X (Messages and Codes)

 DFS1996I (Messages and Codes)

 DFS1998I (Messages and Codes)

 DFS1999I (Messages and Codes)

Chapter 2, "IMS problem determination," on page 13

## 0017

**Explanation:** The IMS system log writer detected an irregular sequence of the log buffers for write and read operations.

**Analysis:** This is a standard abend that can be issued by the post buffer process routine, DFSFDLB0. When issued, the program status word (PSW) at entry-to-abend points to the instruction within label ABEND017 which issues the abend (SVC 13).

Register 12 in the abend SVRB registers is the base register for this module. Register 11 and register 10 contain the address of the system contents directory (SCD) and the log control directory (LCD), respectively. Register 14 contains the address at which this abend condition was first detected. Register 15 contains the reason code, indicating an internal error, for the abend.

Key	Label	Description
Reg15=X'08' Reg9=A(LBUFFER)	WRTE0700	The checked buffer was not in the work-to-do queue chain (LPWKTDQ in LCD), which is used to maintain the order in which the buffer should be written to the system log.
Reg15=X'0C' Reg9=A(LBUFFER) Reg8=A(LDSET)	WRTE1700	The checked block was not the next block that was last written to the log data set. LBBKLCNT in the LBUFFER (register 9) was not equal to LDSBSEQL+1 in LDSET (register 8).
Reg15=X'10' Reg9=A(LBUFFER)	READ0200 READ1300	The read buffer was not in the proper chain from the LDSET. The used OLDS must be in the DSET table (LDSETPTR in the LCD) and must be in the chain from LDSRBUFF in LDSET.

**System action:** IMS terminates abnormally.

**Operator response:** Emergency restart IMS to resume processing.

**Problem determination:** 1, 4, 5, 35

**Source:** IMS abends

**Module:** DFSFDLB0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

## 0019

**Explanation:** The component number in the Communication Name Table (CNT) was greater than the maximum allowed, as determined by the routine that sets the appropriate component-inoperable bit in the communication terminal block (CTB).. Register 10 contains the address of the CNT in error.

**Analysis:** This is a standard abend issued by the SESSIONC completion router, DFSCVRB0, to set the COMPINOP bit in the CTB based on the component number in the CNT. The program status word (PSW) at entry-to-abend points to the instruction in module DFSCVRB0, within label ABEND19, from which the abend (SVC 13) is issued.

Register 10 in the abend SVRB registers contains the address of the CNT in error. The invalid component number can be located at field CNTCMPNT in the CNT. Only four components are allowed; the only valid values for this field are X'00', X'01', X'02', and X'03'.

Key	Label	Description
Reg10=address of CNT Reg7=address of CTB Reg1=Abend completion code, X'80000013'	VRB05	A compare is made of the output component number (CNTCMPNT) in the CNT. If the value is higher than X'03', a branch is taken to label ABEND19 to abend.

**System action:** IMS terminates abnormally.

**Operator response:** Emergency restart IMS to resume processing.

**Source:** IMS abends

**Module:** DFSCVRB0

## 0020

**Explanation:** The IMS system or the specified external subsystem task was terminated by the use of the MODIFY command.

**Analysis:** 0020 is a standard abend issued by DFSFCTT0.

**Possible cause:** A user entered a z/OS MODIFY command to terminate abnormally all of IMS or a specific task representing an external subsystem connection, such as DB2®.

**System action:** All of IMS or the specified external subsystem task terminates. If an external subsystem was specified, only that task will terminate, while the remainder of IMS continues processing.

**Operator response:** Restart IMS if an external subsystem task terminated. You can restart it by issuing the /START SUBSYS command.

**Programmer response:** You should determine the cause of modifying IMS and take corrective action.

**Source:** IMS abends

**Module:** DFSFCTT0

## 0021

**Explanation:** The IMS system log writer was passed an invalid request or detected an illegal condition. An invalid request element was posted to the physical log master ITASK or the set up ITASK, or an invalid log record address was passed to the RSR log data send routine in batch.

**Analysis: For DFSFDLG0:**

This is a standard abend that can be issued from the physical log writer-master routine, DFSFDLG0. When issued, the program status word (PSW) at entry-to-abend points to the instruction from which the abend (SVC 13) is issued.

Register 12 in the abend SVRB registers is the base register for these modules. Register 11 and register 10 contain the addresses of the system contents directory (SCD) and the log control directory (LCD), respectively. Register 14 contains the address at which the abnormal condition occurred. A reason code indicates an internal error.

Key	Label	Description
Reg15=X'04' Reg2=A(AWE)	MAIN0600	While processing the routine in the AWE request, the AWLFUNC field contained an invalid code.
Reg15=X'08'	MSTRCLSE MSTRFEOV	In the FEOV/CLOSE log data set routine, one or more buffers still remain in the work-to-do-queue (LPWKTDQ in LCD). Prior to FEOV/CLOSE, this field must be 0.
Reg15=X'0C'   Reg9=A(OLDS   buffer prefix)	WRTE1700	During OLDS write processing, register 9 points to a buffer prefix whose block sequence number is not 1 greater than the block sequence number written to the prior OLDS.
Reg15 = X'10'   Reg6 = A(ECB) that   failed INITECB   Reg7 = Return code   from INITECB   Reg9 = A(Buffer prefix)	CKCP0817 WRTE1110	An INITECB call returned a non-zero return code for an ECB associated with a Logger ITASK.

**For DFSFDLS0:**

This is a standard abend that can be issued from the physical log writer—setup routine, DFSFDLS0. When issued, the program status word (PSW) at entry-to-abend points to the instruction from which the abend (SVC 13) is issued.

## 0022

Register 12 in the abend SVRB registers is the base register for this module. Register 11 and register 10 contain the address of the system contents directory (SCD) and the log control directory (LCD), respectively. Register 14 contains the address at which this abnormal condition occurred. Register 15 contains the reason code, indicating an internal error, for this abend.

Key	Label	Description
Reg15=X'04' Reg2=A(AWE)	SUBT0600	In the routine that processes the AWE request, DFSFDLS0, the AWLFUNC field contained an invalid code.
Reg15=X'08'	SOPN0200	In the routine that first opened the next OLDS, DFSFDLS, there was no DCB available in the LCD. The LCD contained 2 sets of DCBs (LDCBP1/LDCBS1 and LDCBP2/LDCBS2), but both sets were in an opened status (DCBOFLGS-X'10').
Reg15=X'0C'	DFSFDLS5	In the timer exit routine, DFSFDLS5, the target SCD could not be located from the current TCB. The register 1 field of the first register save area must point to the IMS dispatcher work area.

### For DFSLLRD0:

This is a standard abend that can be issued from the RSR log data send routine, DFSLLRD0. Register 15 contains the reason code for this abend.

Key	Label	Description
Reg15=X'10'	None.	In a batch IMS region, the address of the log data to be sent to the RSR tracker was outside of the boundary of the IMS log buffers.

**System action:** IMS terminates abnormally.

**Operator response:** Emergency restart IMS to resume processing.

**Problem determination:** 1, 4, 15, 35

**Source:** IMS abends

**Module:** DFSFDLG0, DFSFDLS0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

## 0022

**Explanation:** On SESSIONC completion, the session control code in the request parameter list (RPL) is invalid (not STSN, SDT, or CLEAR).

**Analysis:** This is a standard abend issued by the 3770/3767 SESSIONC completion router, module DFSCVRG0. The program status word (PSW) at entry-to-abend points to the instruction within label ABEND from which the abend (SVC 13) is issued.

Register 12 in the abend SVRB registers is the base register. Register 6 contains the address of the VTAM buffer containing the RPL in error. Register 15 contains one of the following return codes:

### Codes

#### Explanation

**X'00'**

CLEAR completion.

**X'04'**

Start data traffic (SDT) completion.

**X'08'**

Set and test sequence number terminal (STSN) completion.

Key	Label	Description
Reg6=address of VTAM buffer with RPL in error Reg1=Abend completion code, X'80000016'	CVRG010	The function code is compared to the operation type. If the operation is not a CLEAR, an SDT, or an STSN, a branch is taken to abend.

**System action:** IMS terminates abnormally.

**Programmer response:** No action is required.

**Problem determination:** 11, 35

At the time of the abend, register 6 contains the address of the buffer that contains the RPL in error.

**Source:** IMS abends

**Module:** DFSCVRG0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

## 0023

**Explanation:** The DL/I subordinate address space option was selected. This abend indicates that the specification for the ACBLIB data sets in the IMS procedure did not match the specification in the DL/I subordinate address space procedure. Message DFS0404W precedes this abend and provides further information.

**Analysis:** The specification of the ACBLIB DDNAMES IMSACBA and IMSACBB must be corrected. The ACBLIB data sets (both active and inactive) specified in the DL/I subordinate address space procedure must be the same as specified in the IMS procedure and the concatenation order must be identical.

Key	Label	Description
Refer to discussion below.	DLI60	Refer to discussion below.

The z/OS Scheduler Work Area blocks must be included in the memory dump. To include these blocks, specify SDATA=SWA on the z/OS CHNGDUMP command.

Message DFS404W precedes this abend and provides reason codes. Applicable data areas are as follows:

- - FOR THE CONTROL MEMORY:
    - ACBCK in module DFSSDLC0 points to the address of DFSACBCK.
    - DFSACBCK+8 is the address of the DSNAME/VOLSER list for IMSACBA.
    - DFSACBCK+X'10' is the address of the list for IMSACBB.
- - FOR THE DL/I MEMORY:
    - AATIOT in module DFSSDLC0 is the address of the first TIOT entry for
    - IMSACBA. BBTIOT is the address for IMSACBB.

If message DFS0404W indicates reason code 02:

- AAERTIOT is the address of the TIOT entry associated with IMSACBA that did not match the IMS procedure specification.
- AAERJFCB is the address of the job file control block (JFCB) associated with AAERTIOT.
- AAERCTL is the address of the data for the control memory used for the compare.
- BBERTIOT, BBERJFCB, and BBERCTL are used for IMSACBB.

**System action:** The DL/I subordinate address space terminates abnormally with a 0023, causing the control region to terminate with a 0150 abend.

**Programmer response:** The ACBLIB data sets specified in the DL/I subordinate address space start-up procedure must be the same as specified in the control region JCL and the concatenation order must be identical. This applies to both the active and inactive ACBLIBs. Correct the specification of the ACBLIB DDNAMES IMSACBA and IMSACBB. If it is necessary to analyze the storage dump, the MVS™ Scheduler Work Area blocks must be included. Specify SDATA=SWA on the z/OS CHNGDUMP command to include these blocks.

## 0024 • 0026

**Source:** IMS abends

**Module:** DFSSDLCO

**Related information:**

 DFS0404W (Messages and Codes)

IMS abend 0150

---

### 0024

**Explanation:** Either a DL/I subordinate address space region or a DBRC region attempted to connect to the IMS control region. The jobname passed on the connection request did not match the DLINM= (for the DL/I subordinate address space region) or the DBRCNM= (for the DBRC region) specification. These values are obtained from the IMSCTRL system definition macro or from the control region JCL. The defaults for these keywords are DLISAS and DBRC, respectively.

This abend is issued from label TESTSAS.

**System action:** The region that attempted the connection abends with 0024.

**Programmer response:** Start a procedure with the correct name, or restart the control region, overriding the expected name.

**Source:** IMS abends

**Module:** DFSRRA00

---

### 0025

**Explanation:** Either a DL/I subordinate address space region or a DBRC region attempted to connect to the IMS control region. The control region already has an active connection to that region type. This error is caused, for example, by starting the DL/I subordinate address space region twice.

This abend is issued from label TESTSAS.

**System action:** The region that attempted the connection terminates with abend 0025.

**Programmer response:** No action is required.

**Source:** IMS abends

**Module:** DFSRRA00

---

### 0026

**Explanation:** IMS was unable to successfully issue the (E)STAE macro. This problem is an IMS system error.

**Analysis:** 0026 is a standard abend that can be issued from three different modules: DFSPCC20, DFSPCC30, or DFSPR000.

The program status word (PSW) at entry-to-abend isolates the failure to a particular module. The PSW points to the label/routine within the failing module from which the abend (SVC 13) is issued. Register 15 in the abend SVRB registers always contains the return code from the unsuccessful (E)STAE (SVC 60):

#### Codes

##### Explanation

**X'00'**

Successful completion of (E)STAE request.

**X'04'**

(E)STAE OV was specified with a valid exit address, but the current exit routine is either nonexistent, not owned by the user's request block (RB), or is not an (E)STAE exit routine.

**X'0C'**

Cancel (an exit address equal to zero) was specified and either there are no exit routines for this task control block (TCB), the most recent exit routine is not owned by the caller, or the most recent exit routine is an (E)STAE exit.

**X'10'**

An unexpected error was encountered while processing this request.

**X'14'**

(E)STAE was unable to obtain storage for a session control block (SCB).

**For DFSPCC20:**

0026 is a standard abend that can be issued by the MPP/BMP program controller, DFSPCC20. In this instance, the program status word (PSW) at entry-to-abend points to the instruction within label PSABEND from which abend (SVC 13) is ultimately issued, and is branched to by the routine that detected the error.

At the time of abend, register 15 in the abend SVRB registers contains the nonzero return code from the unsuccessful (E)STAE.

Key	Label	Description
Reg3=address of (E)STAE exit routine	PC00	During initialization of the interregion communication facility, an unsuccessful (E)STAE (SVC 60) was issued. Register 15 is tested for a return code; anything other than zero causes a branch to label PCAB026 to handle the abend.
Reg1=completion code, X'8000001A'	PC01	

**For DFSPCC30:**

0026 is a standard abend that can be issued by the batch application program controller, DFSPCC30. The program status word (PSW) at entry-to-abend points to the instruction within label PSABEND from which the abend (SVC 13) is issued. It is branched to by the routine within this module that detected the error.

At the time of abend, register 15 in the abend SVRB registers contains the nonzero return code from the unsuccessful (E)STAE (SVC 60).

Key	Label	Description
Reg15=nonzero return code from (E)STAE Reg1=completion code, X'8000001A'	PC1A	During batch initialization, a (E)STAE was issued that was not successful. Register 15 is loaded with the return code and, because it is nonzero, a branch is taken to label PSTAEND to handle the abend.

**For DFSPR000:**

The abend from this module occurs in non-z/OS systems *only*.

This abend is a standard abend that can be issued by the batch application program request handler, DFSPR000. In this instance, the program status word (PSW) at entry-to-abend points to the instruction within label PRABEND from which the abend (SVC 13) is issued, and is branched to by the routine that detected the error.

Register 15 in the abend SVRB registers contains the nonzero return code from the unsuccessful E(STAE) (SVC 60).

Key	Label	Description
Reg5=pointer to first (E)STAE control block Reg1=completion code, X'8000001A'	STATEST	During the setup and checking of initial conditions, a situation was encountered that caused STAE to be issued (that is, not the first DL/I call; no STAE active; the current STAE does not belong to this module). The STAE was not successful, as indicated by the nonzero return code in register 15, and a branch is taken to label PRAB3 to handle the abend.

**System action:** IMS terminates abnormally.

## 0027 • 0029

**Programmer response:** This problem is an IMS internal error.


**Problem determination:** 4, 35

Register 15 contains the return code from (E)STAE.

**Source:** IMS abends

**Module:** DFSPCC20, DFSPCC30, DFSPR000

**Related reference:**

 z/OS: ESTAE return and reason codes

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

### 0027

**Explanation:** While processing a DFSDUMP request, the IMS dump service routine received a system or IMS abnormal termination request. This abend is issued after the dump process is complete to force normal IMS recovery and termination. The reason code field in the first line of message DFS629I contains the original abend code.

**System action:** The error is passed to the normal IMS recovery routines, the current DFSDUMP request is forced to TERM=YES, and a software LOGREC entry is requested for the original termination request.

**Programmer response:** To analyze the original abend referred to by abend 0027, use both the LOGREC entry for the original abend and the dump in progress when the original abend was deferred.

**Source:** IMS abends

---

### 0028

**Explanation:** The IMS control region terminated because the DUMP keyword was included in the /CHECKPOINT command.

This abend is a result of a /CHE FREEZE, PURGE, or DUMPQ ABDUMP command.

**Analysis:** 0028 is a standard abend issued at normal termination of the IMS control region by module DFSTERM0. This abend is not the result of any IMS or programming error, but is merely the result of a request by the IMS operator, who requested a memory dump for diagnostic or informational purposes. IMS issues the abend in order to produce the dump and terminate *normally*.

Register 11 in the abend SVRB registers contains the SCD address. Register 13 contains the save area address.

Key	Label	Description
Reg11=address of SCD	SKIPCLSE	The SCDSTOP1 + 1 field of the SCD is tested with a X'10' to determine if a dump was requested at the time of control region termination. If set, the abend is issued.

**System action:** IMS issues the abend, produces a memory dump, and terminates normally.

**Programmer response:** Determine the reason why the operator specified the DUMP option to the /CHECKPOINT command.

**Problem determination:** None.

**Source:** IMS abends

**Module:** DFSTERM0

---

### 0029

**Explanation:** Abend 0029 is a standard abend issued by DL/I subordinate address space initialization. It is issued when a requested service, either IMS or z/OS, returns a nonzero return code. At the time of the abend, register 3 contains a reason code. Register 15 contains the return code from the requested service. Register 14 points to the location where the error was detected.



Analysis: For DFSXDL00:

Table 2. Codes for 0029 from module DFSXDL00

Key	Label	Description
Reg3=X'01'	ABCOD1	IMODULE GETMAIN failed for temporary save sets.
Reg3=X'02'	ABCOD2	IMODULE GETMAIN failed for an initialization parameter block.
Reg3=X'03'	ABCOD3	Unable to load module DFSFDLI0.
Reg3=X'04'	ABCOD4	The job step TCB ESTAE was not established.
Reg3=X'05'	ABCOD5	DFS4200 was started to build the SSETs for the DL/I subordinate address space.
Reg3=X'06'	ABCOD6	OS load for module DFSDRCL0 failed.
Reg3=X'07'	ABCOD7	Module DFSXDL00 is processing the DL/I subordinate address space preload list (DFSXDLLL) and could not load a required module. Register 4 points to the DLST entry. The first 8 bytes of this entry are the module name.
Reg3=X'08'	ABCOD8	Unable to load DFSFXC10.
Reg3=X'09'	ABCOD9	Unable to load DFSSDL80.
Reg3=X'0A'	ABCOD10	DFSBCB00 was called to obtain a QSAV block.
Reg3=X'0B'	ABCOD11	DFSKDP00 was called to create the dispatcher work area for the DL/I subordinate address space job step TCB.
Reg3=X'0C'	ABCOD12	Unable to load DFSBCB60.
Reg3=X'0D'	ABCOD13	DFSBCB00 was called to obtain a CMWU block.
Reg3=X'0E'	ABCOD14	DFSCIR00 was called to create an ITASK structure for storage compression.
Reg3=X'0F'	ABCOD15	Unable to load DFSXDL10.
Reg3=X'10'	ABCOD16	DFSCWU00 was called to create the background write ITASK structure.
Reg3=X'11'	ABCOD17	Unable to load DFSSDL40.
Reg3=X'12'	ABCOD18	DFSCWU00 was called to create the DFSSDL40 ITASK structure.
Reg3=X'13'	ABCOD19	Unable to obtain the common services work area (DFSCSSWK).
Reg3=X'14'	ABCOD20	Unable to load DFSCSS00
Reg3=X'15'	ABCOD21	DFSCIR00 was called to create the common services ITASK structure.
Reg3=X'16'	ABCOD22	DFSBCB00 was called to release a QSAV block.
Reg3=X'17'	ABCOD23	Unable to release temporary save sets.
Reg3=X'18'	ABCOD24	Unable to load DFSXRPS0
Reg3=X'19'		An authorization index reserve (AXRES) request failed.
Reg3=X'1A'		An authorization index set (AXSET) request failed.
Reg3=X'1B'		A linkage index reserve (LXRES) request failed.
Reg3=X'1C'		An entry table create (ETCRE) request failed.
Reg3=X'1D'		An entry table connect (ETCON) request failed.
Reg3=X'1E'		An authorization table set (ATSET) request failed.
Reg3=X'20'		A load for module DFSNOTB0 failed.
Reg3=X'21'		A load for module DFSCPY00 failed.
Reg3=X'22'	ABCOD34	IMODULE LOAD for DFSRSM0 failed.
Reg3=X'23'	ABCOD35	DFSRSM0 call failed.
Reg3=X'24'	ABCOD36	IMODULE LOAD for DFS5DL60 failed.

Table 2. Codes for 0029 from module DFSXDL00 (continued)

Key	Label	Description
Reg3=X'25'	ABCOD37	A DFSCWU call failed to create an ITASK for DFS5DL60.

**For DFSXDL10:**

Table 3. Codes for 0029 from module DFSXDL10

Key	Label	Description
Reg3=X'01'	ABCOD1	Module DFSIINS0 was called to perform storage management initialization.
Reg3=X'02'	ABCOD2	Unable to GETMAIN a work area for the DFSSDL20 ITASK.
Reg3=X'03'	ABCOD3	Unable to load DFSSDL20.
Reg3=X'04'	ABCOD4	DFSBCB00 was called to obtain a QSAV block.
Reg3=X'05'	ABCOD5	DFSCIR00 was called to create the DFSSDL20 ITASK structure.
Reg3=X'06'	ABCOD6	DFSBCB00 was called to release a QSAV block.
Reg3=X'07'	ABCOD7	DFSIIIND0 was called to perform database initialization.
Reg3=X'07'	ABCOD7B	DFSRDB30 was called to perform program specification block (PSB) and data management block (DMB) directory initialization.
Reg3=X'08'	ABCOD8	DFSDVBI0 was called to perform OSAM and VSAM buffer initialization.
Reg3=X'09'	ABCOD9	DFSIFIX0 was called to process page fix options.
Reg3=X'0A'	ABCOD10	Unable to load DFSNOTB0.
Reg3=X'0B'	ABCOD11	DFSBCB00 was called to obtain a QSAV block.
Reg3=X'0C'	ABCOD12	DFSBCB00 was called to release a QSAV block.
Reg3=X'0D'	ABCOD13	Unable to load DFSSDL30.
Reg3=X'0E'	ABCOD14	DFSSDL30 was called to initialize.
Reg3=X'0F'	ABCOD15	DFSBCB00 was called to free an asynchronous work element (AWE).
Reg3=X'10'	ABCOD16	DFSXRAC0 was unable to set RACF <sup>®</sup> user ID (see information about message DFS0841I for a description of the specific problem.)

**For DFSXDT10:**

The following return codes are issued in decimal format by DFSXDT10:

**Codes****Explanation**

- 81** An error return code was returned from DFSBCB GET indicating that SAVEAREA could not be got during initialization.
- 82** An error return code was returned from DFSCDSP indicating that dispatcher work area could not be created.
- 83** During initialization, an error return code was returned from IMODULE LOAD indicating that the tracking module cannot be loaded.
- 84** During initialization, an error return code was returned from DFSCWU indicating that the tracking ITASK could not be created.
- 85** During initialization, an error return code was returned from IMODULE GETMAIN indicating that storage could not be obtained in subpool 0 for the DFSXDT10 work area.
- 86** During initialization, an error return code was returned from IMODULE GETMAIN indication that storage could not be obtained in subpool 0 for the termination ITASKS.
- 87** During initialization, an error return code was returned from DFSBCB GET indicating that storage could not be obtained for the save area needed by the termination ITASKS.

**88** During initialization, an error return code was returned from DFSCIR indicating that termination ITASKS could not be created.

**For DFSSDLC0:**

*Table 4. Codes for 0029 from DFSSDLC0*

Key	Label	Description
Reg3=X'01'	ABCOD1	Unable to obtain common storage area (CSA) space for the ACBLIB check area.

**System action:** The DL/I subordinate address space region terminates with abend 0029, causing the control region to terminate with abend 0150.

**Programmer response:** Determine the cause using the tables above and correct the problem.

**Source:** IMS abends

**Module:** DFSXDL00, DFSXDL10, DFSXDT10, DFSSDLC0

**Related information:**

 DFS0841I (Messages and Codes)

IMS abend 0150

## 0031

**Explanation:** While trying to locate a block in error on one log data set, CSECT DFSFLTP0 returned a nonzero return code in register 15.

**Analysis:** 0031 is a standard abend issued by the Log Recovery utility, which is a composite module. DFSULTR0 is the load module name, and DFSULG10 is its entry CSECT. If DFSULG20 encounters a read error on the current input log, it BALRs to module DFSFLTP0, which puts the error return code into register 15 and returns to DFSULG20.

At the time of failure, the program status word (PSW) at entry-to-abend points to the instruction within label SWAPABND in CSECT DFSULG20, from which ABEND (SVC 13) is issued. Register 15 in the abend SVRB registers contains the return code from module DFSFLTP0. Return codes can be determined by looking at the branch table at label SWAP1040 in module DFSULG20.

**System action:** Module DFSULG20 terminates abnormally with a memory dump.

**Programmer response:** Check the return code in register 15 and correct the problem.

**Source:** IMS abends

**Module:** DFSULG20

## 0032

**Explanation:** A log sequence error was detected by the Log Recovery utility.

**Analysis:** 0032 is a standard abend issued by the Log Recovery Utility. If DFSULG40 encounters an erroneous log, it issues a DFS3288I message and abends. The detected errors are an incorrect sequence number or a log record that is too short. The error can be determined from the DFS3288I message issued.

At the time of failure, the program status word (PSW) points to the instruction from which the abend (SVC 13) was issued. Register 2 contains the address of the log record, register 3 contains the input log DCB address, and register 4 contains the record length. If the sequence number is incorrect, register 5 contains the expected log sequence number (LSN) and register 6 contains the LSN received.

**System action:** The Log Recovery utility issues abend 0032.

**Programmer response:** Make sure that the order of the log data sets in the JCL is correct. If the order is correct, correct the log data set using the DUP and the REP modes of the Log Recovery utility. Then rerun the job.

**Source:** IMS abends

**Module:** DFSULG40

## 0034 • 0035

### Related information:

 [DFS3288I \(Messages and Codes\)](#)

---

### 0034

**Explanation:** Module DFSSCBT0 or DFSBCB30 was entered to get or release the VTCB, VLQB, or CBTS latch. In the case of a get-latch request, the common latch manager module DFSCLM00 returned with a nonzero return code. In the case of a release-latch request, the common latch manager module DFSCLM10 returned with a nonzero return code.

**Analysis:** 0034 is a standard abend issued by module DFSSCBT0 after a bad return code from a GET latch request to module DFSCLM00 or from a RELEASE latch request to module DFSCLM10. Register 15 contains the return code from the latch request.

To determine which latches were owned by the unit of work (UOW) at the time of the abend, check the Common Latch List Element (CLLE) block that is attached to SAP using SAPACLE. Refer to mapping macro DFSCLE for a description of the formatted CLLE area.

The following return codes are issued in decimal format by DFSCLM00:

#### Codes

##### Explanation

**RC = 12**

Requested latch equals current highest-held latch and the resource headers are identical (latch already owned).

**RC = 16**

Requested latch is less than the current highest held latch (hierarchy violation).

**RC = 20**

Unable to grant request for exclusive latch and requester unwilling to wait (WAIT=NO specified).

**RC = 24**

Unable to grant request for shared latch and requester unwilling to wait (WAIT=NO specified).

**RC = 28**

Requested latch equals current highest held latch and the resource headers are *not* identical (internal system error).

The following return code is issued in decimal format by DFSCLM10:

#### Codes

##### Explanation

**RC = 12**

Requested resource is not allocated (there is no owner of this resource in the system).

**RC = 16**

Requested resource is allocated but not to this requester.

**System action:** IMS terminates abnormally with abend 0034.

**Programmer response:** Abend 0034 is identical to abend 0780 issued by the common latch manager (DFSCLM00 and DFSCLM10). See above and abend 0780 for the explanation of the abend subcode found in register 15.

**Source:** IMS abends

**Module:** DFSSCBT0

### Related information:

IMS abend 0780

---

### 0035

**Explanation:** This abend is issued if the Database Control (DBCTL) phase 1 sync point processor (DFSDSC00) detects a failure. DFSDSC00 sets up the abend for a failure in a Fast Path call or a failure by DL/I to perform phase 1 sync point processing.

**Analysis:** If the failure was in Fast Path, then field PAPLSTCD in the PAPL contains the Fast Path status code, which explains the reason for the failure.

If the failure was in DL/I, then the field PAPLSTCD contains blanks and the field PAPLPLRC contains the IMS abend code (set up by DL/I as the reason for the failure).

Both of these fields appear in the phase 1 PAPL in the DRA SDUMP/SNAP. The CCTL can produce diagnostics based on the information returned to it by the DRA SDUMP/SNAP.

**System action:** DBCTL terminates the unit of recovery, backing out any database changes.

**Programmer response:** See the SNAP dump produced by the database resource adapter (DRA). This dump contains the DRA Architected Parameter List (PAPL) and other diagnostic information.

**Problem determination:** 38

**Source:** IMS abends

**Module:** DFSDSC00

**Related information:**

Chapter 2, "IMS problem determination," on page 13

## 0036

**Explanation:** An unexpected posting of a VTAM request parameter list (RPL) resulting from either a VTAM, an IMS logic error, or both occurred.

**Analysis:** 0036 is a standard abend issued by the communication device module, DFSCVCK0, for the VTAM RPL check routine. The program status word (PSW) at entry-to-abend points to the instruction within the label ABEND from which the abend (SVC 13) is issued.

Register 12 in the abend SVRB registers is the base register. Register 9 has the address of the CLB (DECB). Register 1 has the abend completion code, X'80000024'.

Key	Label	Description
Reg9=DECB address	DFSCVCK0	The DECSDECB field of the event control block (ECB) is tested. If the field contains other than a X'00' or a X'40', the ECB is bad, and a branch is taken to abend.

**System action:** IMS terminates abnormally.

**Operator response:** Save the pertinent information for IBM Software Support, and restart IMS.

**Problem determination:** 4, 5, 6, 11, 14, 35

**Source:** IMS abends

**Module:** DFSCVCK0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

## 0037

**Explanation:** During takeover, module DFSHTKRO called Internal Resource Lock Manager (IRLM) for takeover request processing, and IRLM returned a return code greater than 4. Register 15 contains the return code, and field RLPFCODE in the IRLM parameter list contains the reason code.

**Analysis:** 0037 is a standard abend issued by module DFSHTKR0. The return code is contained in register 15, and the reason code is located in the IRLM parameter list field, RLPFCODE. The IRLM parameter list address can be obtained from the partition specification table (PST) field PSTIRMLA. To determine the cause of the failure, see the IRLM request return and reason codes information.

**Possible cause:** A failure in the IRLM is a possible cause of this abend.

**System action:** IMS terminates abnormally.

**Programmer response:** Correct the error and restart the IMS alternate system.


## 0038 • 0039

**Problem determination:** 1, 4, 10, 35

**Source:** IMS abends

**Module:** DFSHTKR0

**Related reference:**

 IRLM request return and reason codes

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

### 0038

**Explanation:** An attempt to release the locks held by the subsystem from the previous execution at the completion of an emergency restart, or database back out execution, resulted in a bad return code from the IRLM. These locks are the locks held by the subsystem from the previous execution. Message DFS038I precedes this abend. Register 15 contains the return code from IRLM. To determine the cause of the failure, refer to:

- The PURGE request explanation in the IRLM request return and reason codes information to determine the cause of the failure.
- The information about message DFS038I.

*Possible cause:* A failure in the IRLM is a possible cause of this abend.

**System action:** IMS terminates abnormally.


**Programmer response:** Correct the problem determined by the return code and restart IMS.

**Problem determination:** 1, 4, 10, 35

**Source:** IMS abends

DFSPCCC0, DFSRST00

**Related reference:**

 IRLM request return and reason codes

**Related information:**

 DFS038I

Chapter 2, "IMS problem determination," on page 13

---

### 0039

**Explanation:** An IDENTIFY request was issued to the IRLM, and the request failed. The return code is contained in register 15, and the reason code is located in the IRLM parameter list field RLIFCODE (offset X'8F'). Refer to the IDENTIFY request in the IRLM request return and reason codes information to determine the cause of the failure.

**Analysis:** 0039 is a standard abend issued by DFSRST00 or DFSRDSH0 (online) or DFSPCCC0 (batch). With the exception of normal restart, in which case a return code of X'04' also causes the abend, any return code greater than X'04' results in this abend. The IRLM parameter list address can be obtained from the partition specification table (PST) field PSTIRLMA. The return code is contained in register 15, and the feedback status information is located in the IRLM parameter list.

If the IRLM is not active, a return code of X'08' and a subcode of X'40' are returned to IMS. ABENDU0039 is issued after the OS operator has responded CANCEL or DUMP to message DFS039A.

If SCOPE=LOCAL was specified and DBRC=YES, IRLM releases the locks. Run back out and retry the identify request process.

**System action:** IMS terminates abnormally.


**Programmer response:** Ensure that the subsystem name (IMSID or JOBNAME) is not currently being used by another active IMS subsystem and that the IRLM has been started and is active for this system. Back out might be needed.

**Problem determination:** 1, 4, 10, 35

**Source:** IMS abends

**Module:** DFSPCCC0, DFSRDSH0, DFSRST00

**Related reference:**

 [IRLM request return and reason codes \(Messages and Codes\)](#)

**Related information:**

Chapter 2, “IMS problem determination,” on page 13

---

## 0040

**Explanation:** The acquisition of the global command lock resulted in an invalid return code. This lock is used and required for communication between all sharing subsystems.

**Analysis:** This is a standard abend issued by modules DFSPCCC0, DFSRDSH0, and DFSRST00. The IRLM reason code is contained in register 15, and the IMS return code can be found at offset +X'347' under PSTLRXRC in the PST DL/I data sharing section. The parameter list used to issue the request is pointed to by the restart PST. The IRLM parameter list address can be obtained from PST field PSTIRLMA. To determine the cause of the failure, see the IRLM request return and reason codes information.

**Possible cause:** A failure in the IRLM is a possible cause of this abend.

**System action:** IMS terminates abnormally.


**Programmer response:** Ensure that the IRLM is still active.

**Problem determination:** 1, 4, 10, 35

**Source:** IMS abends

**Module:** DFSPCCC0, DFSRDSH0, DFSRST00, DBFLHCK0

**Related reference:**

 [IRLM request return and reason codes \(Messages and Codes\)](#)

**Related information:**

Chapter 2, “IMS problem determination,” on page 13

---

## 0041

**Explanation:** A sign-on request to DBRC failed. The DBRC return codes define the error. To determine the cause of the failure, refer to:

- The DBRC request return codes information to determine the cause of the failure.
- The information for message DFS041I

**Analysis:** 0041 is a standard abend issued by DFSRST00 (online), DFSPCCC0, or DFSPCC30 (batch). The return code is set in register 15 when issuing the abend. The subsystem name for the online IMS subsystem is located in the SCD. For the batch subsystem, the subsystem name is located in PXPARMS. The parameter list address can be retrieved from the DFSBRLSC macro (PRMAREA=*parameter*). The parameter list contains the information passed to DBRC on the sign-on request.


**System action:** IMS terminates abnormally.

**Programmer response:** Ensure that the online subsystem name IMSID or batch job name was correctly specified in the start-up procedure. Also, ensure that the proper RECON data set was allocated to the IMS subsystem. If you are restarting the IMS alternate subsystem, issue the DBRC command CHANGE.SUBSYS SSID(name) NOBACKUP. This command resets the flags that indicate the alternate subsystem has signed on.

**Source:** IMS abends

**Module:** DFSPCCC0, DFSPCC30, DFSRST00

**Related reference:**

 [DBRC request return codes \(Messages and Codes\)](#)

**Related information:**

 [DFS041I \(Messages and Codes\)](#)

**0042**

**Explanation:** Backout was required for this startup and the DBRC signon request indicated that an entry did not exist for this subsystem. During an emergency restart, it is necessary to establish the same environment. Message DFS042I precedes this abend.

**Analysis:** 0042 is a standard abend issued by DFSRLP00. The subsystem name and status of the DBRC are located in the SCD in SCDIMSNM and SCDSHFL2. Check the JCL procedure to ensure that the same RECON data set was specified.

**Possible cause:** An invalid subsystem name, DBRC= specification, or RECON data set was specified in the startup procedure.

**System action:** IMS terminates abnormally.

**Programmer response:** Ensure that the subsystem name IMSID or DBRC= has been properly specified in the startup procedure and that the proper RECON data set has been allocated to the IMS subsystem.

**Problem determination:** 1, 4, 10

**Source:** IMS abends

**Module:** DFSRLP00

**Related information:**

 [DFS042I \(Messages and Codes\)](#)

Chapter 2, "IMS problem determination," on page 13

---

**0043**

**Explanation:** A signon request for "recovery end" was issued to DBRC, and the request failed. This request is issued after the initial checkpoint at the end of emergency restart and notifies DBRC of emergency restart completion. This allows DBRC to remove any pending information held from the previous execution of IMS. The DBRC return codes specify the error. Message DFS043I precedes this abend. To determine the cause of the failure, refer to:

- DBRC request return codes information to determine the cause of the failure.
- The information on message DFS043I.

**Analysis:** 0043 is a standard abend issued by DFSRST00 (online) or DFSPCCC0 (batch). The return code is set into register 15 when issuing the abend. The subsystem name for the online IMS subsystem is located in the SCD. For the batch subsystem, the subsystem name is located in PXPparms. The parameter list address can be located in the checkpoint PST listing with all the information required to issue the signon request.

**System action:** IMS terminates abnormally.


**Programmer response:** Determine the cause of the error and correct the problem.

**Problem determination:** 1, 4, 10, 35

**Source:** IMS abends

**Module:** DFSPCCC0, DFSRST00

**Related reference:**

 [DBRC request return codes \(Messages and Codes\)](#)

**Related information:**

 [DFS043I \(Messages and Codes\)](#)

Chapter 2, "IMS problem determination," on page 13

---

**0044**

**Explanation:** A normal or emergency restart was specified. DBRC was active during the previous execution, but is not active now. The DBRC name or subsystem name was not the same as that found on the X'4001' checkpoint record.

DBRC=FORCE was specified in the Installation Defaults module DFSIDEF0, and an execution of IMS was requested with an execution parameter of DBRC=N.



If you are using HALDB partitioning, this abend indicates that DBRC is not active when it is required. Message DFS0415W, with a reason code of X'74', precedes the abend.

**Analysis:** 0044 is a standard abend issued by DFSRLP00 and DFSXBAT0.

**Possible cause:** A possible cause of this abend is that DBRC=N was specified.

**System action:** IMS terminates abnormally.

**Programmer response:** Ensure that DBRC is present for this execution and that DBRC=N was not specified. The same environment used in the previous execution, namely utilization of DBRC, must be used in this environment for emergency restart and database backout. Validate the input tapes for this execution or the correctness of the checkpoint value specified in the restart command.

**Problem determination:** 1, 4, 10

**Source:** IMS abends

**Module:** DFSRLP00, DFSXBAT0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

## 0045

**Explanation:** The previous execution of IMS had utilized IRLM, but the IRLM was not present for this execution. For emergency restart, it is required that the same IRLM be present for this execution if it was present during the previous execution. Message DFS045I precedes this abend.

**Analysis:** 0045 is a standard abend issued out of DFSRLP00. The existence of the IRLM system name in the X'4001' record implies that the IRLM was active in the previous execution. The IRLM system name existing in the SCD indicates whether the IRLM is active or not for this execution.

**Possible cause:** An incorrect specification of IRLM= or IRLMNM= was given.

**System action:** IMS terminates abnormally.

**Programmer response:** Ensure that IRLM=N was not specified or that the same environment used by the previous execution of IMS exists for this execution.

**Problem determination:** 1, 4, 10

**Source:** IMS abends

**Module:** DFSRLP00

**Related information:**

Chapter 2, "IMS problem determination," on page 13

## 0046

**Explanation:** One of the DBPCBs contained in the PSB, which appears in the third positional operand of the PARM field on the EXEC control statement, had an incompatible PROCESSING INTENT with the ACCESS parameter defined in the DATABASE macro statement for IMS system definition. The PROCESSING INTENT is derived from the PROCOPT operand specifications in the PSBGEN.

**Analysis:** This is a pseudoabend set by module DFSSBMP0 and issued from module DFSPCC20. See the information on message DFS046A for the reason for the failure.

Key	Label	Description
	BMPAB046	A schedule failure code was set to X'09' in the PSTSCHDF field by module DFSDBLM0. The code indicates that an incompatible processing intent was detected in the PSB.

**System action:** Error message DFS046A is issued. The batch region terminates abnormally. Dependent regions continue processing with limited access to the database.

**Programmer response:** One of the following responses can be taken:

## 0047

- Contact the master terminal operator to change the database ACCESS parameter to the level compatible with your PROCESSING INTENT
- change the PROCOPT operand specifications to the level compatible with the ACCESS parameter defined in the DATABASE statement for the IMS system definition.

Rerun ACBGEN, and then rerun the job.

**Problem determination:** 6, 10, 19

**Source:** IMS abends

**Module:** DFSPCC20, DFSSBMP0

**Related information:**

 DFS046A

Chapter 2, "IMS problem determination," on page 13

---

## 0047

**Explanation:** One of the DBPCBs in the named PSB, appearing in the third positional operand of the PARM field on the EXEC control statement, had referenced the database and failed to obtain the database authorization through DBRC, or failed to obtain a larger work storage area in which to build the database authorization request list during the PSB scheduling.

**Analysis: For DFSDBAU0:**

0047 is a standard abend issued by module DFSDBAU0 for a DL/I or DBB batch region. The PSB names appear in the message text if the IMS is an online control region. See the information on message DFS047A.

Key	Label	Description
In DFSDBAU0	ABND47	A DL/I or DBB batch region failed to obtain a database authorization.

**For DFSPCC20, DFSSBMP0:**

0047 is a pseudoabend set by module DFSSBMP0 and issued from module DFSPCC20. See message DFS047A for the reason for the failure.

Key	Label	Description
	BMPAB047	A schedule failure code was set to X'0A' in the PSTSCHDF field by module DFSDBLM0. The code indicates that a database authorization request failed for the PSB.

**System action:** Error message DFS047A or DFS049I is issued and the database is stopped. The batch region terminates abnormally. Dependent regions continue processing, but access to the database is prohibited.

**Programmer response:** See error message DFS047A or DFS049I for the reason for the failure and the response to the failure. For message DFS047A, one of the following responses should be taken:

- Contact the master terminal operator to close the database in other IMS subsystems that had held the database authorization by entering a /DBR command.
- Wait until another IMS subsystem that had held the database authorization has terminated, and rerun the job.

**Problem determination:** 11, 19

**Source:** IMS abends

**Module:** DFSDBAU0, DFSPCC20, DFSSBMP0

**Related information:**

 DFS047A

 DFS049I

Chapter 2, "IMS problem determination," on page 13

**0048**

**Explanation:** An error occurred during initialization of Database Recovery Control (DBRC). The return code is in register 15.

**Analysis:** For DFSXDRC0:

0048 is issued for all abnormal conditions. Register 15 contains the following reason codes:

**Codes****Explanation****X'10'**

DBRC initialization (INIT1) issued a nonzero return code. Register 5 = DBRC return code.

**X'14'**

DFSBCB quick save obtain failed. Register 5=DFSBCB return code.

**X'18'**

DFSCIR ITASK create failed. Register 5 = DFSCIR return code.

**For DFSXRIC0:**

0048 is issued for all abnormal conditions. Register 15 contains the following reason codes. An error occurred during initialization in register 15 of the database recovery control for one of the following reasons:

Code	Reason	Action
X'04'	IMODULE GETMAIN for the IMS-DBRC control block (DFSRCWK) failed.	Increase the IMS control region size.
X'08'	IMODULE LOAD of DFSRCQM0, DFSRCQR0, or DSPCRTR0 failed.	Correct the JOB/STEPLIB DD statement, or bind the DBRC modules into the correct library.
X'0C'	First call for initialization (INIT-0) had a nonzero return code.	Determine the reason for the DBRC initialization failure by referring to the appropriate DBRC database recovery control documentation.
X'10'	Second call for initialization (INIT-1) had a nonzero return code.	Determine the reason for the DBRC initialization failure by the second call for DBRC initialization (INIT-1), which had a nonzero return code.

Take appropriate action according to the error code.

**System action:** IMS terminated abnormally.

**Programmer response:** Take appropriate action according to the error code described above.

**Problem determination:** 1, 2, 3, 4, 8, 12

**Source:** IMS abends

**Module:** DFSXDRC0, DFSXRIC0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

**0049**

**Explanation:** This abend is issued when an OS RDJFCB error or an out-of-storage condition occurs. When given for an out-of-storage condition, this abend only occurs while using the IRLM and an IRLM failure or an IRLM communication failure has occurred.

**Analysis:** 0049 is a standard abend issued by module DFSDBAU0. The program status word (PSW) at entry-to-abend points to one of the following locations: after label ST307, at label AB049 in module DFSDBAU0, or at label DBFSTATS in DBFDDBAU0.

## 0050

Key	Label	Description
Reg8=size of storage needed	ST307	This amount of storage is needed to process an IRLM or communication failure.
PSW at label AB049	AB049	Register 15 contains a return code other than X'00' or X'04' from RDJFCB supervisor call (SVC).

**System action:** IMS terminates abnormally.

**Problem determination:** 4, 12, 35

**Source:** IMS abends

**Module:** DFSDBAU0, DBFDDBAU0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

## 0050

**Explanation:** An IMS services request (DFSBCB for QSAV, DFSCDSP or DFSCWU), required for external trace initialization, received an unexpected nonzero return code.

**Analysis:** The external trace TCB was abended, allowing OLDS external tracing only. Register 2 contains the external trace request that failed. Register 14 contains the BAL REG pointing to the routine that had the error. Register 15 contains the reason code to identify the lower level service that failed.

- Register 2 identifies the IMS services request that failed.

**Code (hex)**

**Meaning**

- 01 DFSBCB for QSAV request failed.
- 02 DFSCDSP request failed.
- 03 DFSCWU request failed.
- 04 IMODULE request failed.

- Register 14 is the address where the error is detected.
- Register 15 contains the return code from the failing service.

**System action:** The external trace TCB terminates. Only OLDS external tracing is permitted until IMS is restarted.

**Source:** IMS abends

**Module:** DFSXTRA0

---

## Chapter 6. IMS abend codes 0051 - 0100

This topic identifies and explains the following abend codes issued by IMS. Each code is 4 decimal digits.

---

### 0055

**Explanation:** The application program requested a buffer from the Shared Virtual Storage Option (SVSO) buffer pool, but this request was not satisfied because the SVSO pool had already reached its expansion limit. This abend is independent of the normal buffer allocation (NBA) and overflow buffer allocation (OBA) usage.

**Analysis:** This is a standard abend issued by DBFIRC10, the Fast Path inter-region communication control module, when the SVSO buffer pool limit has been reached.

**System action:** The dependent region ends abnormally with pseudo code 0055.

**Programmer response:** Increase the SVSO buffer pool size to accommodate concurrent dependent region access.

**Source:** IMS abends

**Module:** DBFIRC10, DBFSGAB0

---

### 0056

**Explanation:** An error occurred in a Database Control (DBCTL) subsystem while Fast Path was trying to access buffers for a Coordinator Controller (CCTL) request. This error can occur during a CCTL identify, terminate, thread schedule, or thread sync point request.

**Analysis:** If the failure occurs during a CCTL thread schedule (DBFINTE0) or thread sync point (DBFINTT0) request, the IMS control region terminates abnormally. Register 12 in the abend SVRB indicates which module issued the abend. In both cases, this abend is issued in one location. For both requests, the error occurs because Fast Path was unable to find the CBUF control block for the CCTL making the request. Fast Path needs this block to get buffers (schedule) and release buffers (sync point).

If the failure occurs during a CCTL terminate request, DBFDDBF10 issues an abend for two conditions. To determine which condition caused the abend, examine register 15 in the abend SVRB.

**Reg15=4**

Fast Path failed to find a CBUF for the CCTL name. The IMS control region terminates abnormally.

**Reg15=8**

Fast Path failed to obtain an AWE. This situation can occur in either the IMS control region or the database resource adapter (DRA) TCB. Either the IMS control region or the DRA TCB terminates abnormally.

If the failure occurs during a CCTL identify request, DFS DASIO sets up pseudoabend 0056 to enable the DRA to tell the CCTL that the request failed for reason 0056. The DRA then takes a SNAP dump. The header title indicates that the identification caused the 0056 failure. For this request, the abend is also issued because Fast Path was unable to get buffers for the CCTL. Field SSPICODE in the SSOB contains a character string that identifies the exact reason for the failure. These strings and reasons are:

**SSPICODE**

**Explanation**

**AWE**

Unable to build AWE block because no CSA storage is available.

**CBUF**

Unable to build CBUF block because no ECSA storage is available.

**NBA**

Not enough buffers are available to meet CCTLNBA request.

**System action:** If the failure occurs during a schedule or sync point request, the IMS control region abends with 0056. Sync point errors occur after the phase 2 abort or commit.

## 0069

If the failure occurs during an identify request, the identify request fails with pseudoabend 0056, but IMS and the CCTL are unaffected. A snap dump is created.

If the failure occurs during a terminate request, the terminate request fails with pseudoabend 0056, but IMS and the CCTL are unaffected. A snap dump is created. The CCTL is no longer identified to IMS, but its FP buffers are not released.

**Programmer response:** Determine which type of request failed.

If schedule or sync point failed, an internal error occurred.

If identify failed, IMS probably does not have enough FP buffers to satisfy the number of buffers requested. See CCTLNBA in the DRA startup table to determine how many buffers you requested.

If terminate failed, an internal system error or a storage shortage problem probably occurred.

**Problem determination:** 1, 4, 38

**Source:** IMS abends

**Module:** DBFDBFI0, DFSDASI0, DBFINTE0, DBFINTT0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

## 0069

**Explanation:** A recursive IRC entry was detected by program DBFIRC10 or DFSDCPY0, or Fast Path (FP) was unavailable, and a call to Fast Path was found by DBFIRC10.

**Analysis:** 0069 is a standard abend issued from DBFIRC10, DFSCPY00, and DFSDCPY0. The PSTREP bit in the PST is checked to determine a recursive entry. If the PSTREP is on, the abend is issued. The error subcode is in register 15.

Key	Label	Description
Reg15=X'04'		Recursive entry to inter-region communication (IRC) occurred. Register 8 contains the address of instruction that detected the error. (Module DFSDCPY0 does not set register 8.) Register 15 contains the abend subcode.
Reg15=X'08'		Fast Path was unavailable, and a Fast Path call was detected. Register 8 contains the address of instruction that detected the error. Register 15 contains the abend subcode.

**Possible cause:** Internal program interface error: Application program issued a DL/I call from an (E)STAE routine.

**System action:** The dependent region is terminated abnormally. Other dependent regions and the IMS control region continue running.

**Programmer response:** Reason code 4 is a user error. Ensure that a user STAE is not issuing an IMS call or an MVS STAE retry, and then issuing a call. Other reason codes should be treated as IMS system errors.

Reason code 8 should have been preceded by message DFS2768W, and an error occurred at IMS startup time. Examine the console log for errors during IMS start for the original error, such as message DFS2713I or message DFS2718I. Any application making calls to Fast Path will be terminated with this abend until IMS is restarted.

**For APAR processing:** Provide the dependent region abend dump.

**Problem determination:** 1, 5, 6, 11, 35

**Source:** IMS abends

**Module:** DBFIRC10, DFSCPY00, DFSDCPY0

**Related information:**

 DFS2768W (Messages and Codes)

Chapter 2, "IMS problem determination," on page 13

## 0070

**Explanation:** An IMODULE failure occurred during system initialization or subsequent execution.

**Analysis:** 0070 is a standard abend issued from one of twelve modules—DFSSTKMG, DBFICI10, DBFIFIX0, DFSIINF0, DFSIRST0, DFSTMOD0, DFSXBAT0, DFSXCIC0, DFSXLGI0, or DFSXSTM0. The program status word (PSW) at entry-to-abend identifies which module issued the abend SVC.

In each module, the IMS/VIS SVC (from the SCD) was just issued using an Execute instruction. The IMS/VIS SVC that services GETMAIN is DFSMODU0. The execution library load module name is IGC14xxx, where xxx is the user-specified IMS/VIS SVC number. The IMS/VIS SVC (DFSMODU0) returns an error code in register 15 (bytes 0 and 1).

Register 15 contains a return code from the IMODULE function. For a description of the IMODULE return codes, see the IMS system services return codes information.

Register 15 (bytes 0 and 1) might contain a second return code from the DFSBCB GET function. For an explanation of the DFSBCB return code, see the IMS system services return codes information.

This is a standard abend that is issued by the Fast Path emergency restart module, DBFNRSST0. The PSW (program status word) at entry-to-abend points to the instruction where the abend was issued.

If you are using HALDB partitioning, this abend indicates either:

- Allocation of storage for control block DFSPCA fails.
- Loading of partition support modules fails. If this module is DFSDT150, message DFS0415W with reason code X'45' precedes this abend. If it is a module other than DFSDT150, no message precedes this abend.

#### For DFSSTKMG:

The IMS Stack Manager received a nonzero return code from an IMODULE GETMAIN.

Abend 0070 is a standard abend that may be issued by the Stack Manager, DFSSTKMG. When this abend is issued the program status word (PSW) at entry-point-to-abend points to the instruction following the abend (SVC 13) instruction.

Register 12 in the abend SVRB registers is the base register for the Stack Manager routine.

Register 15 contains the nonzero return code from IMODULE.

Key	Label	Description
Reg15=return code from IMODULE		An IMODULE GETMAIN request failed.

#### For DBFICI10:

This is a standard abend that can be issued by the IMS/VIS Fast Path ITASK initialization module, DBFICI10. The PSW at entry-to-abend points to the instruction label ABEND where the abend was issued. Register 14 in the abend SVRB is used as the KEY. Register 12 is the base register, register 4 contains the size of the GETMAIN request, register 2 contains the AWE address, and register 15 contains an error code.

Key	Label	Description
Reg14=BAL Reg1=completion code Reg2=AWE address Reg4=size of the request Reg15=DFSQCSS FUNC=STORAGE return code (AWSIRCD)	ABEND1	This routine initializes IMS/VIS/Fast Path COMMUNICATION ROUTER ITASK. The work area (DBFWORK2) for this ITASK is required by DFSQCSS macro with FUNC=STORAGE. A nonzero return code in AWSIRCD from the DFSQCSS macro results in an abend.

#### For DBFIFIX0:

## 0070

0070 is a standard abend that can be issued by the IMS/VS Fast Path fix mechanism module, DBFIFIX0. The PSW at entry-to-abend points to the instruction label ABEND1 where the abend was issued.

Register 14 in the abend SVRB is used as the KEY. Register 12 is the base register, register 4 contains the size of the GETMAIN request, register 2 contains the AWE address, and register 15 contains an error code.

Key	Label	Description
Reg14=BAL Reg1=completion code Reg2=AWE address Reg4=size of the request Reg15=DFSQCSS FUNC=STORAGE return code (AWSIRCD)	ABEND1	This routine fixes Fast Path control blocks according to the default or the user supplied page fixing parameter. The storage requested by the DFSQCSS macro with FUNC=STORAGE is used for the page fixing list. A nonzero return code in AWSIRCD from the DFSQCSS macro results in an abend.

### For DFSIINF0:

0070 is a standard abend that can be issued by the message format block pool initialization module, DFSIINF0. If the error is detected in this module, the program status word (PSW) at entry-to-abend points to the instruction within label ABEND9D from which ABEND (SVC 13) is issued.

Register 11 in the abend SVRB registers is the base register for this module.

Key	Label	Description
Reg15=nonzero return code Reg12=subpool number for GETMAIN Reg9=number of bytes for GETMAIN request Reg1=completion code, X'80000046'	IMOD	The IMODULE GETMAIN (IMS/VS SVC) has been issued and register 15 is loaded with the return code. If the return code is not zero, a branch is taken to label PFERR, which loads the GETMAIN error indicator (IMODERR) into register 3 and branches to label ABEND9D to handle the abend.

### For DFSIRST0:

An IMS IMODULE LOAD failed.

0070 is a standard abend that can be issued by the restart service ITASK module DFSIRST0. Register 15 contains one of the error return codes listed in the table. (This code is not from IMODULE LOAD.)

Key	Label	Description
Return code=1	RSTR010	An IMODULE LOAD failed for XRF module, DFSHRCL0.
Return code=2	RSTR010	An XRF MSSF TCB ATTACH failed.
Return code=3	RSTR030	An IMODULE LOAD failed for IMS restart module, DFSRST00.
Return code=4	RSTR020	An IMODULE LOAD failed for RSR system tracker module, DFSST500.
Return code=5	RSTREST	An IMODULE LOAD failed for module DFSRRSI0, which is required for protected conversation processing.
Return code=7	RST0170	A page fixing failure occurred.

### For DFSTMOD0:

0070 is a standard abend that can be issued by the modify/terminate task initialization module, DFSTMOD0. The program status word (PSW) at entry-to-abend points to the instruction within label GMFAIL from which abend (SVC 13) was issued.



Register 12 is the base register, and register 11 is the pointer to the SCD. Register 15 contains the failure subcode.

Key	Label	Description
Subcode 4 Reg1=X'80000046' Reg4=size of the request	ITRFAIL	This routine attempts to obtain subpool 0 storage for the modify/terminate work area.
Subcode 8	ITRFAIL	IMODULE LOAD of DFSFMOD0 failed.
Subcode 12	ITRFAIL	ITASK create for DFSFMOD0 failed.

#### For DFSXBAT0:

A request for storage allocation failed.

0070 is a standard abend that can be issued by the batch TCB ITASK create module, DFSXBAT0. When this abend is issued, the program status word (PSW) points to the instruction from which ABEND (SVC 13) is issued.

Register 12 at the time of abend is the base register, and register 4 points to the system contents directory (SCD).

Key	Label	Description
Reg14=BAL Reg15=return code Reg4=A(SCD)	ABEND	A nonzero return code was passed to DFSXBAT0 after a storage allocation request. Register 14 points to the BAL instruction following detection of the nonzero return code.

#### For DFSXCIC0:

0070 is a standard abend that can be issued by the control task initialization controller, DFSXCIC0. The program status word (PSW) at entry-to-abend points to the instruction within label ABEND1, from which ABEND (SVC 13) was issued.

Register 14 in the abend SVRB is used as the key, register 12 is the base register, and register 2, where applicable, contains the size of the GETMAIN request.

Key	Label	Description
Reg14=BAL	ABEND1	This abend occurs during control task initialization for the following reasons. <ul style="list-style-type: none"> <li>• IMODULE LOAD for DFSSDLO0 failed.</li> <li>• AWE request (DL/I SAS INIT) failed.</li> <li>• QUICKSAVE request failed.</li> <li>• DPST request failed.</li> </ul>

#### For DFSXSTM0:

0070 is a standard abend that can be issued by the storage management controller, DFSXSTM0. The program status word (PSW) at entry-to-abend points to the instruction within label ABEND1, from which abend (SVC 13) was issued.

Register 14 in the abend SVRB should be used to isolate to a specific label. Register 12 is the base register and register 15 contains an error code.

Key	Label	Description
Reg14=BAL	XSTM0030	IMODULE GETMAIN for DFSTSAV failed
Reg14=BAL	XSTM0050	IMODULE GETMAIN for DFSIPB failed
Reg14=BAL	XSTM0210	IMODULE GETMAIN for DFSSFVT failed
Reg14=BAL	XSTM040	IMODULE GETMAIN for DFSSLX failed
Reg14=BAL	XSTM0500	IMODULE GETMAIN for DFSTAB failed

## 0071

Key	Label	Description
Reg14=BAL	XSTM0744	IMODULE GETMAIN for RSWB failed
Reg14=BAL	RESUINI1	IMODULE GETMAIN for XSL0ECBS failed
Reg14=BAL	XSTM0720	DFSBCB FUNC=GET for QS24 failed
Reg14=BAL	XSTM0810	IMODULE GETMAIN or DFSBCB FUNC=GET failure during CREATE_TASKS routine

### For DFSXLG10:

0070 is a standard abend that can be issued by the logical/physical logger initialization module, DFSXLG10. The program status word (PSW) at entry-to-abend points to the instruction within label ABEND070, from which ABEND (SVC 13) is issued.

Message DFS2205I is sent to the IMS/VS master console before the abend.

Register 0 in the abend SVRB should be used to isolate a specific label. Register 12 is the base register, and register 15 contains the error return code.

Key	Label	Description
Reg15=return code from IMODULE. Reg14=address at which IMODULE is issued.	ABEND070	This routine is a GETMAIN by IMODULE.

### For DFSRESP0:

Module DFSRDB30 was not loaded. Abend 0070 is issued if it was found that DFSRDB30 had not been previously loaded by DFSRST00. Contact the IBM support center.

*Possible cause:* DFSRST00 has an error that bypassed loading DFSRDB30.

**System action:** IMS terminates abnormally.


**Programmer response:** Take appropriate action according to the error code indicated above.

**Problem determination:** 1, 4, 11, 35

**Source:** IMS abends

**Module:** DFSSTKMG, DBFICI10, DBFIFIX0, DFSIINF0, DFSIRST0, DFSTMOD0, DFSXBAT0, DFSXCIC0, DFSXLG10, DFSXSTM0, DFSRESP0, DFSFDLC0

### Related reference:

 [IMS system services return codes \(Messages and Codes\)](#)

### Related information:

Chapter 2, "IMS problem determination," on page 13

---

## 0071

**Explanation:** During system initialization or subsequent execution, an error occurred in a lower-level module, which might cause one of several messages to be issued before this abend. Refer to the messages issued before this abend from either the DBRC region or the control region.

If there is no DFSMDA member for the active ACBLIB, IMS control region initialization terminates with this abend and message DFS0887A is issued.

**Analysis:** 0071 is a standard abend.

*Possible cause:* A common cause of this abend is a failure to close the OLDS logs before restarting IMS by issuing an /ERE COLDSYS command. Alternatively, the lower-level module detected an internal IMS error.

### For DBFXFP00:

0071 is a standard abend issued by DBFXFP00. DBFXFP00 invokes lower-level modules to perform specific initialization. The 0071 is the result of a nonzero return code passed back to DBFXFP00 from one of the lower-level modules.

The PSW at entry-to-abend points to the instruction label ABEND where the abend was issued. Register 14 in the abend SVRB is used as the KEY. Register 12 is the base register and register 15 contains the return code.

Key	Label	Description
Reg1=completion code Reg14=BAL Reg15=DFSBCB FUNC=STORAGE return code=DFSCDSP FUNC=CREATE return code=DFSICIR0	ABEND1	DBFXFP00 is the IMS/Fast Path initialization control task. This module issues DFSBCB FUNC=GET to get a quick save area for the lower-level modules. DFSCDSP FUNC=CREATE creates the dispatcher work area for this Fast Path task and calls DFSICIR0 to initialize ITASKs. The nonzero return code passed from these three functions results in an abend.

#### For DBFXFP10:

0071 is a standard abend issued by DBFXFP10. DBFXFP10 invokes lower-level modules to perform specific initialization. The 0071 is the result of a nonzero return code passed back to DBFXFP10 from one of the lower-level modules. The PSW at entry-to-abend points to the instruction label ABEND where the abend was issued. Register 14 in the abend SVRB is used as the KEY. Register 12 is the base register and register 15 contains the return code.

Key	Label	Description
Reg1=completion code Reg14=BAL Reg15=DFSBCB FUNC=GET return code=DFSCBTS FUNC=FINF return code=DFSCBTS FUNC=ALTER return code=DFSCWU FUNC=CWU return code=DBFICI10 return code=DBFINI20 return code=DBFIFIX0	ABEND1	DBFXFP10 is the IMS/Fast Path initialization module. This module issues DFSBCB FUNC=GET to get a quick save area for the lower-level modules. DFSCBTS issues FUNC=FINF to find FSRB entry for setting FSRB length to CBTE entry, DFSCBTS FUNC=ALTER to set FSRB length to CBTE entry, DFSCWU FUNC=CWU to create WORK UNIT for Fast Path, EPST, and DFSCWu=RWU to release WORK UNIT for Fast Path EPST. It also calls DBFICI10 to initialize Fast Path ITASKs, DBFINI20 to initialize Fast Path control blocks and DBFIFIX0 to fix Fast Path control blocks. The nonzero return code passed from the called modules in DBFXFP10 results in an abend.

#### For DBFXFP20:

0071 is a standard abend issued by DBFXFP20. DBFXFP20 invokes lower-level modules to perform specific initialization. The 0071 is the result of a nonzero return code passed back to DBFXFP20 from one of the lower-level modules.

Key	Label	Description
Reg15=X'0001'	LOWRAB01	The LOCESCD macro could not locate the Fast Path ESCD control block.
Reg15=X'0002'	LOWRAB02	A DFSBCB GET for a QSAV block failed. R8 at abend contains the return code from DFSBCB.
Reg15=X'0003'	LOWRAB03	A DFSCDSP call to create the dispatcher blocks for the FP2 TCB failed. R8 at abend contains the return code from DFSCDSP.

#### For DFSCSL10:

0071 is a standard abend that can be issued by the Common Service Layer Processing module, DFSCSL10.

## 0071

Key	Label	Description
Reg15=X'0002'		IMODULE LOAD failed for DFSDFN20. R2 is the return code from the IMODULE macro.
Reg15=X'0003'		Parameter validation failed. The abend is preceded by one or more DFS3305E messages describing the error.

### For DFSDFDB0:

Abend 0071 is a standard abend, issued by module DFSDFDB0.

Key	Label	Description
Reg15=X'1008'		IMODULE failed to load.
Reg15=X'2008'		An error was encountered while parsing the DATABASE section of the DFSDFxxx PROCLIB member.
Reg15=X200C'		The system was unable to get storage for a copy of the BPEPARSE grammar to parse the DFSDFxxx data set.
Reg15=X'2010'		The system was unable to get storage for the BPEPARSE output area.
Reg15=X'2014'		The system was unable to get storage for the UNREGCATLG list or DFSDDWA control block.
Reg15=X'3000'		BPESTART failed.
Reg15=X'3002'		BPERDPDS failed.

### For DFSDFN20:

0071 is a standard abend that can be issued by the DFSDFxxx and Alternate PROCLIB Member Parsing module, DFSDFN20.

Key	Label	Description
Reg15=X'3004'		Unable to get storage for a copy of the BPEPARSE grammar. Register 2 contains the return code from the MVS STORAGE macro.
Reg15=X'3008'		Unable to get storage for BPEPARSE output area. Register 2 contains the return code from the MVS STORAGE macro.
Reg15=X'300C'		Error reading the alternate PROCLIB member. Register 2 is the return code from BPERDPDS. This abend might be preceded by message BPE0002E with additional information about the error.
Reg15=X'3010'		Error parsing the specified section of the DFSDFxxx PROCLIB member. Register 2 is the return code from BPEPARSE. This abend might be preceded by message BPE0003E with additional information about the error.
Reg15=X'3014'		Error parsing the alternate PROCLIB member. Register 2 is the return code from BPEPARSE. This abend might be preceded by message BPE0003E with additional information about the error.

### For DFSFDLB0:

The IMS physical logger received a nonzero return code from the lower-level modules.

0071 is a standard abend that can be issued by the physical logger, DFSFDLB0. When this abend is issued, the program status word (PSW) at entry-to-abend points to the instruction from which the abend (SVC 13) is issued.

Register 12 in the abend SVRB registers is the base register for this module. Register 11 and register 10 contain the addresses of the system contents directory (SCD) and the log control directory (LCD). Register 14 contains the address at which the failure was detected. Register 15 contains the nonzero return code from DFSBCB or DFSBRLSC macro.

Key	Label	Description
Reg14=A(caller) DFSBCB Reg15=return code	GETA0100	The DFSBCB macro failed to obtain an asynchronous work element (AWE) area and returned a nonzero code in register 15.
Reg14=A(caller) DFSBRLSC Reg15=return code	WRTE0540	The DFSBRLSC macro returned a nonzero return code in register 15 after a DBRC failure. Byte 2 in register 15 has an X'04' return code and the DBRC return code is set in byte 3 in register 15.

#### For DFSFDLD0:

The IMS log timer task received a nonzero return code from a DFSBCB macro.

0071 is a standard abend that can be issued by the DFSBCB macro. When this abend is issued, the program status word (PSW) at entry-to-abend points to the instruction under label DLDA0100 from which the abend (SVC 13) is issued.

Register 12 in the abend SVRB registers is the base register for this module. Register 11 and register 10 are the addresses of the system contents directory (SCD) and the log control directory (LCD). Register 15 contains the nonzero return code from the DFSBCB macro.

Key	Label	Description
Reg15=return code	DLDA0100	When the asynchronous work element (AWE) was requested, the DFSBCB macro failed and returned a nonzero return code in register 15.

#### For DFSFDLG0:

The IMS physical logger received a nonzero condition code from either an STCK or LRA instruction, or an IMODULE or DFSBCB request.

0071 is a standard abend that can be issued by the physical logger, DFSFDLG0. When this abend is issued, the program status word (PSW) at entry-to-abend points to the instruction from which the abend (SVC 13) is issued.

Register 12 in the abend SVRB registers is the base register for this module. Register 11 and register 10 contain the addresses of the system contents directory (SCD) and the log control directory (LCD). Register 14 contains the address at which the failure was detected. For a description of the IMODULE return codes, see the IMS system services return codes information..

Key	Label	Description
Reg14=A(caller) IMODULE Reg15=return code from IMODULE	MSTROPEN MSTRREST	The IMODULE load for the restart log read routine failed and returned a nonzero code in register 15.
Reg15=X'80'	CKCP0600	The STCK instruction set a nonzero condition code in the program status word (PSW). The time-of-day value was not set properly, or the time-of-day timer was inoperative.

## 0071

Key	Label	Description
Reg15=X'84'	CKCP0700	The LRA instruction set a nonzero condition code in the program status word (PSW). The z/OS paging mechanism was damaged.
Reg14=A(caller) DFSBCB Reg15=return code	GETA0100	The DFSBCB macro failed to obtain an asynchronous work element (AWE) area, and returned a nonzero return code in register 15.

### For DFSFDLS0:

0071 is a standard abend issued by DFSFDLS0 when an error return code is received from the DBRC exits (for example, OPEN, CLOSE, SWITCH, and STATUS).

Refer to the messages issued to indicate the failing DBRC exit and return code.

### For DFSFLG0:

The IMS logical logger received a nonzero return code from a DFSBCB macro.

0071 is a standard abend that can be issued by the DFSBCB macro. When this abend is issued, the program status word (PSW) at entry-to-abend points to the instruction in label LOGAB071 from which the abend (SVC 13) is issued.

Register 12 in the abend SVRB registers is the base register for this module. Register 11, register 10, and register 9 contain the addresses of the system contents directory (SCD), the log control directory (LCD), and the DECB. Register 15 contains the nonzero return code from the DFSBCB macro.

Key	Label	Description
Reg15=return code	LAW0100	When the asynchronous work element (AWE) was requested, the DFSBCB macro failed and returned a nonzero return code in register 15.
Reg14=A(caller) Reg15=X'80'	LGET0500	The STCK instruction set a nonzero condition code in the program status word (PSW). The time-of-day value was not set properly, or the time-of-day timer was inoperative.

### For DFSHSRV0:

The IMS/VS hot standby dasd/link surveillance module received a nonzero return code from a lower module.

0071 is a standard abend that can be issued by the hot standby surveillance module, DFSHSRV0. DFSHSRV0 invokes lower-level modules to perform a get memory, release memory, and create ITASK. A failure by one of the modules, indicated by a nonzero return code, results in this abend. Register 14 is used to isolate to the specified low-level module that caused the abend.

Register 12 is the base register. Register 11 contains the address of the system contents directory (SCD). Register 14 contains the address at which the failure was detected. Register 15 contains the nonzero return code.

Key	Label	Description
Reg14=A(caller) DFSBCB FUNC=GET or REL Reg15=return code	SRVADSD1 SRVALNK1	The DFSBCB macro failed to obtain or release an AWE area. It returned a nonzero return code.
Reg14=A(caller) DFSCIR FUNC=ITASK Reg15=return code	SRVADSD1 SRVALNK1	The DFSCIR macro failed to create an ITASK successfully and returned a nonzero return code.

**For DFSPCC30:**

0071 is a standard abend issued by DFSPCC30. DFSPCC30 invokes lower-level modules to perform specific initialization. 0071 is the result of a nonzero return code passed back to DFSPCC30 from one of the lower-level modules. These include DFSDLBL0, or DFSIIND0 for CICS®. Check any accompanying messages for the reason of a nonzero return code.

Key	Label	Description
Reg3=A(PXPARMS) Reg1=CCCCUUUU CCCC=completion code UUUU=user abend type Reg15=return code		Failure to initialize PDIRs or DDIRs, failure to build DL/I control blocks, or both.

**For DFSRCFS0:**

The RACF (RCF TCB) AWE processor module received a nonzero return code from a lower module.

0071 is a standard abend that can be issued by DFSRCFS0. When this abend is issued, the program status word (PSW) at entry-to-abend points to the instruction following the location where ABEND (SVC 13) was issued.

Register 12 in the abend SVRB is the base register for this module. Register 11 contains the address of the system contents directory (SCD). Register 15 contains the abend subcode.

Key	Label	Description
IMODULE Reg15=return code		An IMODULE GETMAIN failed and returned a nonzero return code.

**For DFSRDB30:**

0071 is a standard abend that is issued by DFSRDB30 when ACBLIB problems are detected while DMBs and PSBs are being loaded.

Key	Label	Description
Reg15=4		DFSRDB30 is called to initialize the PSB and DMB directories. The system issues message DFS822I, DFS824I, DFS825I, or DFS831I, which describes the cause of the failure.

**For DFSRESP0:**

0071 is a standard abend issued by DFSRESP0. DFSRESP0 invokes lower-level modules to perform specific functions. 0071 is the result of a nonzero return code being passed back to DFSRESP0 from one of the lower-level modules. The PSW at entry-to-abend points to label ABND071, which issues the abend. Register 14 in the abend SVRB is used as the KEY.

12 is the base register and register 15 contains the return code.

Key	Label	Description
Reg14=BAL Reg15=DFSBCB return code	M015I1	The DFSBCB macro failed to obtain the residual recovery element (RRE) area and returned a nonzero code in register 15.

Key	Label	Description
Reg14=BAL Reg15=DFSBCB return code	M10151	The DFSBCB macro failed to obtain RRE area and returned a nonzero code in register 15.
Reg14=BAL Reg15=DFSBCB return code	SIDXBLD	The DFSBCB macro failed to obtain the subsystem index entry (SIDX) area and returned a nonzero code in register 15.

The return codes (decimal) from the DFSBCB FUNC=GET are located in register 15. For a description of the DFSBCB return codes, see the IMS system services return codes information.

Key	Label	Description
Reg14=BAL Reg15=DFSCBTS return code	M015I91	The DFSCBTS macro failed to enqueue RRE for TYPE=SIDX and returned a nonzero return code in register 15.
Reg14=BAL Reg15=DFSCBTS return code	M101515	The DFSCBTS macro failed to enqueue RRE for TYPE=SIDX and returned a nonzero return code in register 15.
Reg14=BAL Reg15=DFSCBTS return code	SIDXBLD0	The DFSCBTS macro failed to enqueue SIDX for TYPE=SIDX and returned a nonzero return code in register 15.

The return codes (decimal) for the DFSCBTS macro are located in register 15 are shown in the following list:

#### Codes

##### Explanation

- 4 End of chain (SCAN)
- 8 The block was not found.
- 12 Invalid function code
- 16 The element type was not set.
- 20 The element address was not set.
- 24 The element to be dequeued was not on chain.
- 28 The control block table (CBT) entry was not found.
- 32 Invalid page number/buffer offset

Key	Label	Description
Reg14=BAL Reg15=IMODULE return code	SCAN5	The IMODULE GETMAIN macro failed to obtain storage and returned a nonzero return code in register 15. For a description of the IMODULE return codes, see the IMS system services return codes information.

For DFSRST00:



0071 is a standard abend that can be issued by DFSRST00. DFSRST00 attempted to register and connect with the shared queues subsystem in an Extended Recovery Facility (XRF) active system and received a nonzero return code. The PSW at entry-to-abend points to the abend SVC.

Key	Label	Description
R3=return code R12=DFSRST00 base Reg14=call return address	RESTA019	A call to register and connect to the shared queues system failed.

#### For DFSRLP00:

0071 is a standard abend that can be issued by DFSRLP00. DFSRLP00 attempted to register and connect with the shared queues subsystem in an XRF active system and received a nonzero return code. The PSW at entry-to-abend points to the abend SVC.

Key	Label	Description
R3=return code R12=DFSRLP00 base Reg14=call return address	RESTA019	A call to register and connect to the shared queues system failed.

#### For DFSSQ012:

0071 is a standard abend that can be issued by the Shared Queues Parameter Processing module, DFSSQ012.

Key	Label	Description
Reg15=X'0001'		IMODULE GETSTOR failed for the DFSSQM block. Register 2 is the return code from the IMODULE macro.
Reg15=X'0002'		IMODULE LOAD failed for module DFSDFN20. Register 2 is the return code from the IMODULE macro.
Reg15=X'0003'		Parameter validation failed. This abend is preceded by one or more DFS0779E messages describing the error.
Reg15=X'0004'		IMODULE GETSTOR failed for an IB block. Register 2 is the return code from the IMODULE macro.

#### For DFSXBAT0:

A lower-level initialization request failed.

0071 is a standard abend that can be issued by the batch TCB ITASK create module, DFSXBAT0. When this abend is issued, the program status word (PSW) points to the instruction from which ABEND (SVC 13) is issued.

Register 12 at the time of abend is the base register, and register four points to the system contents directory (SCD).

Key	Label	Description
Reg4=A(SCD) Reg14=BAL Reg15=return code	ABEND	A nonzero return code was passed to DFSXBAT0 after an initialization request. Register 14 points to the BAL instruction following detection of the nonzero return code.

Key	Label	Description						
Reg14=BAL Reg15=000000C	ABEND	R14 points to the BAL after label USESSM. R15 indicates that the subsystem member (SSM) in the PROCLIB specifies two or more external subsystems. Only one external subsystem is allowed for DL/I or DBB batch jobs.						
Reg14=BAL Reg15=xxxxyyyy	XLIC0135	DFSFTIN0 encountered an error during initialization. <i>xxxx</i> indicates a reason code shown in the following list, followed by the return code from the lower-level service: <table border="0" style="margin-left: 20px;"> <tr> <td><b>Code</b></td> <td><b>Meaning</b></td> </tr> <tr> <td>0004</td> <td>IMODULE GETMAIN failed. <i>yyyy</i> is the IMODULE return code.</td> </tr> <tr> <td>0008</td> <td>DFSLOADL failed. <i>yyyy</i> is the DFSLOADL return code.</td> </tr> </table>	<b>Code</b>	<b>Meaning</b>	0004	IMODULE GETMAIN failed. <i>yyyy</i> is the IMODULE return code.	0008	DFSLOADL failed. <i>yyyy</i> is the DFSLOADL return code.
<b>Code</b>	<b>Meaning</b>							
0004	IMODULE GETMAIN failed. <i>yyyy</i> is the IMODULE return code.							
0008	DFSLOADL failed. <i>yyyy</i> is the DFSLOADL return code.							

#### For DFSXBAT0 on behalf of DFS3DF10:

0071 is a standard abend that can be issued by module DFSXBAT0 on behalf of module DFS3DF10. DFS3DF10 abends are related to IMS catalog initialization.

Key	Label	Description
Reg15=X'1003'		Parameter validation failed. An abend with this reason code is typically caused by a missing ALIAS parameter in the DFSDFxxx member of the IMS.PROCLIB data set.
Reg15=X'1004'		GETMAIN failed.
Reg15=X'1008'		IMODULE failed to load.
Reg15=X'3000'		BPESTART failed.
Reg15=X'3002'		BPERDPDS failed.
Reg15=X'3004'		Unable to get storage to store a copy of the BPEPARSE grammar to parse the DFSDFxxx member.
Reg15=X'3008'		Unable to get storage for the BPEPARSE output area.
Reg15=X'3010'		Error parsing the specified section of the DFSDFxxx PROCLIB member.

#### For DFSXCIC0:

0071 is a standard abend issued by the control task initialization controller, DFSXCIC0. DFSXCIC0 invokes lower-level modules to perform control task initialization functions. A failure by one of these modules, indicated by a nonzero return code in register 15, results in the abend 0071. Use register 14 in the abend SVRB to isolate to the specific label shown in the following table. Register 12 is the base register, and register 11 is the pointer to the SCD.

Key	Label	Description
Reg14=BAL Reg15=X'04' or X'0C'	CALLIIND	DFSIIND0 is called to initialize the PSB and DMB directories. If the return code in register 15 is X'04', the system issues one of the following messages that provide the cause of the failure: DFS822I, DFS823I, DFS824I, DFS825I, or DFS831I. If the return code is X'0C', DFSIIND0 detected an internal logic error when preparing to load the resident DMB.
Reg15=return code	NODLIS1	DFSXIIOB0 is called to initialize the VSAM and OSAM buffer pools and to page fix the message queue buffer pool.
Reg14=BAL	NODLIS1	DFSIINB0 is called to initialize the communications facilities.

Key	Label	Description
Reg14=BAL	NOCANCVT	DFSIFIX0 is called to fix storage as directed by the user.
Reg15=return code	XCIC5100	DFSPCR00 was called to connect the API component to the PC Router Service.
Reg14=BAL Reg15=Return code from IMODULE	NODLIS4	Unsuccessful IMODULE LOAD of module DFSCST00.
Reg14=BAL Reg15=reason code	SKCSL340	DFSSQ020 was called to perform Shared Queues registration and connection. Message DFS1936E, or message DFS4455E and DFS3308E, are issued with this abend code to indicate the cause of the failure. Reason codes are: <b>X'0101'</b> CQSREG request failed. <b>X'0102'</b> CQSCONN request failed. <b>X'0103'</b> CQSCONN request failed with a message queue (MSGQ) structure error. <b>X'0104'</b> CQSCONN request failed with an expedited message handler queues (EMHQ) structure error. <b>X'0105'</b> CQSCONN request failed with MSGQ and EMHQ structure errors. <b>X'0106'</b> MSGQ structure attribute WAITRBLD is incorrect. <b>X'0107'</b> EMHQ structure attribute WAITRBLD is incorrect. <b>X'0108'</b> IMODULE GETSTOR request failed. <b>X'0109'</b> CQSUNLCK FORCE failed for MSGQ or EMHQ structure.

**For DFSXCTL0:**

0071 is a standard abend that can be issued by DFSXCTL0.

Key	Label	Description
Reg15=return code	ABEND2	A lower-level service returned a nonzero return code. Register 14 points to the location in module DFSXCTL0 that detected the error.

**For DFSXCTX0:**

0071 is a standard abend, issued by module DFSXCTX0.

DFSXCTX0 invokes lower-level modules to perform specific initialization. Abend 0071 results from a nonzero return code passed back to DFSXCTX0 from one of the lower-level modules.

The PSW at entry to abend points to the instruction where the abend was issued. Register 15, in the SVRB, contains the abend subcode.

Key	Label	Description
Reg15=X'01'		DFSBCB GET for a QSAV failed. Register 8 contains the return code from DFSBCB.
Reg15=X'02'		DFSCDSP failed to create an IMS dispatcher work area for the control auxiliary TCB. Register 8 contains the DFSCDSP return code.

**For DFSXDCC0:**

## 0071

0071 is a standard abend issued by DFSXDCC0. DFSXDCC0 invokes lower-level modules to perform specific initialization tasks. The 0071 is the result of a nonzero return code passed back to DFSXDCC0 from one of the lower-level modules. Use register 14 in the abend SVRB to isolate to a specific label. Register 12 is the base register; register 15 contains the return code.

Key	Label	Description
Reg14=BAL Reg15=xxxxyyyy	XDC8725	DFSFTIN0 encountered an error during initialization. <i>xxxx</i> indicates a reason code shown in the following list, followed by the return code from the lower-level service.  <b>Code Meaning</b> <b>0004</b> IMODULE GETMAIN failed. <i>yyyy</i> is the IMODULE return code. <b>0008</b> DFSLOADL failed. <i>yyyy</i> is the DFSLOADL return code.

### | For DFSXDGS0:

| The /DIAGNOSE command DGS TCB initialization routine, DFSXDGS0, received a nonzero return code from one of the lower-level modules called during initialization.

| 0071 is a standard abend that can be issued by DFSXDGS0. When an abend is issued, the program status word (PSW) at entry-to-abend points to the abend instruction (SVC 13). In the abend registers, register 12 points to the data area for the module, register 13 points to the work area, register 15 contains the abend reason code, and register 8 contains the return code from the lower-level module.

Key	Label	Description
Reg15=X'0001'		ABENDU0071: Subcode 01 DFSBCB GET for QSAV failed Register 8 contains the return code from DFSBCB
Reg15=X'0002'		ABENDU0071: Subcode 02 DFSCDSP failed for DGS DWA Register 8 contains the return code from DFSCDSP
Reg15=X'0003'		ABENDU0071: Subcode 03 IMODULE DGS80 load failed Register 8 contains the return code from IMODULE
Reg15=X'0004'		ABENDU0071: Subcode 04 DFSCWU ITASK create failed Register 8 contains the return code from DFSCWU
Reg15=X'0005'		ABENDU0071: Subcode 05 IPOST POST for DGS failed Register 8 contains the return code from IPOST
Reg15=X'0006'		ABENDU0071: Subcode 06 DFSBCB GET for AWE failed Register 8 contains the return code from DFSBCB
Reg15=X'0007'		ABENDU0071: Subcode 07 DGSAQ INIT enqueue failed Register 8 contains the return code from DFSBCB

### For DFSXDLG0:

The IMS system log initialization module received a nonzero return code from a lower module.

0071 is a standard abend that can be issued by the log initialization module, DFSXDLG0. When this abend is issued, the program status word (PSW) at entry-to-abend points to the instruction following the location where ABEND (SVC 13) was issued.

Register 12 in the abend SVRB is the base register for this module. Register 11 contains the address of the system contents directory (SCD). Register 14 contains the address at which the failure was detected. Register 15 contains the nonzero return code.

Key	Label	Description
Reg14=A(caller) DFSBCB FUNC=GET Reg15=return code	DFSXDLG0	The DFSBCB macro failed to obtain a QSAV area and returned a nonzero code in register 15.
Reg14=A(caller) IMODULE Reg15=return code	DFSXDLG0	An IMODULE LOAD failed for module DFSXDLG0 and returned a nonzero code in register 15.
Reg14=A(caller) DFSCDSP FUNC=CREATE Reg15=return code	DISP0100	DFSCDSP failed to create a dispatcher work area and returned a nonzero code in register 15.
Reg14=A(caller) DFSCIR FUNC=ITASK Reg15=return code	ENQ90236 DLOG0300 DLOG0400 MLOG0200	DFSCIR failed to create an ITASK and returned a nonzero code in register 15.

#### For DFSXDYA0:

The IMS Dynamic Allocation initialization routine received a nonzero return code from a lower-level module.

0071 is a standard abend that can be issued by DFSXDYA0. The program status word (PSW) at-entry-to-abend points to the instruction from which the abend is issued.

Register 12 in the abend SVRB registers is the base register for this module. Register 14 contains the address at which the failure was detected. Register 15 contains the nonzero return code from the service that failed.

Key	Label	Description
Reg14=A(caller) Reg15=return code	XDYA010	DFSBCB GET function has failed. R15 contains the DFSBCB return code.
Reg14=A(caller) Reg15=return code	XDYA030 XDYA040	IMODULE GETMAIN function has failed. R15 contains the IMODULE return code.
Reg14=A(caller) Reg15=return code	XDYA120 XDYA125  XDYA140 XDYA145	DFSCDSP CREATE function has failed. R15 contains the DFSCDSP return code.

## 0071

Key	Label	Description
Reg14=A(caller) Reg15=return code	XDYA130 XDYA135 XDYA150	DFSCWU CREATE Work Unit has failed. R15 contains the DFSCWU return code.
Reg14=A(caller) Reg15=return code	NTGETMN	A NAME TOKEN SERVICES CREATE request (IEANTCR) has failed. R15 contains the IEANTCR return code.

### For DFSXLGI0:

The IMS system log initialization module received a nonzero return code from a lower module, or the LRA instruction failed.

0071 is a standard abend that can be issued by the log initialization module, DFSXLGI0. When this abend is issued, the program status word (PSW) at entry-to-abend points to the instruction within label ABEND from which the abend (SVC 13) is issued.

Register 12 in the abend SVRB is the base register for this module. Register 11 contains the address of the system contents directory (SCD). Register 14 contains the address at which the failure was detected. Register 15 contains the nonzero return code from IMSAUTH or DFSQCSS.

Key	Label	Description
Reg14=A(caller) IMSAUTH Reg15=return code Reg10=A(LCD)	ILCD1300	The IMSAUTH FUNC=PGFIX for the LCD failed and returned a nonzero code in register 15.
Reg14=A(caller) DFSQCSS Reg15=return code	LOGB0320 XLG0260	The IMODULE LOAD for the batch logger failed and returned a nonzero code in register 15.
Reg14=A(caller)	XLG0080	The OLDSDEF or WADSDEF control statement is missing in the DFSVSMxx member of IMS PROCLIB. Message DFS2205I was issued before the abend.

### For DFSXLIC0:

0071 is a standard abend issued by DFSXLIC0. DFSXLIC0 invokes lower-level modules to perform specific initialization tasks. The abend 0071 is the result of a nonzero return code passed back to DFSXLIC0 from one of the lower-level modules. Use register 14 in the abend SVRB to isolate to a specific label. Register 12 is the base register; register 15 contains the return code. Where the specify task abnormal exit (STAE) work area cannot be established, the possible STAE return codes are:

#### Codes

##### Explanation

X'00'

Work area created

X'04'

Request canceled

X'08'

Work area overlaid

Key	Label	Description						
Reg14=BAL		Invokes DFSIINS0 to build and format permanent pools.						
Reg14=BAL Reg15=xxxxyyyy	XLIC0135	DFSFTIN0 encountered an error during initialization. xxxx indicates a reason code shown in the following list, followed by the return code from the lower-level service.  <table border="0"> <tr> <td><b>Code</b></td> <td><b>Meaning</b></td> </tr> <tr> <td>0004</td> <td>IMODULE GETMAIN failed. yyyy is the IMODULE return code.</td> </tr> <tr> <td>0008</td> <td>DFSLOADL failed. yyyy is the DFSLOADL return code.</td> </tr> </table>	<b>Code</b>	<b>Meaning</b>	0004	IMODULE GETMAIN failed. yyyy is the IMODULE return code.	0008	DFSLOADL failed. yyyy is the DFSLOADL return code.
<b>Code</b>	<b>Meaning</b>							
0004	IMODULE GETMAIN failed. yyyy is the IMODULE return code.							
0008	DFSLOADL failed. yyyy is the DFSLOADL return code.							
Reg14=BAL	LIC3	DFSTMOD0 is called to perform terminate/modify ITASK initialization.						
Reg15=zero		For Fast Path, DBFIFLI0 was not included in the binding of DFSXLIC0.						
Reg15=IDM1 Reg15=IDM2 Reg15=IDM3 Reg15=IDM4 Reg15=IDM5  Reg15=IDM6		DFSIIDM0 returns one of the following causes: <ul style="list-style-type: none"> <li>• Allocation of DFSPPOOL AOIP failed.</li> <li>• AOIP storage requests for control blocks failed.</li> <li>• Standard User Exit definition failed for DFSAOE00.</li> <li>• AOIP storage request for DFSAOE00 work area failed.</li> <li>• AOIP storage request for DFSAOE00 interface block failed.</li> <li>• Load of DFSAOSC0, ICMD security module, failed.</li> </ul>						
Reg14=BAL		DFSIIOC0 is called to process online change blocks.						

#### For DFSXRCF0:

The IMS system initialization module received a nonzero return code from a lower module.

0071 is a standard abend that can be issued by the system initialization module, DFSXRCF0. When this abend is issued, the program status word (PSW) at entry-to-abend points to the instruction following the location where ABEND (SVC13) was issued.

Register 12 in the abend SVRB is the base register for this module. Register 11 contains the address of the system contents directory (SCD). Register 15 contains the abend subcode. For more information about these subcodes, see message DFS2930I.

Key	Label	Description
Reg15=X'01'		DFSBCB GET failed for a QSAV. Register 8 contains the nonzero return code from DFSBCB.
Reg15=X'02'		DFSCDSP failed to create an IMS dispatcher work area for the RACF (RCF) TCB. Register 8 contains the nonzero return code from DFSCDSP.
Reg15=X'06'		An IMODULE GETMAIN failed for the DFSRCFS0 work block. Register 8 contains the nonzero IMODULE return code.
Reg15=X'07'		DFSCIR failed to create an ITASK. Register 8 contains the nonzero return code from DFSCIR.

#### For DFSXSL10:

DFSXSL10 received a nonzero return code from a lower-level module.

0071 is a standard abend issued by the DFSXSL10. DFSXSL10 invokes lower-level modules to perform task initialization functions. A failure by one of these modules, indicated by a nonzero return code in register 15, results in the 0071. Use register 14 in the abend SVRB to isolate to the specific label. Register 12 is the base register and register 11 is the pointer to the SCD.

The PSW at entry to abend points to the instruction where the abend was issued. The address where the failure was detected is contained in Register 14. Register 15, in the SVRB, contains the abend subcode.

## 0071

Key	Label	Description
Reg14=A(caller) DFSBCB Reg15=1		The DFSBCB macro failed to obtain a QSAV area and returned a nonzero code in register 15.
Reg14=A(caller) DFSCDSP Reg15=2		DFSCDSP failed to create a dispatcher work area and returned a nonzero code in register 15.
Reg14=A(caller) DFSCIR Reg15=3		DFSCIR failed to create an ITASK for DFSOCM0 and returned a nonzero code in register 15.
Reg14=A(caller) DFSCIR Reg15=5		DFSCIR failed to create an ITASK for DFSSINP0 and returned a nonzero code in register 15.

### For DFSXSTM0:

0071 is a standard abend that can be issued by DFSXSTM0, which invokes lower-level modules to perform specific initialization functions.

If DFSXSTM0 detects that shared queues initialization failed, messages DFS2930I and DFS1936E are issued to the system console before this abend. The messages indicate the reason for the failure. The reason and error codes in message DFS2930I are also provided in the dump. Register 15 contains reason code X'16' that identifies a shared queues initialization failure; register 3 contains the error code that identifies the failing function. For more information about these codes, see message DFS2930I.

If DFSCSL00 detects an error during the common service layer (CSL) initialization, module DFSXSTM0 issues message DFS2930I with reason code X'18' and 0071 occurs.

The following reason codes result from errors detected by DFSCSL00:

#### Codes

##### Explanation

#### 1004

RC from GETMAIN

#### 1008

RC from IMODULE Load

#### 100C

RC from DFSLOADL

#### 1010

RC from DFSBCB

#### 1014

RC from ATTACH

#### 1018

RC from DFSBCB

#### 101C

RC from DFSSQPP

The following reason codes result from errors detected by DFSCSL10:



**Codes****Explanation****2004**

A problem with DFSCGxxx PROCLIB member was encountered. For more information, see message DFS3305E.

**200C**

OLC=GLOBAL was specified but OLCST was not specified.

**For DFSXWTR0:**

Abend 0071 is a standard abend, issued by module DFSXWTR0.

DFSXWTR0 invokes lower-level modules to perform specific initialization. Abend 0071 results from a nonzero return code passed back to DFSXWTR0 from one of the lower-level modules. The PSW at entry to abend points to the instruction where the abend was issued. Register 15 at abend contains the abend subcode. Register 8 at abend contains the return code from the failing service.

Key	Label	Description
R15=X'01'	LOWRAB01	DFSBCB GET for a QSAV failed. Register 8 contains the return code from DFSBCB.
R15=X'02'	LOWRAB02	DFSCDSP failed to create an IMS dispatcher work area for the WTR TCB. Register 8 contains the DFSCDSP return code.

**System action:** IMS terminates abnormally.

**Operator response:** Emergency restart IMS to resume processing.

**Problem determination:** 1, 4, 15, 35

**Source:** IMS abends

**Module:** DBFXFP00, DBFXFP10, DFSCSL10, DFSDFN20, DFSFDLB0, DFSFDLD0, DFSFDLG0, DFSFDLS0, DFSFLLG0, DFSHSRV0, DFSPCC30, DFSRCFS0, DFSRDB30, DFSRESP0, DFSSQ012, DFSXBAT0, DFSXCIC0, DFSXCTL0, DFSXCTX0, DFSXDLG0, DFSXDLGI0, DFSXLIC0, DFSXRCF0, DFSXSL10, DFSXSTM0, DFSXWTR0, DFSDFDB0, DFSXDGS0, DBFXFP20

**Related reference:**

[IMS system services return codes \(Messages and Codes\)](#)

**Related information:**

[DFS0887A \(Messages and Codes\)](#)

[DFS2930I](#)

Chapter 2, "IMS problem determination," on page 13

**0072**

**Explanation:** A nonzero return code was issued by DEVTYPE or TRKCALC macro issued for a log data set. The data set (ddname) might have been IEFORDER or IEFORDER2 for IMS batch, or DFSOLP*nn*, DFSOLS*nn*, or DFSWADS*n* for IMS online.

**Analysis:** 0072 is a standard abend that can be issued by DFSXDLGI0. When this abend is issued, the program status word (PSW) at entry-to-abend points to the instruction within label ABEND from which the abend (SVC 13) is issued.

Register 12 in the abend SVRB registers is the base register for this module. Register 11 contains the address of the system contents directory (SCD). Register 14 contains the address at which the failure was detected. Register 15 contains the nonzero return code from DEVTYPE or TRKCALC macro.

Key	Label	Description
Reg14=A(caller) DEVTYPE Reg15=return code	ABEND072	The DEVTYPE macro returned a nonzero return code in register 15.
Reg14=A(caller) TRKCALC Reg15=return code	WADS0200	The TRKCALC macro returned a nonzero return code in register 15.

**System action:** IMS terminates abnormally.

**Programmer response:** Make sure there is a valid DD statement defining the log data sets. Make sure DASD devices are specified for the online log data sets, OLDS, and for the write-ahead data sets, WADS. Make sure the same DASD device type is specified for all WADS.

**Problem determination:** No action is required.

**Source:** IMS abends

**Module:** DFSXLGI0

### 0073

**Explanation:** An insufficient number of valid log data sets existed at online initialization or restart time. IMS online requires at least 3 online log data sets (OLDS) and 1 write-ahead log data set (WADS). OLDS block size is not large enough to handle the log record X'3A'.

If DBRC is not active, the required IEFORDER DD statement was missing.

If DBRC is active, either the IEFORDER DD statement was missing or was specified as DD DUMMY OR DSN=NULLFILE (null data set). Register 15 contains a reason code.

Alternatively, the OLDS block size that is specified on the BLKSZ= parameter of the OLDSDEF statement in the DFSVSMxx PROCLIB member was invalid.

**Analysis: For DFSFDLS0:**

IMS was given an insufficient number of log data sets.

0073 is a standard abend that can be issued by the physical logger, DFSFDLS0, during a warm or emergency restart. To protect IMS recovery resources, this abend does not generate a memory dump.

| The OLDS block size is not large enough. The minimum block size is the sum of the sizes of the following items:

| *Table 5. Calculating the minimum block size of OLDS*

Item	Where to get the value of the item	Current value
Block descriptor word	A constant	4 bytes
X'4800' record	PARECLEN in part DFSPALOG	58 bytes
X'3A' record	L3ABASE in part QLFXFREE, the record suffix length in part DFSLOGRC, plus the large message queue record size	32 bytes plus the large message queue record size

| Before this restart, some I/O errors might have occurred in the online data sets (OLDS) and the number of required new or error-free OLDS was not enough (fewer than 2).

**For DFSXLGI0:**

There was an insufficient number of log data set(s) given to IMS.

0073 is a standard abend that can be issued by the log initialization module, DFSXLGI0. When this abend is issued, the program status word (PSW) at entry-to-abend points to the instruction within label ABEND from which the abend (SVC 13) is issued. Before this abend, message DFS2205I is sent to the system console.

Register 12 in the abend SVRB registers is the base register for this module. Register 11 contains the address of the system contents directory (SCD). Register 14 contains the address at which the failure was detected. Register 15 contains the reason code.

Check the DD statements and VTOC of the preallocated online log data set (OLDS).

Key	Label	Description
Reg15=X'04'	ABEND073	There was no WADS DD statement with the name DFSWADSx.
Reg15=X'08'	ABEND073	For an online system, fewer than 3 OLDS DD statements were specified. For a batch system, the required IEFORDER DD statement was missing.
Reg15=X'0C'	ABEND073	The OLDS block size was either too small or not a multiple of 2048.
Reg15=X'10'	ABEND073	The BLKSZ= parameter of the OLDSDEF statement in the DFSVSMxx PROCLIB member was invalid. You must specify a value that is a multiple of 2048, and that is in the range 6144 - 30,720.

#### For DFSXLIC0:

The DD DUMMY parameter or DSN=NULLFILE was specified and is not supported for the system log in a DB/DC environment, in a batch environment with IRLM, nor with DBRC when processing intent is updated.

0073 is a standard abend issued by module DFSXLG10 at label ABEND4. The program status word (PSW) at entry-to-abend and the register 10 BAL in the abend SVRB should be used to determine the applicable label. Register 12 is the base register.

Key	Label	Description
Reg10=BAL	GETJFCB	This routine ensures that IEFORDER is not specified as DD DUMMY. The DEVTYPE macro supplies as its parameters a DDNAME and an output area (DEVAREA). This area is 5 words in length, and is used by the operating system to supply device characteristics to IMS. A check is made to determine whether this output area has had any data moved into it after completion of the SVC 24. The check is done by the "And Characters" instruction (NC): The instruction adds zeros into itself. A zero resultant, indicating that no data was supplied, results in abend.
Reg6=JFCB Reg10=BAL	(Just before label) GETJF1	This routine ensures that the DSNAME is not NULLFILE. A compare is made between constant NULLFILE and the JFCBDSNM field. An equal compare results in abend.

**System action:** IMS terminates abnormally.

**Programmer response:** Rerun the job with sufficient valid DD statements for the required log data sets.

**Source:** IMS abends

**Module:** DFSFDLS0, DFSXLG10, DFSXLIC0

#### 0074

**Explanation:** A nonzero return code for the RDJFCB macro was issued for the system log or the monitor log.

**Analysis:** 0074 is a standard abend that can be issued by DFSXLG10. When this abend is issued, the program status word (PSW) at entry-to-abend points to the instruction from which abend (SVC 13) is issued.

Register 12 in the abend SVRB registers is the base register for this module. Register 11 contains the address of the system contents directory (SCD). Register 14 contains the address at which the failure was detected. Register 15 contains a nonzero return code from the RDJFCB macro.

Key	Label	Description
Reg5=A(DCB) Reg15=return code	ABEND074	The RDJFCB macro failed for this DCB. The DDNAME is found in DCBDDNAM field.

**System action:** IMS terminates abnormally.

## 0075 • 0076

**Operator response:** Contact your IMS system programmer.

**Problem determination:** 4, 35

**Source:** IMS abends

**Module:** DFSXLGI0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

### 0075

**Explanation:** Initialization of the IMS log detected an invalid log device characteristic. On a tape device, the IMS batch system log must have a standard label or a standard user label. The IMS OLDS must be on direct access devices. Both the primary and secondary OLDS in a dual pair must be on the same device type. All the Write-Ahead Log Data sets must be on the same direct access device type.

**Analysis:** 0075 is a standard abend that can be issued by the system log initialization module, DFSXLGI0. When this abend is issued, the program status word (PSW) at entry-to-abend points to the instruction from which the abend (SVC 13) is issued.

Register 12 in the abend SVRB registers is the base register for this module. Register 11 contains the address of the system contents directory (SCD). Register 14 contains the address at which the failure was detected.

Key	Label	Description
Reg3=A(DDNAME)	TIOT0200	The WADS data set with ddname DFSWADSx is not on a DASD device or the device types are different when multiple WADSs are present.
Reg3=A(DDNAME)	TIOT0300 TIOT1100	The OLDS data set with the ddname DFSOLPnn/DFSOLSnn is not on the DASD device type.
Reg4=A(JFCB) Reg5=A(DCB)	MAKELOGB	The batch primary log (IEFRDER) is on tape, but it does not have a standard label or a standard user label.
Reg4=A(JFCB) Reg5=A(DCB)	LOGB0340	The batch secondary log (IEFRDER2) is on tape, but it does not have a standard label or a standard user label.

**System action:** IMS terminates abnormally.

**Programmer response:** Correct the DD statements for the log data sets, and rerun the job.

**Source:** IMS abends

**Module:** DFSXLGI0

---

### 0076

**Explanation:** This abend is issued for coupling facility services by one of the following initialization processes:

- Control address space initialization
- DL/I subordinate address space initialization

Register 3 contains the abend reason code. Register 15 contains the return code from the system service that found the error (for example, IMODULE).

**Analysis:** For DFSTRA00:

Key	Label	Description
Reg3=X'01'	ABEND76	IMODULE GETSTOR failed for key 7 CFB.

For DFSXCFB0:

Key	Label	Description
Reg3=X'02' Reg3=X'03'	ABEND76	IMODULE LOAD failed for DFSDCFR0. IMODULE LOAD failed for DFSDXES0. IMODULE LOAD failed for DFSDENF0. IMODULE LOAD failed for DFSDMAW0. IMODULE GETMAIN failed for CFR AWE. ENFREQ LISTEN request failed.
Reg3=X'04' Reg3=X'05'	ABEND76	
Reg3=X'06' Reg3=X'07'	ABEND76	
	ABEND76	
	ABEND76	

**System action:** IMS terminates abnormally.

**Programmer response:** Determine the cause of the failure.

**Problem determination:** 1, 4, 35

**Source:** IMS abends

**Module:** DFSTRA00, DFSXCFB0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

## 0077

**Explanation:** A DELETE failure occurred during initialization.

**Analysis:** 0077 is a standard abend that can be issued by the storage management task initialization controller (DFSXSTM0) and the batch TCB ITASK create module (DFSXBAT0). A call to the DFSBCB macro resulted in a nonzero return code.

**For DFSXBAT0:**

This is a standard abend issued by the batch TCB ITASK create module, DFSXBAT0. When this abend is issued the program status word (PSW) points to the instruction from which ABEND (SVC 13) is issued.

Register 12 at the time of abend is the base register, and register 4 points to the system contents directory (SCD).

Key	Label	Description
Reg4=A(SCD) Reg14=BAL Reg15=return code	ABEND	A nonzero return code was passed to DFSXBAT0 after an IMODULE DELETE request. Register 14 points to the BAL instruction following detection of the nonzero return code.

**For DFSXSTM0:**

0077 is a standard abend that can be issued by the storage management task initialization controller, DFSXSTM0. The program status word (PSW) at entry-to-abend isolates the failure to a particular module, and points to the instruction within the routine at label ABEND from which the abend (SVC 13) is issued. This routine is unconditionally branched to by the routine at label ABEND8, which is conditionally branched to by the routine that detected the error.

Register 12 in the abend SVRB registers is the base register for this module. Register 1 contains the abend completion code, X'8000004D'. Register 15 contains the nonzero return code from the DPSBCB release macro.

Key	Label	Description
Reg15=return code from DPSBCB	ABEND	In the routine to release an asynchronous work element (AWE), register 15 is tested for the return code; if nonzero, a branch is taken to label ABEND8 to handle the abend.

**System action:** IMS terminates abnormally.

**Problem determination:** 4, 35

**Source:** IMS abends

**Module:** DFSXBAT0, DFSXSTM0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

**0078****Explanation:**

This abend results if the control block modules (load module DFSBLK00, PI control block module DFSFXC00, or OTMA control block modules DFSYINI0 and DFSYIMI0) could not be loaded during control region initialization. The last character of the module name is the suffix of the IMS system, determined by the SUF parameter of the startup procedure.

If this abend is preceded by one or more occurrences of message DFS1921I, check the syntax of the DFSPBxxx member.

**Analysis:** 0078 is a standard abend issued by the control block module loader, DFSXDBI0. An IMODULE LOAD is issued for modules DFSYINI0, DFSBLK0x, or DFSFXC00 where x is the nucleus suffix determined by the SUF parameter of the IMS startup procedure. An IMODULE GETMAIN is issued for DFSYIMI0 and DFSYINI0 to establish OTMA TSCD, MTE, and MCB control blocks. If the storage for Open Transaction Manager Access (OTMA) control blocks cannot be obtained, the abend can occur.

The program status word (PSW) at entry-to-abend can be used to isolate the location of the abend. Register 12 in the abend SVRB registers is the base register for this module. Register 15 contains the IMODULE LOAD return code indicating the cause of the failure. For a description of the IMODULE return codes, see the IMS system services return codes information.

If this abend is preceded by one or more occurrences of MSGDFS1921I, check the syntax of the DFSPBXXX member according to the instructions on specifying PROCLIB members in the IMS system definition information.

Key	Label	Description
Reg15=0	ABEND078	The IMODULE LOAD SVC has been issued, and register 15 is loaded with the IMODULE LOAD return code. If register 15 contains a nonzero return code, the routine abends.
Reg9=IMODULE LOAD subcode		Subcode=4 DFSBLKx Subcode=8 DFSFXCx  Subcode=12 DFSYINI0 or DFSYIMI0 Subcode=X'C' Failure to load module other than DFSBLKx or DFSFXCx.


**System action:** IMS terminates abnormally.

**Problem determination:** 35

**Source:** IMS abends

**Module:** DFSXDBI0

**Related reference:**

 IMS system services return codes (Messages and Codes)

**Related information:**

Chapter 2, "IMS problem determination," on page 13

**0079**

**Explanation:** This abend occurs if the IMS nucleus, load module DFSVNUCs or DFSCNUCs (s=suffix of the IMS system), could not be loaded during control region initialization. Register 15 contains the return code from IMODULE LOAD. For a description of the IMODULE return codes, see the IMS system services return codes information.

**Analysis:** 0079 is a standard abend issued by the module, DFSXNCL0, for nucleus load and address resolution. An IMODULE LOAD issued for module name DFSVNUCs or DFSCNUCs failed.

The program status word (PSW) at entry-to-abend can be used to isolate to label ABEND079, from which the abend is issued. Register 12 in the abend SVRB registers is the base register. Other SVRB registers pertinent to the resolution of this problem are indicated below. Register 15 has the IMODULE return code indicating the cause of the failure.

Key	Label	Description
Reg3=RGPparms address Reg4=IPB address Reg8=address of module name if IMODULE LOAD for DFSVNUCs or DFSCNUCs failed. Reg11=SCD address Reg15=0	ABEND079	The IMODULE LOAD SVC has been issued, and register 15 is loaded with the return code. When tested, if register 15 contains a nonzero return code, the routine abends.


**System action:** IMS terminates abnormally.

**Problem determination:** 35

**Source:** IMS abends

**Module:** DFSXNCL0

**Related reference:**

 IMS system services return codes (Messages and Codes)

**Related information:**

Chapter 2, "IMS problem determination," on page 13

## 0080

**Explanation:** An error occurred during execution of the Overflow Sequential Access Method (OSAM) OPEN/CLOSE/EOV processing.

**Analysis:** This is a standard abend issued by module DFSAOSF0 and DFSAOS10. The program status word (PSW) at entry-to-abend points to the instruction in the routine from which the abend (SVC 13) is issued. This routine is branched to from various locations in DFSAOSF0 upon detection of an error situation.

The fourth byte of register 2 in the abend SVRB registers contains the failure code, and the second byte of the register 2 contains a value corresponding to the failing process.

### Reg 2, Second Byte Failing processing

**X'10'**  
Entry processing.

**X'50'**  
Termination processing

Register 8 contains the private area address.

Register 6 points to the private area; at offset X'68' in that area, you can find the address of the TIOT entry. This entry contains the DD name.

In the processing types listed below, register 2 in the abend SVRB registers contains the failure code. Register 14 is the BAL register to the abend routine and contains the address of the location from which control was passed.

### Entry processing (label ENTABxx)

#### Codes Explanation

**X'01'**  
Incompatible operating system

## 0081

X'02'

Invalid parameter list was passed.

X'03'

Invalid function code detected.

X'04'

Zero SCD address in vector.

### Termination processing (label OTRMABxx)

#### Codes

##### Explanation

X'01'

IMODULE GETMAIN failed for CSA DEB.

X'02'

Page fix failed for CSA DEB.

X'03'

Page fix failed for final DEB.

**System action:** IMS terminates abnormally.

**Operator response:** Restart IMS and call your IMS system programmer for assistance.

**Problem determination:** 2, 3, 4

Obtain a memory dump showing both the module and the work area. If necessary, change the memory dump parameters or use a SYSABEND statement to obtain the above information.

**Source:** IMS abends

**Module:** DFSAOSF0, DFSAOS10

#### Related information:

Chapter 2, "IMS problem determination," on page 13

---

## 0081

**Explanation:** The IMS IMSAUTH SVC BLDSSCT function was unable to build the subsystem control table (SSCT) or was unable to build or set the SCD address into the termination control block. Register 14 contains the address of the program invoking the abend. Register 15 contains the IMSAUTH BLDSSCT return code. For an explanation of the IMSAUTH BLDSSCT return codes, see the IMS system services return codes information.

**Analysis:** 0081 is a standard abend issued by the storage management initialization controller, DFSXSTM0. The program status word (PSW) at entry-to-abend points to the instruction within the routine at label abend from which the abend (SVC 13) is issued. This routine is unconditionally branched to by the routine at label ABEND9, which is conditionally branched to by the routine that detected the error.

Register 12 in the abend SVRB registers is the base register for this module. Register 1 contains the abend completion code, X'80000051'. Register 15 contains a return code from the IMSAUTH macro. For an explanation of the IMSAUTH BLDSSCT return codes, see the IMS system services return codes information.

Key	Label	Description
Reg15=return code Reg14=address of SCD		In the routine to get the global control blocks, the IMSAUTH macro is issued to set SCD address into SSCT built at label IMSSCT. Register 15 is loaded with, then tested for, the return code. If the return code is other than zero, a branch is taken to label ABEND9 to handle the abend.

**System action:** IMS initialization terminates abnormally.

**Programmer response:** If the caller is not authorized, make sure that the JOB or STEP libraries are authorized and DFSPCC30 is bound with the re-entrantable (RENT) attribute.


**Problem determination:** 1, 4, 8, 35



**Source:** IMS abends

**Module:** DFSXSTM0

**Related reference:**

 [IMS system services return codes \(Messages and Codes\)](#)

**Related information:**

Chapter 2, “IMS problem determination,” on page 13

### 0083

**Explanation:** This abend occurs when ABEND is appended to message DFS1920I. The execution parameter was set with a bad value. This type of error is unrecoverable for the following IMS control region execution parameters:

- TRN
- SGN
- RCF
- ISIS

**System action:** IMS terminates abnormally.

**Programmer response:** Correct the parameter and error and rerun IMS.

**Source:** IMS abends

**Related information:**

 [DFS1920I \(Messages and Codes\)](#)

### 0084

**Explanation:** A batch message processing (BMP) program tried to access an undefined data set.

**Source:** IMS abends

### 0085

**Explanation:** The message format data set (ddname FORMATA or FORMATB) directory block that was read online conflicts with the directory block that was read at IMS initialization.

**Analysis:** 0085 is a standard abend issued by module DFSFDSC0, which pre-fetches or immediate fetches blocks into storage. The program status word (PSW) at entry-to-abend points to the instruction within label abend from which the abend (SVC 13) is issued.

Register 11 in the abend SVRB registers is the base register. It contains the entry point address of the module DFSFDSC0 at the time of abend. The subroutine at label SRCHDICT in DFSFDSC0 returns the disk address (TTR) of the requested block. There is no resident directory (register 8 is zero), so the partitioned data set (PDS) index is used to locate the member name. Register 5 contains a pointer to the member in the PDS index, and register 10 contains a pointer to the fetch request element (FRE) dsect.

Key	Label	Description
Reg5=pointer to member in the PDS index Reg9=address of end of directory I/O area	DPDSCOMP	Register 5 and register 9 are compared. If the address in register 5 points beyond the register 9 address, it is an indication that the PDS has probably been damaged. If the addresses in the two registers are equal, it means that the end-of-block has been reached without finding the member name, indicating that the member name has probably been deleted since the in-storage indexes were built. In either case, the abend is issued.

**Possible cause:** The message format data set was updated while IMS is active, without a corresponding update to the in-storage indexes.

**System action:** IMS terminates abnormally.

**Operator response:** Emergency restart IMS to resume processing.

## 0088

**Programmer response:** Make sure that the Message Format data set does not update while IMS is active. Determine whether the format library is shared between processing units. Verify that active format library is a PDS file. PDS-E is not supported for Message Format data sets in IMS.

**Problem determination:** 8, 11

**Source:** IMS abends

**Module:** DFSFDSCO

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

## 0088

**Explanation:** Initialization for the Local Storage Option (LSO) failed.

**Analysis:** 0088 is issued by modules DFSKLSO0 and DFSXLSM0 for all abnormal conditions.

Register 15 contains the following reason codes.

**Attention:** Abend 4095 is issued in DFSKETXR to propagate any LSO abends.

**For DFSXLSM0:**

**Codes**

**Explanation**

**X'01'**

The DFSBCB quick save obtain macro failed.

**X'02'**

The DFSBCB GET command for the LSO work area failed. (Register 3 = DFSBCB return code)

**X'03'**

The DFSQSS load for the LSO control module, DFSKLSO0, failed. (Register 3 = DFSQSS return code)

**X'04'**

The DFSCIR macro for the ITASK CREATE failed. (Register 3 = DFSCIR return code)

| **X'15'**

| The IMODULE load for module DFS5DL60 failed. (Register 3 = IMODULE load return code)

| **X'16'**

| The DFSCWU macro for the DFS5DL60 ITASK CREATE failed. (Register 3 = DFSCWU return code)

**For DFSKLSO0:**

**Codes**

**Explanation**

**X'05'**

The asynchronous work element (AWE) enqueued to LSO has an invalid function. Register 3 indicates the invalid function and register 9 indicates the address of AWE.

**X'06'**

At signon, the DFSBCB quick save obtain failed. (Register 3 = DFSBCB return code)

**X'07'**

At signon, the DFSBCB GET command for the LSO block failed. (Register 3 = DFSBCB return code)

**X'08'**

At signoff, no LSO was specified for PST. (Register 3 = specified PST)

**X'09'**

The system terminated because the dependents were not signed off. (Register 2 = control word contents; register 3 = first active LSO block address)

**X'0A'**

The system terminated because an invalid LSO control work update occurred. (Register 2 = control word contents)

**For DFSXLSD0:****Codes****Explanation****X'11'**

The DFSBCB quick save obtain macro failed. (Register 3 = DFSBCB return code)

**X'12'**

The DFSCIR macro for the ITASK CREATE failed. (Register 3 = DFSCIR return code)

**X'13'**

The ESTAE create failed for DFSFLSD0. (Register 3 = ESTAE service return code.)

**For DFSKLS0:****Codes****Explanation****X'14'**

The DFSBCB quick save obtain macro failed (Register 3 = DFSBCB return code).

**System action:** IMS terminates abnormally.

**Operator response:** Probably an IMS internal error. Contact your system programmer for help.

**Problem determination:** 1, 2, 3, 4, 8, 12, 35

**Source:** IMS abends

**Module:** DFSXLSM0, DFSKLSO0, DFSXLSD0, DFSKLS0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

**0090**

**Explanation:** An IMS internal post failed. A Communication Line Block (CLB) that was waiting for a format block could not be posted (nonzero return code from IPOST).

Register 8 contains the CLB address. Register 10 contains the FRE address where FRE+0 has the format block name.

**Analysis:** Register 8 in the abend SVRB registers contains the CLB address. Register 10 contains the fetch request element (FRE) address, where FRE+0 has the format block name.

**For DFSFFET0:**

0090 is a standard abend that can be issued by the format block immediate fetch module, DFSFFET0. If the error is detected in this module, the program status word (PSW) at entry-to-abend points to the instruction within label IPOSTM1 from which the abend (SVC 13) is issued. The abend is issued because IPOST failed for a request waiting for load.

Register 11 in the abend SVRB registers is the base register. Register 8 contains the address of the event control block (ECB) to be posted. Register 1 contains the abend completion code, X'8000005A'.

Key	Label	Description
Reg8=address of ECB (CLB)	IPOSTM1 (\$IPC0006)	An IPOST macro has been issued to post the ECB specified in register 8 to the appropriate events table. Register 15 is loaded with, and tested for, the return code from IPOST. If register 15 is not zero, the abend is issued.
Reg15=nonzero return code from IPOST		

**For DFSFPRF0:**

0090 is a standard abend that can be issued by module, DFSFPRF0, which prefetches blocks into storage. If the error

## 0092 • 0095

is detected in this module, the program status word (PSW) at entry-to-abend points to the instruction within label IPOSTM1 from which the abend (SVC 13) is issued. The abend is issued if the return code from the IPOST macro is not zero.

Register 11 in the abend SVRB registers is the base register. Register 8 contains the address of the ECB to be posted. Register 1 contains the abend completion code, X'8000005A'.

Key	Label	Description
Reg8=address of ECB (CLB) Reg15=nonzero return code from IPOST	IPOSTM1 (\$IPC0017)	An IPOST macro has been issued to post the ECB specified in register 8 to the appropriate events table. Register 15 is loaded with, and tested for, the return code from IPOST. If register 15 is not zero, the abend is issued.

**System action:** IMS terminates abnormally.

**Operator response:** Emergency restart IMS to resume processing.

**Problem determination:** 1, 11, 35

**Source:** IMS abends

**Module:** DFSFFET0, DFSFPRF0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

### 0092

**Explanation:** An IMODULE GETMAIN request for storage in the private or extended private area of the IMS control region was not successful.

**Analysis:** The private or extended private storage area maintains a list of nodes with input sequence numbers that must be updated during emergency restart. The set and test sequence number (STSN) nodes are obtained from the type 28 log records during emergency restart.

Register 1 contains the IMODULE parameter list address. Register 8 contains the amount of storage requested in subpool 0 of the IMS control region. Register 15 contains the return code from the IMODULE GETMAIN request.


**System action:** The IMS control region terminates abnormally.

**Programmer response:** For a description of the IMODULE GETMAIN return codes, see the IMS system services return codes information. For storage allocation errors, increase the size allocated to the private or extended private area and restart IMS.

**Source:** IMS abends

**Module:** DFSCRPO0, DFSRLP00

**Related reference:**

 IMS system services return codes (Messages and Codes)

---

### 0095

**Explanation:** An error occurred while initializing the logon, user, and MSC descriptors. This error is unrecoverable.

**Analysis:** See message DFS3658X, which was issued before the abend, for the reason for the failure.

**System action:** IMS terminates abnormally with a memory dump.

**System programmer response:** See the response for message DFS3658X.

**Problem determination:** See the problem determination for message DFS3658.

**Source:** IMS abends

**Module:** DFSIILD0

**Related information:**

 DFS3658X (Messages and Codes)

---

0097

**Explanation:** The VSAM DLVRP macro request, issued by DFSPCC30 to delete the VSAM local resource pools, failed.

**Analysis:** The VSAM return code is in register 15.

Key	Label	Description
Reg14=address of location in DFSPCC30 that detected the error Reg15=return code	ABEND97	VSAM did not satisfy the DLVRP request.

**System action:** IMS terminates abnormally.

**Programmer response:** The VSAM return code is in register 15.

**Problem determination:** 4, 35

**Source:** IMS abends

**Module:** DFSPCC30

**Related reference:**

 z/OS: DLVRP return codes

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

0098

**Explanation:** IMS issues this abend when it cannot find the user edit routine name in the user edit routine address table.

**Analysis:** While creating checkpoint records, the system was unable to locate the user edit routine name in the user edit routine name and address table. The descriptor was ignored and message DFS368OW was returned.

**System action:** IMS terminates abnormally with a memory dump.

**System programmer response:** Ensure that the terminal descriptors describe the same user edit routines that were in effect before the restart.

**Problem determination:** 1, 4, 35

**Source:** IMS abends

**Module:** DFSRCP30

**Related information:**

 DFS3680W (Messages and Codes)

Chapter 2, "IMS problem determination," on page 13

---

0100

**Explanation:** The IMS control region ended abnormally while attempting to build a type X'47' log record. Register 15 contains one of the following reason codes:

**Code**    **Meaning**

X'01'    A type X'47' log record cannot be created because the changed database list of a BMP exceeds the capacity of the log record in which it is to be built.

X'02'    An IMODULE GETMAIN request failed to acquire storage to build a type X'47' log record.

**Analysis:**

## 0100

### R15=X'01'

This abend prevents overlays in the event that BMP or IMS batch regions have corrupted database lists that will not fit into IMS log records.

### R15=X'02'

An IMODULE GETMAIN failed while attempting to acquire storage to build the type X'47' log record. Register 10 contains the IMODULE GETMAIN return code. For a description of IMODULE return codes, see the IMS system services return codes information.

**System action:** IMS ends abnormally with abend code 0100.

**Operator response:** After correcting the problem, restart IMS with the /ERESTART command.

**Programmer response:** Contact IBM Software Support.

**Source:** IMS abends

**Module:** DFSRCP10

**Related reference:**

 [IMODULE return codes \(Messages and Codes\)](#)

---

## Chapter 7. IMS abend codes 0101 - 0150

This topic identifies and explains the following abend codes issued by IMS. Each code is 4 decimal digits.

---

### 0101

**Explanation:** An error occurred during Java™-dependent region processing.

This abend can occur for non-Java-dependent regions as well, for example MPP, BMP, IFP regions.

**Analysis:** Examine the dependent region job output for the cause of the failure by searching on the character string DFSJVM00: for all instances of this abend.

#### For DFSRCJB0:

This code indicates a standard abend issued by DFSRCJB0.

Key	Label	Description
Reg15=X'3'		Required ENVIRON=member not specified.
Reg15=X'4'		Required JVMOPMAS=member not specified.
Reg15=X'6' Reg3=BPX1SDD return code Reg4=BPX1SDD reason code		BPX1SDD failed.

#### For DFSPCJB0:

0101 is a standard abend issued by DFSPCJB0.

Key	Label	Description
Reg15=X'1' Reg3=CEEPIPI return code		Unable to create LE enclave (CEEPIPI error).
Reg15=X'2' Reg3=Possible CEEPIPI return code		Unable to create Java Virtual Machine (JVM).
Reg15=X'3' Reg3=Possible CEEPIPI return code		JVM unable to start application (CEEPIPI error).
Reg15=X'4'		JVM unable to start application.
Reg15=X'B' Reg5=LE enclave termination return code Reg3=LE enclave termination reason code Reg6=LE enclave termination feedback code		The LE enclave terminated due to an unhandled condition. IMS treats this as a backout situation. The application cannot be allowed to commit because it might not have completed all its work; thus any commit would be a partial commit that is not correct.

#### For DFSPCJM0:

0101 is a standard abend issued by DFSPCJM0.

## 0101

Key	Label	Description
Reg15=X'1' Reg3=CEEPIPI return code		Unable to create LE enclave (CEEPIPI error).
Reg15=X'2' Reg3=possible CEEPIPI return code		Unable to create JVM.
Reg15=X'3' Reg3=CEEPIPI return code		JVM unable to start application (CEEPIPI error).
Reg15=X'4'		JVM unable to start application.
Reg15=X'B' Reg5=LE enclave termination return code Reg3=LE enclave termination reason code Reg6=LE enclave termination feedback code		The LE enclave terminated due to an unhandled condition. IMS treats this as a backout situation. The application cannot be allowed to commit because it might not have completed all its work; thus any commit would be a partial commit that is not correct.

### For DFSRCJM0:

0101 is a standard abend issued by DFSRCJM0.

Key	Label	Description
Reg15=X'3'		Required ENVIRON=member not specified.
Reg15=X'4'		Required JVMOPMAS=member not specified.
Reg15=X'6' Reg3=BPX1SDD return code Reg4=BPX1SDD reason code		BPX1SDD failed.

### For DFSRRC10, DFSRRC40:

0101 is a standard abend issued by DFSRRC10 or DFSRRC40.

Key	Label	Description
Reg15=X'4'		Required JVMOPMAS= <i>member</i> was not specified.
Reg15=X'6' Reg3=BPX1SDD return code Reg4=BPX1SDD reason code		BPX1SDD failed.

### For DFSPCC20:

0101 is a standard abend issued by DFSPCC20.

Register 7 and Register 8 together contain the value of field RCPGM (program name) at the time of abend.

Key	Label	Description
Reg15=X'1' Reg3=CEEPIPI return code		Unable to create the LE enclave (CEEPIPI error).
Reg15=X'2' Reg3=Possible CEEPIPI return code		Unable to create the JVM.



Key	Label	Description
Reg15='X'3' Reg3=Possible CEEPIPI return code		CEEPIPI was unable to start the application.
Reg15='X'5' Reg3=Possible CEEPIPI return code		Unable to add an entry (CEEPIPI error).  If Reg3='X'0C', message DFS650E precedes this abend.
Reg15='X'7' Reg3=IMODULE GETMAIN return code		Failed to obtain storage for LE @LOAD table.
Reg15='X'8' Reg3=IMODULE GETMAIN return code		Failed to obtain storage for LE PIPI table.
Reg15='X'9' Reg3=CEEPIPI return code		Unable to call LE @LOAD table scrub routine (PC20LETS).
Reg15='X'A'		An unhandled Java exception existed after the application returned to IMS. IMS captures and displays the exception in the job output. IMS treats this as a backout situation. The application cannot be allowed to commit because it might not have completed all its work; thus any commit would be a partial commit that is not correct.
Reg15='X'B' Reg5=LE enclave termination return code Reg3=LE enclave termination reason code Reg6=LE enclave termination feedback code		The LE enclave terminated due to an unhandled condition. IMS treats this as a backout situation. The application cannot be allowed to commit because it might not have completed all its work; thus any commit would be a partial commit that is not correct.
Reg15='X'C' Reg3=CEEPIPI return code		CEEPIPI was unable to perform the add_entry function for the APPTERMEXIT. A possible cause is that the user exit specified on the APPTERMEXIT= option does not exist or the user exit name was misspelled.
Reg15='X'D' Reg3=CEEPIPI return code		CEEPIPI was unable to invoke the user exit specified on the APPTERMEXIT= option.

#### For DBFFPPR0, DFSTMPR0:

0101 is a standard abend issued by DBFFPPR0 or DFSTMPR0.

Key	Label	Description
Reg15='X'A'		An unhandled Java exception existed after the application returned to IMS. IMS captures and displays the exception in the job output. IMS treats this as a backout situation. The application cannot be allowed to commit because it might not have completed all its work; thus any commit would be a partial commit that is not correct.

For CEEPIPI errors, see *z/OS Language Environment Programming Guide*.

**System action:** The dependent region is terminated abnormally unless the abend is issued by DFSPCJM0 with a reason code of 3 or 4.


**Programmer response:** For all instances of this abend, examine the dependent region job output for the cause of the failure by searching on the character string DFSJVM00:.

**Source:** IMS abends

## 0102

**Module:** DFSRCJB0, DFSRCJM0, DFSPCJB0, DFSPCJM0, DFSRCJM0, DBFFPPR0, DFSTMPR0

**Related reference:**

 z/OS: Application invocation (CEEPIPI)

---

## 0102

**Explanation:** An error was detected in doing the positioning for an extended restart.

0102 is a standard abend. The program restart handler can issue this abend when attempting to locate the symbolic checkpoint records from the OLDS or IMSLOGR data set. Message DFS1000I is issued before this abend. The program status word (PSW) points to the instruction within the module from which ABEND (SVC 13) is issued. Register 14 is the BAL return register to the error discoverer. Register 2 contains the error reason code set by the error discoverer. This error reason code corresponds to the label name at the failing location.

For additional information, see message DFS1000I.

**Analysis:** During online batch processing, IMS determines the OLDS position from the last complete checkpoint of the BMP or JBP application from a Residual Recovery Element (RRE) entry in an internal IMS table. The RRE entry identifies a BMP or JBP application by the job, PSB, and program name. For IMS to locate this table entry when the BMP or JBP application restarts, the BMP or JBP application must have the same job, PSB, and program name that it had when the program terminated abnormally.

After finding the table entry, if a data set with a ddname of IMSLOGR has been allocated in the JCL of the BMP or JBP region, the application restart process attempts to access the checkpoint records only from the data set with the IMSLOGR ddname. Otherwise, IMS tries to locate the last checkpoint records from the OLDS, if the OLDS has not been overwritten since the last checkpoint was recorded.

In a batch environment, the checkpoint records are accessed from the data set with a ddname of IMSLOGR.

**For DFSZSR00:**

Key	Label	Description
Reg2=E2D90002	E2D90002	Nonzero return code was returned from the internal extended restart (XRST) call to module DFSDLA00. A PSEUDO ABEND is set in DFSCPY00. This problem is an IMS internal error.
Reg2=E2D90003	E2D90003	Nonblank status code was returned from the internal XRST call to module DFSDLA00. The actual segment (or path of segments) in the PSB was longer than the length coded (in binary) in the area pointed to by the XRST parameter I/O area gen.
Reg2=E2D90004	E2D90004	LAST was specified as the parameter from which to process the XRST call, but the process cannot be completed. An error reason code was returned by module DFSDLA00 in register 3.
Reg3=C4D3A001		First time error
Reg3=C4D3A002		No RRE was found for this BMP. 1. A cold start was performed. 2. A BMP with the same job name, program name, and PSB name ran to normal completion. 3. No checkpoints were taken for the failed BMP. 4. The PSB name, program name, and job name specified during the extended restart did not match the PSB name, program name, and job name of the BMP in which the original abend occurred.
Reg3=C4D3A003		A nonzero return code was received from the ILOG <sup>®</sup> read of the OLDS.
Reg3=C4D3A004		An error was returned from the log read exit routine, LOGREXIT.
Reg2=E2D90005 Reg9=//IMSLOGR DCB address	E2D90005	After issuing the OPEN macro for the IMSLOGR data set, the DCBOFLGS indicated the data set was not opened. A wrong data set was mounted or the //IMSLOGR DD statement is missing.  You must provide the correct or recovered IMSLOGR data set and rerun the job.

Key	Label	Description
Reg2=E2D90006	E2D90006	LAST was invalidly specified as the parameter from which to restart in a batch environment.  Rerun the job, specifying the correct parameter.
Reg2=E2D90007 Reg9=//IMSLOGR	E2D90007	The SYNAD exit routine (label IOERROR) for the IMSLOGR data set was taken because an unrecoverable error was detected.  You must provide the correct or recovered IMSLOGR data set and rerun the job.
Reg2=E2D90008	E2D90008	The record returned by the READ of the IMSLOGR data set was not formatted according to the BSAM rules of variable blocking. The data set referred to by the //IMSLOGR DD statement is probably not a valid IMS log.  You must provide the correct or recovered IMSLOGR data set and rerun the job.
Reg2=E2D90009 Reg5=a (checkpoint ID) Reg9=//IMSLOGR DCB address	E2D90009	An end-of-data (EOD) on the //IMSLOGR data set was reached without finding a type X'18' log record for the checkpoint ID that was passed with the XRST call. <ul style="list-style-type: none"> <li>• An incorrect checkpoint ID was specified.</li> <li>• An incorrect or insufficient number of volume serials was specified.</li> <li>• The program name on the execute statement does not match the program name in the type X'18' log records.</li> <li>• An incorrect job name was used. The restarted job name must match the original job name.</li> </ul> You must provide the correct or recovered IMSLOGR data set and rerun the job.
Reg2=E2D9000A	E2D9000A	An invalid call was made to the SVC interface. This problem is an IMS internal error.
Reg2=E2D9000B	E2D9000B	A restart is being attempted from a checkpoint ID that is not the last one taken. There is a full-function database PCB within the PSB. The restart must start from the last checkpoint taken.  Rerun the job, specifying the correct parameter.
Reg2=E2D9000C	E2D9000C	A restart is being attempted from a checkpoint ID that is not found in the OLDS.  Rerun the job, specifying an IMSLOGR DD statement that references the data set that contains the required checkpoint records.

**For DFSZSR10:**

Key	Label	Description
Reg2=E2D91001 Reg7=a(DBPCB) Reg8=a(bucket within the X'18' log record)	E2D91001	The PCBs of the PSB under which the program is restarting are not in the same order, or do not specify the same DBDs or logical terminals as those of the PSB when the program checkpointed. The DBD name in the PCB is compared with the data set DBD name in the X'18' log record. They must be equal.  A user block GEN was done. Rerun with a prior GEN.

Key	Label	Description
Reg2=E2D91002	E2D91002	End-of-data (EOD) was reached while repositioning a Generalized Sequential Access Method (GSAM) non-DASD data set. Message DFS0780I is issued specifying the ddname.  The GSAM data set is not the same as when the checkpoint was taken.
Reg2=E2D91003 Reg6=length of checkpointed area from X'18' log record Reg9=address of the length passed with the XRST call	E2D91003	The XRST call area list was specified incorrectly. The list length is insufficient to contain the addresses and lengths of all areas, or the user-supplied area is not the same length as the checkpointed data.
Reg2=E2D91004 Reg15=Pseudo Abend code from DFSCPY00	E2D91004	DFSCPY00 has returned a pseudoabend code in register 15.
Reg2=E2D91005	E2D91005	Nonblank status code from internal XRST call to module DFSDLA00.
Reg2=E2D91006  Reg3=C4D3A003  Reg3=C4D3A004	E2D91006	Error reading checkpoint records from OLDS. An error reason code was returned by module DFSDLA00 in register 3.  A nonzero return code was received from the ILOG read of the OLDS.  An error was returned from the log read exit routine, LOGREXIT.
Reg2=E2D91007 Reg9=//IMSLOGR DCB address	E2D91007	The SYNAD exit routine (label IOERROR) for the IMSLOGR data set was taken because an unrecoverable error was detected.  You must provide the correct or recovered IMSLOGR data set and rerun the job.
Reg2=E2D91008	E2D91008	The record returned by the READ of the IMSLOGR data set was not formatted according to the BSAM rules of variable blocking. The data set referred to by the //IMSLOGR DD statement is probably not a valid IMS log.
Reg2=E2D91009 Reg5=a (checkpoint ID) Reg9=//IMSLOGR DCB address	E2D91009	An end-of-data (EOD) on the IMSLOGR data set was reached without finding a type X'18' log record for the checkpoint ID that was passed with the XRST call. <ul style="list-style-type: none"> <li>• An incorrect checkpoint ID was specified.</li> <li>• An incorrect or insufficient number of volume serials was specified.</li> </ul> You must provide the correct or recovered IMSLOGR data set and rerun the job.
Reg2=E2D9100A Reg15=Return Code	E2D9100A	Nonzero return code on internal PC call to inter-region communication (IRC).
Reg2=E2D100B	E2D9100B	Invalid call to position a GSAM database.

Key	Label	Description
Reg2=E2D9100C	E2D9100C	The PCB name field in one of the database PCBs at restart time does not match the PCB name in the checkpoint record. If the names are valid, but mismatched, then a PSBGEN might have added or deleted PCBs from the PSB between checkpoint and restart. When one of the names is invalid, it is often the result of an overlay of the PCB caused by application or PSB member changes to a prior PCB in storage. Examine the name field for possible clues.

#### For DFSZDC00

Key	Label	Description
Reg2=C4C30001	C4C30001	<p>During XRST processing, GSAM determined that the output data set being repositioned was empty. To successfully resubmit the XRST job:</p> <ol style="list-style-type: none"> <li>1. Issue a /STA command for the PSB stopped by the abending job.</li> <li>2. Correct the DD statement for the GSAM output data set to point to the same data set in use when the abend requiring an XRST occurred.</li> <li>3. Resubmit the XRST job.</li> </ol> <p><b>Note:</b> Probably the most common cause of this problem is associated with the use of Generation Data Group (GDG) data sets. After the initial job fails, the XRST job is submitted. However, you failed to change the generation number on the GSAM output data set from (+1) to (0). As a result, XRST repositioning is performed on a new (+1) data set, not the current (0 at abend time) copy of the data set.</p> <p><b>Attention:</b> If an XRST job is executed specifying the wrong restart data set, XRST completes successfully but valid data is overwritten.</p>

**System action:** The program terminates abnormally.

**Operator response:** Determine the cause of the error, correct the problem, and rerun the job.

**Source:** IMS abends

**Module:** DFSZSR00, DFSZSR10, DFSZDC00

**Related information:**

 DFS1000I (Messages and Codes)

---

#### 0103

**Explanation:** The length field for one of the user areas pointed to by the parameter list for the CHKP call is either zero or negative.

This is a standard abend that the extended checkpoint module might issue in attempting to write the symbolic checkpoint records to the OLDS or IMSLOGR data set. Message DFS1000I is issued before this abend. The program status word (PSW) points to the instruction within the module from which the ABEND (SVC 13) is issued. Register 14 is the BAL return register to the error discoverer. Register 2 contains the error reason code set by the error discoverer. This error reason code corresponds to the label name at the failing location.

**Analysis:** The extended checkpoint module writes symbolic checkpoint records which contain database positioning data (RRA or SSA). The records also contain any area specified by the user positioning parameter in the CHKP call.

Key	Label	Description
Reg2=E2C30001	E2C30001	An attempt was made to issue a CHKP call and the application has a main storage database (MSDB) area defined within it. This is not supported.

**0106**

Key	Label	Description
Reg2=E2C30002	E2C30002	There was an attempt to write a checkpoint LOG record greater than the maximum record size allowed.
Reg2=E2C30003	E2C30003	The internal ILOG call from the BMP region received a nonzero return code. Register 15 contains the return code.
Reg2=E2C30004	E2C30004	A nonblank status code was returned from the internal ILOG call. Register 3 contains the status code.
Reg2=E2C30005 Reg9=address of length in parameter list	E2C30005	The length field for one of the user areas pointed to by the parameter list for the CHKP call is either zero or negative. Register 5 contains the value.
Reg2=E2C30006	E2C30006	The MOVE LONG (MVCL) instruction within the USERLOG routine has detected a destructive overlay. The user area length specified in the CHKP call is too long.

**System action:** The application program terminates abnormally.

**Programmer response:** Check the length fields for the user areas that you have specified in the parameter list for the CHKP call. If one is found to be invalid, change it and rerun the job.

**Problem determination:** 2, 3, 11

**Source:** IMS abends

**Module:** DFSZSC00

**Related information:**

Chapter 2, "IMS problem determination," on page 13

**0106**

**Explanation:** A nonzero return code was issued in response to an IMODULE LOAD request during dependent region initialization. IMODULE LOAD for the Environment Controller (DFSECP10 or DFSECP20) failed with a nonzero return code. This is a standard abend issued by module DFSPCC20. Register 15 contains the return code. For a description of the IMODULE LOAD return codes, see the IMS system services return codes information.

**Analysis:** This is a standard abend issued by module DFSPCC20 as a result of failure to load. Register 15 contains the return codes in the IMODULE LOAD prolog.

Key	Label	Description
Reg1=X'8000006A'	PCAB106	IMODULE error.

**System action:** The application program terminates abnormally.


**Programmer response:** Correct the error and restart IMS.

**Problem determination:** 4

**Source:** IMS abends

**Module:** DFSPCC20

**Related reference:**

 IMS system services return codes (Messages and Codes)

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

**0107**

**Explanation:** Resource access security initialization encountered a problem. One of the following problems occurred:

- One of the security modules failed to load completely.
- If RACF is used (z/OS only), RACF is at the wrong level or is not installed on the system.

The hexadecimal error subcode in register 15 indicates why the initialization failed.

**Analysis:** This is a standard abend issued by module DFSXLIC0. Register 15 contains one of the following codes:

**Codes****Explanation****X'01'**

IMODULE LOAD failed for module DFSSCHR0.

**X'02'**

Reserved.

**X'03'**

IMODULE LOAD failed for module DFSRAS00.

**X'04'**

An attempt to define the Resource Access Security user exit (RASE) to IMS failed.

**X'05'**

An attempt to get storage for the initialization call of the Resource Access Security user exit (RASE) failed.

**System action:** The application program terminates abnormally.

**Programmer response:** Correct the error and restart IMS.

**Source:** IMS abends

**Module:** DFSXLIC0

---

**0108**

**Explanation:** A timer request failure occurred during an LU 6.2 shutdown. The LU 6.2 shutdown is not completed and is unable to proceed. DFSTIMER has returned with a bad return code.

**Analysis:** 0108 is a user abend issued by DFSASLT1 to prevent loss of LU 6.2 messages during normal IMS shutdown.

**System action:** The shutdown is terminated abnormally.

**Programmer response:** Contact IBM Software Support.

**Problem determination:** 1, 5, 6, 8, 10, 27, 34, 36

**Source:** IMS abends

**Module:** DFSASLT1

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

**0109**

**Explanation:** For modules DBFSLM62 and DFSTMS00:

IMS failed to deliver a response for the APPC synchronous transaction or OTMA send-then-comment (CM1) transaction.

Or, the LU 6.2 device or OTMA client returned a negative acknowledgment (NACK) for an output message of send-then-commit (CM1) transaction with synchronization level of CONFIRM (SL1) by using XCF communication.

The DFS554A or DFS555I message contains the abend reason code if issued.

The reason code is shown in the following table:

## 0110

Table 6. Possible reason codes for DBFSLM62, DFSTMS00

Code	Meaning
X'01'	A NACK is received for the SEND of the output message of a send-then-commit (CM1) transaction with sync level of NONE or CONFIRM (SL0 or SL1) where XCF communication is used.

### For modules DBFGHU10 and DFSDLA30:

The GU process discards an APPC or OTMA send-then-commit (CM1) transaction with sync level of NONE or CONFIRM (SL0 and SL1) where XCF communication is used, when one of the following conditions is detected:

- The front-end IMS system has terminated.
- The transaction message timestamp is earlier than the FE IMS system restart timestamp.

The DFS554A or DFS555I message contains the abend reason code if issued.

The reason code is shown in the following table:

Table 7. Possible reason codes for DBFGHU10, DFSDLA30

Code	Meaning
X'02'	A send-then-commit (CM1) transaction with sync level of NONE or CONFIRM (SL0 or SL1) where XCF communication is used, is discarded.

**Analysis:** 0109 abend ended this instance of the transaction.

**System action:** If this abend occurs during dependent region processing, the dependent region terminates abnormally.

**Programmer response:** Determine the cause of the negative acknowledgment.

**Problem determination:** 1, 5, 6, 8, 10, 27, 34, 36


**Source:** IMS abends

**Module:** DBFHGU10, DBFSLM62, DFSDLA30, DFSTMS00

**Related information:**

 DFS554A (Messages and Codes)

 DFS555I (Messages and Codes)

 IMS problem determination (Messages and Codes)

---

## 0110

**Explanation:** A Coordinator Control (CCTL) subsystem issued an INIT call to connect to a Database Control (DBCTL) subsystem. The connection routines sent a request to the IMS control region to load the DBCTL modules, but the load request failed.

**System action:** IMS terminated the connection process and issued message DFS697I.

**System programmer response:** See message DFS697I.

**Problem determination:** 6, 8, 10, 38

**Source:** IMS abends

**Related information:**

 DFS697I (Messages and Codes)

Chapter 2, "IMS problem determination," on page 13

---



---

**0111**

**Explanation:** A Coordinator Control (CCTL) subsystem issued an INIT call to connect to a Database Control (DBCTL) subsystem. DBCTL built a subsystem status index entry (SIDX) for this CCTL and tried to enqueue it to the SIDX chain, but the enqueue request failed.

**Analysis:** 0111 is a pseudoabend set in module DFSDASIO.

Check the return code in field SSPSCODE of the subsystem options block (SSOB) from the SNAP or SDUMP data set. The return codes have the following meanings:

**Codes****Explanation****X'14'**

No SIDX element for enqueue

**Others**

Internal errors

**System action:** IMS terminated the connection process.

**System programmer response:** Bring down the IMS control region with a memory dump and perform an emergency restart.

**Problem determination:** 1, 4, 6, 11, 15, 35, 38

**Source:** IMS abends

**Module:** DFSDASIO

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

**0112**

**Explanation:** A Coordinator Control (CCTL) subsystem requested termination (purge) of a Database Control (DBCTL) subsystem thread.

**System action:** IMS purged the thread and the thread TCB.

**Source:** IMS abends

---

**0113**

**Explanation:** An IMS abnormal termination was detected while DL/I or DC was processing a call in a dependent region. The diagnostic area for the region causing this abend contains the program status word (PSW), general-purpose registers, and a copy of the IMS-related blocks with eye-catchers at entry to the abend.

**Note:** If offline memory dump formatting is chosen, the memory dump is created during the dependent region abend processing and shows the dependent region abend code. The control region then terminates with abend 0113, but no additional memory dump is created unless the SDUMP was not successful. Analyzing dependent region SDUMPs is the same as analyzing other IMS SDUMPs.

| You can implement the MVS IEFUTL Timeout exit routine to capture and override abends that result from CPU or  
| wait time limits being exceeded. DFSUTL is a sample exit source that is provided with IMS. If a BMP must be  
| canceled, you can wait for a time when the impact will be minimal.

**Analysis: For DFSASK00, DFSFUNL0:**

An abend occurred in a message region or batch message region while DL/I or DC was processing a call in the message region.

This is a standard abend primarily issued by module DFSASK00. Other modules from IRC/ISI service calls (Fast Path, external subsystem, and normal DL/I or DC database) issue this abend.

IMS latch management (DFSFUNL0) also issues this abend. Module DFSFUNL0 comprises three CSECTS: DFSFUNL0 (get a latch), DFSFLAT0, and DFSFLRC0 (latch recovery). Latch recovery actually issues this abend. CSECT DFSFLRC0 issues abend 0113.

## 0113

The partition specification table (PST) diagnostic area for the region causing this abend contains program status word (PSW) and general-purpose registers, and a copy of the PST at entry to the abend.

The service request block (SRB) routine, ASKSRBE, in DFSASK00, stores the address of DFSABSAV in field SRBEP after it gets control, but before it establishes the IRB that issues the abend 0113.

Locate the diagnostic work areas as follows:

1. Locate the control region mother task load list or CDE and find module DFSBLKXX.  
DFSBLKXX = SCD address
2. Locate the system diagnostic work area in the SCD (SCDU113).
  - The SCDU113 field points to the U113 SRB. The address of the diagnostic area is located in the SRB + X'14'.
  - Locate the CDE chain under the job step TCB. Find module DFSABSAV, which is the system diagnostic work area.
  - Field SRBEP also contains the address of DFSABSAV.
3. The formatted section for abend 0113 or the non-IMS formatted memory dump, locate the system diagnostic work area for the message region that terminated abnormally during IMS processing. The following three areas outline the format of the system diagnostic work area:  
SDWA  
PST  
SAP
4. Locate the associated message region dump (ABENDU0002) and follow standard diagnostic procedures.

**Attention:** If offline memory dump formatting is chosen, the memory dump is taken during the dependent region abend processing and shows the dependent region abend code. The control region is then terminated with 0113, but no additional memory dump is taken unless the offline memory dump formatting was not successful. Analyzing dependent region SDUMPs is the same as analyzing other IMS SDUMPs.

### For DFSSDA20:

The IMS end of task (EOT) exit routine discovered a dependent region active in a DL/I or DC call. DFSABSAV does not contain abend 0113 diagnostic information. Instead, at the time of the abend register 4 contains the address of the PST of the original abend.

Locate the PST pointed to by register 4 at the time of the abend. Get the job name from the PSTNJOB field of the PST in the control region memory dump. Use the job name to locate the dependent region in which the original abend occurred. The dump contains PSW registers, the completion code for the original abend, and an RTM2WA for the 0002.

Take the appropriate action indicated by the completion code of the original abend.

An additional test of the IMS Dispatcher Cross-Memory flag (SAPS3CXM) also issues this abend if both the LPS and Dispatcher Cross-Memory flags are on.

### For DBFATRM0:

Fast Path detected an 'in Fast Path' condition for a dependent region that is terminating. DFSABSAV does not contain abend 0113 diagnostic information. Instead, at the time of the abend, register 4 contains the address of the PST of the original abend.

Locate the PST pointed to by register 4 at the time of the abend. Get the job name from the PSTNJOB field of the PST in the control region memory dump. Use the job name to locate the dependent region in which the original abend occurred. The dump contains PSW registers, the completion code for the original abend, and an RTM2WA for the 0002.

Take the appropriate action indicated by the completion code of the original abend.

An additional test of the IMS Dispatcher Cross-Memory flag (SAPS3CXM) also issues this abend if both the LPS and Dispatcher Cross-Memory flags are on.

**System action:** The IMS control region terminates abnormally.

**Operator response:** Emergency restart IMS to bring up the control region.

**Problem determination:** 1, 4, 6, 14, 35

Do not submit an APAR if the abend was the result of MVS CANCEL, SMF JOB STEP TIMING, /STOP REGION CANCEL, or other user-initiated abends.

**Source:** IMS abends

**Module:** DFSASK00, DFSSDA20, DBFATRM0, DFSFUNL0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

#### 0114

**Explanation:** An IMODULE GETMAIN request failed to obtain the DFSSCHR work area.

**Analysis:** 0114 is a standard abend issued by module DFSSCHR0. An IMODULE GETMAIN failure occurred while trying to obtain the DFSSCHR work area. Register 15 contains the IMODULE GETMAIN return code. For a description of the IMODULE return codes, see the IMS system services return codes information.


**System action:** The dependent region is terminated.

**Programmer response:** Increase the region size and resubmit the job.

**Source:** IMS abends

**Module:** DFSSCHR0

**Related reference:**

 [IMS system services return codes \(Messages and Codes\)](#)

#### 0119

**Explanation:** An LU 6.2 session failure occurred on an LU 6.2 synchronous conversation or a SEND failure occurred on an OTMA SEND-THEN-COMMIT transaction. This abend can occur when:

- An LU 6.2 session failure occurred on an LU 6.2 synchronous conversation. Either the User Destination exit routine elected to end the transaction rather than send the output message asynchronously, or a User Destination exit routine was not provided.
- A SEND failure occurred on an OTMA SEND-THEN-COMMIT transaction. This can happen when:
  - The TPIPE on which the reply is to be sent is stopped.
  - OTMA is stopped.
  - The OTMA client has left the Cross System Coupling Facility (XCF) group.
  - The OTMA output message segment size was greater than 32K.

The 0119 code was used to end this instance of the transaction. The program and transaction were not stopped.

**System action:** If the abend was issued for an LU 6.2 session failure, the shutdown continues normally. For an OTMA error, both the incoming SEND-THEN-COMMIT message to IMS, and the REPLY message to the client are lost.

**Programmer response:** Determine the cause of the session or SEND failure.

**Problem determination:** 1, 5, 6, 8, 10, 27, 34, 36

**Source:** IMS abends

**Module:** DFSTMS00

**Related information:**

 [DFS555I \(Messages and Codes\)](#)

Chapter 2, "IMS problem determination," on page 13

## 0120

**Explanation:** An error was detected by module DFSXNCL0 while loading the IMS nucleus. The error is one of the following:

- When DFSXNCL0 loads the IMS nucleus, it performs a validity check on the IMS product number stored within the nucleus. The nucleus product number must match the product number stored within DFSXNCL0. This check is to ensure that the IMS nucleus was built using the same IMS release macros as the IMS that is executing. If the nucleus product number does not match, then DFSXNCL0 issues an abend 0120. Register 15 at abend will be zero for this case. You must ensure that the IMS system definition is done with the IMS macro libraries for the IMS release that you are executing.
- When IMS Fast Path is installed, DFSXNCL0 performs address resolution on the SCD extensions. If an SCD extension cannot be located (if the LOCESCD service fails), then DFSXNCL0 issues an abend 0120. Register 15 at abend will be non-zero, and will point to the address of label XNCLABND within module DFSXNCL0.

**Analysis:** 0120 is a standard abend issued from module DFSXNCL0, the module for nucleus load and address resolution, during control region initialization. The program status word (PSW) at entry-to-abend points to the instruction within label XNCLABND from which the abend (SVC 13) is issued. Register 12 in the abend SVRB registers is the base register; register 11 contains the address of the SCD.

Key	Label	Description
Reg6=address of entry point of load module DFSVNUCs (where 's' is the nucleus suffix) as returned from IMODULE LOAD Reg15=0	NCL0100	When the control region nucleus is loaded, its validity is verified by checking the copyright statement in the IMS nucleus load module. If these copyright statements are not compatible, this abend is issued. The nucleus entry point DFSIOPL0 is at displacement 0 in the nucleus. Ensure that the IMS system definition is done with the IMS macro libraries for the IMS version that you are executing. The discrepancy between the IMS macro libraries and IMS version used is a common reason for this problem.
Reg6=address of ESCD ID in option list Reg15=address of label XNCLABND	XNCLLOCE	When IMS Fast Path is installed, the address resolution is also performed on the SCD extensions. When the LOCESCD macro fails, the abend is issued. Register 6 points to the ESCD ID of the ESCD that was not located (that is, not chained to the SCD using SCDESCDQ).

**System action:** The IMS control region terminates abnormally.

**Problem determination:** 4, 35

**Source:** IMS abends

**Module:** DFSXNCL0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

## 0121

**Explanation:** When the IMS control blocks module (DFSBLK00) was loaded, the SCD (CSECT DFSISCD) was not the entry point, as required.

**Analysis:** 0121 is a standard abend issued from DFSXDBI0, the control block module loader, during control region initialization. The program status word (PSW) at entry-to-abend can be used to isolate label DBI2000, from which the abend is issued.

Register 12 in the abend SVRB registers is the base register. Other pertinent SVRB registers are detailed in the following table:

Key	Label	Description
Reg8=address of module name Reg9=BAL Reg10=load module entry point	DBI2000 (prior to label)	When module DFSBLK00 is loaded, the first few bytes are checked to determine if the SCD is the entry point for this module, that is, the first byte is compared for the first half of the user IMS SVC (0A), and the third and fourth bytes are compared for a branch (07FE). If either or both of these bytes do not compare, module DFSBLK00 is of an invalid format, and a branch is taken to label DBI2000 to abend.

**Possible cause:** A modification of the binder control statements was made.

**System action:** IMS terminates abnormally.

**Programmer response:** Rebind DFSBLK0x with the correct entry point.

**Source:** IMS abends

**Module:** DFSXDBI0

### 0123

**Explanation:** A deadlock condition that cannot be retried was detected while an IMS database call for a modified DL/I application program was being processed. The modified standard DL/I program had allocated at least one LU 6.2 conversation when the deadlock condition was detected. This abend is similar to abend 0777, but cannot be retried; the input message is discarded. CPI communications driven programs cannot be retried.

The application program had allocated at least one CPI-C conversation when the deadlock was detected. This abend is similar to abend 0777, but is not retrievable and the input message is discarded. CPI communications driven application programs are always non-retriable.

**Analysis:** 0123 is a pseudoabend, set by module DFSTMR00 and issued by modules DFSECP10 and DFSECP20.

**System action:** The modified application DL/I program terminates abnormally, and the input message is discarded.

**Programmer response:** Reenter the transaction.

**Source:** IMS abends

**Module:** DFSECP10, DFSECP20

### 0124

**Explanation:** A U2478 condition that cannot be retried was detected while an IMS database call for a modified application DL/I program was being processed. The application program had allocated at least one one CPI-C conversation when the number of PI waiters exceeded 63. This abend is similar to abend 2478, but cannot be retried; the input message is discarded. CPI communications driven programs cannot be retried.

**Analysis:** 0124 is a pseudoabend, set by module DFSTMR00 and issued by modules DFSECP10 and DFSECP20.

**System action:** The modified application program terminates abnormally and the input message is discarded.

**Programmer response:** Reenter the transaction.

**Source:** IMS abends

**Module:** DFSECP10, DFSECP20

### 0125

**Explanation:** A LOCK REJECT condition that cannot be retried was detected while an IMS database call for a modified application DL/I program was being processed. The application program had allocated at least one CPI-C conversation when the lock reject occurred. This abend is similar to abend 3303, but cannot be retried; the input message is discarded. CPI communications driven programs cannot be retried.

**System action:** The modified application program terminates abnormally and the input message is discarded.

**Programmer response:** Reenter the transaction.

## 0126 • 0129

**Source:** IMS abends

**Module:** DFSECP10, DFSECP20

---

### 0126

**Explanation:** An abend 2479 condition that cannot be retried was detected for a modified application DL/I program. The modified standard DL/I program had allocated at least one CPI-CI conversation when IMS detected that insufficient storage exists for a Buffer Queue Element (BQEL). This abend is similar to abend 2479, but cannot be retried; the input message is discarded. CPI communications driven programs cannot be retried.

**Analysis:** 0126 is a pseudoabend, set by module DFSTMR00 and issued by modules DFSECP10 and DFSECP20.

**System action:** The modified application program terminates abnormally, and the input message is discarded.

**Programmer response:** Reenter the transaction.

**Source:** IMS abends

**Module:** DFSECP10, DFSECP20

---

### 0127

**Explanation:** An SAA communications driven program terminated with IMS resources in an uncommitted state. This condition occurs when the CPI communications driven program fails to issue CPI-RR COMMIT(SRRRCMIT) or BACKOUT(SRRBACK) calls before terminating. IMS attempted to perform an implicit commit but the commit failed. IMS managed resources were in an uncommitted state. All of the uncommitted changes were backed out. CPI communications driven programs cannot be retried.

**Analysis:** 0127 is a standard abend, issued by DFSPCC20.

**System action:** The program terminates abnormally.

**Programmer response:** Ensure the resources are recovered.

**Source:** IMS abends

**Module:** DFSPCC20

---

### 0128

**Explanation:** An APPC/MVS ASSOCIATE call failed. An ASSOCIATE call informs APPC/MVS where the LU 6.2 conversation is processed.

**Analysis:** 0128 is a pseudoabend set by module DFSASK00 or module DFSCPY00 and issued by modules DFSECP10 and DFSECP20. A X'67D0' log record is written to indicate the return code from the attempted call. The input message is put back on the message queue.

**System action:** The program terminates abnormally. The input message is requeued for scheduling, and the conversation is changed to asynchronous.

**Programmer response:** Reenter the transaction.

**Source:** IMS abends

**Module:** DFSECP10, DFSECP20

---

### 0129

**Explanation:** A RACF or equivalent call to create a copy of an accessor environment element (ACEE) failed. This call establishes the security environment for the dependent region.

**Analysis:** 0129 is a pseudoabend set by modules DFSASK00 or module DFSCPY00, and issued by modules DFSECP10 and DFSECP20. A X'67D0' log record is written to indicate the return code from the attempted call. The input message is discarded from the system if RACF or equivalent indicates an actual security violation. If the input message has been discarded, the DFS554A message indicates that neither the transaction nor the program were stopped. If the input message is put back on the message queue (for example, RACF or equivalent is not active), the DFS554A message indicates that both the transaction and program were stopped.

The input message is discarded from the system if RACF or equivalent indicates an actual security violation. If the input message has been discarded, the DFS554A message indicates that neither the transaction nor the program were stopped. If the input message is put back on the message queue (for example, RACF or equivalent is not active), the DFS554A message indicates that both the transaction and program were stopped.

**System action:** The program terminates abnormally. The input message is discarded from the system if RACF or equivalent indicates an actual security violation. If the input message has been discarded, the DFS554A message will indicate that neither the transaction nor the program was stopped. If the input message is put back on the message queue (for example, RACF or equivalent is not active), the DFS554A message will indicate that both the transaction and program were stopped.

**Programmer response:** Ensure RACF or equivalent is available and start the transaction.

**Source:** IMS abends

**Module:** DFSECP10, DFSECP20

**Related information:**

 DFS554A

### 0130

**Explanation:** An message processing program (MPP) region attempted to connect to a DBCTL region. The attempt is invalid because DBCTL does not allow a connection from MPP regions.

**Analysis:** 0130 is a pseudoabend, issued by modules DFSASK00 during the SIGNON attempt by an MPP region.

**System action:** The MPP receives pseudoabend 0130.

**Programmer response:** Do not start an MPP that attempts to connect to an IMSID that is associated with an active DBCTL region.

**Source:** IMS abends

**Module:** DFSASK00

### 0131

**Explanation:** In the identify function of DFSASK00, a nonzero return code was returned from an IMODULE GETMAIN request for SP230 storage for DFSSRB.

**Analysis:** This is a pseudoabend issued by module DFSASK00 and DFSDASIO.

**System action:** The dependent region is terminated.

**Problem determination:** 4, 35

**Source:** IMS abends

**Module:** DFSASK00, DFSDASIO

**Related information:**

Chapter 2, "IMS problem determination," on page 13

### 0132

**Explanation:** A nonzero condition code was received following a store clock (STCK) instruction during IDENTIFY.

**Analysis:** This is a pseudoabend issued by module DFSASK00.

**Possible cause:** A hardware or a system control program (SCP) problem can cause this abend.

**System action:** The dependent region is terminated.

**Programmer response:** Refer to *System 370 Principles of Operation (GA22-7000)* for the store clock, STCK, failure return code.

**Problem determination:**

**Source:** IMS abends

## 0134 • 0137

**Module:** DFSASK00

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

### 0134

**Explanation:** Dependent region issued an IDENTIFY request when the control region shutdown was in progress. This is normal when a shutdown is in process.

**Analysis:** This is a pseudoabend issued by module DFSASK00.

**System action:** The dependent region is terminated.

**Problem determination:** 4, 35

**Source:** IMS abends

**Module:** DFSASK00

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

### 0135

**Explanation:** Invalid request code passed to DFSASK00.

**Analysis:** This is a pseudoabend issued by module DFSASK00.

**Possible cause:** This is probably an IMS system error.

**System action:** The dependent region is terminated.

**Problem determination:** 4, 35

**Source:** IMS abends

**Module:** DFSASK00

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

### 0136

**Explanation:** A second IDENTIFY request was issued from the same dependent region.

**Analysis:** This is a pseudoabend issued by module DFSASK00.

**Possible cause:** This is probably an IMS system error.

**System action:** The dependent region is terminated. For a Database Control (DBCTL) subsystem, the Coordinator Control (CCTL) subsystem connection process is terminated.

**Problem determination:** 4, 35

**Source:** IMS abends

**Module:** DFSASK00

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

### 0137

**Explanation:** A Coordinator Control (CCTL) subsystem issued a second IDENTIFY while the cleanup for the previous connection was still in progress. The same address space control block (ASCB) address or the same CCTL ID detected the error. To determine which one detected the error, examine the SSPSCODE field of the SSOB. This field contains either NAME or ASCB.

**System action:** The connection process is terminated.



**System programmer response:** Wait until the cleanup completes for the previous connection before attempting another connection. You can use the /DIS CCTL command to check the CCTL status.

**Problem determination:** 1, 5, 8, 38

**Source:** IMS abends

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

#### 0138

**Explanation:** A Coordinator Control (CCTL) subsystem issued an INIT request. The database resource adapter (DRA) loads a module that is called to notify the DRA about DBCTL events such as DBCTL abending, DBCTL terminating normally (/CHE FREEZE), or /STOP REGION commands for DRA threads. DBCTL was unable to locate the DRA notification module.

**System action:** The CCTL INIT request is rejected.

**Programmer response:** Verify that the library defined in the DRA startup table includes all of the DRA modules.

**Source:** IMS abends

---

#### 0139

**Explanation:** An error return code was received from the SYSEVENT SVC.

**Analysis:** This is a pseudoabend issued by module DFSASK00.

**Possible cause:** This is probably an SCP or hardware problem.

**System action:** The dependent region is terminated.

**Problem determination:** 4, 35

**Source:** IMS abends

**Module:** DFSASK00

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

#### 0140

**Explanation:** An application making a DL/I call was not identified. The IDENTIFY token is not valid.

This abend is also issued if a dependent region is not found in the PST ID table while making a DL/I call.

**Analysis:** This abend is a pseudoabend issued by DFSCPY00. This abend is actually issued by DFSECP10 (for an MPP), or DFSECP20 (for a BMP).

**Possible cause:** This is probably an IMS system error.

**System action:** The dependent region is terminated.

**Problem determination:** 4, 35

**Source:** IMS abends

**Module:** DFSCPY00, DFSECP10, DFSECP20

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

#### 0141

**Explanation:** The signon token was not valid. The requester could not keep track of the token.

**Analysis:** This is a pseudoabend issued by module DFSASK00.

**System action:** The dependent region is terminated.

## 0142 • 0143

**Problem determination:** 4, 35

**Source:** IMS abends

**Module:** DFSASK00

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

### 0142

**Explanation:** The work unit blocks (LCRE, OSWA, DPST, GMQW, XPST, D1WA, EPST, or D2WA) assigned to the dependent region could not be created. The storage for the dependent PST were not available. This abend can also occur when the maximum partition specification table (MAXPST) limit is reached.

**Analysis:** 0142 is a standard abend issued by DFSRRA00. The signon failed for lack of resources or internal error.

**Possible causes:**

1. ECSA storage was not available for one of the work unit blocks.
2. MAXPST is reached.
3. Internal IMS error with BCB IPAGE control block get or release. Module SIGNSAVE+X'58' may contain the control block name (first 3 bytes) plus BCB return code in low order byte.
4. There were not enough BCB blocks in one of the following blocks:

LCRE	OSWA
DPST	GMQW
XPST	D1WA
EPST	D2WA

**System programmer response:** If IMS can still function with acceptable throughput, no action is required.

If MAXPST has been reached as shown through IMS commands DISPLAY ACTIVE, or DISPLAY CCTL, consider increasing MAXPST value, if system resources can be tuned for the increase. MAXPST indicated internally by SCDPSTCT being equal to SCDPSTMX. This change requires an IMS recycle with FORMAT RS option recommended at restart.

If IMS throughput is not acceptable, and CSA is full, CSA utilization must be understood and resolved. This may require system monitor display information analysis or dump analysis.

If IMS throughput is not acceptable, the threads can be reset by a recycle of IMS with FORMAT RS option specified at restart. This clears most internal errors related to control block overlays. Create a memory dump of the IMS control region prior to recycling IMS and report the problem to the IBM Software Support.

**Problem determination:** 4, 35

**Source:** IMS abends

**Module:** DFSRRA00

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

### 0143

**Explanation:** Insufficient virtual storage was available for dependent region initialization.

**Analysis:** This is a pseudoabend issued by module DFSASK00 during SIGNON or CREATE-THREAD processing.

**System action:** The dependent region is terminated.

**Programmer response:** Increase the region (partition) size and resubmit the job.

**Source:** IMS abends

**Module:** DFSASK00

---

---

**0144**

**Explanation:** A dependent region IDENTIFY request failed, probably because the storage request for the IDENTIFY table failed. This abend is issued if a DFSBCB FUNC=GET fails while trying to obtain:

- An asynchronous work element (AWE) or storage for an IDENTIFY table for a dependent region IDENTIFY request. No IDENTIFY table is available.
- An AWE during dependent region SIGNOFF.
- An AWE for a DBCTL resync call.

**Analysis:** This is a pseudoabend issued by module DFSASK00, DFSDASI0, DFSDASS0, and DFSDASR0.

If this abend is issued during an IDENTIFY, reissue the request after some of the regions have terminated. For DBCTL, the SSPSCODE field in the SSOB usually contains a 3-byte failure eye-catcher indicating which block was being obtained and a 1-byte return code from BCB GET.

**System action:** The dependent region is terminated.

**Operator response:** Try to start more dependent regions than specified in the IMS definition, up to a maximum of 255, if the resources are available. When the region attempted the IDENTIFY request, the maximum number of dependent regions defined in the IMS control region were active.

**Programmer response:** Increase the region (partition) size and resubmit the job.

**Problem determination:** 4, 35

**Source:** IMS abends

**Module:** DFSASK00, DFSDASI0, DFSDASS0, DFSDASR0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

**0145**

**Explanation:** The caller of DFSASK00 is not a valid caller. During the processing of a call, either the identify token passed on the call did not match that in the IDT, or the caller was not in the same address space as the identifier.

**Analysis:** This code indicates a pseudoabend detected by DFSASK00 and issued by one of the following modules: DFSRRA00, DFSPCC20, DFSRRC10, or DFSRRC40. 0145 can be issued for one of the following reasons:

- During the processing of a SIGNON call, DFSASK00 determined that the identify token passed on the SIGNON call (SSCNTOKI) did not match the identify token placed in the IDT (IDTIDTKN) during IDENTIFY processing for the dependent region.
- During the processing of a CREATE THREAD call, DFSASK00 determined one of the following:
  - The caller was not in the same address space as the identifier because the ASCB address did not match the address space control block (ASCB) address that was placed in the PST (PSTID) during dependent region SIGNON.
  - The PST associated with the CREATE THREAD token did not match the PST address that was placed in the PST Index Table (KIT) during dependent region SIGNON.
- During the processing of a TERMINATE THREAD call, DFSASK00 determined that the caller was not in the same address space as the identifier. The ASCB address did not match the ASCB address that was placed in the PST (PSTID) during dependent region SIGNON.
- During the processing of a SIGNOFF call, DFSASK00 determined that the caller was not in the same address space as the identifier. The ASCB address did not match the address that was placed in the PST (PSTID) during dependent region SIGNON.
- During the processing of a TERMINATE call, DFSASK00 determined that the identify token passed on the TERMINATE call (SSTMTOKI) did not match the identify token that was placed in the IDT (IDTIDTKN) during IDENTIFY processing for the dependent region.

**System action:** The dependent region is terminated.

**Problem determination:** 4, 35

**Source:** IMS abends

**Module:** DFSASK00

**Related information:**

Chapter 2, "IMS problem determination," on page 13

0147

**Explanation:** A SIGN-OFF request was received while one or more threads were still active. SIGN-OFF issued a TERMINATE THREAD. This can happen when a prior TERMINATE-THREAD aborted and a SIGNOFF is the next request progressing back from the dependent IMS hierarchy.

**Analysis:** This is a pseudoabend issued by module DFSASK00.

**System action:** The dependent region is terminated.

**Problem determination:** 4, 35

**Source:** IMS abends

**Module:** DFSASK00

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

0148

**Explanation:** A nonzero return code was returned in response to an IMODULE GETMAIN or a QSAV request during dependent region termination (SIGN-OFF processing).

**Analysis:** This is a standard abend issued by module DFSASK00 during dependent region termination (SIGNOFF).

**System action:** The IMS dependent region terminates abnormally.

**Problem determination:** 4, 35

**Source:** IMS abends

**Module:** DFSASK00

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

0149

**Explanation:** An ISWITCH to the control region failed. DFSASK00 received a nonzero return code in register 15 from DFSIDSP0 after an ISWITCH TO=CTL request.

**Analysis:** This is a pseudoabend established by DFSASK00.

**Possible cause:** The IMS dispatcher does not permit an ISWITCH TO=CTL if a /STO REG ABDUMP is in progress for that region. If a /STO REG ABDUMP command was not entered, then this is an internal system problem. The IMS dispatcher trace and the pseudoabend dump from the log are required to analyze the problem. Examine the SAPSCNTL flag SAPSISWT to see if an ISWITCH was already in progress.

**System action:** The dependent region is terminated.

**Operator response:** If a /STOP REGION ABDUMP command was not entered for the abended region contact the IMS systems programmer.

**Programmer response:** Determine the problem and correct it. For more details, see the Analysis section above.

**Problem determination:** 4, 35

**Source:** IMS abends

**Module:** DFSASK00

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

---

**0150**

**Explanation:** When either the DBRC region, the DL/I subordinate address space region, or the IMS control region abnormally terminates, the other regions are terminated with abend 0150.

**Analysis:** *For DBRC termination:*

- If the control region abnormally terminates, the DBRC region is notified with a SIGNOFF ABNORMAL request issued by the control region job step ESTAE, DFSFSTM0. DFSRCQM0 processes this request and abends the DBRC region with 0150.
- If DBRC abnormally terminates, the DBRC ESTAE, DFSXRID0, terminates the control region using CALLRTM. The CTL TCB ESTAE, DFSFCST0, or the control region job step ESTAE, DFSFSTM0, converts the system 4095 abend into 0150. CALLRTM is issued to the job step if the CTL TCB has not yet initialized.

*For DL/I subordinate address space termination:*

- If the control region abnormally terminates, the control region issues a CALLRTM to terminate the DL/I subordinate address space. The ESTAE for the DL/I subordinate address space job step TCB, DFSFDLI0 then converts the system 4095 abend into 0150. This CALLRTM can be issued from DFSFSTM0 (control region job step ESTAE), DFSFMOD0 (control region TCB ETXR routines), or DFSABND0 (formatted dump).
- If the DL/I subordinate address space region terminates, a CALLRTM is issued to the control region. The CTL TCB ESTAE, DFSFCST0, or the control region job step ESTAE, DFSFSTM0, then converts the system 4095 abend into 0150. CALLRTM is issued to the job step if the CTL TCB has not yet initialized. This CALLRTM can be issued from DFSSDL20 (DL/I subordinate address space TCB ETXR routines), DFSFDLI0 (DL/I subordinate address space job step ESTAE) or DFSABND0.
- Abend 0150 is issued by DL/I subordinate address space initialization module DFSXDL00 if the control region is in the process of abending while DL/I subordinate address space is initializing.

**System action:** IMS, DBRC, and DL/I subordinate address space terminate abnormally.

**Programmer response:** Provide a memory dump of the DBRC, DL/I subordinate address space, and IMS control regions.

**Source:** IMS abends

**Module:** DFSFSTM0, DFSRCQM0, DFSXDL00, DFSFDLI0, DFSFSCT0



---

## Chapter 8. IMS abend codes 0151 - 0200

This topic identifies and explains the following abend codes issued by IMS. Each code is 4 decimal digits.

---

### 0151

**Explanation:** The //DFSCTL data set has invalid data set attributes. Message DFS1071I is issued along with this abend.

**Analysis:** This is a standard abend issued by DFSSBI00. Register 3 points to the data control block (DCB) that describes the //DFSCTL data set.

Key	Label	Description
Reg3=address of DCB	BADDCB	After opening the DCB for the //DFSCTL data set, DFSSBI00 verifies that the record format is fixed-length and the record length is 80. If it is not, DFSSBI00 issues this abend.

**System action:** The application program terminates abnormally.

**Programmer response:** Correct the data set attributes based on information in message DFS1071I.

**Source:** IMS abends

**Module:** DFSSBI00

**Related information:**

 DFS1071I

---

### 0152

**Explanation:** The Sequential Buffering (SB) COMPARE option detected that the buffer content that the SB buffer handler wanted to return to the OSAM buffer handler did not match the content of the block as stored on DASD. Message DFS1070I is issued along with this abend.

**Analysis:** 0152 is a pseudoabend detected by the COMPARE subroutine of module DFSSBCR0. After detecting the mismatch, DFSSBCR0 calls module DFSSBSN0 to create a SNAP memory dump containing diagnostic information, and then requests the abend.

For IMS systems, SNAP records are written to the IMS log as X'67EF' log records. For CICS systems, SNAP records are written to the CICS system log, Journal 01, which can be on tape or disk. For batch regions, SNAP output is written (depending on user-provided specifications on a SNAPDEST control statement in the //DFSCTL data set) either to the IMS log or to a data set chosen by the user. For information about how to print both IMS and CICS log records, see File Select and Formatting Print utility (DFSERA10) (System Utilities). CICS also provides a utility (DFHJUP) to print SNAP log records.

The SNAP output consists of the following items:

- A summary (the PSB name, the DBD name, and the ddname, for example)
- The IBFPRF buffer control block of the OSAM buffer handler
- The displacement within the buffer of the first nonmatching byte
- The block as stored on DASD
- The buffer that the SB buffer handler tried to return to the OSAM buffer handler
- DL/I control blocks and buffer pools.

**System action:** The application program terminates abnormally.

**Programmer response:** See message DFS1070I for the appropriate response.

**Source:** IMS abends

## 0153 • 0155

**Module:** DFSSBCR0

**Related information:**

 [DFS1070I \(Messages and Codes\)](#)

---

### 0153

**Explanation:** IMS was unable to acquire storage for an area, or load an IMS system module. IMS issues message DFS1075A, DFS1076A, or DFS2347A before this abend to indicate the cause of the problem.

**System action:** IMS terminates abnormally.

**Programmer response:** Correct the error based on information in message DFS1075A, DFS1076A, or DFS2347A.

**Source:** IMS abends

**Related information:**

 [DFS1075A \(Messages and Codes\)](#)

 [DFS1076A \(Messages and Codes\)](#)

 [DFS2347A \(Messages and Codes\)](#)

---

### 0154

**Explanation:** An unexpected interface error occurred between the OSAM access method and the Sequential Buffering (SB) buffer handler. An interface error between the OSAM access method and DFSSBIO0 prevented the OSAM access method from starting I/O. A SNAP dump is written to the IMS log or to a user-designated data set.

**Analysis:** 0154 is a standard abend issued by DFSSBIO0. Before issuing this abend, DFSSBIO0 calls DFSSBSN0 to create a SNAP dump containing diagnostic information. SNAP output is written (depending on user-provided specifications on a SNAPDEST control statement in the //DFSCCTL data set) either to the IMS log as an X'67EF' log record, or to a data set chosen by the user.

The SNAP output consists of the following items:

- A 'Summary' containing the text INTERFACE ERROR BETWEEN DFSSBIO0-MODULE AND OSAM
- DL/I control blocks and buffer pools.

Key	Label	Description
Reg5=address of the DECB used by DFSSBIO0 in the RWOS macro call to OSAM Reg10=address of the PST	CHECKS80	The CHECKS subroutine of DFSSBIO0 issues an ICHKOS macro to check the completion of a previously issued RWOS TYPE=READMULT OSAM Read I/O operation. A nonzero return code in register 15 signals a problem, and the CHECKS subroutine tests the DECBNOIO bit. If the DECBNOIO bit is on, an interface error occurred. The CHECKS subroutine saves the registers in PSTSAVL, calls DFSSBSN0 to create a SNAP dump, and issues an abend (batch regions only).

**System action:** The application program terminates abnormally.

**Programmer response:** See Analysis above for detailed diagnostic information.

**Source:** IMS abends

**Module:** DFSSBIO0

---

### 0155

**Explanation:** The Sequential Buffering (SB) test program (DFSSBHD0) detected an error during execution.

Before terminating abnormally, the SB test program issues a message describing the problem, and places a reason code in the low-order byte of register 15.

**Analysis:** 0155 is a standard abend issued by DFSSBHD0. If the SB test program detects a problem, IMS places a reason code in the low-order byte of register 15 and issues an error message. Each reason code, explained in the table



below, is associated with an error message that identifies the type of error. See the message that corresponds to the reason code in the following table:

Key	Label	Description
Reg15=X'00' Message DFS1080A	TCBFSA	DFSSBHD0 verifies that it is running in a batch region by testing fields in PXPARMs. If not, DFSSBHD0 issues an error message and an abend.
Reg15=X'01' Message DFS1081A	BATCH	DFSSBHD0 tests fields in the PST to verify that SB has been properly initialized. If not, DFSSBHD0 issues an error message and an abend.
Reg15=X'02' Message DFS1082A		DFSSBHD0 determines if the SYSPRINT data set was successfully opened. If not, DFSSBHD0 issues an error message and an abend.
Reg15=X'03' Message DFS1083A		DFSSBHD0 determines if the SYSIN data set was successfully opened. If not, DFSSBHD0 issues an error message and an abend.
Reg15=X'04' Message DFS1084A		DFSSBHD0 determines if the SYSUT1 data set was successfully opened. If not, DFSSBHD0 issues an error message and an abend.
Reg15=X'05' Message DFS1085A		DFSSBHD0 determines if the record format of the SYSUT1 data set is variable. If not, DFSSBHD0 issues an error message and an abend.
Reg2=A(PSB) Reg15=X'06' Message DFS1086A	GETUTS30	DFSSBHD0 determines if the PSB has at least one database PCB. If not, DFSSBHD0 issues an error message and an abend.
Reg3=A(image capture log record) Reg15=X'07' Message DFS1087A		DFSSBHD0 determines if the image capture log record contains a valid PCB number. If not, DFSSBHD0 issues an error message and an abend.
Reg3=A(image capture log record) Reg15=X'08' Message DFS1088A	FINDPCB5	DFSSBHD0 determines if the image capture log record contains a valid DBD name. If not, DFSSBHD0 issues an error message and an abend.
Reg3=A(image capture log record) Reg15=X'09' Message DFS1089A		DFSSBHD0 determines if the image capture log record contains a valid relative data set group control block (DSG) number. If not, DFSSBHD0 issues an error message and an abend.
Reg2=A(DSG) Reg3=A(image capture log record) Reg15=X'0A' Message DFS1090A		DFSSBHD0 determines if the data set organization of a database data set has changed to VSAM. If so, DFSSBHD0 issues an error message and an abend.
Reg3=A(image capture log record) Reg5=A(DMB) Reg15=X'0C' Message DFS1092A		DFSSBHD0 checks the ddname stored in the image capture log record. If it is not valid, DFSSBHD0 issues an error message and an abend.

Key	Label	Description
Reg3=A(image capture log record) Reg15=X'0D' Message DFS1093A		DFSSBHD0 checks the subrecord type in the image capture log record. If it is not valid, DFSSBHD0 issues an error message and an abend.
Reg15=X'0F' Message DFS1095A	EODLOG	DFSSBHD0 cannot find the image capture log record (in the //SYSUT1 data set) that describes the start of the application execution.
Reg3=A(image capture log record) Reg15=X'13' Message DFS1099A		DFSSBHD0 checks the ddname stored in image capture log records. If it is not valid, DFSSBHD0 issues an error message and an abend.
Reg15=X'14' Message DFS2343A	EODSYSIN	DFSSBHD0 checks the test program control statements provided in the //SYSIN data set. If one or more control statements are not valid, DFSSBHD0 issues an error message and an abend.

**System action:** IMS terminates abnormally.

**Programmer response:** See the message that corresponds to the reason code for the appropriate response.

**Source:** IMS abends

**Module:** DFSSBHD0

#### 0156

**Explanation:** An application program made a DL/I call. The caller was in supervisor state.

**Analysis:** This is a pseudoabend set up by DFSCPY00. This abend is actually issued by DFSECP10 (for an MPP), or DFSECP20 (for a BMP).

**Possible cause:** An error occurred in the application program.

**System action:** The dependent region is terminated.

**Problem determination:** 4, 35

**Source:** IMS abends

**Module:** DFSCPY00

**Related information:**

Chapter 2, "IMS problem determination," on page 13

#### 0157

**Explanation:** An application program made a DL/I call. The caller was in a different key from the key in the task control block (TCB).

**Analysis:** This is a pseudoabend set up by DFSCPY00. This abend is actually issued by DFSECP10 (for an MPP), or DFSECP20 (for a BMP).

**Possible cause:** An application program error may have been caused by setting the storage protect key.

**System action:** The dependent region is terminated.

**Problem determination:** 4, 35

**Source:** IMS abends

**Module:** DFSCPY00

**Related information:**

Chapter 2, "IMS problem determination," on page 13

## 0158

**Explanation:** The GSAM initialization module, DFSZDI20, encountered a problem executing the type 13 RDJFCB macro to obtain DD statement information for the GSAM data sets.

**Analysis:** The following table shows the register information and explanations:

Key	Label	Description
Reg14=X'01'	AB15801	Register 15 contains a nonzero return code from the RDJFCB macro. This could indicate that the DD statement is not present in the step. However, a prior DEVTYPE macro indicated that the DD statement is present.
Reg14=X'02'	AB15802	On return from the RDJFCB macro, the ARLAREA, the address of the ARA area, obtained by RDJFCB is 0. In this case, ARLCODE is probably 8. For a description of ARLCODE=8, see Reg14=X'03'.
Reg14=X'03'	AB15803	On return from the RDJFCB macro, ARLCODE at R5 + X'1C' is nonzero. If ARLCODE=4, the ARL was not properly initialized prior to issuing the RDJFCB macro. If ARLCODE=8, there was not enough virtual memory to return the ARA area. Increase the maximum virtual memory available for the region and execute the step again. The ARA area is equal to approximately 200 bytes times the number of data sets included in the DD statement.

**System action:** The IMS batch or BMP region terminates abnormally.

**Programmer response:** Additional virtual memory is needed to execute the RDJFCB macro. Ensure that the size of the region is adequate.

**Source:** IMS abends

**Module:** DFSZDI20

**Related reference:**

 z/OS: RDJFCB macro specification

## 0160

**Explanation:** An application that was not executing in an IMS-dependent region issued a DL/I call, an SRRCMIT call, or an SRRBACK call. One of the following occurred:

- The application issued a DL/I program call, which was processed by module DBFIRC10, DFSCPY00, or DFSCPY50.
- The application issued an SRRCMIT or an SRRBACK call, which was processed by DFSCPY70.

One of the following failed a validity check in DFSCPY00, DFSCPY50, or DFSCPY70:

PST  
DIRCA  
Signon token  
Address space control block (ASCB)  
Thread token

**Analysis:** This code indicates a pseudoabend when issued by modules DFSCPY00, DFSCPY50, and DBFIRC10. It is a standard abend when issued by module DFSCPY70.

DBFIRC10 issues this pseudoabend if the current TCB is not an IMS-dependent region TCB.

If this abend was issued by DFSCPY70, register 15 at the time of abend contains one of the following character strings in EBCDIC:

PST (X'D7E2E340')  
DIRC (X'C4C9D9C3')  
VTDE (X'E5E3C4C5')  
ASCB (X'C1E2C3C2')  
THRD (X'E3C8D9C4')

**System action:** The application is terminated.

## 0166

**Source:** IMS abends

**Module:** DFSCPY00, DFSCPY50, DFSCPY70, DBFIRC10

---

### 0166

**Explanation:** IMS security options specified RACF to be used for signon verification and command or transaction authorization or for RAS authorization, but IMS RACF security initialization either failed or is not present in the system.

**Analysis:** 0166 is a standard abend issued by module DFSIRAC0.

A return code in register 15 at the time of abend identifies the cause of RACF initialization failure. Register 6 contains the value returned by RACF in register 0. If the error occurred on the RACF RACLIST call, register 5 contains a value that indicates which class failed:

#### Codes

##### Explanation

- 4 CIMS
- 5 TIMS
- 6 IIMS
- 7 LIMS
- | 8 PIMS
- | 9 AIMS

#### Reg 15

##### Explanation

X'04'

Unable to perform the requested function

X'08'

Class specified not defined to RACF

X'0C'

RACLIST processing error

X'10'

RACF not active or class not active

X'14'

RACLIST installation exit routine error

X'18'

Parameter list error

X'1C'

RACF not installed or insufficient level of RACF

**System action:** IMS terminates abnormally.

**Operator response:** Activate RACF security for this IMS system.

**Programmer response:** A code in register 15 at the time of ABEND identifies the cause of the failure in RACF initialization. If an error occurred on the RACF RACLIST call, register 5 will contain a value to indicate which class failed (4 - CIMS, 5 - TIMS, 6 - IIMS, and 7 - LIMS).

If RAS security is activated, either by a RAS definition in the security macro or by ISIS= A, C or R in the DFSPBxx member, and class IIMS, LIMS or TIMS are not active, IMS terminates abnormally with abend 0166 with RC=4 in reg 15 and RC=5, 6 or 7 in reg 5.

**Source:** IMS abends

**Module:** DFSIRAC0

---

## 0168

**Explanation:** A problem was encountered during restart. One of the following errors occurred:

- The checkpointed blocks do not match the loaded blocks. This condition occurs if your IMS control block structure has been changed since the specified checkpoint was taken.
- A checkpointed communication name table (CNT) was not found on a restart. This condition is the result of an IMS code failure.

**Analysis:** For DBFERST0, DBFNRST0:

The checkpointed blocks do not match the loaded blocks. This condition occurs if the IMS block structure has changed since the specified checkpoint was taken.

0168 is a standard abend issued from the Fast Path restart module DBFNRST0 and from modules called by the Fast Path emergency restart module DBFERST0. The program status word (PSW) at entry-to-abend points to the instruction within the module from which the abend (SVC 13) is issued.

Register 12 in the abend SVRB registers is the base register. Register 2 in the abend SVRB registers contains an EBCDIC code indicating the type of error that occurred. Register 9 contains the SCD address. The field SCDCWRK in the SCD contains a pointer to the log record being processed when the abend occurred. See the list of log records in *IMS Version 13 Diagnosis* to see the valid Fast Path log records (type X'59').

Key	Label	Description
Reg2=C'ADSC'	NRST4087	The checkpointed area is not defined in the restarting system.
Reg2=C'ADSC'	EROC5921	A data entry database (DEDB) area data set (ADS) open log record was read, but the DEDB ADSC named in the log record existed in the system (field ADSCDDN).
Reg2=C'ADSC'	EROC5922	A DEDB ADS close log record was read, but the area data set control block (ADSC) named in the log record does not exist in the restarting system.
Reg2=C'ADSC'	EROC5923	A DEDB ADS status log record was read, but the ADSC named in the log record does not exist in the restarting system.
Reg2=C'ADSC'	EROC5924	ADSC for checkpointed error queue element (EQE) does not exist.
Reg2=C'AVAQ'	NRST4086	Normal restart was unable to obtain a buffer to rebuild an in-doubt control interval (CI).
Reg2=C'NRSR'	NRST4080	An RSR tracker has received a checkpointed log record written by an active IMS system.
Reg2=C'NACT'	NRST4080	An active IMS system has received a checkpointed log record written by an RSR tracker.
Reg2=C'BMMM'	PROC4080	Buffer manager mismatch occurred. The buffer manager of the tracking system is different from the buffer manager of the active system as defined by FPBP64=Y N and DBBF, DBFX, BSIZ. The buffer manager of the tracking system must be the same as the buffer manager of the active system.
Reg2=C'DBFR'	NRST4086	A buffer containing an in-doubt CI or the current sequential dependent segment (SDEP) buffer was logged at checkpoint; the buffer was too large for a single log record. Moreover, the multiple log records were created incorrectly and cannot be processed by restart.
Reg2=C'DMAC'	NRST4084	The checkpointed DEDB name or the checkpointed area name cannot be found in the restarting system.
Reg2=C'DMAC'	NRST4088	A checkpointed IEEQE contains in-doubt data for an area that is not defined on the restarting system.
Reg2=C'DMCB'	various	The DEDB name in a log record cannot be found in the system DEDB master control block (DMCB) list.

Key	Label	Description
Reg2='C'DMHR'	NRST4080	Fast Path buffer definition (number of buffers, buffer size) on the restarting system does not match checkpointed information. For 31-bit Fast Path systems (FPBP64=N), DBBF values must be the same on both FDBR and the active IMS.
Reg2='C'IEQE'	NRST4086	During warm start, a checkpoint record for an in-doubt CI was seen, but the checkpoint record for the corresponding IEEQE was not seen.
Reg2='C'IEQE'	NRST4088	A log record containing the continuation of a checkpointed IEEQE was seen, but the initial log record was not seen.
Reg2='C'MSDB'	NRST4089	An error occurred in DBFDBDL0 while trying to load the main storage databases (MSDBs) from the image copy data sets, MSDBCP1/2.
Reg2='C'MSDB'	EROCS920	An error occurred in DBFDBDL0 while trying to load the MSDBs from the image copy data sets, MSDBCP1/2.
Reg2='C'RCTE'	NRST4083	The number of checkpointed RCTEs is not equal to the number of routing code table entries (RCTEs) in the extended system contents directory (ESCD) (field ESCDNRCE). Alternately, corresponding RCTEs do not have the same routing code (field RCTEcode).

#### For DBFNDC00:

A FIND request that was done for a Communication Name Table (CNT) name that is contained in a checkpoint record was unsuccessful.

0168 is a standard abend issued from the Fast Path restart FIND DC control blocks module DBFNDC00. The program status word (PSW) at entry-to-abend points to the instruction within the module from which the abend (SVC 13) is issued.

Register 12, in the abend SVRB registers, is the base register. Register 2, in the abend SVRB registers, contains an EBCDIC code indicating the type of error that occurred. Register 9 contains the SCD address. The field SCDCWRK in the SCD contains a pointer to the log record being processed when the abend occurred. A list of valid Fast Path log records appears in the IMS diagnosis information.

Key	Label	Description
Reg2='C'CNTE'	NRST4082	DBFNDC00 was unable to locate the CNT. R3 and R4 contain the CNT name.
Reg2='C'CNTE'	NRST4085	DBFNDC00 was unable to locate the CNT. R3 and R4 contain the CNT name.
Reg2='C'CNTE'	ASOC4089	DBFNDC00 was unable to locate the CNT. R3 and R4 contain the CNT name.
Reg2='C'CNTE'	ERMG5903	DBFNDC00 was unable to locate the CNT. R3 and R4 contain the CNT name.
Reg2='C'CNTE'	ERMG5911	DBFNDC00 was unable to locate the CNT. R3 and R4 contain the CNT name.
Reg2='C'CNTE'	ERMG5936	DBFNDC00 was unable to locate the CNT. R3 and R4 contain the CNT name.

#### For DBFTORS0:

A FINDDMAC call failed to locate the corresponding DMAC address for a DEDB area in the log record. This condition occurs if the corresponding DMAC address of a DEDB area from a log record is not in the restarted system.

0168 is a standard abend issued from the I/O toleration Fast Path restart module DBFTORS0 that is called by the I/O toleration restart module DFSTORS0. The program status word (PSW) at entry-to-abend points to the instruction within the module from which the abend (SVC 13) is issued.

**For DFSFDR10:**

Fast Path log records detected by an FDBR region which does not contain Fast Path.

0168 is a standard abend issued from the FDBR module DFSFDR10. This FDBR region does not include Fast Path, but the log tapes being processed contained a checkpoint that contains Fast Path checkpoint log records. Either this region was processing the incorrect log stream, or this FDBR region must specify FP=Y to include Fast Path.

**For DFSRLP00:**

The checkpointed blocks do not match the loaded blocks. This condition occurs if the user has changed the IMS block structure since the specified checkpoint was taken.

0168 is a standard abend issued from module DFSRLP00, the restart log processor. The program status word (PSW) at entry-to-abend points to the instruction from which the abend (SVC 13) is issued. Register 12 in the abend SVRB registers is the base register.

This abend is issued when an inconsistency is found between the checkpointed blocks and the loaded blocks. Register 15 in the abend SVRB registers contains a code indicating which block is inconsistent.

Error codes 1 and 2 are detected in module DFSRDBP0. Error code 3 is detected in DFSRLP00. Error codes X'6' through X'11' are detected in DFSCRSP0 and DFSCRPB0.

**Codes**

**Explanation**

**X'01'**

The count of checkpointed DDIRs is not equal to the loaded count.

**X'02'**

The count of checkpointed PDIRs is not equal to the loaded count.

**X'03'**

Fast Path checkpoint records were encountered, but Fast Path is not included in the system.

**X'04'**

An error occurred while processing an X'22' log record. Either storage could not be obtained to process the log record, or a subsequent segment of an X'22' log record was received before the first segment.

**X'06'**

The count of checkpointed CNTs is not equal to the loaded count, or the length of the checkpointed CNT is not equal to the loaded CNT.

**X'07'**

The CNT names in the checkpointed record do not match the loaded CNT names.

**X'08'**

The count of checkpointed SMBs is not equal to the loaded count; or the length of the checkpointed SMB is not equal to the loaded SMB; or the SMB is not in order; or the module failed to get a new SMB.

**X'09'**

The SMB names in the checkpointed record do not match the loaded SMB names.

**X'0A'**

The count of checkpointed communication terminal blocks (CTBs) is not equal to the loaded count. Alternately, the length of the checkpointed CTB is not equal to the loaded CTB.

**X'0B'**

The X'0B' error code can indicate the following problems:

- The count of checkpointed CLBs is not equal to the loaded count
- The length of the checkpointed CLB is not equal to the loaded CLB

## 0168

### X'0C'

The length of the checkpointed conversation control block (CCB) is not equal to the length of the loaded CCB.

### X'0D'

The count of the checkpointed CCBs is not equal to the loaded count.

### X'0E'

The X'0E' error code can indicate the following problems:

- The count of the checkpointed link control blocks (LCBs) is not equal to the loaded count
- The length of the checkpointed LCB is not equal to the loaded LCB.

### X'0F'

The count of the checkpointed CRBs is not equal to the loaded count, or the length of the checkpointed CRB is not equal to the loaded communication restart block (CRB).

### X'10'

The count of the checkpointed subpool queue blocks (SPQBs) is not equal to the loaded count. Alternately, the length of the checkpointed SPQB is not equal to the loaded SPQB.

### X'11'

The SPQB names in the checkpointed record do not match the loaded SPQB names.

### X'12'

The count of the checkpointed VTAM terminal control block (VTCB) is not equal to the loaded count. Alternately, the length of the checkpointed VTCB is not equal to the loaded VTCB; or the VTCB names in the checkpointed records do not match the loaded VTCB names; or the SPQB name checkpointed within a VTCB is not found in the loaded SPQB names.

### X'13'

DFSBCB could not obtain a UOWE for X'4040' log record processing.

### X'14'

Neither the IMS conversation bitmap nor the CCB block could be obtained for X'4024' (TIB) log record processing.

### X'15'

Neither the IMS conversation bitmap nor the CCB block could be obtained for X'4034' (YTIB) log record processing.

### X'16'

HIOP storage was unavailable during SID table rebuild following X'4003' log record processing.

### X'18'

This code is issued while trying to recover a VTCB from an X'4021' log record.

### X'19'

A mismatch between IMS and the checkpoint from which IMS is being restarted was detected. Either IMS was restarted with FP=Y and the checkpoint records used for the restart did not contain Fast Path resources, or FP=N was specified on the restart and the checkpoint records used for the restart contained Fast Path resources. If FP=Y is specified, IMS must be restarted from a checkpoint containing Fast Path. If FP=N is specified, IMS must be restarted from a checkpoint without Fast Path.

### X'1A'

You tried to change the MODBLKS=OLC|DYN parameter across a restart. This parameter can only be changed across a cold start.

### X'1B'

An attempt was made to change the FPBP64=Y|N parameter across a restart. This parameter can only be changed across a cold start.

### X'1C'

One of the following conditions has occurred:

- An attempt was made to disable the IMSRSC repository across a warm start. A warm start allows a change in which the IMSRSC repository is enabled, but fails when an attempt is made to disable it. To disable the IMSRSC repository, a cold start of IMS must be performed.
- An attempt was made to enable or disable the IMSRSC repository across an emergency restart. Enabling or disabling the IMSRSC repository across an emergency restart is not allowed.



To enable the repository when it was not previously enabled, perform an emergency restart (/ERE) with the IMSRSC repository disabled, and use the UPDATE IMS command to enable it after the restart completes.

To disable the IMSRSC repository when it was previously enabled, a cold start of IMS must be performed.

- In an XRF environment, the IMSRSC repository must be enabled on the IMS active and IMS alternate systems or it must be disabled on them both. RC=X'1C' is returned when the IMSRSC repository is enabled on one of the IMS systems and not on the other. To correct this situation, modify the DFSDFxxx member at the IMS active or the IMS alternate system to enable or disable the IMSRSC repository consistently on both systems.

Make this decision based on how the active IMS system is being restarted and whether the IMSRSC repository had been enabled before the restart.

**System action:** IMS terminates abnormally.

**Programmer response:** If the error is because checkpointed blocks do not match the loaded blocks, re-IPL IMS and either enter a checkpoint that reflects the current block structure, or cold start the system.

If the error is because a checkpointed CNT could not be found on a restart or if you encounter a reason code of X'04', contact IBM Software Support.

**Problem determination:** 4, 5, 10, 35, 36

**Source:** IMS abends

**Module:** DFSRCP00, DBFERST0, DBFNDC00, DBFNRS0, DFSFDR10, DFSRLP00, DBFTORS0

**Related reference:**

 Log records (Diagnosis)

**Related information:**

Chapter 2, "IMS problem determination," on page 13

## 0172

**Explanation:** There is insufficient space in the work area pool (WKAP) to satisfy restart work space requirements.

**Analysis:** 0172 is a standard abend issued from DFSRST00. The program status word (PSW) at entry-to-abend and the Reg14 BAL in the abend SVRB should be used to isolate to the label. Register 11 is the base register.

The ICREATE routine is called from many different points within DFSRST00. Register 10 is the BAL register and should be used to determine who the caller is.

Key	Label	Description
Reg2=POOL ID Reg3=size of the pool request Reg9=SCD Reg14=BAL (to ICREATE)	ICREATE	A variable pool request returned with a nonzero return code in register 15. An abend is issued.

**System action:** IMS terminates abnormally.

**Programmer response:** Increase the size of the IMS work area pool (WKAP), and perform another restart.

**Source:** IMS abends

**Module:** DFSRST00

## 0175

**Explanation:** Unrecoverable errors have been encountered while processing the input log during restart. Refer to messages DFS0739A and DFS0739X, which you might receive before this abend.

This abend is related to abend 0176.

**Analysis: For DFSFDLY0:**

This abend is issued in module DFSFDLY0 in response to errors detected by the following modules:

## 0175

### DFSFDLN0

Allocate an OLDS or SLDS

### DFSFDLQ0

Subroutines used by DFSFDLQ0

### DFSFDLP0

SLDS read

### DFSFDLQ0

STATE routines used by DFSFDLX0

### DFSFDLR0

Restart-read driver

### DFSFDLT0

Format WADS

### DFSFDLU0

Terminate OLDS from WADS

### DFSFDLV0

OLDS read STATE transition table used by DFSFDLX0 contains no executable code, yet this is the heart of the OLDS read logic.

### DFSFDLX0

OLDS read driver

### DFSFDLY0

Subroutines used by all restart-read

### DFSFDLZ0

Used in emergency restart to switch between OLDS read and SLDS read

During restart, either the input log data set encountered an unrecoverable error or log data set processing encountered a *should-never-occur* logic error. Message DFS0739I or DFS0739X is issued.

0175 is a standard abends that can be issued from the modules listed for this abend. The program status word (PSW) at entry-to-abend points to the instruction from which ABEND (SVC 13) is issued.

Regardless of which of the two modules actually encounters an error, all 0175 abends are accomplished by calling the common ABEND subroutine in csect DFSFDLY2 in module DFSFDLY0. Register 14 contains the return address of the caller, which should be very close to the actual point of failure. The low halfword of register 15 contains a return code that identifies the reason for the abend.

If a DBRC error is involved, the high halfword of register 15 contains the DBRC return code.

If a DFSMDA error is involved, the high halfword of register 15 contains the DFSMDA return code. (Equate statements defining the abend return codes can be found in macro DFSFRLWA.) The remaining abend registers have been saved at RLWAREGS in the restart-read work area, RLWORK. RLWORK is located using register 9 (or using LRESTWK in the LCD) and is mapped by macro DFSFRLWA.

If working from a formatted dump, the registers at entry to DFSFDLY2 will be close, but not necessarily identical, to the registers stored at RLWAREGS.

The following abend registers have the same meanings for all occurrences of 0175:

- Reg8** Second base register for module (if needed)
- Reg9** RLWORK - restart-read work area
- Reg10** LCD
- Reg11** SCD
- Reg12** First base register for module
- Reg13** IMS pre-chained save area
- Reg14** BALR return address - where the error actually occurred
- Reg15** Return codes

**RLWORK** is the key data area for all the logger restart-read modules and is mapped by DSECT macro DFSFRLWA. The following fields within RLWORK are especially useful:

### RLWQDECB

A copy of the parameter list passed to restart-read

**RLWOLTFE**  
 Pointer to the first entry in a table of OLDS DECB and buffer addresses. These entries are mapped by DSECT OLAT within DFSFRLWA.

**RLWGLOBL**  
 Restart-read global status flags

**RLWAREQ**  
 Last log allocation request (type) passed to DBRC

**RLWALLOC**  
 Type of log allocated

**RLWDBRET**  
 DBRC work area address

**RLWDSETE**  
 DSET entry for OLDS currently being read

**RLWEN**  
 Allocation information for log currently being read

**RLWESTAT**  
 Log status flags

**RLWEEXCP**  
 Log exception flags

**RLWEAVL**  
 Log availability flags

**RLWEOPN**  
 Log open flags

**RLWEMODE**  
 Log mode flags

**RLWEOFOK**  
 Log end-of-file (EOF) written flags

**RLWECNT**  
 Number of good blocks read on current log

**RLWOSUFF**  
 OLDS block suffix for last good block read. The OLDS suffix is mapped by DSECT SUFFIX within DFSFRLWA.

**RLWPCNT**  
 Relative block count for POINT

**RLWOPDCB**  
 OLDS primary DCB

**RLWOSDCB**  
 OLDS secondary DCB

**RLWSDCB**  
 SLDS DCB

**RLWWDCB**  
 WADS DCB

**RLWAREGS**  
 Copy of caller's registers at abend

**RLWTSUFF**  
 OLDS block suffix for first block on current OLDS. The OLDS suffix is mapped by DSECT SUFFIX within DFSFRLWA.

**RLWDSUFF**  
 OLDS block suffix for last duplicate block read. The OLDS suffix is mapped by DSECT SUFFIX within DFSFRLWA.

**RLWTFLG1**  
 OLDS error toleration flags

**RLWXSTAT**  
 OLDS XRF status flags

**RLWXSTT2**  
 OLDS XRF status flags (byte 2)

**RLWHSRRT**  
 OLDS restart-read transition table address. For more information about the OLDS restart-read transition table, please refer to the prolog for module DFSFDLV0. The OLDS restart-read transition table is mapped by DSECTs HSRRTT, XSTATBL, and XSTATENT within DFSFRLWA.

## 0175

### RLWXDOMA

OLDS domain mapping (prior state/event and current state/event)

### RLWPPART

OLDS prior domain partition (within DFSFDLV0)

### RLWCPART

OLDS current domain partition (within DFSFDLV0)

### RLWNPART

OLDS next domain partition (within DFSFDLV0)

### RLWXTRCN

OLDS next transition trace entry address

### RLWXTRCE

OLDS end of transition trace table address

### RLWXTRC

OLDS transition trace table (25 entries - 5 words each). The trace table entries are mapped by DSECT XTRCENT within DFSFRLWA. (If it becomes necessary to analyze the entries in this trace table, please contact the IBM Support Center for assistance.)

**XSXX** OLDS transition STATE equate statements

**XEXX** OLDS transition EVENT equate statements

### XPTINIT

OLDS transition PARTITION equate statements

### RLWWPARM

WADS read parameter list passed to module DFSFDLW0

### RLWABC

ABEND code save area. Equate statements for various restart-read abend codes and return codes follow RLWABC.

### For DFSFDLN0:

Module DFSFDLN0 allocates the log data sets required by the restart-read process. A request is made to DBRC to locate a particular type of log data set. If the data set is located successfully, it is allocated. The following table shows the codes and an explanation for the registers:

Key	Label	Description
Reg15=RLWLLLF RLWDBRC=DBRC return code	LTA1430	DBRC could not find the latest OLDS.
Reg15=RLWAF RLWAREGF=RDJFCB return code	LTA1578	RDJFCB failed while attempting to allocate a new OLDS for output.
Reg15=RLWLLNF	LTA1585	A new OLDS is needed for output to complete the log close process. A reusable OLDS was not found. Either DBRC could not be checked for an available OLDS, or an operator replied "ABEND" to message DFS0739I.
Reg15=RLWDBRCF RLWDBRC=DBRC return code	LTA1590	The DBRC status exit routine returned a nonzero return code.
Reg15=RLWAF RLWMDRC=DFSMDA return code	LTA1620	Allocation failed for a system log data set (SLDS). If DFSMDA failed, RLWMDRC is nonzero. Otherwise, the SLOPN subroutine in DFSFDLY0 failed to open the SLDS.
Reg15=RLWAF RLWMDRC=DFSMDA return code	LTA21200	Allocation failed for an SLDS. If DFSMDA failed, RLWMDRC is nonzero. Otherwise, the SLOPN subroutine in DFSFDLY0 failed to open the SLDS.
Reg15=RLWAF RLWMDRC=DFSMDA return code	LTA22200	Allocation failed for an SLDS. If DFSMDA failed, RLWMDRC is nonzero. Otherwise, the SLOPN subroutine in DFSFDLY0 failed to open the SLDS.

Key	Label	Description
Reg15=RLWAF RLWDBRC=DBRC return code	LTA2220	DBRC is unable to locate the secondary log data set.
Reg15=RLWLGCE	ALOPENOL	Should-never-occur logic error.
Reg15=RLWLGCE	ALOPENSL	Should-never-occur logic error.
Reg15=RLWAF	LTD320	Final checking within the ALOPENOL subroutine determined that OLDS allocation failed.

#### For DFSFDL00:

Module DFSFDL00 contains several subroutines used by the OLDS read transition STATE routines in module DFSFDLQ0.

#### XREAD

Issues a READ for the current OLAT entry and issues a CHECK for the outstanding READ against the next OLAT entry.

#### XREREAD

Resets BSAM DCB status (CHECKs unchecked READs), issues a POINT for the last good block read, issues a READ and CHECK for the last good block, and issues a READ and CHECK for the next block (the block that was in error the last time XREAD was invoked).

#### XBLKCHK

Examines the results of the last CHECK. Tests for EOF, READ errors, and sequence errors. Sets the CURRENT EVENT to indicate the relative success of the last READ/CHECK. Selects the NEXT PARTITION to be used in the restart-read transition table (DFSFDLV0).

The following table shows the codes and an explanation for the registers:

Key	Label	Description
Reg15=RLWLGCE RLWEOPN=open OLDS flags	XRER0020	The CLROLD subroutine in DFSFDLY0 has returned a nonzero return code. It is not possible to reread the current OLDS and an alternate OLDS is not available.
Reg15=RLWLGCE RLWCSTAT=current state RLWTFLG1=error toleration flags	XBLK0016	XBLKCHK is attempting to analyze the last BSAM CHECK. Error toleration is required for STATE XS18 and an unknown condition occurred.
Reg15=RLWLGCE RLWCSTAT=current state RLWTFLG1=error toleration flags	XBLK0026	XBLKCHK is attempting to analyze the last BSAM CHECK. Error toleration is required for STATE XS12 and an unknown condition occurred.
Reg15=RLWLGCE	XBLKEOF	Should-never-occur logic error. XBLKCHK is attempting to analyze the last BSAM CHECK. EOF was detected and the OLDCALC subroutine returned a bad return code while trying to determine where EOF occurred within the data set.
Reg15=RLWLGCE	XBLKREAD	Should-never-occur logic error. XBLKCHK is attempting to analyze the last BSAM CHECK. A read error was detected and the OLDCALC subroutine returned a bad return code while trying to determine where the read error occurred within the data set.
Reg15=RLWLGCE RLWCSTAT=current state	XBLK3500	XBLKCHK is attempting to analyze the last BSAM CHECK. A first block read error was detected and RLWCSTAT is bad.
Reg15=RLWLGCE RLWCSTAT=current state	XBLK4500	XBLKCHK is attempting to analyze the last BSAM CHECK. A middle block read error was detected and RLWCSTAT is bad.
Reg15=RLWLGCE RLWCSTAT=current state	XBLK5500	XBLKCHK is attempting to analyze the last BSAM CHECK. A last block read error was detected and RLWCSTAT is bad.

Key	Label	Description
Reg15=RLWLSE RLWLOGSQ=last good log record sequence number Reg6 at entry to DFSFDLY2 points to the log buffer holding the out-of- sequence log record	XBLK6140	XBLKCHK successfully read the first block in an OLDS but found that the sequence number of the first record in the block is not equal to the previous OLDS/SLDS log record sequence number +1.
Reg15=RLWLGCE RLWCSTAT=current state	XBLK7500	XBLKCHK is attempting to analyze the last BSAM CHECK. An older data sequence error was detected and RLWCSTAT is bad.
Reg15=RLWLGCE LWAREG2=current block suffix address RLWTSUFF=first block of last good OLDS suffix RLWTFLG1=error toleration flags RLWTSKIP=BSN of block being analyzed when the newer data sequence error was detected.	XBLK8075	XBLKCHK is attempting to analyze the last BSAM CHECK. A newer data sequence error was detected because of a skipped OLDS. However, the key fields indicate an unexpected condition.
Reg15=RLWLGCE RLWAREG2=current block address suffix  RLWTSUFF=first block of last good OLDS suffix RLWTFLG1=error toleration flags RLWTSKIP=BSN of block being analyzed when the newer data sequence error was detected.	XBLK8085	XBLKCHK is attempting to analyze the last BSAM CHECK. A newer data sequence error was detected because of a skipped OLDS. The skipped OLDS was located. While performing a forward read of the skipped OLDS, a second newer data sequence error was encountered before returning to the skipped OLDS processing starting point, RLWTSKIP.
Reg15=RLWLGCE RLWCSTAT=current state	XBLK8500	XBLKCHK is attempting to analyze the last BSAM CHECK. A newer data sequence error was detected and RLWCSTAT is bad.

#### For DFSFDLP0:

Module DFSFDLP0 reads system log data sets (SLDS) during warm start or emergency restart. The following table shows the codes and an explanation for the registers:

Key	Label	Description
Reg4=RLWSDECB Reg15=RLWLRE	DFSFDLP0	An I/O error occurred while reading an SLDS.
Reg15=RLWLGCE RLWEMODE	LSA200	An I/O error occurred while reading the secondary SLDS. Recover the SLDSs using the Log Recovery utility and try the restart again.
Reg15=RLWAF RLWALLOC	LSA210	DBRC was unable to locate the secondary SLDS.

Key	Label	Description
Reg15=RLWLSE SAVELSN=last LSN read from secondary SLDS while trying to match up with the primary SLDS. (SAVELSN is located within DFSFDLP0) RLWLOGSQ=last LSN read from primary SLDS.	LSA235	EOF was reached on the secondary SLDS without finding a record to match the last record read from the primary SLDS.
Reg4=RLWSDECB Reg15=RLWLRE	LSA240	An I/O error occurred on the secondary SLDS while attempting to match the primary SLDS.
Reg2=first LSN in block Reg3=RLWLOGSQ+1 Reg15=RLWLSE	LSA250	A sequence error occurred on the secondary SLDS while attempting to match the primary SLDS. The log sequence number (LSN) in register 2 is more than one greater than the last LSN read from the primary SLDS.
Reg15=RLWLSE	LSA21R	A sequence error occurred on the secondary SLDS while attempting to truncate the block that would match up with the primary SLDS.

#### For DFSFDLQ0:

Module DFSFDLQ0 contains all the STATE routines used to read an OLDS. There is an 8-byte eye catcher at the beginning of each STATE routine.

Module DFSFDLX0 reads the STATE table, DFSFDLV0, to determine which STATE to execute next. DFSFDLX0 calls DFSFDLQ0 with the address of the STATE routine in register 0. The STATE routine is executed and control is returned to DFSFDLX0, along with a return code. The following table shows the codes and an explanation for the registers:

Key	Label	Description
Reg15=RLWLRE or RLWDRE RLWEOPN=open data set flags RLWTFLG1=error toleration flags	XS020035	Routine XSTATS02 was called to analyze a read error. The CLROLD subroutine failed to reset BSAM status using CLOSE and OPEN. Since all required OLDS data sets are unavailable, restart-read abends.
Reg15=abend code (variable) RLWXDOMA=last STATE flags RLWCPART=last partition	XSTATS03	The STATE table, DFSFDLV0, determined that a U0176 abend is required.
Reg15=RLWLGCE	XS040100	Should-never-occur logic error. The last LSN in a good block was zero.
Reg15=RLWLGCE RLWDBRC=DBRC return code	XS050300	Should-never-occur logic error. XSTATS05 received an unexpected response from DBRC on a LOCATE LAST request.
Reg15=RLWLGCE RLWDBRC=DBRC return code	XS050700	Should-never-occur logic error. XSTATS05 received an unexpected response from DBRC on a LOCATE NEXT request.

Key	Label	Description
Reg15=RLWLGCE RLWDBRC=DBRC return code	XS070025	Should-never-occur logic error. XSTATS07 received an unexpected response from DFSFDLN0 on an ALLOCATE NEXT request.
Reg15=RLWLGCE	XS160200	Should-never-occur logic error. XSTATS16 received an unexpected return code from DFSFDLU0.
Reg15=abend code (variable) RLWXDOMA=last STATE flags RLWCPART=last partition	XS21EXIT	The STATE table, DFSFDLV0, determined that a U0175 abend is required.
Reg15=RLWLGCE	XS290025	Should-never-occur logic error. XSTATS29 received an unexpected response from DFSFDLN0 on an ALLOCATE PRIOR request.
Reg15=RLWLGCE	XS320025	Should-never-occur logic error. XSTATS29 received an unexpected response from DFSFDLN0 on an ALLOCATE LAST request.
Reg15=abend code (variable) RLWXDOMA=last STATE flags RLWCPART=last partition RLWXTRC=transition trace table	XSTATS99	Should-never-occur logic error. A search of the STATE table, DFSFDLV0, did not produce a match. A U0175 or U0176 abend is forced.

#### For DFSFDLR0:

Module DFSFDLR0 is the driver module for the log restart-read function. DFSFLLG0 branches to DFSFDLR0 under the restart TCB to perform one of the following functions:

**FIND** Locate the most current type X'42' log record (checkpoint ID table) and pass it back to restart.

The OLDS might not be terminated from the WADS during this request.

**OPEN** OPEN the log containing the restart checkpoint and locate the start of the checkpoint.

**READ** READ the log forward, beginning with the restart checkpoint. Pass one block of data at a time back to restart. If necessary, terminate the OLDS from the WADS.

**CLOSE** After EOF on the last log, merge the data set entry tables, DSETs, (initial DSET from log initialization, DSET from the latest X'4301' log record, DSET built during forward log read) and allocate all OLDS.

#### FORMAT

Format the WADS as requested by the /STA WADS, /NRE, or /ERE commands.

The following table shows the codes and an explanation for the registers:

Key	Label	Description
Reg15=RLWLRE	LOA200	The OPEN function encountered a read error while attempting to read the first block from the OLDS containing the restart checkpoint.
Reg15=RLWLNT RLWGLOBL=global flags	LCA10E	The CLOSE function was requested and RLWGLOBL does not indicate that the OLDS was terminated from the WADS.



Key	Label	Description
Reg2=log record address Reg3=log record length Reg15=RLWLSE RLWGLOBL=global flags	LCA21210	The CLOSE function was requested. While searching for a DSET (X'4301' log record) within the block, a zero length record was found.
Reg15=RLWLGCE	LCA2300	Should-never-occur logic error.
Reg2=log record address Reg4=log record length Reg15=RLWLSE RLWGLOBL=global flags	LOB110	The OPEN function was requested. While searching for the beginning checkpoint record (X'4001' log record) within the block, a zero length record was found.
Reg15=RLWLRE	LFB010	The FIND (or CLOSE) function encountered a read error while attempting to read the first block from the latest OLDS.
Reg15=RLWLLLF RLWALLOC=allocation flags	LFB190	The FIND (or CLOSE) function received an unexpected result from DFSFDLN0 on a request to allocate the latest log.
Reg2=log record address Reg4=log record length Reg15=RLWLSE RLWGLOBL=global flags	LFC10R	The FIND (or CLOSE) function was requested. While searching a block for an X'42' (or X'4301') log record, a zero length record was found.

#### For DFSFDLR0:

Module DFSFDLU0 terminates the latest OLDS from the WADS. The following table shows the codes and an explanation for the registers:

Key	Label	Description
Reg15=RLWLLLF RLWDBRC=DBRC return code	LBA100	DFSFDLU0 is trying to verify that restart-read is correctly positioned on the latest OLDS. The locate-latest-OLDS request to DBRC failed.
Reg15=RLWRLGB	LBA106	DFSFDLU0 failed in an attempt to reread the last good block from the data set on which it was originally found.
Reg15=RLWRLGB	LBA110	DFSFDLU0 failed in an attempt to reread the last good block from the data set on which it was originally found.
Reg15=RLWLGCE RLWXSTAT=first XRF status flags byte RLWXST2=second XRF status flags byte	LBA208	The OLDS is being terminated as part of an XRF takeover. RESERVEs of the OLDS and WADS are required, but have not been established.
Reg15=RLWLNT RLWEOPN=open OLDS flags	OLDSCLR	An attempt to clear BSAM status (using CLOSE and OPEN) for the OLDS to terminate failed.
Reg15=RLWLNT RLWEOPN=open OLDS flags	LDX0100	An attempt to clear BSAM status (using CLOSE and OPEN) for the OLDS to terminate failed.

Key	Label	Description
Reg15=RLWLNT RLWEOPN=open OLDS flags	LDX0200	An attempt to clear BSAM status (using CLOSE and OPEN) for the OLDS to terminate failed.

**For DFSFDLX0:**

Module DFSFDLX0 is the driver for the OLDS read process. The following table shows the codes and an explanation for the registers:

Key	Label	Description
Reg2=loop counter Reg3=next entry in DFSFDLV0 address Reg15=RLWLGCE	XTRANEXT	Should-never-occur logic error. DFSFDLX0 is searching transition entries in a partition of the STATE transition table, DFSFDLV0. All entries in the partition were searched without finding a match. All partitions should have as their last entry an entry that matches any condition. This entry should direct processing to STATE XS99, which abends. In addition, all partitions have a count of the number of entries in the partition. This count is used to initialize register 2.
Reg15=RLWLGCE	XTRA3700	Should-never-occur logic error. An unexpected return code was returned by DFSFDLQ0.

**For DFSFDLY0:**

Module DFSFDLY0 contains all the subroutines commonly used by the restart-read modules. The following table shows the codes and an explanation for the registers:

Key	Label	Description
Reg15=RLWTFCF RLWAREGF=return code from TRKCALC	ANALEOF	The ANALEOF subroutine received a nonzero return code from TRKCALC.
Reg15=RLWLGCE RLWEOPN=open flags for OLDS	LBN120	Should-never-occur logic error. The OLSETM subroutine found no open OLDS.
Reg2=log record address Reg3=end of buffer address Reg4=log record length Reg15=RLWLSE	LLC10R	The RCDSCH subroutine encountered a log record with an invalid length.
Reg15=RLWTFCF	LDG200	The OLDSCALC subroutine received a nonzero return code from TRKCALC.
Reg15=RLWTFCF	OLPNT	The OLPNT subroutine received a nonzero return code from TRKCALC.
Reg15=RLWDBRCF RLWDBRC=DBRC return code RLWAREQ=request passed to DBRC	LTH150	The DBRCEXEC subroutine received an unexpected return code from DBRC.
Reg5=DCB Reg15=RLWLFCF RLWEEXCP=exception flags	LSC105	The SLCLS subroutine received an error attempting to CLOSE an SLDS.
Reg15=RLWLGCE	LBP130	Should-never-occur logic error.

Key	Label	Description
Reg2=temporary DSET entry Reg15=RLWLGCE RLWDSETE=DSET entry	LBP130	Should-never-occur logic error. Either RLWDSETE or the temporary DSET is bad.

**For DFSFDLZ0:**

Module DFSFDLZ0, during warm start or emergency restart, performs either OLDS read or SLDS read. The following table shows the codes and an explanation for the registers:

Key	Label	Description
Reg15=RLWLLFF RLWALLOC=allocation flags	LLA330	EOF occurred on a nonlast log data set. A request to DFSFDLN0 to allocate the next log data set failed.
Reg15=RLWLSE	LLA440	EOF occurred on a nonlast log data set. A sequence error occurred while trying to locate the next record.
Reg15=RLWLSE	LLA500	EOF occurred on a nonlast log data set. A sequence error occurred while trying to locate the next record.
Reg15=RLWLSE	LLA600	EOF occurred on a nonlast log data set. A sequence error occurred while trying to locate the next record.

**System action:** IMS terminates abnormally.

**Problem determination:** 1, 5, 8, 11, 35

Save a copy of the OLDSs and WADSs.

List the DBRC RECON data set.

**Source:** IMS abends

**Module:** DFSFDLY0, DFSFDLN0, DFSFDL00, DFSFDLP0, DFSFDLQ0, DFSFDLX0, DFSFDLZ0

IMS abend 0176

 DFS0739I

 DFS0739X

Chapter 2, "IMS problem determination," on page 13

**0176**

**Explanation:** While processing the input log during restart within an Extended Recovery Facility (XRF) complex, the IMS alternate system encountered errors from which recovery was impossible. Fast Database Recovery also issues this abend when it encounters errors. Refer to messages DFS0739I and DFS0739X, which you might receive before this abend.

This abend is related to abend 0175.

**Analysis: For DFSFDLY0:**

This abend is issued in module DFSFDLY0 in response to errors detected by the following modules:

**DFSFDLN0**

Allocate an OLDS or SLDS

**DFSFDL00**

Subroutines used by DFSFDLQ0

**DFSFDLP0**

SLDS read

**DFSFDLQ0**

STATE routines used by DFSFDLX0

## 0176

### DFSFDLR0

Restart-read driver

### DFSFDLT0

Format WADS

### DFSFDLU0

Terminate OLDS from WADS

### DFSFDLV0

OLDS read STATE transition table used by DFSFDLX0 contains no executable code, yet this is the heart of the OLDS read logic.

### DFSFDLX0

OLDS read driver

### DFSFDLY0

Subroutines used by all restart-read

### DFSFDLZ0

Used in emergency restart to switch between OLDS read and SLDS read

During XRF tracking or takeover, either the input log data set encountered an unrecoverable error or log data set processing encountered a *should-never-occur* logic error. Message DFS0739I or DFS0739X is issued.

0176 are standard abends that can be issued from the modules listed for this abend. The program status word (PSW) at entry-to-abend points to the instruction from which ABEND (SVC 13) is issued.

Regardless of which of the two modules actually encounters an error, all 0175 and 0176 abends are accomplished by calling the common ABEND subroutine in csect DFSFDLY2 in module DFSFDLY0. Register 14 contains the return address of the caller, which should be very close to the actual point of failure. The low halfword of register 15 contains a return code that identifies the reason for the abend.

If a DBRC error is involved, the high halfword of register 15 contains the DBRC return code.

If a DFSMDA error is involved, the high halfword of register 15 contains the DFSMDA return code. (Equate statements defining the abend return codes can be found in macro DFSFRLWA.) The remaining abend registers have been saved at RLWAREGS in the restart-read work area, RLWORK. RLWORK is located using register 9 (or using LRESTWK in the LCD) and is mapped by macro DFSFRLWA.

If working from a formatted dump, the registers at entry to DFSFDLY2 closes, but not necessarily identical, to the registers stored at RLWAREGS.

The following abend registers have the same meanings for all occurrences of 0176:

- Reg8** Second base register for module (if needed)
- Reg9** RLWORK - restart-read work area
- Reg10** LCD
- Reg11** SCD
- Reg12** First base register for module
- Reg13** IMS pre-chained save area
- Reg14** BALR return address - where the error actually occurred
- Reg15** Return codes

RLWORK is the key data area for all the logger restart-read modules and is mapped by DSECT macro DFSFRLWA. The following fields within RLWORK are especially useful:

### RLWQDECB

A copy of the parameter list passed to restart-read

### RLWOLTFE

Pointer to the first entry in a table of OLDS DECB and buffer addresses. These entries are mapped by DSECT OLAT within DFSFRLWA.

### RLWGLOBL

Restart-read global status flags

### RLWAREQ

Last log allocation request (type) passed to DBRC

### RLWALLOC

Type of log allocated

**RLWDBRET**  
DBRC work area address

**RLWDETE**  
DSET entry for OLDS currently being read

**RLWEN**  
Allocation information for log currently being read

**RLWESTAT**  
Log status flags

**RLWEEXCP**  
Log exception flags

**RLWEAVL**  
Log availability flags

**RLWEOPN**  
Log open flags

**RLWEMODE**  
Log mode flags

**RLWEFOK**  
Log end-of-file (EOF) written flags

**RLWECNT**  
Number of good blocks read on current log

**RLWOSUFF**  
OLDS block suffix for last good block read. The OLDS suffix is mapped by DSECT SUFFIX within DFSFRLWA.

**RLWPCNT**  
Relative block count for POINT

**RLWOPDCB**  
OLDS primary DCB

**RLWOSDCB**  
OLDS secondary DCB

**RLWSDCB**  
SLDS DCB

**RLWWDCB**  
WADS DCB

**RLWAREGS**  
Copy of caller's registers at abend

**RLWTSUFF**  
OLDS block suffix for first block on current OLDS. The OLDS suffix is mapped by DSECT SUFFIX within DFSFRLWA.

**RLWDSUFF**  
OLDS block suffix for last duplicate block read. The OLDS suffix is mapped by DSECT SUFFIX within DFSFRLWA.

**RLWTFLG1**  
OLDS error toleration flags

**RLWXSTAT**  
OLDS XRF status flags

**RLWXSTT2**  
OLDS XRF status flags (byte 2)

**RLWHSRRT**  
OLDS restart-read transition table address. For more information about the OLDS restart-read transition table, please refer to the prolog for module DFSFDLV0. The OLDS restart-read transition table is mapped by DSECTs HSRRTT, XSTATBL, and XSTATENT within DFSFRLWA.

**RLWXDOMA**  
OLDS domain mapping (prior state/event and current state/event)

**RLWPPART**  
OLDS prior domain partition (within DFSFDLV0)

**RLWCPART**  
OLDS current domain partition (within DFSFDLV0)

**RLWNPART**  
OLDS next domain partition (within DFSFDLV0)

**RLWXTRCN**  
OLDS next transition trace entry address

## 0176

### RLWXTRCE

OLDS end of transition trace table address

### RLWXTRC

OLDS transition trace table (25 entries - 5 words each). The trace table entries are mapped by DSECT XTRCENT within DFSFRLWA. (If it becomes necessary to analyze the entries in this trace table, please contact the IBM Support Center for assistance.)

**XSXX** OLDS transition STATE equate statements

**XEXX** OLDS transition EVENT equate statements

### XPTINIT

OLDS transition PARTITION equate statements

### RLWWPARM

WADS read parameter list passed to module DFSFDLW0

### RLWABC

ABEND code save area. Equate statements for various restart-read abend codes and return codes follow RLWABC.

### For DFSFDLN0:

Module DFSFDLN0 allocates the log data sets required by the restart-read process. A request is made to DBRC to locate a particular type of log data set. If the data set is located successfully, it is allocated. The following table shows the codes and an explanation for the registers:

Key	Label	Description
Reg15=RLWLLLF RLWDBRC=DBRC return code	LTA1430	DBRC could not find the latest OLDS.
Reg15=RLWAF RLWAREGF=RDJFCB return code	LTA1578	RDJFCB failed while attempting to allocate a new OLDS for output.
Reg15=RLWLLNF	LTA1585	A new OLDS is needed for output to complete the log close process. A reusable OLDS was not found. Either DBRC could not be checked for an available OLDS, or an operator replied "ABEND" to message DFS0739I.
Reg15=RLWDBRCF RLWDBRC=DBRC return code	LTA1590	The DBRC status exit routine returned a nonzero return code.
Reg15=RLWAF RLWMDRC=DFSMDA return code	LTA1620	Allocation failed for a system log data set (SLDS). If DFSMDA failed, RLWMDRC is nonzero. Otherwise, the SLOPN subroutine in DFSFDLY0 failed to open the SLDS.
Reg15=RLWAF RLWMDRC=DFSMDA return code	LTA21200	Allocation failed for an SLDS. If DFSMDA failed, RLWMDRC is nonzero. Otherwise, the SLOPN subroutine in DFSFDLY0 failed to open the SLDS.
Reg15=RLWAF RLWMDRC=DFSMDA return code	LTA22200	Allocation failed for an SLDS. If DFSMDA failed, RLWMDRC is nonzero. Otherwise, the SLOPN subroutine in DFSFDLY0 failed to open the SLDS.
Reg15=RLWAF RLWDBRC=DBRC return code	LTA2220	DBRC is unable to locate the secondary log data set.
Reg15=RLWLGCE	ALOPENOL	Should-never-occur logic error.
Reg15=RLWLGCE	ALOPENSL	Should-never-occur logic error.
Reg15=RLWAF	LTD320	Final checking within the ALOPENOL subroutine determined that OLDS allocation failed.

### For DFSFDL00:

Module DFSFDL00 contains several subroutines used by the OLDS read transition STATE routines in module DFSFDLQ0.

**XREAD**

Issues a READ for the current OLAT entry and issues a CHECK for the outstanding READ against the next OLAT entry.

**XREREAD**

Resets BSAM DCB status (CHECKs unchecked READs), issues a POINT for the last good block read, issues a READ and CHECK for the last good block, and issues a READ and CHECK for the next block (the block that was in error the last time XREAD was invoked).

**XBLKCHK**

Examines the results of the last CHECK. Tests for EOF, READ errors, and sequence errors. Sets the CURRENT EVENT to indicate the relative success of the last READ/CHECK. Selects the NEXT PARTITION to be used in the restart-read transition table (DFSFDLV0).

The following table shows the codes and an explanation for the registers:

Key	Label	Description
Reg15=RLWLGCE RLWEOPN=open OLDS flags	XRER0020	The CLROLD subroutine in DFSFDLY0 has returned a nonzero return code. It is not possible to reread the current OLDS and an alternate OLDS is not available.
Reg15=RLWLGCE RLWCSTAT=current state RLWTFLG1=error toleration flags	XBLK0016	XBLKCHK is attempting to analyze the last BSAM CHECK. Error toleration is required for STATE XS18 and an unknown condition occurred.
Reg15=RLWLGCE RLWCSTAT=current state RLWTFLG1=error toleration flags	XBLK0026	XBLKCHK is attempting to analyze the last BSAM CHECK. Error toleration is required for STATE XS12 and an unknown condition occurred.
Reg15=RLWLGCE	XBLKEOF	Should-never-occur logic error. XBLKCHK is attempting to analyze the last BSAM CHECK. EOF was detected and the OLDCALC subroutine returned a bad return code while trying to determine where EOF occurred within the data set.
Reg15=RLWLGCE	XBLKREAD	Should-never-occur logic error. XBLKCHK is attempting to analyze the last BSAM CHECK. A read error was detected and the OLDCALC subroutine returned a bad return code while trying to determine where the read error occurred within the data set.
Reg15=RLWLGCE RLWCSTAT=current state	XBLK3500	XBLKCHK is attempting to analyze the last BSAM CHECK. A first block read error was detected and RLWCSTAT is bad.
Reg15=RLWLGCE RLWCSTAT=current state	XBLK4500	XBLKCHK is attempting to analyze the last BSAM CHECK. A middle block read error was detected and RLWCSTAT is bad.
Reg15=RLWLGCE RLWCSTAT=current state	XBLK5500	XBLKCHK is attempting to analyze the last BSAM CHECK. A last block read error was detected and RLWCSTAT is bad.
Reg15=RLWLSE RLWLOGSQ=last good log record sequence number Reg6 at entry to DFSFDLY2 points to the log buffer holding the out-of- sequence log record	XBLK6140	XBLKCHK successfully read the first block in an OLDS but found that the sequence number of the first record in the block is not equal to the previous OLDS/SLDS log record sequence number +1.
Reg15=RLWLGCE RLWCSTAT=current state	XBLK7500	XBLKCHK is attempting to analyze the last BSAM CHECK. An older data sequence error was detected and RLWCSTAT is bad.

Key	Label	Description
Reg15=RLWLGCE LWAREG2=current block suffix address RLWTSUFF=first block of last good OLDS suffix RLWTFLG1=error toleration flags RLWTSKIP=BSN of block being analyzed when the newer data sequence error was detected.	XBLK8075	XBLKCHK is attempting to analyze the last BSAM CHECK. A newer data sequence error was detected because of a skipped OLDS. However, the key fields indicate an unexpected condition.
Reg15=RLWLGCE RLWAREG2=current block address suffix  RLWTSUFF=first block of last good OLDS suffix RLWTFLG1=error toleration flags RLWTSKIP=BSN of block being analyzed when the newer data sequence error was detected.	XBLK8085	XBLKCHK is attempting to analyze the last BSAM CHECK. A newer data sequence error was detected because of a skipped OLDS. The skipped OLDS was located. While performing a forward read of the skipped OLDS, a second newer data sequence error was encountered before returning to the skipped OLDS processing starting point, RLWTSKIP.
Reg15=RLWLGCE RLWCSTAT=current state	XBLK8500	XBLKCHK is attempting to analyze the last BSAM CHECK. A newer data sequence error was detected and RLWCSTAT is bad.

**For DFSFDLP0:**

Module DFSFDLP0 reads system log data sets (SLDS) during warm start or emergency restart. The following table shows the codes and an explanation for the registers:

Key	Label	Description
Reg4=RLWSDECB Reg15=RLWLRE	DFSFDLP0	An I/O error occurred while reading an SLDS.
Reg15=RLWLGCE RLWEMODE	LSA200	An I/O error occurred while reading the secondary SLDS. Recover the SLDSs using the Log Recovery utility and retry restart.
Reg15=RLWAF RLWALLOC	LSA210	DBRC was unable to locate the secondary SLDS.
Reg15=RLWLSE SAVELSN=last LSN read from secondary SLDS while trying to match up with the primary SLDS. (SAVELSN is located within DFSFDLP0) RLWLOGSQ=last LSN read from primary SLDS.	LSA235	EOF was reached on the secondary SLDS without finding a record to match the last record read from the primary SLDS.
Reg4=RLWSDECB Reg15=RLWLRE	LSA240	An I/O error occurred on the secondary SLDS while attempting to match the primary SLDS.



Key	Label	Description
Reg2=first LSN in block Reg3=RLWLOGSQ+1 Reg15=RLWLSE	LSA250	A sequence error occurred on the secondary SLDS while attempting to match the primary SLDS. The log sequence number (LSN) in register 2 is more than one greater than the last LSN read from the primary SLDS.
Reg15=RLWLSE	LSA21R	A sequence error occurred on the secondary SLDS while attempting to truncate the block that would match up with the primary SLDS.

#### For DFSFDLQ0:

Module DFSFDLQ0 contains all the STATE routines used to read an OLDS. There is an 8-byte eye catcher at the beginning of each STATE routine.

Module DFSFDLX0 reads the STATE table, DFSFDLV0, to determine which STATE to execute next. DFSFDLX0 calls DFSFDLQ0 with the address of the STATE routine in register 0. The STATE routine is executed and control is returned to DFSFDLX0, along with a return code. The following table shows the codes and an explanation for the registers:

Key	Label	Description
Reg15=RLWLRE or RLWDRE RLWEOPN=open data set flags RLWTFLG1=error toleration flags	XS020035	Routine XSTATS02 was called to analyze a read error. The CLROLD subroutine failed to reset BSAM status using CLOSE and OPEN. Since all required OLDS data sets are unavailable, restart-read abends.
Reg15=abend code (variable) RLWXDOMA=last STATE flags RLWCPART=last partition	XSTATS03	The STATE table, DFSFDLV0, determined that a U0176 abend is required.
Reg15=RLWLGCE	XS040100	Should-never-occur logic error. The last LSN in a good block was zero.
Reg15=RLWLGCE RLWDBRC=DBRC return code	XS050300	Should-never-occur logic error. XSTATS05 received an unexpected response from DBRC on a LOCATE LAST request.
Reg15=RLWLGCE RLWDBRC=DBRC return code	XS050700	Should-never-occur logic error. XSTATS05 received an unexpected response from DBRC on a LOCATE NEXT request.
Reg15=RLWLGCE RLWDBRC=DBRC return code	XS070025	Should-never-occur logic error. XSTATS07 received an unexpected response from DFSFDLN0 on an ALLOCATE NEXT request.
Reg15=RLWLGCE	XS160200	Should-never-occur logic error. XSTATS16 received an unexpected return code from DFSFDLU0.
Reg15=abend code (variable) RLWXDOMA=last STATE flags RLWCPART=last partition	XS21EXIT	The STATE table, DFSFDLV0, determined that a U0175 abend is required.
Reg15=RLWLGCE	XS290025	Should-never-occur logic error. XSTATS29 received an unexpected response from DFSFDLN0 on an ALLOCATE PRIOR request.

Key	Label	Description
Reg15=RLWLGCE	XS320025	Should-never-occur logic error. XSTATS29 received an unexpected response from DFSFDLN0 on an ALLOCATE LAST request.
Reg15=abend code (variable) RLWXDOMA=last STATE flags RLWCPCPART=last partition RLWXTRC=transition trace table	XSTATS99	Should-never-occur logic error. A search of the STATE table, DFSFDLV0, did not produce a match. A 0175 or 0176 abend is forced.

#### For DFSFDLR0:

Module DFSFDLR0 is the driver module for the log restart-read function. DFSFLLG0 branches to DFSFDLR0 under the restart TCB to perform one of the following functions:

**FIND** Locate the most current type X'42' log record (checkpoint ID table) and pass it back to restart.

The OLDS might be terminated from the WADS during this request.

**OPEN** OPEN the log containing the restart checkpoint and locate the start of the checkpoint.

**READ** READ the log forward, beginning with the restart checkpoint. Pass one block of data at a time back to restart. If necessary, terminate the OLDS from the WADS.

**CLOSE** After EOF on the last log, merge the data set entry tables, DSETs, (initial DSET from log initialization, DSET from the latest X'4301' log record, DSET built during forward log read) and allocate all OLDS.

#### FORMAT

Format the WADS as requested by the /STA WADS, /NRE, or /ERE commands.

The following table shows the codes and an explanation for the registers:

Key	Label	Description
Reg15=RLWLRE	LOA200	The OPEN function encountered a read error while attempting to read the first block from the OLDS containing the restart checkpoint.
Reg15=RLWLNT RLWGLOBL=global flags	LCA10E	The CLOSE function was requested and RLWGLOBL does not indicate that the OLDS was terminated from the WADS.
Reg2=log record address Reg3=log record length Reg15=RLWLSE RLWGLOBL=global flags	LCA21210	The CLOSE function was requested. While searching for a DSET (X'4301' log record) within the block, a zero length record was found.
Reg15=RLWLGCE	LCA2300	Should-never-occur logic error.
Reg2=log record address Reg4=log record length Reg15=RLWLSE RLWGLOBL=global flags	LOB110	The OPEN function was requested. While searching for the beginning checkpoint record (X'4001' log record) within the block, a zero length record was found.
Reg15=RLWLRE	LFB010	The FIND (or CLOSE) function encountered a read error while attempting to read the first block from the latest OLDS.

Key	Label	Description
Reg15=RLWLLLF RLWALLOC=allocation flags	LFB190	The FIND (or CLOSE) function received an unexpected result from DFSFDLN0 on a request to allocate the latest log.
Reg2=log record address Reg4=log record length Reg15=RLWLSE RLWGLOBL=global flags	LFC10R	The FIND (or CLOSE) function was requested. While searching a block for an X'42' (or X'4301') log record, a zero length record was found.

#### For DFSFDLR0:

Module DFSFDLU0 terminates the latest OLDS from the WADS. The following table shows the codes and an explanation for the registers:

Key	Label	Description
Reg15=RLWLLF RLWDBRC=DBRC return code	LBA100	DFSFDLU0 is trying to verify that restart-read is correctly positioned on the latest OLDS. The locate-latest-OLDS request to DBRC failed.
Reg15=RLWRLGB	LBA106	DFSFDLU0 failed in an attempt to reread the last good block from the data set on which it was originally found.
Reg15=RLWRLGB	LBA110	DFSFDLU0 failed in an attempt to reread the last good block from the data set on which it was originally found.
Reg15=RLWLGCE RLWXSTAT=first XRF status flags byte RLWXSTT2=second XRF status flags byte	LBA208	The OLDS is being terminated as part of an XRF takeover. RESERVEs of the OLDS and WADS are required, but have not been established.
Reg15=RLWLNT RLWEOPN=open OLDS flags	OLDSCLR	An attempt to clear BSAM status (using CLOSE and OPEN) for the OLDS to terminate failed.
Reg15=RLWLNT RLWEOPN=open OLDS flags	LDX0100	An attempt to clear BSAM status (using CLOSE and OPEN) for the OLDS to terminate failed.
Reg15=RLWLNT RLWEOPN=open OLDS flags	LDX0200	An attempt to clear BSAM status (using CLOSE and OPEN) for the OLDS to terminate failed.

#### For DFSFDLX0:

Module DFSFDLX0 is the driver for the OLDS read process. The following table shows the codes and an explanation for the registers:

Key	Label	Description
Reg2=loop counter Reg3=next entry in DFSFDLV0 address Reg15=RLWLGCE	XTRANEXT	Should-never-occur logic error. DFSFDLX0 is searching transition entries in a partition of the STATE transition table, DFSFDLV0. All entries in the partition were searched without finding a match. All partitions should have as their last entry an entry that matches any condition. This entry should direct processing to STATE XS99, which abends. In addition, all partitions have a count of the number of entries in the partition. This count is used to initialize register 2.

Key	Label	Description
Reg15=RLWLGCE	XTRA3700	Should-never-occur logic error. An unexpected return code was returned by DFSFDLQ0.

**For DFSFDLY0:**

Module DFSFDLY0 contains all the subroutines commonly used by the restart-read modules. The following table shows the codes and an explanation for the registers:

Key	Label	Description
Reg15=RLWTFCF RLWAREGF=return code from TRKCALC	ANALEOF	The ANALEOF subroutine received a nonzero return code from TRKCALC.
Reg15=RLWLGCE RLWEOPN=open flags for OLDS	LBN120	Should-never-occur logic error. The OLSETM subroutine found no open OLDS.
Reg2=log record address Reg3=end of buffer address Reg4=log record length Reg15=RLWLSE	LLC10R	The RCDSCH subroutine encountered a log record with an invalid length.
Reg15=RLWTFCF	LDG200	The OLDSCALC subroutine received a nonzero return code from TRKCALC.
Reg15=RLWTFCF	OLPNT	The OLPNT subroutine received a nonzero return code from TRKCALC.
Reg15=RLWDBRCF RLWDBRC=DBRC return code RLWAREQ=request passed to DBRC	LTH150	The DBRCEXEC subroutine received an unexpected return code from DBRC.
Reg5=DCB Reg15=RLWLCF RLWEEXCP=exception flags	LSC105	The SLCLS subroutine received an error attempting to CLOSE an SLDS.
Reg15=RLWLGCE	LBP130	Should-never-occur logic error.
Reg2=temporary DSET entry Reg15=RLWLGCE RLWDSETE=DSET entry	LBP130	Should-never-occur logic error. Either RLWDSETE or the temporary DSET is bad.

**For DFSFDLZ0:**

Module DFSFDLZ0, during warm start or emergency restart, performs either OLDS read or SLDS read. The following table shows the codes and an explanation for the registers:

Key	Label	Description
Reg15=RLWLLFF RLWALLOC=allocation flags	LLA330	EOF occurred on a nonlast log data set. A request to DFSFDLN0 to allocate the next log data set failed.
Reg15=RLWLSE	LLA440	EOF occurred on a nonlast log data set. A sequence error occurred while trying to locate the next record.
Reg15=RLWLSE	LLA500	EOF occurred on a nonlast log data set. A sequence error occurred while trying to locate the next record.

Key	Label	Description
Reg15=RLWLSE	LLA600	EOF occurred on a nonlast log data set. A sequence error occurred while trying to locate the next record.

**System action:** IMS terminates abnormally.

**Problem determination:** 1, 5, 8, 11, 35

Save a copy of the OLDSSs and WADSSs, and List the DBRC RECON data set.

**Source:** IMS abends

**Module:** DFSFDLY0, DFSFDLN0, DFSFDL00, DFSFDLP0, DFSFDLQ0, DFSFDLX0, DFSFDLZ0

IMS abend 0175

➡ DFS0739I

➡ DFS0739X

Chapter 2, "IMS problem determination," on page 13

## 0181

**Explanation:** Duplicate data space tokens found. A new data space was acquired that has the same data space token as an existing data space.

**System action:** The IMS control region abnormally terminates.

**System programmer response:** This problem should not occur and represents a z/OS error if it does.

**Source:** IMS abends

**Module:** DBFVIDS0

## 0182

**Explanation:** A must-not-occur condition was detected in one of the Fast Path I/O modules.

**Analysis:** Register 15 contains the diagnostic code: bytes 0-2 contain the module ID and byte 3 contains the subcode.

**For module ID F0R (DBFFORI0), the subcodes are:**

- 01 A process running at SYNCPOINT added a buffer on the output thread for which the write-must-complete state was not set. This condition should not occur.
- 02 Data space unpin buffer after I/O failed.
- 03 Data space I/O in-process count does not match the number of Virtual Storage Option (VSO) buffers being written.

**For module ID MAI (DBFUMAI0), the subcodes are:**

- 01 The resource ID for the pre-allocated SDEP CI equals the current CI relative byte address (RBA) or HWM CI RBA.

**For module ID MI0 (DBFMIOS0), the subcodes are:**

- 02 A process running at SYNCPOINT added a buffer on the output thread for which the write-must-complete state was not set. This condition should not occur.

**For module ID SRB (DBFMSRB0), the subcodes are:**

- 01 DMAC does not support VSO.
- 02 Data space control block structure is invalid.
- 03 Area contained in data space control block structure is invalid.
- 04 RBA requested is too large (GE SDEP portion of area)

**For module ID V0C (DBFVOCIO), the subcodes are:**

- 01 Data space map list (DSML) not passed by caller.
- 02 Dummy DMHRs for write staging area do not exist in DSML.
- 03 No I/O, data space unpin failed.
- 04 No DSML has completed I/O.
- 05 After I/O completed, data space unpin failed.

## 0184

- 06 The control interval (CI) address value of the current cursor is out of the bitmap range.
- 07 Address of L56X block in DSML DSME L56X is zero.
- 08 The relative byte address (RBA) of the data space CI does not match the RBA of the bitmap CI.
- 09 The relative byte address (RBA) of the data space CI exceeds the limit.

### For module ID V50 (DBFVSOW0), the subcodes are:

- 01 The DMAC pointed to in the DMHR did not specify VSO.
- 02 The DMAC and DSME pointers did cross-check.
- 03 The RBA in the DMHR was beyond the scope of the DSME.
- 04 The block passed to DBFVSOW0 was not a DMHR.
- 05 The return code from BCB was nonzero.

### For module ID MAI (DBFUMAI0), the subcodes are:

- 01 The resource ID for the pre-allocated SDEP CI equals the current CI RBA or HWM CI RBA.

### For module ID NCB (DBFNCBS0), the subcodes are:

- 01 The resource ID for the pre-allocated SDEP CI equals the current CI RBA or HWM CI RBA.

### For module ID MFL (DBFMFL10), the subcodes are:

- 01 The user's field search argument (FSA) for the hexadecimal data field exceeds the length allowed by the IMS I/O work area.
- 02 The user's field search argument (FSA) for decimal data field exceeds the IMS I/O work area.
- 03 The user's field search argument (FSA) for the data type is different than the hexadecimal or decimal and exceeds the IMS I/O work area.

### For module ID LG2 (DBFSLG20), the subcodes are:

- 01 Invalid DMHRFLG1 flag byte was found during SYNCPOINT Phase 2 processing. Block on EPSTAWAQ was not EMHB or DMHR.

### For module ID SLG (DBFSLOG0), the subcodes are:

- 01 The database buffer header (DMHR) block that was obtained using an exclusive control resource block (XCRB) chain had an invalid DMHRFLG1 flag byte.
- 02 The DMHR block, that was obtained using UOW exclusive resource control block (UXRB) chain, had an invalid DMHRFLG1 flag byte.
- 03 The Expedited Message Handler Buffer (EMHB) block, that was obtained using EPSTEMHB, had an invalid DMHRFLG1 flag byte.
- 04 After a failed synchronization, the DMHR block, that was obtained using an XCRB chain, had an invalid DMHRFLG1 flag byte.

### For module ID VSR (DBFVSOR0), the subcodes are:

- 01 The DEDB area control list (DMAC) pointed to in the database buffer header (DMHR) did not specify VSO.
- 02 The DMAC and data space mapping entry (DSME) pointers did not cross-check.
- 03 The DSME and DMAC have different data space map list (DSML) pointer.
- 04 The relative byte address (RBA) in the DMHR was beyond the scope of the DSME.
- 05 The block passed to DBFVSOR0 was not a DMHR.

### System action:

The IMS control or dependent region terminates abnormally.

**System programmer response:** If the problem persists, contact IBM Software Support.

**Source:** IMS abends

**Module:** DBFMRSB0, DBFVOCIO, DBFVSOW0, DBFMIOS0, DBFFORI0, DBFUMAI0, DBFNCBS0, DBFMFL10, DBFSLG20, DBFVSOR0, DBFSLOG0

---

## 0184

**Explanation:** IMS could not allocate a structure for long busy toleration in the coupling facility at startup.

**System action:** IMS terminates abnormally.

**Programmer response:** See message DFS1552A. Correct the problem and restart IMS. If the MADS I/O timing function is not required, delete the MADSIOT specification from the DFSVSMxx PROCLIB member and restart IMS.

**Source:** IMS abends

---

**0195**

**Explanation:** A must-not-occur condition was detected in one of Fast Path modules.

**Analysis:** Register 15 contains the subcode for this abend. Register 13 is the address of a save area that contains registers at the time of abend if the module that detects the error saves the registers before issuing the abend.

Key	Description
Reg15=X'01'	DBFARD30 detects that the number of areas in parameter list for authorized areas returned from DBRC is greater than 200.
Reg15=X'02'	DBFHSRT0 detects that there is no EMHB for output message ISRT and there is no INIT STATUS GROUP A

**System action:** The IMS dependent region or the IMS control region abnormally terminates, depending on the abend subcode.

**System programmer response:** Call IBM Software Support.

**Source:** IMS abends

**Module:** DBFARD30, DBFHSRT0

---

**0199**

**Explanation:** A BMP or IMS batch region has exceeded the count of databases that can be changed within a commit cycle.

**System action:**

The BMP or IMS batch region ends abnormally with abend code U0199.

**Operator response:** After correcting the problem, restart the BMP or IMS batch region.

**System programmer response:** Ensure that BMP or IMS batch regions do not make alterations to more databases than will fit in the largest log record allocated for the IMS log data set (that is, IMS log block size) without issuing a SYNCPOINT request. Or, increase the size of the IMS log data set to accommodate the number of changed databases.

An estimate of the maximum number of changed databases that will fit into the largest log record can be determined by using the following decimal formula:

Changed database limit =  $(64 * N - 15)$  databases

where  $N$  is the decimal number of 1024-byte blocks allocated for the IMS log data set (that is, IMS log block size).

**Source:** IMS abends

**Module:** DFSECP20

---

**0200**

**Explanation:** A GET type function was issued using the AIB interface that was expecting data to be returned in the I/O area. However, the length of the I/O area was too small to receive the data. If the abend occurred in a Java dependent region (that is, JBP or JMP), this could be a storage related issue.

**System action:** The application program terminates abnormally.

**Programmer response:** Increase the size of the I/O area to allow the data to be returned to the application. If the abend occurred in a Java dependent region (that is, JBP or JMP), increase the region size of JBP or JMP dependent regions.

**Source:** IMS abends





---

## Chapter 9. IMS abend codes 0201 - 0250

This topic identifies and explains the following abend codes issued by IMS. Each code is 4 decimal digits.

---

### 0201

**Explanation:** One of the following situations may have occurred:

1. At least one HALDB input statement that was read and validated contained syntax errors. Message DFS3777A with a RC=04 or 08 is issued. A RC=04 could also indicate that there were no syntax errors but more than 20 statements were supplied.
2. Single partition processing in an IFP, JMP, or MPP region is not allowed. Message DFS3776A is issued.
3. DFSHALDB failed to open or its attributes are not valid. The LRECL might not be equal to 80 or the RECFM is not fixed block format. The corresponding message is DFS3775A.
4. There is not enough storage to continue processing. Message DFS3774A is issued and indicates a GETMAIN failure for the control block that appears in the message.

**System action:** The job terminates abnormally in module DFSHSPI0 with one of the following reason codes:

#### Codes

##### Message Number

- 04 DFS3777A RC=4
- 08 DFS3777A RC=08
- 12 DFS776A IFP/JMP/MPP
- 16 DFS3775A RECFM
- 20 DFS3775A LRECL
- 24 DFS3667A open failure of DFSHALDB
- 28 DFS3774A insufficient storage

**Programmer response:** Take one of the following actions:

- For syntax errors, review the report in SYSHALDB. Correct the HALDB input statements and resubmit the job. If no syntax errors exist, limit the input to 10 statements and resubmit the job.
- For HALDB input statements that are supplied to DFSHALDB in an IFP, JMP, or MPP region, delete DFSHALDB or remove the input statements and resubmit the job.
- Investigate the attributes of the data set associated with DFSHALDB.
- Increase the size value of the REGION parameter.

**Problem determination:** 1, 17h

**Source:** IMS abends

**Module:** DFSHSPI0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

### 0202

**Explanation:** For any of the statements whose syntax was successfully validated, the PCB number does not correspond to any DBPCB within the PSB or the partition name is not valid. The partition name could be either misspelled or it could be a valid name that cannot be found in the database referenced by the DBPCB number that was supplied. This abend is preceded by messages DFS3379E and DFS3779A.

**System action:** Pseudoabend 0202 is issued from module DFSHSPP0

**Programmer response:** Correct the invalid PCB number or partition name and resubmit the job.

**Problem determination:** 1, 17b, 17h

## 0203 • 0204

**Source:** IMS abends

**Module:** DFSHDPP0

**Related information:**

 DFS3379E (Messages and Codes)

 DFS3779A (Messages and Codes)

Chapter 2, "IMS problem determination," on page 13

---

### 0203

**Explanation:** While running the File Select and Formatting Print utility, DFSERA10, the parameter list constructed from the OPTION statements was found to have been modified. An error might have occurred in the user exit routine (OPTION control statement, operand EXITR=). The error caused the routine to address and modify storage outside the legitimate address space of the program.

**Analysis:** 0203 is a standard abend issued from the Format Print utility, DFSERA10. The registers in the abend SVRB are valid. You can use them together with the program status word (PSW) at entry-to-abend, to aid in identifying the problem.

Register 6 might contain a pointer to the first list element. If it does not, an error occurred during initialization of the option process.

Key	Label	Description
Reg6=pointer to first list element	SELGET	Register 6 is loaded, then tested, for a pointer to the first list element. If the pointer is missing, a branch is taken to the routine at label SEL03, which issues abend 0203.
Reg11=address of routines		

**Possible cause:** There could be an error in the user exit routine (OPTION control statement, operand EXITR=) that caused it to either or both address and modify storage outside the legitimate address space of the program.

**System action:** The program terminates abnormally.

**Programmer response:** Correct any indicated errors, and rerun the job.

**Problem determination:** 3, 4, 35

**Source:** IMS abends

**Module:** DFSERA10

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

### 0204

**Explanation:** An attempt to open a data set defined by the TRCPUNCH DD statement failed. The data set is for output for the DL/I call trace exit routine from DFSERA10.

**Analysis:** This is a standard abend issued by module DFSERA50.

**Possible cause:** There might be an error in the TRCPUNCH DD statement.

**System action:** The job terminates abnormally.

**Programmer response:** Ensure that TRCPUNCH DD statement exists and is correct in the DFSERA10 input stream.

**Source:** IMS abends

**Module:** DFSERA50

---

**0206**

**Explanation:** The IMS.PSBLIB or IMS.DBDLIB libraries could not be opened.

**Analysis:** This is a standard abend issued by module DFSPCC30. Register 1 contains the completion code X'800000CE', as shown in the following table. Register 2 contains the address of PXPparms; label PCDCBADR in PXPparms contains the DCB address.

Key	Label	Description
Reg1=X'800000CE'	BADOPEN	OPEN failure for PSBLIB or DBDLIB.

**System action:** The IMS batch region terminates abnormally.

**Programmer response:** Check the batch processing region JCL for the correct IMS ddnames. Correct the JCL in the DFSERA10 input stream, and resubmit the job.

**Problem determination:** 2, 3

**Source:** IMS abends

**Module:** DFSPCC30

**Related information:**

Chapter 2, "IMS problem determination," on page 13

**0209**

**Explanation:** A region type of ULU or UDR was specified, but the DBD name specified was a logical DBD or the access method was invalid.

**Analysis:** 0209 is a standard abend issued by module DFSDLBL0, when module DFSDPB0 has detected an error.

The program status word (PSW) at time of abend points to the instruction within label SETPSEU from which the abend (SVC 13) is issued. The registers in the abend SVRB should be used for problem isolation.

DFSDLBL0 links to DFSDPB0, at label NOLOAD, if it detects a special region type has been specified (UDR or ULU). The save area pointed to by register 13 in the abend SVRB has the registers stored on entry to DFSDPB0. Register 1 contains the address of a parameter list with the following format:

- Address of region parameter list
- Address of DBD
- Address of constructed PSB
- Address of program control parameters

Register 2 has the address of the name used to load the DBD.

Key	Label	Description
Reg9=address of PREFIX ACCESS X'0F' Reg10=address of loaded DBD	GETDBD	This routine validity checks the access method. If PREACCESS (register 9 + X'0C') without the high-order X'80' bit on is greater than X'0F', indicating GSAM or an MSDB, the abend is issued for an invalid access method.
Reg9=address of PREFIX PRENODSG=0 Reg10=address of loaded DBD	CONT2	The access method is being validity checked. PRENODSG (register 9 + X'0D'), the number of data sets, is determined to be zero. No data set specification means that this is a logical DBD, so the abend is issued.

**System action:** IMS terminates abnormally.

**Programmer response:** Correct the JCL, and resubmit the job.

**Source:** IMS abends

**Module:** DFSDLBL0, DFSDPB0

---

**0210**

**Explanation:** A CCTL subsystem issued an UNSCHEDULE call or a TERMINATE THREAD call to the DBCTL subsystem. DBCTL detected the unit of recovery with outstanding updates, or an ODBA thread issued a DPSB call with no prior commit or rollback, or issued a DPSB call without the PREP (prepare) subfunction followed by a commit or rollback.

IMS issues a DFS554A message when this problem is encountered; an example of the message:

```
DFS554A DDLT40H 00001 CINTDDLT STLDDLT1(5) 000,0210 2008 +
/241 15:45:15 RTKN= ODBA00230000000500000000
```

**System action:** IMS terminates the thread abnormally. The updates are backed out if the DBCTL thread is in flight. The updates are held if the DBCTL thread is in doubt.

**System programmer response:** See message DFS554A for additional information to correct the problem.

**Problem determination:** 1, 3, 5, 33, 36

**Source:** IMS abends

**Related information:**

 DFS554A

Chapter 2, "IMS problem determination," on page 13

---

**0214**

**Explanation:** The parameter field specified a region type of ULU, UDR, DBB, or DLI. The program name supplied is not authorized to use this region type.

**Analysis:** 0214 is a standard abend issued by module DFSDLBL0 when module DFSDPSB0 has detected an error.

The program status word (PSW) at the time of abend points to the instruction within label SETPSEU from which the abend (SVC 13) is issued. The registers in the abend SVRB should be used for problem isolation.

Module DFSDLBL0 links to module DFSDPSB0, at label NOLOAD. It detects if a special region type, UDR or ULU, was specified. Using the save area pointed to by Register 13 in the abend SVRB, these are the registers stored on entry to DFSDPSB0. Register 1 contains the address of a parameter list with the following format:

- Address of region parameter list
- Address of DBD
- Address of constructed PSB
- Address of program control parameters

Key	Label	Description
Reg3=PCPARMS address	AUTHLOOP	The application program name, RCPGM field in RCPARMS, is compared against the entries in the authorized program list. If the end of the list is reached (APLPGM=X'FF') with no match found, the program name supplied is not authorized to use a region type of ULU or UDR, so the abend is issued.
Reg8=authorized program list address		
RCPGM ^= APLPGM		
APLPGM=X'FF'		

**System action:** IMS terminates abnormally.

**Programmer response:** Correct the JCL, and resubmit the job.

**Source:** IMS abends

**Module:** DFSDLBL0, DFSDPSB0

---

**0215**

**Explanation:** The IMS disk logical logger has encountered an internal error.

**Analysis:** Register 15 contains the following return codes:

**Codes****Explanation****X'04'**

Invalid buffer address

**X'08'**

Incorrect post code

0215 is a standard abend issued by module DFSFDLL0.

The following indicates the contents of the registers:

**Register**

	<b>Contents</b>
2	ECB contents if RC=08
6	Invalid buffer address if RC=04
9	Address of ECB
14	Address of detecting routine
15	Return codes

*Possible cause:* DFSFDLL0 was posted erroneously while waiting for a buffer to be made available.

**System action:** IMS terminates abnormally.

**Operator response:** Emergency restart IMS to resume processing.

**Problem determination:** 1, 3, 4, 5, 6, and 25

**Source:** IMS abends

**Module:** DFSFDLL0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

**0216**

**Explanation:** The Log Recovery utility (DFSULTR0) detected a bad return code from DBRC, and DBRC=YES was requested.

**Analysis:** 0216 is a standard abend issued by module DFSULG10.

The following indicates the contents of the registers:

**Register**

	<b>Contents</b>
14	Address of detecting routine
15	Return code from DBRC

*Possible cause:* A bad DBRC PROLOG of the SECLOG record occurred. See the DBRC messages issued preceding this abend.

**System action:** The Log Recovery utility terminates abnormally.

**Operator response:** Contact your IMS system programmer.

**Programmer response:** Correct the DBRC problem or code DBRC=N on the EXEC statement to ignore DBRC.

**Problem determination:** 1, 2, 3, 4, 8, and 25

**Source:** IMS abends

**Module:** DFSULG10

**Related information:**

Chapter 2, "IMS problem determination," on page 13

## 0219

**Explanation:** The command processor, DFSICLD0, detected an error that was unrecoverable or detected an invalid interface with a command action module.

**Analysis:** This abend is a standard abend issued by the /DISPLAY controller, DFSICLD0. The program status word (PSW) at entry-to-abend points to the instruction within label ACTION from which the abend (SVC 13) is issued.

Abend 0219 is issued by DFSICLD0 when, upon return from a call to a specific DISPLAY action module, the length field of the queue buffer passed by DFSICLD0 to the action module is found to have been modified by that action module, and to be greater in size than the original size of the buffer acquired (132 bytes). This implies that the action module wrote over the queue buffer, potentially destroying some other queue buffer and thus the message queues, so the abend is issued.

Register 9 in the abend SVRB registers contains the address of the SPAD (scratch pad work area), and register 5 contains a pointer to the queue buffer. Information in the SPAD (specifically, the SPADCALL and SPADCOND) is sufficient to identify the action module and the original command causing the problem. Register 5 points to the display output line that the action module was attempting to build.

Field SPADCALL in the SPAD contains an index value in the table of addresses of /DISPLAY action modules (label ACTIONAD of DFSICLD0). The condensed form of the command as entered by the console operator is pointed to by the field SPADCOND in the SPAD. Field SPADNTRY is frequently used by the /DISPLAY action modules to determine the action module entry that will build a specific detail line of the display. If this is the case, SPADNTRY can be used to isolate the code that set the invalid length field. The following table gives the index values that could be found in SPADCALL, and the corresponding command and action modules.

Index value	Command	Action module
X'00'	/RDISPLAY	DFSIRD10
X'04'	/DISPLAY STATUS	DFSIDP10
X'08'	/DISPLAY ACTIVE	DFSIDP20
X'0C'	/DISPLAY QUEUE	DFSIDP30
X'10'	/DISPLAY TRAN   LTERM	DFSIDP40
X'14'	/DISPLAY DB	DFSIDP50
X'18'	/DISPLAY LINE	DFSIDP60
X'1C'	/DISPLAY ASMT	DFSIDP70
X'20'	/DISPLAY CONV	DFSIDP80
X'24'	/DISPLAY SHUTDOWN	DFSIDP90
X'28'	/DISPLAY POOL	DFSIDPA0
X'2C'	/DISPLAY NODE	DFSIDPB0
X'30'	/DISPLAY ASMT   LINK   MSNAME   SYSID   MSPLINK	DFSIDPC0
X'34'	/DISPLAY LINK	DFSIDPD0
X'38'	/DISPLAY RTCODE	DBFCDRC0
X'3C'	/DISPLAY DBD	DBFCDDB0
X'40'	/DISPLAY PSB	DBFCDPS0
X'44'	/DISPLAY SUBPOOL	DFSIDPE0
X'48'	/DISPLAY SUBSYS   OASN   CCTL	DFSIDPF0
X'4C'	/DIS MODIFY	DFSIDPG0
X'50'	/DIS OLDS	DFSIDPH0
X'54'	/DIS AREA	DBFCDAR0
X'58'	/DIS HSB	DFSIDPI0
X'5C'	/DIS DB BKERR	DFSIDPJ0

Index value	Command	Action module
X'60'	/DIS TRACE	DFSIDPK0
X'64'	/DIS HSSP	DBFPDHS0
X'68'	/DIS TIMEOVER	DFSIDPL0
X'6C'	/DISPLAY APPC   DESC	DFSIDPM0
X'70'	/DISPLAY LUNAME TPNAME	DFSIDPN0
X'74'	/DISPLAY FPV	DBFCDVS0
X'78'	/DISPLAY PROGRAM	DFSIDPP0
X'7C'	/DISPLAY TRACKING STATUS	DFSIDPO0
X'80'	/DISPLAY AOITOKEN	DFSIDPQ0
X'84'	/DISPLAY OTMA	DFSYPDPO
X'88'	/DISPLAY FDR	DFSIDPZ0

The following table provides an explanation of the registers:

Key	Label	Description
Reg2=pool ID for ICREATE Reg5=address of queue buffer Reg9=address of SPAD	ACTION1 and DOCREATE	Both of these routines branch unconditionally to label ACTION, which BALRs to a specific action module based on the value in field SPADCALL. On return from the action module, if the length of the message built in the action module is greater than the output buffer size, the abend is issued.

**System action:** IMS terminates abnormally.

**Operator response:** Note the activity on the terminal that caused the message, and contact your IMS system programmer for assistance.

**Problem determination:** 1, 4, 5, 6, 35

**Source:** IMS abends

**Module:** DFSICLD0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

## 0223

**Explanation:** A DBT thread issued a DLI call immediately following a successful commit. Although the commit completed successfully, a /DBD (DBDUMP) or /DBR (DBRECOVERY) command was found to be active against one or more of the databases associated with the scheduled program (PSB).

IMS issues a DFS554A message when this problem is encountered; an example of the message:

```
DFS554A DDLT5J 00001 G PLAPJK04(7) 000,0223 2014 +
/280 11:13:06 RTKN= ODBA00250000000500000001
```

**System action:** The DLI call was not processed and the DBT thread was abnormally terminated to allow the outstanding command(s) to complete.

**Programmer response:** See message DFS554A for additional information to correct the problem.

**Problem determination:** 1, 3, 5, 33, 36

**Module:** DBFIRC10, DFSCPY00

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

**0225**

**Explanation:** The Database Image Copy 2 utility (DFSUDMT0) issues this abend when it determines that invalid information was received from DFSMSdss. The utility also issues message DFS3144A, which identifies the database data set for which the problem occurred.

**Analysis:** 0225 is a standard abend issued by module DFSUDUI1. Register 15 contains the reason code for the termination. The reason codes have the following meanings:

**Codes****Explanation**

**4** The output volume serial number from DFSMSdss on the exit option 26 invocation is zeros or is invalid.

The utility continues processing for other database data sets if DFSMSdss is able to recover from the abend. The return code for the utility execution is 8 or higher.

**System action:** The Database Image Copy 2 utility (DFSUDMT0) continues processing for other database data sets if DFSMSdss is able to recover from the abend. The return code for the utility is 8 or higher.

**Programmer response:** Check ADR $nnn$  messages issued by DFSMS to aid in diagnosis. See *z/OS MVS System Messages, Volume 1* for a description of ADR messages. If you cannot determine the solution, contact IBM Software Support. Have the abend documentation available.

**Source:** IMS abends

**Module:** DFSUDUI1

**Related reference:**

 z/OS: ADR messages

**Related information:**

 DFS3144A

---

**0229**

**Explanation:** Abend 0229 is issued by module DFSRLP00 for the following reason:

A normal or emergency restart (not cold start type) command was issued, and the logs that are used for restart reached end-of-file (EOF) processing without encountering the log records that are needed to reconstruct necessary resource blocks.

**Analysis:** 0229 is a standard abend issued from module DFSRLP00. The program status word (PSW) at entry-to-abend points to the instruction from which the abend (SVC 13) is issued. If the abend was issued because of an EOF condition, register 15 contains an address.

**Possible cause:** The required set of system checkpoint log records that contain the necessary resource blocks were not read.

**System action:** IMS terminates abnormally.

**Programmer response:** Ensure that the log record data sets (WADS/OLDS) are complete and in the correct sequence to restart IMS. If this is not possible, consider a cold start type of restart.

**Note:** Database and Transaction Manager integrity issues can exist when an IMS cold start is performed after an abnormal termination of the control region.

**Problem determination:** 1, 25, 30, 35, 40, LOGREC entries

**Source:** IMS abends

**Module:** DFSRLP00

---



---

**0230**

**Explanation:** This abend is issued by module DFSRLP00 for these reasons:

- An /ERE BLDQ command was issued and the log used for restart reached EOF before the end of checkpoint record (4099) was read.
- During restart an attempt to get storage for a work area from the MAIN (WKAP) storage pool failed.

**Analysis:** 0230 is a standard abend issued from module DFSRLP00. The program status word (PSW) at entry-to-abend points to the instruction from which the abend (SVC 13) is issued. If the abend was issued because of an EOF condition, register 15 contains an address. If the abend was issued because of a failure to acquire the work area, register 15 contains the return code of the storage request.

**Possible cause:** For the EOF failure, the required checkpoints may have spanned volumes and the second volume was not read. For the storage request failure, the size of the MAIN (WKAP) pool may be too small to satisfy the request.

**System action:** IMS terminates abnormally.

**Programmer response:** Specify an earlier checkpoint at which to restart IMS, or determine the cause of the error and rerun this execution.

**Problem determination:** 1, 4, 5, 35

**Source:** IMS abends

**Module:** DFSRLP00

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

**0231**

**Explanation:** IMS was not able to resynchronize with the CQS subsystem.

**Analysis:** 0231 is a standard abend issued from module DFSRLP00. Register 15 contains the return code.

For more information on inconsistent use of EMHQ across IMS executions, see the IMS diagnosis information.

The abend subcode in register 15 provides the reason for the failure:

**Subcode****Explanation**

- 01** The operator replied ABORT to message DFS3909A.
- 02** IMS received a return code other than 0 and 4 from CQS. This return code is provided in register 14 in the dump.
- 03** An attempt was made to warm or emergency start IMS with CQS, but CQS was not active in the prior IMS execution. A change in the usage of CQS can only be made during a cold start.
- 04** An attempt was made to warm or emergency start IMS without CQS, but CQS was active in the prior IMS execution. A change in the usage of CQS can only be made during a cold start.
- 05** An attempt was made to warm or emergency start IMS with a different CQS than was active in the prior IMS execution. A change of this type can only be made during a cold start.
- 06** An attempt was made to warm or emergency start IMS with a different CQS MSGQNAME than was used in the prior IMS execution. A change of this type can only be made during a cold start.
- 07** An attempt was made to warm or emergency start IMS with a different CQS EMHQNAME than was used in the prior IMS execution. A change of this type can only be made during a cold start.
- 08** An EMHQ structure was used in the previous IMS execution and no EMHQ structure is used in the current execution, or no EMHQ structure was used in the IMS execution and an EMHQ structure is used in the current execution. A change of this type can only be made during a cold start. If abend 0231 is caused by inconsistent use of EMHQ across IMS executions, you can:
  - Either add or remove the EMHQ statement in the DFSSQxxx PROCLIB member for the current IMS execution so that the PROCLIB member matches the EMHQ setting in the previous IMS execution.

## 0233 • 0240

- Restart IMS with a cold start, using either /ERESTART COLDCOMM for an emergency restart, or /NRESTART CHECKPOINT 0 for a full cold start if you want to change EMHQ usage.
- Messages on EMHQ from the previous execution are then discarded. For every message discarded, a X'67D0' subtype 11 trace log record is written.

**Source:** IMS abends

**Module:** DFSRLP00

**Related information:**

 [DFS3909A \(Messages and Codes\)](#)

---

### 0233

**Explanation:** IMS was not able to resume with the current resource structure usage.

**Analysis:** The abend subcode provides the reason for the failure:

#### Subcode

##### Explanation

- 01** An attempt was made to warm or emergency restart IMS with a resource structure, but the resource structure was not in use by the TM component of IMS in the prior IMS execution. This change can only be made during a cold start of the TM component of IMS.
- 02** An attempt was made to warm or emergency start IMS without a resource structure, but the resource structure was in use by the TM component of IMS in the prior IMS execution. This change can only be made during a cold start of the TM component of IMS.

**System action:** IMS terminates abnormally.

**Programmer response:** Determine whether or not IMS should be using a resource structure, then determine whether the resource structure has been defined correctly to the Resource Manager. If IMS must change its use of the resource structure, cold start the Transaction Manager component of IMS.

**Source:** IMS abends

**Module:** DFSRLP00

---

### 0240

**Explanation:** A message processing application program exceeded the allowable execution time. This time includes the processing time used by the dump process if any. You set the processing time at IMS system definition. In the case of a BMP, this abend indicates that the value specified in CPUTIME= has been exceeded.

**Analysis:** 0240 is issued by DFSPCC20 and DFSECP10. When issued by DFSPCC20 the registers point into the application. When issued by DFSECP10 the registers do not point to the application. In this case the register 13 back chain ends in the eye catcher 'F1SA'. Use the Linkage Stack created by the BAKR instruction to obtain the application registers and PSW at the time of abend.

**System action:** A storage dump is created and the dependent region program controller is reattached. In the case of a BMP step, the step is terminated.

**Programmer response:** Refer to the TRANSACT macro in TRANSACT macro (System Definition).

**Problem determination:** 5, 14

Save the log data sets being used and the assembly listings of any user-modified IMS modules.

**Source:** IMS abends

**Module:** DFSPCC20, DFSECP10

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

**0242**

**Explanation:** The system issues this abend for the following reasons:

- The DIRCA parameter (for BMPs) or PCB parameter (for MPPs) specifies a size that is too small to accommodate the PSB to be scheduled.
- The default DIRCA size is not large enough to accommodate the PSB to be scheduled.
- The PCB parameter for the message region is not large enough.
- A Database Control (DBCTL) internal error in building the DIRCA for a Coordinator Control (CCTL) thread occurred.

**Analysis:** This is a standard abend when the error was detected by module DFSASK00.

**System action:** The IMS dependent region terminates abnormally, and the program in progress stops.

**Operator response:** Issue a /START command for the program, and contact your IMS system programmer for assistance.

**Programmer response:** See the IMS system definition information on the default DIRCA size and how it is calculated. Increase the DIRCA size specifications.

**Problem determination:** 1, 2, 3, 4, 5, 6, 8, 9, 17b

**Source:** IMS abends

**Module:** DFSASK00

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

**0243**


**Explanation:** The system issues this abend for one of the following reasons:

- IMS has timed out an APPC transaction. You specify the time-out value in the APPCIOT=(,XX) parameter.
- IMS detected an expired input transaction and canceled the transaction. You can specify the expiration time for the canceled transaction in the OTMA message prefix, using the TRANSACT macro, the IMS destination creation exit DFSINSX0, or the DRD type-2 commands such as CREATE/UPDATE TRAN commands.
- At the application GU time, IMS detected that the maximum number of transaction pipes (tpipes) as defined by the MAXTP= parameter in the OTMA client descriptor is reached, and then IMS canceled an OTMA input transaction that required a new tpipe. The system then issued this abend because TODUMP=YES or TODUMP=U243 is specified in the OTMA client descriptor.

**System action:** A storage dump is created and the dependent region controller is reattached.

**Source:** IMS abends

**Related reference:**

 OTMA client descriptor syntax and parameters (System Definition)

---

**0249**

**Explanation:** An invalid status code (not STATUSGA or status blank) was returned to the DL/I test program, DFSDDL0, during its internal end-of-job status calls in a batch environment. The absence of VSAM buffers in the DFSVSAMP DD card results in a STATUSGE on the status call and causes abend 0249 to be issued from DFSDDL0, if you specify the ABU249 option.

**Analysis:** This abend issued by DFSDDL0, the DL/I test program, in a batch environment. After all DFSDDL0 control statements are processed, DFSDDL0 issues internal end-of-job status calls. DFSDLA00, the DL/I Call Analyzer, calls DFSDV BH0, the buffer handler, for OSAM and VSAM buffer statistics. The return code (register 15) from DFSDV BH0 is non zero.

DFSDLA00 sets DBPCBSTC to 'GE' (STATUSGE) and passes it back to DFSDDL0. DFSDDL0 checks DBPCBSTC for blanks or 'GA'. When DBPCBSTC is not blanks, 0249 is issued, if you specified the ABU249 option. If you did not specify the ABU249 option, printing of the output for the status call is bypassed.

In an online environment, the job terminates with 0479. DFSDLA00 issues the abend and does not return to DFSDDL0 to finish the status calls.

## 0250

**Possible cause:** The user coded ABU249 on the DFSDDLTO option statement, and an invalid status code was received during the DFSDDLTO internal end-of-job status calls.

**Attention:** The absence of VSAM buffers in the DFSVSAMP DD statement results in a STATUSGE on the status call and causes 0249 to be issued from DFSDDLTO, if the ABU249 option is coded.

**System action:** The application program is abnormally terminated.

**Programmer response:** In the dump, locate PSTRTCDE to determine the cause of the error.

**Problem determination:** 17a, 17b, 35

**Source:** IMS abends

**Module:** DFSDDLTO

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

## 0250

**Explanation:** The DL/I test program, DFSDDLTO, issued a conditional GETMAIN macro for an area to be used for the segment I/O area. If running online, the size requested was 32732 bytes. If running in batch, the size was the maximum I/O length determined when the blocks were built for the PSB. The requested storage was not available.

**Analysis:** This is a standard abend issued by module DFSDDLTO.

**System action:** The IMS dependent or batch region terminates abnormally.

**Programmer response:** Provide a larger region size for the step and rerun the job.

**Source:** IMS abends

**Module:** DFSDDLTO

---

## Chapter 10. IMS abend codes 0251 - 0300

This topic identifies and explains the following abend codes issued by IMS. Each code is 4 decimal digits.

---

### 0251

**Explanation:** IMS was unable to open one of the data sets used by the DL/I test program for batch. A message on the SYSPRINT data set indicates the ddname of the data set that could not be opened, unless the failure was in opening the SYSPRINT data set.

**Analysis:** 0251 is a standard abend issued from DFSDDL0 (named DFSDDL0 after being bound with the language interface module DFSLI000). The abend is issued because of a failure to open the DCB for SYSIN, SYSPUNCH, or SYSPRINT.

If the failure is to open SYSIN or SYSPUNCH, a message is written to the SYSPRINT data set prior to the abend.

The program status word (PSW) at entry-to-abend and the registers in the abend SVRB should be used for problem isolation, along with the output from the SYSPRINT data set, if available. The following table provides the register information:

Key	Label	Description
Reg1=completion code, X'800000FB' Reg3=address of SYS2NDCB Reg9=address of SYSINDCB	ISBATCH	The DCBOFLGS field of the SYSIN DCB is tested for a successful OPEN. If the data set did not open, a branch is taken to the routine at label NOOPEN to move the completion code and to write the message "ABEND 251 ISSUED DUE TO UNSUCCESSFUL OPEN OF DDxxxxxxx" to the print data set. A branch is then taken to RETURN at the label EPILOGUE to abend.
Reg1=completion code, X'800000FB'	ISBATCH	The DCBOFLGS field of the PRINT DCB is tested for a successful OPEN. If the data set did not open, the abend code is loaded into register 1 and the abend is issued. No message is issued because the print data set is not available.
Reg1=completion code, X'800000FB' Reg9=address of PUNCH DCB	CDONE	The DCBOFLGS field of the PUNCH DCB is tested for a successful OPEN. If the data set did not open, a branch is taken to the routine at label NOOPEN to move the completion code and issues the message "ABEND 251 ISSUED DUE TO UNSUCCESSFUL OPEN OF DDxxxxxxx" to the print data set. A branch is then taken to RETURN at label EPILOGUE to abend.

**System action:** The IMS dependent or batch region terminates abnormally.

**Programmer response:** Provide a DD statement for the data set and execute the job again.

**Problem determination:** 2, 3, 4, 20

**Source:** IMS abends

**Module:** DFSDDL0, DFSDDL0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

### 0252

**Explanation:** The DL/I test program (DFSDDL0) might issue this abend for the following reasons:

- An abend was requested by specifying ABEND in columns 1 through 5 on an input control card.
- An unknown return code was received after the STAT call was issued to gather statistics before termination.

## 0253

- An unknown return code was received following a message ISRT call to the IMS message queue. No PRINTDD statement was present in the JCL for the job.

**Analysis:** While running the DL/I test program for batch, DFSDDL0 (named DFSDDL0 after being bound with the language interface module, DFSLI000), a control statement requesting an abend has been encountered.

0252 is a standard abend from the DL/I test program, and the registers in the abend SVRB should be used for problem isolation, along with the output from the SYSPRINT data set.

This abend has been coded and requested by the user, and is primarily for diagnostic purposes.

Key	Label	Description
Reg1=completion code, X'80000FC' Reg3=BAL to Message Writer	NOTWTSR	The field CARDID is tested to see if the user requested an abend using a special control statement. If so, a branch is taken to a routine to issue the message "ABEND 252 ISSUED DUE TO ABEND CONTROL CARD 'PRINTDD'," and then branches to RETURN at label EPILOGUE to abend.

**System action:** The IMS batch or dependent region terminates abnormally with abend 0252.

**Programmer response:** If you requested the abend, the abnormal termination is expected. If the abend occurred on a STAT call, the first database PCB contains the return code. If it occurred on an ISRT call to the message queue, the I/O PCB contains the return code. Determine the cause of the return code, correct the error, and rerun the job.

**Source:** IMS abends

**Module:** DFSDDL0, DFSDDL0

---

## 0253

**Explanation:** The name of the database PCB specified, starting at column 16 of the last status statement read by the DL/I test program, matches none of the database PCB names in the PSB.

**Analysis:** 0253 is a standard abend issued from the DL/I test program for batch, DFSDDL0 (DFSDDL0 after being bound with the language interface module DFSLI000). The program status word (PSW) at entry-to-abend and the registers in the abend SVRB should be used for problem isolation, along with the output from the SYSPRINT data set.

Key	Label	Description
Reg1=completion code, X'80000FD' Reg3=BAL to Message Writer Reg5=PCB Address	NOTPCB	Register 5 is tested to see if the address is for the last PCB (negative register). If this is the last, and the requested PCB has not been found, a branch is taken to the routine at NOFOUND to issue the message "DBDxxxxxxx DOES NOT EXIST—ABEND 253 'PRINTDD'". A branch is taken to RETURN to label EPILOGUE to abend.

**Possible causes:** Check the PCB name in the status control statement and ensure that it is the same in the PCBs in the PSB named on the //EXEC statement of the JCL.

An online/BMP transaction might have attempted to process a generated PSB (GPSB). Check whether the PSB specified is a GPSB.

**System action:** The IMS dependent or batch region terminates abnormally.

**Programmer response:** Ensure that the PCB name in the status control statement is correct by comparing the database PCB names in the PSB specified on the EXEC statement. Correct the error, and resubmit the job.

**Problem determination:** 2, 3, 4, 18, 19

**Source:** IMS abends

**Module:** DFSDDL0, DFSDDL0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

**0254**

**Explanation:** The DL/I test program, DFSDDLTO, received an AI status code, indicating that one of the data sets used by DL/I could not be opened. A message on the output data set indicates the ddname of the data set that could not be opened.

**Analysis:** 0254 is a standard abend from the DL/I test program for batch, DFSDDL0 (named DFSDDLTO after being bound with the language interface module, DFSLI000). The program status word (PSW) at entry-to-abend and the registers in the abend SVRB, along with the printed output from the SYSPRINT data set, should be used in problem isolation.

Key	Label	Description
Reg1=completion code, X'80000FE' Reg3=BAL to Message Writer Reg9=PCB Address	ONLINE18	A compare is made of the return code from the call for an AI status code, indicating a data management OPEN failure (any database OPEN failure). If AI was returned, the message DATA MANAGEMENT OPEN ERROR—ABEND 254 is issued to the print data set, and a branch is taken to RETURN to label EPILOGUE to abend.

**System action:** The IMS dependent or batch region terminates abnormally.


**Programmer response:** See the DL/I status code information for possible causes of an AI status code.

**Problem determination:** 2, 3, 4, 18, 19, 20

**Source:** IMS abends

**Module:** DFSDDL0, DFSDDLTO

**Related reference:**

 [DL/I status codes \(Messages and Codes\)](#)

**Related information:**

Chapter 2, "IMS problem determination," on page 13

**0255**

**Explanation:** In the DL/I test program (DFSDDLTO), the specified limit of unequal compares was reached. A nonzero return code was received from the buffer handler.

**Analysis:** 0255 is a standard abend issued by module DFSDDLTO. The following is a list of the buffer handler return codes in the DSECT PST:

PSTRTCDE	DC	1XL1'00'	Status of call
PSTCLOK	EQU	X'00'	EVERYTHING SATISFACTORY
PSTGTDS	EQU	X'04'	RBN BEYOND DATA SET
PSTRDERR	EQU	X'08'	PERMANENT READ ERROR
PSTNOSPC	EQU	X'0C'	NO MORE SPACE IN DATA SET
PSTBDCAL	EQU	X'10'	ILLEGAL CALL
PSTENDDA	EQU	X'14'	END OF DATA SET ENCOUNTERED *NO RECORD RETURNED
PSTNOTFD	EQU	X'18'	REQUESTED RECORD CANNOT BE FOUND
PSTNWBLK	EQU	X'1C'	NEW BLOCK CREATED IN BUFFER POOL
PSTNPLSP	EQU	X'20'	INSUFFICIENT SPACE IN POOL
PSTTRMNT	EQU	X'24'	USER MUST TERMINATE. NO SPACE IN POOL
PSTDUPLR	EQU	X'28'	LOGICAL RECORD ALREADY IN KSDS
PSTWRERR	EQU	X'2C'	PERMANENT WRITE ERROR

PSTRTCDE	DC	1XL1'00'	Status of call
PSTBUFIN	EQU	X'30'	BUFFER INVALIDATED
PSTBIDIN	EQU	X'34'	UNABLE TO ACQUIRE BID/B
PSTPDERR	EQU	X'38'	UNABLE TO LOCATE DDIR/PDIR ENTRY
PSTNOSTO	EQU	X'3C'	STORAGE NOT AVAILABLE

Message DFS255 was returned along with this abend.

**Possible cause:** An invalid value was specified for RCF, SGN, TRN or ISIS.

**System action:** The IMS dependent or batch region terminates abnormally.

**Programmer response:** No action is required.

**Source:** IMS abends

**Module:** DFSDDLTO

**Related information:**

 DFS255

## 0256

**Explanation:** The communication restart processor for conversations detected an error while processing X'11' log records during emergency restart. This problem is probably caused by modifications to the IMS system.

**Analysis:** This abend is a standard abend issued from the IMS/VS communication restart processor (DFSCRVPV0) while reprocessing the conversation start (type X'11') log record during emergency restart.

The program status word (PSW) at entry-to-abend and the registers in the abend SVRB aid in isolating the problem.

Key	Label	Description
Reg2=address of type X'11' log record Reg9=address of RESTART ECB Reg11=address of SCD Reg15=return code from DFSCONS0	CRPV11	Register 15 is tested for a return code from DFSCONS0 (a return code from DFSBCB FUNC=GET for conversation block). A nonzero return code results in an abend.

**System action:** IMS terminates abnormally.

**Operator response:** Run the Database Backout utility for each PSB that was scheduled when the original problem occurred; then cold start IMS.

**Programmer response:** Determine what modifications might be causing this problem.

**Source:** IMS abends

**Module:** DFSCRVPV0

## 0257

**Explanation:** An error occurred while scanning for a VTAM terminal control block when processing a X'11' or X'12' log record during an emergency restart. This problem is probably caused by modifications to the IMS system.

**Analysis:** 0257 is a standard abend issued by module DFSCRVPV0.

Register 2 contains the address of the log record. The log record contains the node name, and if the node is an LU6, the subpool name used in scanning for the VTAM terminal control block.

DSECT LCONVERS, which is part of the ILOGREC macro, defines the following fields:

**LCONNODE**

Node name



**LCONHSQN**

Subpool name

**Possible cause:** This problem is probably caused by modifications to the IMS system.

**System action:** IMS terminates abnormally.

**Operator response:** Run the Database Backout utility for each PSB that was scheduled when the original problem occurred; then cold start IMS.

**Programmer response:** Determine what modifications might have caused this problem.

**Problem determination:** 4, 5, 12, 14

**Source:** IMS abends

**Module:** DFSCRPV0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

**0258**

**Explanation:** This abend is caused by an error detected either in the /HOLD or /RELEASE command processor (DFSICLH0) during emergency restart or in the checkpoint logger (DFSRC30).

**Analysis:** This is a standard abend issued by the /HOLD or /RELEASE command processor during emergency restart or by the checkpoint logger because of an OSAM error reading or writing a disk SPA from or to the SPA data set.

The program status word (PSW) at entry-to-abend and the registers in the SVRB aid in isolating the problem, as shown in the following table:

Key	Label	Description
Reg6=address of CCB Reg9=address of CCB/DECB Reg15=OSAM return code: 04=I/O completed abnormally 08=I/O was not initiated	SPAERR	Register 15 is tested for a return code from OSAM. A nonzero return code results in an abend. DECBSTAT contains additional information.

**Source:** IMS abends

**Module:** DFSICLH0, DFSRC30

**0260**

**Explanation:** The number of parameters (data items named in the USING list) in the application program call to IMS exceeds the allowable limit of 18. This abend can also occur if the checkpoint call is used and too few parameters are specified, or the number of user-specified areas exceeds the number specified on the XRST call, or the user area parameters are not paired (a length and address for each area to be dumped).

This abend is issued for one of the following reasons:

- The number of parameters in the application program call to IMS exceeded the allowable limit of 18 or is equal to 0.
- The checkpoint call was used and one of these explanations applies:
  - Too few parameters were specified.
  - The number of user-specified areas exceeds the number specified on the XRST call.
  - The user area parameters are not paired (a length and address for each area to be dumped).
- The application program overlaid the DL/I function code so that the first character was not alphanumeric. This problem causes the field to be treated as a count field.

**Analysis:** 0260 is a standard abend that can be issued by the MPP application environment controller, DFSECP10. If issued here, the program status word (PSW) at entry-to-abend points to the instruction within label EC1ABEND from which the abend (SVC 13) is issued.

## 0261

Key	Label	Description
Reg1=X'80000104'	EC1ABEND	DFSLIE00 or DFSLIE20 detect that the call-parameter list is too long. A pseudoabend code is set, and DFSECP10 issues the abend.

### For DFSECP20:

When this abend is issued by the BMP environment controller, DFSECP20, the program status word (PSW) at entry-to-abend points to the instruction within label EC2ABEND, from which the abend (SVC 13) is issued.

Key	Label	Description
Reg1=X'80000104'	EC2ABEND	DFSLIE00 or DFSLIE20 detect that the call-parameter list is too long. A pseudoabend code is set, and DFSECP20 issues the abend.

### For DFSPR000:

This abend that can be issued by the batch application program request handler, DFSPR000. If issued here, the program status word (PSW) at entry-to-abend points to the instruction within label PRABEND, from which the abend (SVC 13) is issued.

All 0260 abends from this module are the result of a conditional branch to label PRAB1 by the routine that detected the error condition.

The following labels can be found in module DFSPR000 to determine the level:

PRLAN  
PRIMP  
PRCHKPP  
PRCPX  
LENXLOOP

**System action:** The online dependent or standalone batch region terminates abnormally. Other IMS regions are not affected.

**Programmer response:** If the problem is an incorrect number of parameters, correct the DL/I call from the application program to conform to the format and content.

When using the symbolic checkpoint call, be sure that the number of user-specified areas does not exceed the number specified on the XRST call.

If the application program overlaid the DL/I function code, correct the application program.

**Source:** IMS abends

**Module:** DFSECP10, DFSECP20, DFSPR000

---

## 0261

**Explanation:** One of the values passed in the USING list of the application program call to IMS is invalid. It either exceeds object machine size, does not meet alignment requirements, or violates storage protection boundaries.

One of the parameters passed in the USING list of the application program from the application program DL/I call to IMS is invalid. It either exceeds object machine size, does not meet alignment requirements, or violates storage protection boundaries.

Some possible causes for failure are:

- The call requires an AIB. No AIB was provided.
- The parameter count in the DL/I call is invalid.
- An invalid PCB is detected by IMS. For example:
  - A PL/I program does not match a PSB.
  - The CMPAT parameter might be specified incorrectly. If the application program is designed to use the IOPCB, CMPAT=YES should be specified in the PSGEN statement for the program to run in the DL/I batch region.
  - Incorrect DCB information on the GSAM input data set DD statement.

**Analysis:** 0261 is a standard abend that can be issued from one of these modules: DFSCDLI0, DFSPR000, DFSECP10,

or DFSECP20. The program status word (PSW) at entry-to-abend isolates the failure to a particular module.

There are the conditions that could result in a 0261 abend:

- A parameter list address is not on a word boundary.
- A parameter list address is outside main storage.
- A parameter list address violates the nucleus (protected) boundaries.
- No AIB was provided on the call to AERTDLI.
- The DL/I call list does not have an address for the AIB.

#### For DFSCDLI0:

This is a standard abend that can be issued by the ODBA language interface module DFSCDLI0 AERTDLI. When issued from this module, the program status word (PSW) at entry-to-abend points to the instruction within DFSCDLI0 AERTDLI, from which the abend (SVC 13) is issued.

These are the reasons that the abend can be issued:

- No AIB was provided on the call to AERTDLI.
- The DL/I call list does not have an address for the AIB.

The following is the format for the DL/I call list:

- Optional parmcount
- DL/I call function
- AIB
- Additional call parameters

#### For DFSECP10, DFSECP20:

The parameter list is not on the fullword boundary.

Key	Label	Description
Reg1='X'80000105'	EC1ABEND EC2ABEND	DFSLIE00 or DFSLIE20 detects that the user parameter list is not on a fullword boundary, and sets a pseudoabend code. DFSECP10 and DFSECP20 issue the abend.

#### For DFSPR000:

When issued from this module, the program status word (PSW) at entry-to-abend points to the instruction within label PRABEND, from which the abend (SVC 13) is issued.

All 0261 abends from this module are the result of a conditional branch to label PRAB2 by the routine that detected the error condition.

Key	Label	Description
Reg2=address of PXPARMs (DFSPRPX0) dsect	PRSTAERN	A test is made to see if the user parameter list is on a word boundary. If not, a branch is taken to PRAB2 to handle the abend.
Reg4=highest machine address Reg6=points to the address in parameter list Reg7=lowest nonnucleus address (dynamic area)	VCHK3	A compare is made between register 4 and register 6, and between register 6 and register 7. If the address in the parameter list is higher than the highest machine address, or if the address in the parameter list is lower than the nucleus boundary, a branch is taken to PRAB2 to handle the abend.

**System action:** The online dependent or stand-alone batch region terminates abnormally.

## 0262 • 0263

**Programmer response:** Correct the DL/I call list to conform to the correct format and content.

1. Correct the application program DL/I call statements.
2. Rebind the application program with the ENTRY control statement.

**Problem determination:** 4, 17d

**Source:** IMS abends

**Module:** DFSCDLI0, DFSECP10, DFSECP20, DFSPR000

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

### 0262

**Explanation:** A batch message processing (BMP) program issued a DL/I call after receiving an XD status code. The XD status code was set because the program issued a DL/I checkpoint or SYNC call while the IMS system was undergoing a checkpoint freeze shutdown.

**Analysis:** Status code XD was returned on the DL/I checkpoint or synchronization call, warning the application program not to issue another DL/I call. Check the application program for the DL/I call sequences.

**System action:** The BMP region is terminated abnormally.

**Operator response:** First, contact your IMS system programmer for assistance. At IMS system restart, you can restart the batch message processing program from the DL/I checkpoint.

**Programmer response:** This is an application program error. Status code XD was returned on the DL/I checkpoint call to warn the application program not to issue another DL/I call. Correct the application program, and resubmit the job.

**Source:** IMS abends

**Module:** DFSECP20

---

### 0263

**Explanation:** An invalid call was made to a Database Control (DBCTL) subsystem for a Coordinator Control (CCTL) subsystem thread. If this abend occurred when issuing a call against the I/O program communication block (IOPCB), then the call was in error; otherwise, this abend is an IMS internal error.

**Analysis:** 0263 is a pseudoabend.

The SSPDCODE field in the SSOB block in the thread SDUMP identifies the cause.

SSPDCODE	Module	Reason
ICAL	DFSDPRH0	The function in the DL/I call list is invalid when using the IOPCB.
RINV	DFSDPRH0	This is an internal database resource adapter (DRA) error. The sync point function number the DRA generated and requested is invalid.
	DFSDCPY0	When neither code is in field SSPDCODE, the sync point number requested is invalid. This is an internal IMS error.

**System action:** The CCTL thread is terminated.

**Programmer response:** SNAP or SDUMP the CCTL thread.

**Problem determination:** 38

**Source:** IMS abends

**Module:** DFSDPRH0, DFSDCPY0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

---

**0265**

**Explanation:** An internal error has been detected by DFSIAF20. Upon entry to DFSIAF20 for sync point processing, the value contained in PSTSYNFC was not valid for DFSIAF20. This abend appears when one of the following conditions occurs:

- An interface error occurs in the SPOOL/API processing routines.
- A sync point PCB addressing error occurs in the SPOOL/API processing routines.

**Analysis:** 0265 is a standard abend issued by DFSIAF20. Use the program status word (PSW) at entry-to-abend to isolate the failing module. The PSW at entry-to-abend points to the instructions within the label ABEND265 from which the abend (SVC 13) is issued. The following Register can be used to isolate the invalid Function and caller.

Key	Label	Description
Reg10=function code	ABEND265	An invalid function code was passed to DFSIAF20. Register 14 in DFSIAF20's save area points to the calling module.

**Possible cause:** Check for a coding change or a user modification.

**System action:** The IMS application region terminates abnormally.

**Problem determination:** 1, 3, 4, 5, 8, 11, 27

**Source:** IMS abends

**Module:** DFSIAF20

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

**0271**

**Explanation:** An I/O error was detected while purging buffers during a checkpoint operation. GSAM is abnormally terminated because bad data records might be left in the data set.

**Analysis:** See Message DFS0530I for details about the data set.

**System action:** GSAM terminates abnormally.

**Problem determination:** 4, 35

**Source:** IMS abends

**Module:** DFSZDC00

**Related information:**

 DFS0530I

Chapter 2, "IMS problem determination," on page 13

---

**0272**

**Explanation:** After receiving an AF status code identifying an invalidly formatted BSAM variable-length record, an application program issued a call to a GSAM data set without reinitializing GSAM.

**Analysis:** Message DFS0768I was issued at the time the AF status code was returned identifying the ddname of the data set containing the invalid record. The GSAM control blocks and the buffer containing the invalid record were written to the IMSERR or SYSPRINT data set.

**System action:** GSAM terminates abnormally.

**Programmer response:** See message DFS0768I for details.

**Source:** IMS abends

**Module:** DFSZDC00

**Related information:**

 DFS0768I

## 0273

**Explanation:** An error was detected while repositioning a GSAM data set during an XRST call, or there is a GSAM DCB discrepancy (BLKSIZE/LRECL) between the original execution and the attempted restart execution. GSAM is terminated abnormally because the position in the affected data set is unpredictable.

**Analysis:** After printing the GSAM control block DSECTS, use the GSAM Control Blocks Dump on the IMSERR or SYSPRINT output to determine which GSAM PCB has a nonblank status code.

**System action:** GSAM abnormally terminated because position in the affected data set was unpredictable.

**Programmer response:** Print the GSAM control block DSECTS (member name IGLI in SDFSMAC) for use in analyzing the GSAM control blocks. Determine and correct the reason for the error, and rerun the failing job.

**Source:** IMS abends

**Module:** DFSZDC00

## 0274

**Explanation:** An unexpected interface error occurred between module DFSSBIO0 and its caller.

**Analysis:** 0274 is a standard abend issued by DFSSBIO0. At the request of IMS functions, DFSSBIO0 initiates read I/O operations for multiple consecutive blocks, and waits for the completion of the I/O operations. DFSSBIO0's caller must provide the address of an SRAN control block identifying the range of consecutive blocks in the I/O operation. The caller of DFSSBIO0 must be a module running under the PST that owns the SRAN control block involved in the I/O operation.

Key	Label	Description
Reg7=address of the SRAN Reg6=address of the SDSG owning the SRAN SDSGPSTA=address of the PST owning the SDSG/SRAN Reg10=address of the PST of the caller of DFSSBIO0 Reg14 (within the save area of the calling module)=return address of the caller	AB0274	DFSSBIO0 determines if the PST owning the SRAN and the SDSG involved in the call are equal to the PST of the module calling DFSSBIO0. (SDSGPSTA is compared with register 10.)

**System action:** IMS terminates abnormally.

**Programmer response:** See Analysis above for detailed diagnostic information.

**Source:** IMS abends

**Module:** DFSSBIO0

## 0275

**Explanation:** IMS was not able to acquire an area needed to create a SNAP memory dump of the Sequential Buffering (SB) control blocks and areas. Message DFS2350I is issued along with this abend.

**Analysis:** 0274 is a pseudoabend issued by module DFSSBSN0. DFSSBSN0 needs a work area to process SNAP requests from the calling IMS module. If a work area was not already acquired, DFSSBSN0 calls DFSSBGM0 to acquire a work area. If DFSSBGM0 fails to acquire a work area through an IMODULE/DFSQCSS macro, DFSSBSN0 requests an abend after regaining control from DFSSBGM0.

Key	Label	Description
Reg7=address of SBPST Reg15=0	DFSSBSN0	The SBPST contains call parameters for DFSSBSN0's call to DFSSBGM0. If DFSSBGM0 returns a zero in register 15 instead of a work area address, DFSSBSN0 requests an abend.

**System action:** The application program terminates abnormally.

**Programmer response:** Correct the error based on information in message DFS2350I.

**Source:** IMS abends

**Module:** DFSSBSN0

**Related information:**

 DFS2350I

## 0288

**Explanation:** A Fast Path secondary index database (FPSI DB) was used as input to either the HISAM Reorganization unload or reload utility (DFSURUL0 or DFSURRL0). These utilities do not support Fast Path secondary index databases. See message DFS0288E for additional information.

**Analysis:** Abend 0288 with a reason code of X'00000008' in REG15 is a standard abend issued from module DFSDLBL0. Message, DFS0288E, accompanies this termination.

**System action:** The batch job terminates abnormally.

**Programmer response:** Use an appropriate utility to reorganize or recover Fast Path secondary index databases. To reorganize an FPSI DB, an IDCAMS REPRO is sufficient. To recover an FPSI DB, use either the IMS Recovery Utility, DFSURDB0, or an index builder tool of any IMS vendor product.

**Source:** IMS abends

**Module:** DFSDLBL0

**Related information:**

 DFS0288E (Messages and Codes)

## 0300

**Explanation:** While running under control of the utility control facility (UCF), a message was produced for which the user requested an abend to occur.

This abend is issued in the UCF by module DFSUCP40, only when requested by the user as a diagnostic aid. A dump is produced if a SYSABEND or SYSUDUMP DD statement was included in the procedure for DFSUCF00.

**Analysis:** DFSUCP40 issues a 0300 only while running under UCF and after writing a diagnostic message. The error-point abend is invoked in one of two ways.

- REQUEST=MSGALL was coded on the FUNCTION=OP control statement or entered as a response to the UCF outstanding WTOR, requesting an abend on any **A** or **W** type message.
- MSGNUM=(CCC,...) was coded on the FUNCTION=OP control statement, requesting an abend on any one or more specific message.

Analyzing the dump output, it should be possible to back up to the module that called DFSUCP40. Once there, the cause and validity of a given message can be determined. It should be noted that, while the majority of messages capable of being trapped with a 0300 are initiated from the various IMS utilities, some are issued internally from UCF. Also note, only those messages contained in DFSUCP80 can be trapped. The utilities run as a subtask of UCF. The "bootstrap" module, DFSUCPB0, is attached and it, in turn, loads the requested utility module.

Using the registers in the abend SVRB, locate register 13. This register contains a pointer to the caller's save area data set. Registers 14 through 12 are save at register 13 plus C. The following registers pertain to the caller's save area data set.

Register 15 is a pointer to the entry of DFSUCP40. Check at this address plus 5 for this same literal, which verifies you have the correct save area data set. Register 14 is a BAL and points back to the caller's return address. This calling module issued the diagnostic message. Register 12 is the caller's base register. Register 1 is a pointer to a parameter list to be passed to DFSUCP40. The parameter list is a doubleword in length; the first word contains a 1-byte flag that defines the message number and a 3-byte address of a parameter merge list. The second word is the pointer to the UCFCMVEC CSECT. This CSECT is defined in module DFSUCP70 and is effectively one large control block used by the UCF modules as a common work area or common vector area.

**Possible cause:** Because there are more than 100 messages that can be trapped with a 0300, it would not be feasible to cover them all in detail here. Generally, there are two causes for a failure:

- A user error; probably a missing or invalid UCF control statement, missing DD statement, or other error.
- A UCF or utility software error. While analyzing the reason for a particular message, you may determine a software error. It is possible that all the required documentation will be in the abend 0300 memory dump. However, UCF builds any SYSIN required by the various utilities in the data set described by the DFSCNTRL DD statement. This is done by one of two function interface modules: DFSUCP60 or DFSUCPA0. These two modules also issue the attach macro that gets the utility initiated. Program check traps may be necessary at the attach point for further diagnosis, or APAR documentation, if a bad SYSIN data set is suspected. Also, include a printout of the DFSCNTRL data set.

**System action:** The Utility Control Facility terminates, and a storage dump is produced on the SYSUDUMP or SYSABEND data set.

**Programmer response:** Verify that the objective of the dump has been achieved. This dump is a diagnostic aid.

**Source:** IMS abends

**Module:** DFSUCF00, DFSUCP40



## Chapter 11. IMS abend codes 0301 - 0350

This topic identifies and explains the following abend codes issued by IMS. Each code is 4 decimal digits.

### 0302

**Explanation:** During the execution of the Database Data Set Recovery utility, DFSURDB0, an unrecoverable error occurred.

**Analysis:** 0302 is a standard abend issued from the DL/I utility for Database/Data Set Recovery Utility, DFSURDB0. The registers in the abend SVRB aid in problem isolation.

In almost all cases for a 0302 abend, an IMS message is issued before the abend. That message can be found in the program output, and on the system console, if the user has so specified.

This abend can occur only if the user's SYSIN stream contains an abend option, and if the ABORT error switch has been set by an internal subroutine within the recovery module. The abend is primarily diagnostic, and is issued as a result of a branch and link (BAL) to a routine labeled ABNDTST. The program status word (PSW) at entry-to-abend points to the area within label ABNDTST from which the abend (SVC 13) is issued.

Key	Label	Description
Reg1=Completion code Reg10=BAL to Message Generator	RDBSCANC #@LB92	This subroutine reads a control statement and checks its format. A compare is also made to see if the first character in column 1 is an 'S' to indicate a control statement. If not, message DFS302A is written and the ABORT error switch is set. When this switch is on, the routine within label TCBFSA BALs to ABNDTST, which causes this abend to be issued.
Reg1=Completion code Reg10=BAL to Message Generator	RDBSCANC #@LB93	A compare is made to columns 4 through 11 of the control statement for the database description (DBD) name. If blank, message DFS304A is issued and the ABORT error switch is set. Because this switch is on, the routine within label TCBFSA BALs to ABNDTST to issue the abend.
Reg1=Completion code Reg10=BAL to Message Generator	RDBSCANC #@LB95	This compare is to columns 13 through 20 of the control statement for the ddname of the database. If blank, message DFS307A is issued and the ABORT error switch is set. Because this switch is on, the routine within label TCBFSA BALs to ABNDTST to issue the abend.
Reg1=Completion code Reg2=address of DBD Prefix Reg10=BAL to Message Generator	RDBDBDLD #@120	This subroutine opens DBDLIB to locate the data base to be recovered. The access method field of the DBD prefix is compared for a value less than X'0F'. If the value is greater, it is an indication that this database has an unknown organization, and message DFS316A is issued. The ABORT error switch is set, and the routine within TCBFSA BALs to ABNDTST to abend.
Reg1=Completion code Reg2=address of DBD Prefix Reg10=BAL to Message Generator	RDBDBDLD #@LB114	This subroutine is branched to if the access method is not VSAM. A compare is made for a value not greater than X'07'. If the value is greater, but not VSAM, there is an error, and message DFS316A is issued. The ABORT error switch is set; because of this, the routine within label TCBFSA BALs to ABNDTST to abend.

Key	Label	Description
Reg1=Completion code Reg10=BAL to Message Generator	RDBDBDL #@LB134 CASE3	The ddname in the control statement in the DBD 'dbdname' field could not be located. Message DFS306A is issued and the ABORT error switch is set. Because of this, the routine within label TCBFSA BALs to ABNDTST to issue the abend.
Reg1=Completion code Reg10=BAL to Message Generator	RDBDBDL #@LB112	The DBD name specified in the control statement, columns 4 through 11, could not be found in DBDLIB. Message DFS305A is issued and the ABORT error switch is set. The routine within label TCBFSA BALs to ABNDTST to issue the abend.
Reg1=Completion code Reg5=DCB address Reg10=BAL to Message Generator	RDBDBDL #@LB110	The field DCBOFLGS is tested to see if the OPEN of DBDLIB was successful. If not, Message DFS301A is issued and the ABORT error switch is set, causing the routine within label TCBFSA to BAL to ABNDTST to abend.
Reg1=Completion code Reg10=BAL to Message Generator	RDBDUOPN #@LB188	A return code of 'X04' was received from the DEVTYPE macro expansion in routine at IDUCSTRT, indicating a device type failure, no ddname. Message DFS315A is issued and the ABORT error switch is set. When the routine IDUCSTRT interrogates this switch and finds it on, it BALs to ABNDTST to issue the abend.
Reg1=Completion code Reg5=CUM DCB Address Reg10=BAL to Message Generator	RDBCUPN	The DCBOFLGS field of the CUM DCB is tested for a successful OPEN. If unsuccessful, a branch is taken to NOSYSIN to output message DFS301A and set the ABORT error switch. The routine within COPNSTRT BALs to ABNDTST to issue the abend.
Reg1=Completion code Reg10=BAL to Message Generator	RBDUHDR (BADDMPID)	The routine at EODUMP tested for a dump header and did not find one. Messages DFS312A is issued and the ABORT error switch is set. When the routine at IDUCSTRT interrogates this switch and finds it on, it BALs to ABNDTST to issue the abend.
Reg1=Completion code Reg10=BAL to Message Generator	RBDUHDR #@LB208	A compare indicated that the input data set was for the proper DBD, but not for the proper data set. Message DFS317W is issued. The ABORT error switch is set and the routine within IDUCSTRT BALs to ABNDTST to issue the abend.
Reg1=Completion code Reg7=address of CUMHDR DSECT Reg10=BAL to Message Generator	RDBNDXSP  (DXSPSTRT)	The CUMHDR flag was tested, and it was found that the data set description did not match the DBD description. Message DFS329A is issued and the ABORT error switch is set. The routine within IDUCSTRT BALs to ABNDTST to issue the abend.
Reg1=Completion code Reg5=DCB address (of data set to be recovered) Reg10=BAL to Message Generator	RDBRVOPN (VOPNSTRT)	The DCBOFLGS field of the DCB for the data set to be recovered is tested for a successful OPEN. If unsuccessful, a branch is taken to NOSYSIN to output message DFS301A and set the ABORT error switch. The routine within IDUCSTRT BALs to ABNDTST to issue the abend.

Key	Label	Description
Reg1=Completion code Reg7=address of CUMHDR DSECT Reg10=BAL to Message Generator	RDBDUCIN (CRDSET) #@LB360	A compare is made between the CUM DBDname and the DBD name. If unequal, a missing or invalid header condition is indicated, and Message DFS356A is issued. The ABORT error switch is set and the routine within MRGWSTRT at MERGEI BALs to ABNDTST to issue the abend.
Reg1=Completion code Reg2=Relative key position from start of record Reg7=address of CUM record Reg10=BAL to Message Generator	RDBDUCQK (UCQKSTRT) #@LB377	The key-sequenced data set (KSDS) prime key is compared with the relative position of the key from the start of the CUM record. If unequal, message DFS330A is issued with a reason code ('1') and the ABORT error switch is set. The routine within MRGWSTRT BALs to ABNDTST to issue the abend.
Reg1=Completion code Reg2=Relative RBN from start of record Reg7=address of CUM record Reg10=BAL to Message Generator	RDBDUCQK #@LB390	The OSAM or ESDS RBN is compared with the CUM RBN and found to be unequal. Message DFS330A is issued with a reason code ('2'), and the ABORT error switch is set. The routine within MRGWSTRT BALs to ABNDTST to issue the abend.
Reg1=Completion code Reg9=Pointer to RBA Reg10=BAL to Message Generator	RDBDUCOT (RBAERROR) #@LB468	A compare has found that the RBN of the OSAM record from the ACCUM input data set was beyond the end of the data set. Message DFS332A is issued and the ABORT error switch is set. The routine within MRGWSTRT BALs to ABNDTST to issue the abend.
Reg1=Completion code Reg5=DCB Address Reg10=BAL to Message Generator	RDBNDXLG #@LB509	The DCBOFLGS field is tested to see if the OPEN for log input DCB was successful. If not, a branch is taken to the routine at NOSYSIN, message DFS301A is issued, and the ABORT error switch is set. A BAL is taken to ABNDTST to issue the abend.
Reg1=Completion code Reg10=BAL to Message Generator	RDBNDXLG #@LB515	A compare after the DEVTYPE macro indicated that the log input data set was a dummy file. Because this routine is for log only recovery, DD DUMMY is not allowed, and message DFS324W is issued. The ABORT error switch is set and a BAL is taken to ABNDTST to issue the abend.
Reg1=Completion code Reg6=DSG Address Reg7=DCB Address Reg10=BAL to Message Generator	RDBLDCBO (CKBADCB)	After checking all data set groups (DSGs), the ddname for the DBD could not be located. Message DFS306A is issued and the ABORT error switch is set. When this subroutine returns to RDBNDXLG, a BAL is taken to ABNDTST to issue the abend.

Key	Label	Description
Reg1=Completion code Reg6=Log record address Reg7=Pointer to DBLOG Reg10=BAL to Message Generator	RDBLGRDL #@LB621 and #@LB631	A compare is made between the image log date and the current date, and they are unequal. Message DFS330A is issued with a reason code ('3'), and the ABORT error switch is set. When the routine at RDBLGMRG interrogates this switch and finds it on, it BALs to ABNDTST to issue the abend.
Reg1=Completion code Reg5=Current OSAM RRN Reg6=Highest OSAM RBN encountered Reg10=BAL to Message Generator	RDBDBRDC #@LB679	A compare is made between the current RBN for the OSAM record for the log input data set, and the highest possible RBN. If the compare is unequal, message DFS332A is issued, the ABORT error switch is set, and when the routine at RDBLGMRG regains control, it BALs to ABNDTST to issue the abend.
Reg1=Completion code Reg10=BAL to Message Generator Reg15=Return code from IEHATLAS utility	RDBTRCV0 (IHB423A)	The recovery routine, having linked to the IEHATLAS utility to attempt track recovery, received a return code indicating an alternate track could not be assigned. The ABORT error switch is set, and a BAL is taken to ABNDTST to issue the abend.

**System action:** IMS terminates abnormally with a memory dump.

**Programmer response:** No action is required.

**Problem determination:** 3, 4, 35

**Source:** IMS abends

**Module:** DFSURDB0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

### 0303

**Explanation:** The parameter list constructed from the OPTION statements has been modified (see the IMS system definition information about specifying the IMS buffer pools). An error might have occurred in the user exit routine, which caused the routine to address and modify storage outside its legitimate address space.

**Analysis:** 0303 is a standard abend issued by the format print utility. The abend supervisor request block (SVRB) registers aids in problem isolation. The program status word (PSW) at entry-to-abend points to label SEL04, from which abend (SVC 13) is issued.

Message DFS706I, is also issued to the SYSPRINT data set.

Register 6 contains a pointer to the first list element.

Register 1 contains a pointer to a 2-word parameter list. The first word points to the candidate record; the second word (with the high-order bit on) points to the SYSPRINT data set DCB.

Register 15 contains:

- The exit routine entry address (if the user specified an EXITR=parameter in the OPTION statement); or
- Zero (if no user exit routine is specified, in which case IEFBR14 clears the register); or
- A return code.

Key	Label	Description
Reg6=address of first list element	SEL08	A test is made for the END option. If the condition is not satisfied, a branch is taken to the routine at label SEL04 to abend.
Reg6=address of first list element	SEL10	A test is made for the OFFSET parameter and the VALUE parameter of the OPTION control statement. If neither condition is satisfied, an invalid list element has been detected, and a branch is taken to the routine at label SEL04 to abend.
Reg3=list length Reg6=address of first list element	SEL20	A test is made to see if VALUE and LENGTH were specified on the OPTION control statement. Then a test of register 3 is made for the length of the list. If any or all these tests fail, a branch is taken to the routine at label SEL04 to abend.

**System action:** The program terminates abnormally.

**Programmer response:** Correct any indicated errors, and rerun the job.

**Problem determination:** 3, 4, 35

**Source:** IMS abends

**Module:** DFSERA10

**Related information:**

 DFS706I (Messages and Codes)

Chapter 2, "IMS problem determination," on page 13

## 0305

**Explanation:** Reading a VBS sequential data set, an incomplete spanned record was detected. Either a continuation segment was encountered without being preceded by a starting segment, or an end-of-file condition or a new segment was encountered after a starting segment without an ending segment.

**Analysis:** 0305 only occurs for read requests. This abend is issued by DFSURIO0 when an invalid spanned record is detected. DFSURIO0 is called by one of the following modules:

- DFSURDB0 (recovery), to read a Change Accumulation data set or log records to the Change Accumulation data set.
- DFSUC150 (Change Accumulation), to read and write type X'24' log records to the Change Accumulation data set
- DFSUC350 (Change Accumulation), to read and write records to the Change Accumulation data set

When the abend occurs, register 13 in the abend SVRB points to the save area for DFSURIO0. Offset 4 in the DFSURIO0 save area points to the caller's save area. Register 0 and register 1 contain the parameters passed by the caller, and register 14 contains the return address to the caller.

Key	Label	Description
Reg2=address of input record Reg9=address of DCB	GETLOG	If offset X'02' in the input record equals X'02' (middle of a spanned record) or X'03' (end of a spanned record) and the first segment has not been read, branch to BADSPAN.
Reg9=address of DCB	UC249	
	UC237	If end-of-data is detected before the last segment of a spanned record is read, branch to BADSPAN.

**Possible cause:** This abend might be caused by the following:

1. Log or Change Accumulation is out of sequence or a data set is missing.
2. A system crash occurred, during which the system could not terminate the log correctly.

**System action:** IMS terminates abnormally.

**Programmer response:** Correct the problem, and rerun the job.

## 0306 • 0310

**Problem determination:** 1, 2, 3, 4

**Source:** IMS abends

**Module:** DFSURIO0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

### 0306

**Explanation:** An error was detected by Change Accumulation utility while trying to build the output change accumulation record. The length of the new Change Accumulation output record exceeds output buffer size.

**Analysis:** This abend is issued by module DFSUC350 when it detects that the length of data represented by an entry in the offset/length table exceeds the current maximum database block size of 32760. The abend supervisor request block (SVRB) contains the registers from DFSUC350. The program status word (PSW) at entry-to-abend points to the instruction from which the abend (SVC 13) is issued. The routine that issued the abend is branched to by the routine that detected the error.

Key	Label	Description
Reg3=data length and is > 32K	MERGP2	The data length of a spill record in an offset/length table entry exceeds 32K.
Reg5=data length and is > 32K (Reg3=1)	MORECUM	The data length of a detail record in an offset/length table entry exceeds 32K.
Reg7=address of table entry		
Reg8=address of output record		

**Possible cause:** Invalid data sets used as input to the Change Accumulation utility, or an error occurred in the Change Accumulation utility.

**System action:** Change Accumulation terminates abnormally with a memory dump.

**Programmer response:** Correct any errors in the input to Change Accumulation, and rerun the job.

**Problem determination:** 3, 35, and copies of the input Log and Change Accumulation data sets.

**Source:** IMS abends

**Module:** DFSUC350

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

### 0307

**Explanation:** Batch backout was not run with the same GSG name used when the log records were created.

**System action:** Batch backout abnormally terminates.

**System programmer response:** Rerun batch backout with the correct GSG name.

**Source:** IMS abends

**Module:** DFSBCKI0

---

### 0310

**Explanation:** The partial database reorganization step terminates abnormally because of the condition indicated in the previously issued error message.

**Analysis:** 0310 is a standard abend issued by module DFSPRERR, the PDBR error-message generator. An error message in the range of DFS3000 through DFS3099W always accompanies this abend. Register 5 contains the address of the PDBR common area (COMAREA), which contains addresses of other PDBR blocks.

**System action:** The step is abnormally terminated.

**Programmer response:** See output of PDBR step for the error message that describes this problem.

**Problem determination:** 4, 35

**Source:** IMS abends

**Module:** DFSPRERR

**Related information:**

Chapter 2, "IMS problem determination," on page 13

### 0311

**Explanation:** Surveyor detected an error.

**Analysis:** This is a standard abend issued by module DFSPRSER, the Surveyor error-message generator. An error message in the range of DFS3000 through DFS3099W always accompanies this abend. Register 5 contains the address of the Surveyor common area (CMAREA), which contains addresses of other Surveyor blocks.

**System action:** The step is abnormally terminated.

**Programmer response:** See output of Surveyor for the error message that describes this problem.

**Source:** IMS abends

**Module:** DFSPRSER

### 0315

**Explanation:** IMS received a nonzero return code from an IMSAUTH SVC function.

**Analysis:** Register 15 contains the IMSAUTH return code. For a description of these return codes, see the IMS system services return codes information.

**System action:** IMS terminates abnormally with a memory dump.


**Programmer response:** Take appropriate action according to the codes in register 15.

**Problem determination:** 4, 11, 35

**Source:** IMS abends

**Module:** DFSIRST0, DFSRLP00, DFSXSTM0

**Related reference:**

 IMS system services return codes (Messages and Codes)

**Related information:**

Chapter 2, "IMS problem determination," on page 13

### 0316

**Explanation:** An unexpected or unrecoverable error was detected during coupling facility services initialization or subsequent execution. Register 15 contains the abend reason code.

**Analysis: For DFSDMAW0:**

At the time of the abend, register 15 contained the reason code for the error.

Key	Label	Description
Reg15=X'10'		Caller did not provide SCD or PST.
Reg15=X'11'		IOSB integrity error.
Reg15=X'20'		A batch PST serialization problem was detected at the module exit routine.
Reg15=X'30'		An internal logic error (invalid PC) occurred.
Reg15=X'40'		Caller provided an invalid batch SCD.
Reg15=X'50'		An internal logic error (invalid PT) occurred.

Key	Label	Description
Reg15=X'60'		A batch PST serialization problem was detected before the call was processed.
Reg15=X'70'		IMODULE GETSTOR failed for key 7 PST.
Reg15=X'80'		A batch PST serialization problem was detected after the "dummy" key 7 PST was obtained.
Reg15=X'90'		An unexpected z/OS return code was returned during connection to an OSAM structure.
Reg15=X'A0'		An unexpected z/OS return code was returned during connection to a VSAM structure.

**For DFSDCFC0:**

At the time of the abend, register 15 contained the reason code for the error.

Key	Label	Description
Reg15=X'11'		IOSB integrity error.

**System action:** IMS terminates abnormally.

**Programmer response:** See Analysis above to determine the cause of the problem.

**Problem determination:** 1, 4, 35

**Source:** IMS abends

**Module:** DFSDMAW0, DFSDCFC0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

**0317**

**Explanation:** A coupling facility failure was detected on the XRF alternate subsystem. The failure could be one of the following:

- Connection failure
- Structure rebuild failure

**Analysis:** 0317 is issued by the GOSTPDS routine in DFSDCFR0. Error messages issued before this abend indicate the reason for the failure.

**System action:** IMS terminates abnormally.

**Programmer response:** If the failure was a connection failure, restart the alternate subsystem after its connection to the coupling facility is reestablished.

**Problem determination:** 1, 4, 35

**Source:** IMS abends

**Module:** DFSDCFR0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

**0318**

**Explanation:** Module DFSDVBI0 determined that the coupling facility services function cannot be initialized. To use this function, MVS release level 510 or higher is required. If a VSAM structure is to be used, DFSMS release level 120 or higher is required.

**Analysis:** Register 15 contains the abend reason code.



Key	Label	Description
Reg15=X'01'	ABEND318	Either OSAM, VSAM, or both structure names were specified on CFNAMES control statement, but the required z/OS release level (510 or above) was not used.
Reg15=X'02'	ABEND318	VSAM structure name was specified on CFNAMES control statement, but the required DFSMS release level (120 or above) was not used.

**System action:** IMS terminates abnormally.

**Programmer response:** Verify the z/OS release level, the DFSMS release level, or both.

**Source:** IMS abends

**Module:** DFSDVBI0

### 0319

**Explanation:** Module DFSDVBI0 determined that coupling facility services initialization failed in a batch environment. Message DFS3381E is issued; the DATA part of the message identifies the reason for failure.

**System action:** IMS terminates abnormally. One or more messages are issued identifying the reason for failure.

**Programmer response:** See message DFS3381E.

**Problem determination:** 1, 4

**Source:** IMS abends

**Related information:**

 DFS3381E

Chapter 2, "IMS problem determination," on page 13

### 0320

**Explanation:** A CFNAMES control statement was specified to request use of the coupling facility; however, an IRLM at a release level that does not include coupling facility support was used. To use coupling facility services, IRLM version 2 or higher must be used.

**System action:** IMS terminates abnormally.

**Programmer response:** Verify the IRLM release level used.

**Source:** IMS abends

### 0322

**Explanation:** An unexpected or unrecoverable error occurred in OSAM sequential buffering during execution for coupling facility services. This is an IMS system error.

**Analysis:** Sequential buffering control blocks should be freed when the program terminates. An error was detected during validation of reserved local cache entry (LCE) bits that are held by the transaction; the error occurred before these bits were freed. The program status word (PSW) at entry-to-abend points to the instruction (in routine DFSSBTD1 at label LCERET) from which the abend (SVC 13) is issued.

**System action:** IMS terminates abnormally.

**Problem determination:** 1, 4, 35

**Source:** IMS abends

**Module:** DFSSBTD0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

## 0347

**Explanation:** A severe error condition was detected during HD Reorganization Unload, and a SYSUDUMP or SYSABEND DD statement was provided. The error was caused by one of the following:

- An internal system error
- A missing DD statement
- A missing data set

All HD Reorganization Unload and Reload operations now include PHDAM and PHIDAM databases.

**Analysis:** 0347 is a standard abend issued during execution of the HDAM Reorganization Unload utility, DFSURGU0. The program status word (PSW) at entry-to-abend and the registers in the abend supervisor request block (SVRB) aids in problem isolation.

All abends from module DFSURGU0 result from a conditional branch to either label BADRUN, STOPRUN, or both. After some housekeeping and testing, a branch is ultimately taken to the routine at label ABND, which issues abend 0347. In most cases, an IMS error message has been issued to the IMS master console prior to abend and should be used to isolate the problem to a particular label. The following table shows the messages and explanations:

Key	Label	Description
Message DFS315A	TSTRET2	The DEVTYPE macro is issued from the routine at label NXTSDB. A return code of X'08' indicates that the macro failed, and a branch is taken to BADRUN to handle the system failure.
Message DFS344W	NODD1	A DD statement for primary output was not supplied, and a branch is taken to BADRUN.
Message DFS343W	CHKDUMMY	After the DEVTYPE macro is issued (in routine at label NXTSDB), a check is made of the return code to see if DD DUMMY was specified. If yes, a branch is taken to BADRUN to handle the abend.
Message DFS318A	GETBLKSI	A compare is made to see if LRECL is larger than the BLKSIZE. If so, a branch is taken to BADRUN to handle the abend.
Message DFS301A	OPEN1OPT	The DCBOFLGS field is tested to see if OPEN was successful for the output DCB. If not, a branch is taken to BADRUN to handle the abend.
Message DFS318A	GETBLKSZ	Same as for routine at label GETBLKSI above, except that the test is for the second DD statement.
Message DFS348A	RECREATE	The DBPCBSTC field of the PCB is tested for a valid return code (in this case, GA, GB, GK, or blank is valid). If not a valid return code, a branch is taken to BADRUN to handle the abend.
	STPTEST2	The field UCFCMSTP is tested to see if STOP has been requested. If yes, a checkpoint record is written out, and a branch is taken to BADRUN to handle the abend.
Message DFS301A	SETUP1	Same as for routine at label OPEN1OPT above, except that the test for successful OPEN is made against the CHECKPOINT (input) DCB.
Message DFS377A	RESTFAIL	Reg15 is tested to see if restart completed successfully. A return code of '4' or '8' indicates failure, and a branch is taken to BADRUN to handle the abend.
Message DFS301A	OPINERR	Same as for OPEN1OPT, except that the test for successful OPEN is made against the DCB for edit and output statistics tables.
Message DFS319A	IOERRIN	A permanent I/O error was encountered while outputting a record or a statistics table entry to a data set, and a branch is taken to BADRUN to handle the abend.
Message DFS377A	EOSYSIN	The input for the CHECKPOINT data set was found to be invalid, and a branch is taken to BADRUN to handle the abend.
Message DFS388A	NOTTAB	No statistics table record was found from UNLOAD, and a branch is taken to BADRUN to handle the abend.

Key	Label	Description
Message DFS346A	ERRNOALT	A volume error was encountered on the primary output data set, and no alternate output data set was available. A branch is taken to BADRUN to handle the abend.
Message DFS301A	NONUCF	During execution of an OPEN macro, it was discovered that no SYSOUT data set was available. The address of the WTO area is loaded into register 1 to inform the IMS master console operator of this fact, and a branch is taken to STOPRUN to handle the abend.

**Possible cause:** Either an internal system error caused this abend, or a missing DD statement or data set was detected. Check the IMS Master Console Log Sheets for all messages prior to the abend, and check the JCL being used for this utility.

**System action:** System terminates abnormally.

**Programmer response:** Check all preceding messages for clues to what caused the dump. Correct any errors found, and rerun the job.

**Problem determination:** 1, 2, 3, 4, 17d, 20, 35

**Source:** IMS abends

**Module:** DFSURGU0

**Related information:**

Chapter 2, "IMS problem determination," on page 13



---

## Chapter 12. IMS abend codes 0351 - 0400

This topic identifies and explains the following abend codes issued by IMS. Each code is 4 decimal digits.

---

### 0355

**Explanation:** A severe error condition was detected during HD Reorganization Reload, and a SYSUDUMP or SYSABEND DD statement was provided. This problem is possibly an internal system error.

All HD Reorganization Unload and Reload operations now include PHDAM and PHIDAM databases.

**Analysis:** 0355 is a standard abend issued during execution of the HDAM Reorganization Reload utility, DFSURGL0. The registers in the abend SVRB and the program status word (PSW) at entry-to-abend aids in problem isolation.

In addition, the console log sheets from the IMS master console should be examined for all IMS action or information messages sent to the user prior to the abend. The IMS message number aids in isolating the routine that detected the error condition.

Input to DFSURGL0 is the data set created by the HDAM Reorganization Unload utility, DFSURGU0. Condition codes and messages from unload should be carefully checked if a 0355 abend occurs. The following table shows the messages and explanations:

Key	Label	Description
Message DFS301A	ENQOK	After execution of the OPEN macro, the DCBOFLGS field for the input DCB created by the HDAM Reorganization Unload utility is tested to see if OPEN was successful. If not, a branch is taken to the routine labeled BADRUN to handle the abend.
Message DFS348A	STATUX	The status/return code field (DBPCBSTC) received an invalid return code from a database 'ASRT' call. The failing status code is propagated to the error message, the abend SW field of this utility is updated to indicate an error condition (X'FF'), and a branch is taken to the routine labeled BADRUN to handle the abend.
Message DFS301A	STATAI	A data management OPEN error was encountered, as indicated by the AI status code returned to the DBPCBSTC field of the PCB. The segment name in the DBPCBSFD field of the PCB can indicate the error condition. A branch is taken to the routine labeled BADRUN to handle the abend.
Message DFS386A	EOD1	The HDAM Reorganization Reload utility has encountered an EOF condition before the last status record is read. A branch is taken to the routine labeled BADRUN to handle the abend.
Message DFS331A	TABOK	The input data set passed by the HDAM Reorganization Unload utility was empty, or an immediate EOF condition was encountered on the read. A branch is taken to the routine at label BADRUN to handle the abend.
Message DFS388A	NOTTAB	While processing an input DCB, the HDAM Reorganization Reload utility expected to find a statistics record and did not. A branch is taken to the routine at label BADRUN to handle the abend.
Unload statistics segment count is not equal to reload segment count. Return code equals 8.	LESS	The unload statistics record segment count does not match the reload segment count.
Message DFS358A	MYLOOP1	Segment name found in unload segment record was not found in the database segment table.

**System action:** IMS terminates abnormally.

**Programmer response:** Check all preceding messages for clues to what caused the dump. Correct any errors found, and rerun the job.

**Problem determination:** 1, 2, 3, 4, 17d, 20, 35

**Source:** IMS abends

**Module:** DFSURGL0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

### 0359

**Explanation:** The abend option was selected for the HISAM Reorganization Unload or Reload utility and a severe error occurred.

**Analysis:** 0359 is a standard abend issued during execution of the HISAM Reorganization Unload utility, DFSURUL0. The program status word (PSW) at entry-to-abend points to label TRMOPBAD, which issues the abend (SVC 13).

This abend is a result of the user having coded an OPTIONS=ABEND statement in the JCL for this utility or the HISAM Reorganization Reload utility. During execution, any serious error causes a 0359 abend. Generally, there is an IMS message corresponding to this abend. This message can be found in the program output.

The error message and its explanation can probably provide enough information to identify the problem. In addition, you can locate the subroutine in DFSURUL0 that issues the error message by using the cross-reference table in the program listing. The work registers of the subroutine are saved in the save area labeled LINKSAVE. Registers 14, 15, and 0 through 7 of the routine in process are saved in the location pointed to by LINKPTR.

Register 12 is the base register for all subroutines. The base registers for the mainline processing routine are registers 10, 11, and 12.

**System action:** The utility terminates abnormally.

**Programmer response:** Check all preceding messages for clues to what caused the dump. Correct any errors found, and rerun the job.

**Problem determination:** 1, 2, 3, 4, 17d, 20, 35

**Source:** IMS abends

**Module:** DFSURUL0, DFSURRL0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

### 0360

**Explanation:** Transport Manager Subsystem (TMS) was not running in a supervisor state or a send filter routine failed.

#### Reason code

##### Explanation

**01** TMS was not running in protect key seven (supervisor state). Make sure you have specified KEY(7) in the parameter field of the transport manager EXEC statement.

**02** A send filter exit routine failure has been detected. Recovery is attempted. If successful, filtering is suspended for the conversation.

**Source:** IMS abends

**Module:** ELXINIT0, ELXUISE0

---

**0361**

**Explanation:** A logic error in the Transport Manager Subsystem (TMS) was encountered.

**Reason code****Explanation**

- 01** Internal logic error: the code in user address space does not match the code in TMS address space.  
Module: ELXUTM00
- 02** Internal logic error: invalid vector value in transport manager exit scheduler.  
Module: ELXUSX10
- 03** Internal logic error: an invalid vector value in TMS service routines was detected.  
Module: ELXUS000
- 04** Internal logic error: an invalid vector value in TMS VTAM allocation 1 routine was detected.  
Module: ELXUVA10
- 05** Internal logic error: a send queue Should Not Occur condition in transport manager send support occurred.  
Module: ELXUVS10
- 06** Internal logic error: unsupported what-received status or data length was detected. A recovery is attempted after an SDUMP; the conversation is deallocated. Register 3 at entry to abend points to the RPL.  
Module: ELXUVXR0
- 07** Internal logic error: unsupported RPL6REQ/RPL6QUAL values in a completed RPL were detected. Recovery is attempted after an SDUMP. Register 3 at entry to abend points to the RPL. Any conversation associated with the RPL is likely to be inoperative.  
Module: ELXUVXR0
- 08** RESMGR DELETE produced an error return code (saved in register 2). An SDUMP is created and recovery continues processing.  
Module: ELXUIFO0
- 09** Internal logic error: an invalid AFR function was received.  
Module: ELXUSX20
- 10** Internal logic error: an invalid status exit routine reason code was detected.  
Module: ELXTITX0

**Source:** IMS abends

**Module:** ELXUTM00, ELXUSX10, ELXUS000, ELXUVA10, ELXUVS10, ELXUVXR0, ELXUIFO0, ELXUVXR0, ELXUSX20, ELXTITX0

---

**0362**

**Explanation:** Termination of the Transport Manager Subsystem (TMS) was requested by the operator.

**Reason code****Explanation**

- 01** The ABDUMP command was invoked to create the system dump. The transport manager is abnormally terminated.

**Source:** IMS abends

**Module:** ELXCABN0

---

**0363**

**Explanation:** An unrecoverable error was detected by database tracking logic. Reason codes further identify the type of unrecoverable error:

**Reason code****Explanation**

- 01** During initialization, an error return code was returned by the MVS DSPSERV/ALESERV macro while

attempting to create a data space. The return code from R15 has been saved in R2, the reason code from R0 has been saved in R4.

Module: DFSDT100, DFSDT320

- 03 During initialization, an error return code was returned from IMODULE GETMAIN for L0C=(any,any) private area storage in the DLT region, for the tracking hash table. The return code from R15 has been saved in R2.  
Module: DFSDT150
- 04 During initialization, an error return code was returned from DFSDSM indicating that the tracking data space does not have storage for basic system areas. The return code from R15 has been saved in R2.  
Module: DFSDT100, DFSDT320
- 07 During initialization, an error return code was returned from IMODULE LOAD indicating that a required module could not be loaded. The return code from R15 has been saved in R2.  
Module: DFSSDT20
- 09 During initialization, an error return code was returned from DFSCWU indicating that milestone PST ITASKs could not be created. The return code from R15 has been saved in R2.  
Module: DFSSDT20
- 10 An error return code was returned from DFSBCB GET indicating that AWE could not be created during initialization, milestone/quiesce processing, database status change processing, or abrupt-end-tracking processing. The return code from R15 has been saved in R2.  
Module: DFSDT100, DFSDT160, DFSDT180, DFSDT330
- 11 During initialization, an error return code was returned from MVS TCBTOKEN request. The return code from R15 has been saved in R2.  
Module: DFSDT100
- 12 During initialization, an error return code was returned from MVS SETLOCK request. The return code from R15 has been saved in R2.  
Module: DFSDT100
- 13 During initialization or SAS AWE queue processing, an error return code was returned from DFSBCB GET indicating that storage could not be obtained for a control block or work area. The return code from R15 has been saved in R2.  
Module: DFSDT150, DFSDT300
- 14 During tracking redo processing, an error return code was returned from IMODULE GETMAIN for L0C=(any,any) private area storage in the DLT region, indicating that storage could not be obtained for control blocks needed. The return code from R15 has been saved in R2.  
Module: DFSDT400
- 15 An error return code was returned from IMODULE GETSTOR, indicating that storage could not be obtained by the tracking CTL AWE processor for an AWE. The return code from R15 has been saved in R2.  
Module: DFSDT240
- 16 During initialization, an error return code was returned from IMODULE GETMAIN indicating that storage could not be obtained in subpool 231 for the DL/I database tracking trace table. The return code from R15 has been saved in R2.  
Module: DFSDT100
- 19 During initialization, an error return code was returned from IMODULE GETMAIN for L0C=(any,any) private area storage, indicating that storage could not be obtained for control blocks needed. The return code from R15 has been saved in R2.  
Module: DFSSDT20

**System action:** IMS terminates abnormally with a memory dump.

**System programmer response:** Take appropriate action based on the return code in register 2.

**Source:** IMS abends

**Module:** DFSDT100, DFSDT240, DFSDT300, DFSDT150, DFSDT100, DFSDT320, DFSSDT20, DFSDT100, DFSDT160, DFSDT180, DFSDT330, DFSDT400,



---

**0364**

**Explanation:** A logic error was detected during database tracking. Reason codes further identify the type of logic error:

**Reason code****Explanation**

- 01** An invalid AWE request code was detected by database tracking CTL AWE processor or SAS AWE queue server.  
Module: DFSDT240, DFSDT300
- 02** An invalid function code was detected by milestone/quiesce processing or database status change processing.  
Module: DFSDT160, DFSDT180, DFSDT250
- 03** An invalid type of database redo record was encountered during tracking redo-queue processing.  
Module: DFSDT240

**System action:** IMS terminates abnormally with a memory dump.

**System programmer response:** Contact IBM.

**Source:** IMS abends

**Module:** DFSDT160, DFSDT180, DFSDT250, DFSDT240, DFSDT300,

---

**0365**

**Explanation:** A logic error was detected during isolate log send (ILS) processing. Reason codes further identify the type of logic error:

**Reason Code****Explanation**

- 01** An incorrect function code was detected by ILS.  
Module: DFSIL010, DFSIL210, DFSIL240, DFSIL330, DFSIL340, DFSIL500, DFSIL510
- 02** An ILS internal logic error was detected.  
Module: DFSIL300, DFSIL320
- 03** ILS issued a DBRC macro call and received an invalid return code. See either message DFS4072A or DFS4095A for more information about the error.  
Module: DFSIL300, DFSIL310
- 05** An invalid status exit reason code was detected.  
Module: DFSIL320, DFSIL400

**System action:** IMS terminates abnormally with a memory dump.

**System programmer response:** Contact IBM Software Support.

**Source:** IMS abends

**Module:** DFSIL010, DFSIL210, DFSIL240, DFSIL300, DFSIL310, DFSIL320, DFSIL330, DFSIL340, DFSIL400, DFSIL500, DFSIL510

**Related information:**

 [DFS4072A \(Messages and Codes\)](#)

 [DFS4095A \(Messages and Codes\)](#)

---

**0367**

**Explanation:** An IMS system error was encountered during DL/I call processing for a GU (message), ISRT (message) or SETO call. IMS writes a '67D0' log record indicating the cause of the error. This abend is set by DFSDLA30 and issued by DFSECP10/DFSECP20.

**System action:** The application program terminates abnormally.

**Programmer response:** Check the '67D0' log record to determine the cause of the error and contact IBM Software Support.

## 0368 • 0369

**Problem determination:** 1, 3, 5, 6, 36

**Source:** IMS abends

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

### 0368

**Explanation:** A CQS request failed during DL/I call processing for a GU (message) or an ISRT (message). IMS writes a type X'67D0' log record indicating the cause of the error. This abend is set by DFSDLA30 and issued by DFSECP10/DFSECP20.

**System action:**

The application program terminates abnormally. The input message is discarded if abend 0368 occurs on an ISRT call; the input message is not discarded if this abend occurs on a GU. The transaction is not stopped and the PSB is not stopped.

**Programmer response:** Check the X'67D0' log record to determine the cause of the error. Contact IBM Software Support.

**Problem determination:** 1, 3, 5, 6, 36

**Source:** IMS abend

**Module:** DFSECP10, DFSECP20

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

### 0369

**Explanation:** One of the following situations occurred:

- **Case 1:** The buffer handler module, DFSDVBH0, has detected an internal error. The caller provided the buffer handler with an RBA whose low-order bit contains either '0' or '1'. Bit '0' refers to data sets A-J of a HALDB, and bit '1' refers to data sets M-V of a HALDB. The value is inconsistent with DMBORFL2.
- **Case 2:** The tracking IMS in an RSR tracker environment has encountered an internal error.

**Analysis for Case 1:** At the time of the abend, both register 4 and register 5 indicate what the internal error might be. Both registers contain either a 0 or 1.

**Register value**

**Explanation**

- |   |   |
|---|---|
| 0 | The I/O is for one of the A-J data sets |
| 1 | The I/O is for one of the M-V data sets |

Both registers must contain the same value if processing is to continue. However, if the registers contain different values, processing immediately returns to the caller with PSTSUBCD=X'08' and PSTRTCDE=X'10'.

The input to the buffer handler is the PSTBYTNM that contains the RBA. When calling the buffer handler, DFSDVBH0, the RBA is always an even value for the A-J data sets; therefore, the value of the flag byte, DMBORFL2, of the corresponding data management block (DMB) (A-J DMB) is 0. The RBA is always an odd value for the M-V data sets; its flag byte, DMBORFL2, of the corresponding DMB (M-V DMB) is 1.

If a memory dump can be provided other than the pseudoabend from the X'67FF' log records, the same area chain will help determine the module that called the buffer handler and what the value of PSTFNCTN is.

**Analysis for Case 2:** At the time of the error, R15 contains the return code that indicates the condition that caused the abend.

**Return code (R15)**

**Explanation**

- |   |   |
|---|---|
| 1 | An IMODULE GETSTOR request failed. R10 contains the IMODULE GETSTOR return code. For a description of IMODULE return codes, see the IMS system services return codes information. |
| 2 | A DFSBCB request failed. R10 contains the DFSBCB return code.   |

**System action:** **Case 1:** The task terminates with a pseudoabend 0369.

**Case 2:** The IMS tracker terminates with abend code U0369.

**System programmer response:** **Cases 1 and 2:** Contact IBM Software Support.

**Source:** IMS abends

**Module:** **Case 1:** DFSDVBH0

**Case 2:** DFSST500

**Related reference:**

 IMODULE return codes (Messages and Codes)

### 0370

**Explanation:** A CQS request by a transaction for an uncommitted message failed. Some examples of the request would be a CQSPUT call for an output message or an UPDATE PREFIX call for an APPC/OTMA message. The failure could be due to a structure-full or lost connection to the structure.

This abend is set by DFSFXC40 and issued by DFSECP10, DFSECP20, DFSPCC20, and DFSCPY00.

**System action:** The application program ends abnormally. The transaction and PSB are stopped. The input message is not discarded.

**Programmer response:** After the failing condition is corrected, the input message should be ready for reprocessing. If there is only one program to process the transaction, you might need to start the transaction (/STA TRAN) and the program (/STA PGM); the input message will then be reprocessed.

**Problem determination:** 1, 3, 5, 6, 36

**Source:** IMS abends

**Related information:**

Chapter 2, "IMS problem determination," on page 13

### 0371

**Explanation:** During Phase 1 of syncpoint processing, a DFSQMGR Insert Prefix call was issued to append an IMSID to the queue name of a program-to-program switch message. The DFSQMGR call received a RC=08, which indicates an internal QMGR error.

**System action:** The application program ends abnormally. The transaction and PSB are stopped. The input message is discarded.

**Problem determination:** 1, 3, 5, 6, 36

**Source:** IMS abends

**Module:** DFSFXC40

**Related information:**

Chapter 2, "IMS problem determination," on page 13

### 0378

**Explanation:** IMS was unable to parse the DFSRSRxx member for one of the reasons below. The subcodes can be found in R15 at abend.

#### Reason code

##### Explanation

**01** One or more parameter errors were found while processing the member.

**02** Unable to obtain necessary working storage required while processing. An IMODULE GETMAIN failed; the return code is displayed in the DFS2929A message which precedes this abend. For more information, see the IMS system services return codes information.

**03** The TSO/E IKJPARS service used for parsing returned a nonzero return code indicating an error. The return

code is displayed in the DFS2929A message which precedes this abend. TSO writes an error message to the data set indicated by the SYSTSPRT DD statement in the IMS job. See the *TSO Extensions Version 2 Programming Services manual* for these messages.

**Analysis:** The DFSRSRxx member contains parameters for an RSR system. This member is selected by specifying the two-character suffix on the RSRMBR= exec parameter in the IMS procedure; the default is 00. While processing this member, IMS detected an error. This abend is preceded by one or more DFS2929A messages, which indicate the reason for the abend.

**System action:** IMS terminates abnormally.


**System programmer response:** The following actions correspond to the reason codes listed in the explanation.

1. Correct the error in the DFSRSRxx member and restart IMS. This return code indicates the member was parsed correctly, but the values were not in the allowable range.
2. Increase the IMS region size and restart IMS. If this problem recurs report the problem to IBM.
3. Analyze the TSO error messages and correct the syntax of the DFSRSRxx member, then restart the IMS system.

**Source:** IMS abends

**Module:** DFSRPP00

**Related reference:**

 [IMS system services return codes \(Messages and Codes\)](#)

**Related information:**

 [DFS2929A](#)

## 0379

### Explanation:

A Transport Manager instance (TMI) name was not specified for an IMS batch or utility job for which a global service group (GSG) name and logging were specified. A license certificate could not be found for either the DLT or RLT features.

**Analysis:** Specifying a GSG name for a batch or utility job that creates a log makes the job RSR-capable. However, a TMI name must also be specified for RSR processing. The abend is issued because a TMI name was not specified. Message DFS2924I precedes the abend.

**System action:** IMS terminates abnormally.

**System programmer response:** Correct the IMS system definition to specify the TMI name or supply the name using the TMI= execution parameter.

If you're running on a z/OS operating system that is running a zSeries processor, add the necessary license certificates for RLT, DLT, or both.

**Source:** IMS abends

**Module:** DFSXBAT0, DFSXDCC0

**Related information:**

 [DFS2924I \(Messages and Codes\)](#)

## 0380

**Explanation:** A logic error was detected during log router processing. Reason codes further identify the type of logic error:

### Reason code

#### Explanation

- 01** An invalid AWE function code was detected by a log router ITASK.

Module: DFSLRART, DFSLRRDS, DFSLRIDS, DFSLRSAR, DFSLRDCR, DFSLRRDZ, DFSLRILT, DFSLRMRG, DFSLRETI, DFSLRASC, DFSLRMIL, DFSLRARC, DFSLRLTR, DFSLRTRM, DFSLRCAS, DFSLRMST, DFSLRARP, DFSLROPR, DFSLRDDE, DFSLROIC, DFSLRRDC, DFSLRDSS, DFSLRRRO

- 02** An invalid parameter was detected while allocating a conversation with an active subsystem or isolated log sender by the Transport Manager Subsystem (TMS). Message DFS4018I accompanies the abend code. Register 14 contains the reason code from the TMS allocate conversation request.  
Module: DFSLRCS0, DFSLRICN
- 03** An invalid parameter error code was returned by the transport manager subsystem (TMS) as a result of an allocate conversation request. Message DFS4018I accompanies the abend code. Register 14 contains the reason code from the TMS allocate request.  
Module: DFSLRCS0, DFSLRICN
- 04** The tracking subsystem processed an invalid tracking subsystem log record (49xx) during tracking subsystem restart. Register 14 contains the address of the log record.  
Module: DFSLRARR
- 05** The tracking subsystem detected a missing gap descriptor block while attempting to obtain missing active subsystem log data from an Isolated Log Sender at the active site.  
Module: DFSLRICM, DFSLRIDS
- 06** The tracking subsystem detected an error while attempting to communicate with an active subsystem or the Isolated Log Sender (ILS) at the active site. Register 14 contains the reason code from the send request.  
Module: DFSLRASC, DFSLRIDS, DFSLRSCM
- 07** An invalid parameter was detected while deallocating a conversation with an active subsystem or isolated log sender by the Transport Manager Subsystem (TMS). Register 14 contains the reason code from the TMS deallocate conversation request.  
Module: DFSLRASC, DFSLRCAS, DFSLRCS0, DFSLRICN, DFSLRSTX
- 08** The tracking subsystem detected invalid log sequence numbers when attempting to close a tracking log with DBRC. The first log sequence number of the tracking log is greater than the last log sequence number. Register 14 has the address of a parameter area containing the invalid log sequence numbers.  
Module: DFSLRDDE, DFSLRRDD, DFSLRRDY, DFSLRSFT
- 09** The tracking subsystem attempted to write a log buffer with no data. Register 14 contains the address of the empty buffer.  
Module: DFSLRWRT
- 0A** The tracking subsystem received active subsystem log data out of order.  
Module: DFSLRREX
- 0B** The tracking subsystem received an invalid package descriptor record from an active subsystem.  
Module: DFSLRREX
- 0C** The tracking subsystem Transport Manager Subsystem (TMS) status exit routine received an invalid function request. Register 14 contains the invalid function code.  
Module: DFSLRSTX
- 0D** The tracking subsystem detected that an active subsystem provided log buffer with no data.  
Module: DFSLRIRX, DFSLRREX
- 0E** The tracking subsystem is unable to find the milestone position block containing the log position required for recovery of tracked databases or AREAs.  
Module: DFSLREDT
- 0F** An AWE was received by a tracking subsystem ITASK that could not be processed prior to tracking subsystem initialization.  
Module: DFSLRDSS
- 10** The tracking subsystem detected an invalid request code from the isolated log sender (ILS) at the active site. Register 14 contains the invalid request code.

Module: DFSLRIRX

- 11** The tracking subsystem detected an invalid buffer header in which the buffer header does not contain the last log sequence number contained in the buffer. Register 14 contains the address of the buffer header.

Module: DFSLRRR0

- 12** An OPEN macro was issued for a tracking log with a block size larger than the buffer size available to read it. Message DFS4012I with return code X'FFFF' precedes this abend. That message identifies the tracking log data set with the invalid block size. The buffer size is the maximum of the BLOCK keyword values of the SLDSDEF, ARCHDEF and RLDSDEF parameters of IMS.PROCLIB member DFSRSRxx. See message DFS4012I for more information.

Module: DFSLRDOP

- 13** The tracking subsystem detected a sharing subsystem using an active subsystem log record and has no record of it.

Module: DFSLRMRG

- 14** The tracking subsystem is merging active subsystem log records and is unable to find a control block representing an active subsystem participating in the merge.

Module: DFSLRMRG

- 15** The tracking subsystem is presented with a second sharing log record representing an active subsystem participating in the merge.

Module: DFSLRMRG

- 16** The tracking subsystem detected invalid feedback after routing data to the tracking subsystem trackers while merging log records from multiple active subsystems.

Module: DFSLRMRG

- 18** The tracking subsystem detected an invalid condition while attempting to read a tracking log. Register 14 contains the address of the invalid AWE.

Module: DFSLROPR, DFSLRRR0

- 1A** The tracking subsystem detected an invalid log record from an active subsystem.

Module: DFSLRIRX, DFSLRREX

- 1B** The tracking subsystem detected an invalid STCK value in a log record from an active subsystem.

Module: DFSLRBIR, DFSLRRBF, DFSLRSDS, DFSLRWRT

- 1C** The tracking subsystem is unable to read a set of log records required to complete restart.

Module: DFSLRRRS

- 1D** The tracking subsystem has determined that the position being set for routing a log is not valid. This error could result from an internal logic error or an improper cold start of the tracking subsystem.

Module: DFSLRRR0, DFSLRRRS

- 1E** (Reason code 1E is currently not used.)

- 21** The tracking subsystem received data from the isolated log sender without a "begin data set" notification.

Module: DFSLRIDS

- 22** The tracking subsystem is told to stop receiving data from the isolated log sender, but cannot find the required control blocks.

Module: DFSLRIDS

- 23** The tracking subsystem detected a buffer in the available buffer pool that was not marked as available.

Module: DFSLRASC, DFSLRBIT, DFSLRGLB, DFSLRIDS, DFSLRIRX, DFSLRLTR, DFSLRMRG, DFSLROPR, DFSLRORM, DFSLRORN, DFSLRRDC, DFSLRRDF, DFSLRRDS, DFSLRRDW, DFSLRREX, DFSLRRRD, DFSLRRRS, DFSLRRR0, DFSLRRTO, DFSLRSAR, DFSLRSBR, DFSLRSIN, DFSLRTRS, DFSLRWRT

- 24 The tracking subsystem detected a buffer in the available buffer pool that belongs to the automatic archive buffer pool.  
Module: DFSLRASC, DFSLRBIT, DFSLRGLB, DFSLRIDS, DFSLRIRX, DFSLRLTR, DFSLRMRG, DFSLROPR, DFSLRORM, DFSLRORN, DFSLRRDC, DFSLRRDF, DFSLRRDS, DFSLRRDW, DFSLRREX, DFSLRRRD, DFSLRRRS, DFSLRRR0, DFSLRRTO, DFSLRSAR, DFSLRSBR, DFSLRSIN, DFSLRTRS, DFSLRWRT
- 25 The tracking subsystem was either unable to obtain information about missing log data from DBRC, or obtained invalid information about missing log data from DBRC.  
Module: DFSLRRRS
- 26 The tracking subsystem detected an invalid buffer when processing the buffer to write to a tracking log.  
Module: DFSLRARP DFSLRWRT
- 27 The tracking subsystem needs to close a data set that resides on a mountable device type, but is unable to find the TCB which opened the data set.  
Module: DFSLRDSS
- 28 The tracking subsystem detected an internal error while attempting to obtain a tracking subsystem latch. Register 14 contains the return code from the obtain latch request.  
Module: DFSLRARA, DFSLRARL, DFSLRARR, DFSLRDRC, DFSLRDDE, DFSLRDSS, DFSLRLTS, DFSLRMST, DFSLRRDD, DFSLRRDS, DFSLRRDY, DFSLRRST, DFSLRSAR, DFSLRSDS, DFSLRSFT, DFSLRSIN, DFSLRSWB, DFSLRWRT
- 29 The tracking subsystem detected an internal error while attempting to release a tracking subsystem latch. Register 14 contains the return code from the release latch request.  
Module: DFSLRARA, DFSLRARL, DFSLRARR, DFSLRDRC, DFSLRDDE, DFSLRDSS, DFSLRLTS, DFSLRMST, DFSLRRDD, DFSLRRDS, DFSLRRDY, DFSLRRST, DFSLRSAR, DFSLRSDS, DFSLRSFT, DFSLRSIN, DFSLRSWB, DFSLRWRT
- 2A The tracking subsystem found no logs to start a database or AREA as a result of a /START command, but was notified by DBRC that the database or AREA requires recovery.  
Module: DFSLRORH DFSLRORM DFSLRORN
- 2B The tracking subsystem determined that the start point for a database or AREA was invalid when processing a /START DATABASE, UPDATE DB START(ACCESS), /START AREA or UPDATE AREA START(ACCESS) command.  
Module: DFSLRORM DFSLRORN
- 2C The tracking subsystem found the initial log record for an active subsystem was not in the concurrent image copy (CIC) as a result of processing a /START DATABASE, UPDATE DB START(ACCESS), /START AREA or UPDATE AREA START(ACCESS) command.  
Module: DFSLROPR
- 2D The tracking subsystem determined that DBRC did not supply the start-point information required for online forward recovery (OFR) processing.  
Module: DFSLROIC, DFSLRORH, DFSLRORM
- 2E The tracking system detected an I/O error while accessing a tracking log data set. The error occurred during either the BSAM READ macro or BSAM WRITE macro (not the BSAM CHECK macro). The abend originates in DFSLRRDU if the error occurred during BSAM READ; it originates in DFSLRSWB if the error occurred during BSAM WRITE. Refer to the accompanying access method services message to determine the appropriate action to take.  
Module: DFSLRRDU, DFSLRSWB
- 2F During OFR, the log router detected a log data set descriptor in the list returned by DBRC that represented a log data set that contained data that occurred prior to the current OFR position.  
Module: DFSLRORM

Source: IMS abends

**Module:** DFSLRARA, DFSLRARC, DFSLRARL, DFSLRARP, DFSLRARR, DFSLRART, DFSLRASC, DFSLRBIR, DFSLRBIT, DFSLRCAS, DFSLRCS0, DFSLRDCR, DFSLRDDE, DFSLRDOP, DFSLRDSS, DFSLREDT, DFSLRETI, DFSLRGLB, DFSLRICM, DFSLRICN, DFSLRIDS, DFSLRILT, DFSLRIRX, DFSLRLTR, DFSLRLTS, DFSLRMIL, DFSLRMRG, DFSLRMST, DFSLROIC, DFSLROPR, DFSLRORH, DFSLRORM, DFSLRORN, DFSLRRBF, DFSLRRDC, DFSLRRDD, DFSLRRDF, DFSLRRDS, DFSLRRDU, DFSLRRDW, DFSLRRDY, DFSLRRDZ, DFSLRREX, DFSLRRR0, DFSLRRRD, DFSLRRRD, DFSLRRRS, DFSLRRST, DFSLRRTO, DFSLRSAR, DFSLRSBR, DFSLRSCM, DFSLRSDS, DFSLRSFT, DFSLRSIN, DFSLRSTX, DFSLRSWB, DFSLRTRM, DFSLRTRS, DFSLRWRT

## 0381

**Explanation:** An unrecoverable resource shortage was detected by log router processing logic. Reason codes further identify the type of unrecoverable error:

**Reason code****Explanation**

- 01** The tracking subsystem was unable to obtain an AWE. Register 14 contains the reason code from DFSBCB. Module: DFSLRARA, DFSLRARC, DFSLRARI, DFSLRARP, DFSLRARR, DFSLRART, DFSLRAR1, DFSLRASC, DFSLRAST, DFSLRAS1, DFSLRBIR, DFSLRBIT, DFSLRCAS, DFSLRCS0, DFSLRDAL, DFSLRDBT, DFSLRDCR, DFSLRDDE, DFSLRDSS, DFSLRDTM, DFSLREDT, DFSLRICM, DFSLRICN, DFSLRIDS, DFSLRILT, DFSLRIN1, DFSLRIRX, DFSLRLTC, DFSLRLTR, DFSLRMIL, DFSLRMRG, DFSLRMST, DFSLROCR, DFSLROIC, DFSLROPR, DFSLRORM, DFSLRORN, DFSLRRBF, DFSLRRDA, DFSLRRDB, DFSLRRDC, DFSLRRDD, DFSLRRDF, DFSLRRDH, DFSLRRDK, DFSLRRDN, DFSLRRDR, DFSLRRDS, DFSLRRDT, DFSLRRDU, DFSLRRDW, DFSLRRDX, DFSLRRDY, DFSLRRDZ, DFSLRREX, DFSLRRRS, DFSLRRR0, DFSLRSAR, DFSLRSBR, DFSLRSDS, DFSLRSFG, DFSLRSFT, DFSLRSIN, DFSLRSTX, DFSLRTK0, DFSLRWRT
- 02** The tracking subsystem was unable to obtain BRLSB storage in CSA for communication with DBRC. Register 14 contains the reason code from the storage request. Module: DFSLRAST, DFSLRDAL, DFSLRDBT, DFSLRDCR, DFSLRDSS, DFSLRIN0, DFSLRORS, DFSLRRDB, DFSLRRDC, DFSLRRDN, DFSLRRDZ, DFSLRRRS
- 03** The tracking subsystem was unable to obtain storage for a log buffer(s). Register 14 contains the reason code from the storage request. Module: DFSLRDCR, DFSLRDSS, DFSLRRCR
- 04** The tracking subsystem was unable to create an ITASK. Register 14 contains the reason code. Module: DFSLRASC, DFSLRDAL, DFSLRDCR, DFSLRDSS, DFSLROIC
- 05** The tracking subsystem was unable to obtain extended CSA storage. Register 14 contains the reason code from the storage request. Module: DFSLRARS, DFSLRDBT, DFSLRDSS, DFSLRDTM, DFSLRMIL, DFSLRMST, DFSLRRCR, DFSLRRDB, DFSLRRDC, DFSLRRDH, DFSLRRDN, DFSLRRDS, DFSLRRDX, DFSLRRDZ, DFSLRRRD, DFSLRRST, DFSLRSIN
- 06** The tracking subsystem was unable to obtain storage from BCB to represent missing log data. Register 14 contains the reason code from the storage request. Module: DFSLRASC, DFSLRAS1, DFSLRDDE, DFSLRICM, DFSLRIDS, DFSLRIGP, DFSLRRDH, DFSLRRDX, DFSLRSAR, DFSLRSDS
- 07** The tracking subsystem was unable to obtain storage using IMODULE. Register 14 contains the reason code. Module: DFSLRARA, DFSLRARC, DFSLRARI, DFSLRARL, DFSLRARR, DFSLRAR1, DFSLRASC, DFSLRAS1, DFSLRCAS, DFSLRCS0, DFSLRDAL, DFSLRDCR, DFSLRDDE, DFSLRDIS, DFSLRDSS, DFSLRDTM, DFSLREDT, DFSLRGLB, DFSLRICM, DFSLRICN, DFSLRIDS, DFSLRILT, DFSLRIN0, DFSLRINS, DFSLRIRX, DFSLRLTS, DFSLRMIL, DFSLRMRG, DFSLRMST, DFSLROIC, DFSLRORH, DFSLRORM, DFSLRORN, DFSLRORS, DFSLRRDA, DFSLRRDB, DFSLRRDC, DFSLRRDD, DFSLRRDH, DFSLRRDN, DFSLRRDR, DFSLRRDS, DFSLRRDY, DFSLRRRS, DFSLRRR0, DFSLRRST, DFSLRSAR, DFSLRSDS, DFSLRSFT, DFSLRSIN, DFSLRSTX, DFSLRTRM, DFSLRSWB, DFSLRTRM
- 08** The tracking subsystem was unable to create a conversation manager ITASK. Register 14 contains the reason code from the create work unit request. Module: DFSLRIN0
- 09** The tracking subsystem was unable to load a load module with IMODULE. Register 14 contains the reason code from the imodule func=load request. Module: DFSLRIN0



- 0A The tracking subsystem was unable to obtain a save area set for use by the tracking subsystem Transport Manager exits. Register 14 contains the reason code from BCB.  
Module: DFSLRIN0
- 0B The tracking subsystem was unable to create a tracking log data set manager ITASK. Register 14 contains the reason code from the create work unit request.  
Module: DFSLRIN0
- 0C The tracking subsystem was unable to create a milestone manager ITASK. Register 14 contains the reason code from the create work unit request.  
Module: DFSLRIN0
- 0D The tracking subsystem was unable to create an isolated log Transport Manager Subsystem ITASK. Register 14 contains the reason code from the create work unit request.  
Module: DFSLRIN0
- 0E The tracking subsystem was unable to create an automatic archive ITASK. Register 14 contains the reason code from the create work unit request.  
Module: DFSLRIN0
- 0F The tracking subsystem was unable to create a tracking log read controller ITASK. Register 14 contains the reason code from the create work unit request.  
Module: DFSLRIN0
- 10 The tracking subsystem was unable to attach subordinate TCBs. Register 14 contains the reason code from the dfsattach request.  
Module: DFSLRDAL, DFSLRDCR, DFSLRDSS, DFSLRIN0
- 11 The tracking subsystem was unable to obtain a service group list from DBRC during initialization. Register 14 contains the reason code from DBRC.  
Module: DFSLRIN1
- 12 The Remote Site Recovery tracking subsystem detected an unexpected return code from a deallocate conversation request for a conversation with the isolated log sender at the active site. Register 14 contains the return code from the deallocate conversation attempt.  
Module: DFSLRILT
- 13 An INITECB failure occurred while attempting to read tracking log data in a Remote Site Recovery tracking subsystem. Register 14 contains the reason code from the INITECB request.  
Module: DFSLRBIR, DFSLRRDT
- 14 The tracking subsystem detected an unexpected return code from DBRC on an allocate database/AREA attempt. Register 14 contains the reason code from the DBRC request.  
Module: DFSLRDBT
- 15 The tracking subsystem detected an unexpected return code from DBRC on a deallocate database/AREA attempt. Register 14 contains the reason code from the DBRC request.  
Module: DFSLRDBT
- 16 The Remote Site Recovery tracking subsystem detected an error during the log truncation phase of unplanned takeover processing. An unexpected return code from DBRC when the truncation request is made. Register 14 contains the reason code from the DBRC request.  
Module: DFSLRLTS
- 17 The tracking subsystem detected an unexpected return code from DBRC on a replace tracking logs for truncation attempt for log truncation during unplanned takeover. Register 14 contains the reason code from the DBRC request.  
Module: DFSLRLTR
- 18 The tracking subsystem was unable to obtain BCB storage. Register 14 contains the reason code from the BCB request.  
Module: DFSLRDCR, DFSLRICM, DFSLRICN, DFSLRILT, DFSLRINS, DFSLRIN0, DFSLRMIL, DFSLROIC, DFSLRORS, DFSLRRST
- 19 The tracking subsystem was unable to create an online forward recovery ITASK. Register 14 contains the reason code from the create work unit request.  
Module: DFSLRIN0

- 1B** An error was detected when the RSR tracking IMS tried to create a position data set during RSR tracking IMS shutdown. Register 14 contains one of the following
- 1** An error was detected when the RSR tracking IMS tried to create the position data set. See accompanying messages and resolve the problem identified in the messages before restarting the tracking IMS.
  - 2** Another data set was detected with the same name as the position data set. Delete the position data set identified in the accompanying messages before shutting down the tracking IMS.
  - 3** An error was detected when the RSR tracking IMS tried to open the position data set. See accompanying messages and resolve the problem identified in the messages before restarting the tracking IMS.
  - 4** An error was detected when the RSR tracking IMS tried to write to the position data set. See accompanying messages and resolve the problem identified in the messages before restarting the tracking IMS.
  - 5** An error was detected when the RSR tracking IMS tried to close and deallocate the position data set. See accompanying messages and resolve the problem identified in the messages before restarting the tracking IMS.
  - 6** A read error was detected while processing the secondary position data set during a cold start of the RSR tracking IMS. The read error occurred after an error was detected while processing the primary position data set. The RSR tracking IMS is unable to set the routing position and cannot continue with cold start processing. Resolve the cause of the errors and start the RSR tracking IMS.

Module: DFSLRDCR, DFSLRDOP, DFSLRDSS, DFSLRTRM:

- 1C** The tracking subsystem was unable to log a tracking subsystem log record. Register 14 contains the reason code from the ILOG attempt.  
Module: DFSLRARL, DFSLRARR, DFSLRASC, DFSLRAST, DFSLRAS1, DFSLRCAS, DFSLRDCR, DFSLRDDE, DFSLRDSS, DFSLRD TM, DFSLRLTR, DFSLRLTS, DFSLRMIL, DFSLRMST, DFSLROIC, DFSLRRCR, DFSLRRDD, DFSLRRDH, DFSLRRDY, DFSLRSAR, DFSLRSDS, DFSLRSFT, DFSLRSWB, DFSLRTK0, DFSLRTRM, DFSLRWRT
- 1D** The tracking subsystem failed to create the required output tracking log for log truncation during unplanned takeover processing.  
Module: DFSLRLTS
- 1E** The tracking subsystem was unable to obtain at least 1 log buffer for use by log truncation during unplanned takeover processing.  
Module: DFSLRLTS
- 1F** The tracking subsystem was unable to create a log reader for use by log truncation during unplanned takeover processing. Register 14 contains the return code from the create read request.  
Module: DFSLRLTS
- 20** The tracking subsystem was unable to create an end tracking ITASK. Register 14 contains the reason code from the create work unit request.  
Module: DFSLRIN0
- 21** The tracking subsystem detected an error while processing the position data set. Register 14 contains one of the following the reason codes:
- The tracking subsystem was unable to create a position data set.
  - A data set with the same name as the position data set was found when the tracking subsystem attempted to create it.
  - An error was detected while attempting to open the position data set.
  - An error was detected while attempting to write to the position data set.
  - An error was detected while attempting to close the position data set.
- Module: DFSLRDCR, DFSLRTRM
- 22** The tracking subsystem experienced an unexpected return code from DBRC on a query tracking logs request. Register 14 contains the return code from DBRC.  
Module: DFSLRIGP, DFSLRINI, DFSLRRRS
- 23** The tracking subsystem experienced an unexpected return code on a call to DBRC. Register 14 contains the return code from DBRC.  
Module: DFSLRAST, DFSLRDDE, DFSLRDSS, DFSLRLIT, DFSLRINI, DFSLRRDB, DFSLRRDD, DFSLRRDH, DFSLRRDN, DFSLRRDX, DFSLRRDY, DFSLRRRS, DFSLRSAR, DFSLRSDS, DFSLRTK0, DFSLRTRM

- 24 The tracking subsystem detected an invalid control block. Register 14 contains the address of the control block.  
Module: DFSLRART
- 25 The tracking subsystem requires archive SLDS to be defined before allowing unplanned takeover processing to continue. This abend is preceded by message DFS4030A. See the message description for further details.  
Module: DFSLRLTS, DFSLRTRM
- 27 The log router is unable to create the position data set ITASK. Register 14 contains the create work unit return code.  
Module: DFSLRIN0
- 28 The set owner call to DBRC for a HALDB Online Reorganization at the tracking site failed.  
Module: DFSLRDBT
- 29 The set owner call to DBRC for a HALDB Online Reorganization at the tracking site failed.  
Module: DFSLRDBT
- 2A The reset cursor call to DBRC for a HALDB Online Reorganization at the tracking site failed.  
Module: DFSLRDBT

**Source:** IMS abends

**Module:** DFSLROIC, DFSLRRDX, DFSLRAR1, DFSLRARA, DFSLRARC, DFSLRARI, DFSLRARL, DFSLRARP, DFSLRARR, DFSLRARS, DFSLRART, DFSLRAS1, DFSLRASC, DFSLRAST, DFSLRBIR, DFSLRBIT, DFSLRCAS, DFSLRCS0, DFSLRDAL, DFSLRDBT, DFSLRDCR, DFSLRDDE, DFSLRDIS, DFSLRDOP, DFSLRDSS, DFSLRDTC, DFSLRDTM, DFSLREDT, DFSLRGSL, DFSLRICM, DFSLRICN, DFSLRIDS, DFSLRIGP, DFSLRILT, DFSLRIN0, DFSLRIN1, DFSLRINI, DFSLRINS, DFSLRIRX, DFSLRLTC, DFSLRLTR, DFSLRLTS, DFSLRMIL, DFSLRMRG, DFSLRMST, DFSLROCR, DFSLROIC, DFSLROPR, DFSLRORH, DFSLRORM, DFSLRORN, DFSLRORS, DFSLRRBF, DFSLRRCR, DFSLRRDA, DFSLRRDB, DFSLRRDC, DFSLRRDD, DFSLRRDF, DFSLRRDH, DFSLRRDK, DFSLRRDN, DFSLRRDR, DFSLRRDS, DFSLRRDT, DFSLRRDU, DFSLRRDW, DFSLRRDX, DFSLRRDY, DFSLRRDZ, DFSLRREX, DFSLRRR0, DFSLRRRD, DFSLRRRS, DFSLRRST, DFSLRSAR, DFSLRSBR, DFSLRSDS, DFSLRSFG, DFSLRSFT, DFSLRSIN, DFSLRSTX, DFSLRSWB, DFSLRTRM, DFSLRWRT, DFSLRDC

---

0384

**Explanation:** A service invoked by the Recovery Data Manager returned an unexpected result. Reason codes further identify the type of error:

**Reason code**

**Explanation**

002

Obtain variable length storage error

The Recovery Data Manager is unable to obtain storage required for recovery. Register 14 contains the return code from the obtain storage service.

Modules: FRDRPPI00, FDRPPR00, FDRPPS00, FDRRVAM0, FDRRVGF0, FDRRVIA0, FDRRVLM0, FDRRVPR0, FDRRVQD0, FDRRVS00

003

Load error

The Recovery Data Manager is unable to load modules required for recovery. Register 14 contains the return code from the load service.

Modules: FDRPPI00, FDRPPR00, FDRPPS00, FDRRVMI0, FDRRVS10

004

Obtain fixed length storage

The Recovery Data Manager is unable to obtain fixed length storage. Register 14 contains the return code from the load service.

Modules: FDRRVAC0, FDRRVAM0, FDRRVIA0, FDRRVIC0, FDRRVIR0, FDRRVMI0, FDRRVPR0, FDRRVQD0, FDRRVRB0, FDRRVRC0, FDRRVRD0, FDRRVWI0, FDRRVWR0, FDRRVWT0

## 0384

### 005

#### Create thread error

If the module is FDRRVIC0, the Recovery Data Manager is unable to create an image copy restore dispatchable thread. If the module is FDRRVIR0, the Recovery Data Manager is unable to gain access to image copy restore dispatchable thread blocks. Register 14 contains the return code from the thread service.

Modules: FDRRVIC0, FDRRVIR0

### 007

The Recovery Data Manager is unable to enqueue an asynchronous work element to a dispatchable thread. Register 14 contains the return code from the enqueue service.

Modules: FDRRVAC0, FDRRVAM0, FDRRVIA0, FDRRVIC0, FDRRVIR0, FDRRVM00, FDRRVPR0, FDRRVQD0, FDRRVRB0, FDRRVRC0, FDRRVRD0, FDRRVWI0, FDRRVWR0, FDRRVWT0

### 008

#### Attach TCB error

The Recovery Data Manager image copy restore controller is unable to attach a TCB for an image copy restore instance. Register 14 contains the return code from the attach TCB service.

Modules: FDRRVIC0

### 009

#### Message error

The Recovery Data Manager is unable to issue a message. Register 14 contains the return code from the message service.

Modules: FDRRVAM0, FDRRVDA0, FDRRVDD0, FDRRVID0, FDRRVII0, FDRRVIR0, FDRRVIS0, FDRRVRB0, FDRRVRC0, FDRRVRD0, FDRRVWC0, FDRRVWR0

### 00A

The Recovery Data Manager is unable to obtain a data buffer. Register 14 contains the return code from the obtain buffer service.

Modules: FDRRVMI0, FDRRVPR0

### 00B

#### Set storage length

The Recovery Data Manager is unable to set values for obtaining fixed storage. Register 14 contains the return code from the set storage service.

Modules: FDRRVMI0

### 00C

#### Post Error

The Recovery Data Manager is unable to post a dispatchable thread. Register 14 contains the return code from the post service.

Modules: FDRRVAM0

### 00F

#### Write to IMS Pipe Error

The Recovery Data Manager detected an error sending data to the Database Recovery Manager using an IMS pipe service. Register 14 contains the return code from the wait service.

Modules: FDRRVWR0

### 011

#### Release Storage Error

The Recovery Data Manager is unable to release storage. Register 14 contains the return code from the wait service.

Modules: FDRRVPR0

**012**

## Terminate Thread

The Recovery Data Manager is unable to terminate a thread. Register 14 contains the return code from the wait service.

Modules: FDRRVIR0

**013**

## Create IMS Pipe Error

The Recovery Data Manager is unable to create an IMS pipe for inter-address space communication. Register 14 contains the return code from the wait service.

Modules: FDRRVPR0

**014**

## Address Space Parameter Error

The Recovery Data Manager experienced an error using the ASEX macro to access and interpret parameters passed by the Database Recovery Manager as part of creating the Recovery Data Manager address space. Register 14 contains the return code from the create address space service.

Modules: FDRRVMI0

**015**

## IMS Pipe Create Error in DSPSERV

The IMS pipe create service detected an error from the DSPSERV macro. Register 14 contains the return code from the DSPSERV macro.

Modules: FDRPPIB0

**016**

## IMS Pipe Create Error in ALESERV

The IMS pipe create service detected an error from the ALESERV macro. Register 14 contains the return code from the ALESERV macro.

Modules: FDRPPIB0

**017**

## Establish ESTAE Error in ESTAEX

The IMS pipe service detected an error while using the ESTAEX macro.

Modules: FDRPPIB0, FDRPPRB0, FDRPPSB0

**Source:** IMS abends

**Module:** FDRPPI00, FDRPPIB0, FDRPPR00, FDRPPRB0, FDRPPS00, FDRPPSB0, FDRRVAC0, FDRRVAM0, FDRRVDA0, FDRRVDD0, FDRRVGF0, FDRRVIA0, FDRRVIC0, FDRRVID0, FDRRVII0, FDRRVIR0, FDRRVIS0, FDRRVLM0, FDRRVM00, FDRRVMI0, FDRRVPR0, FDRRVQD0, FDRRVRB0, FDRRVRC0, FDRRVRD0, FDRRVS00, FDRRVS10, FDRRVWC0, FDRRVWI0, FDRRVWR0, FDRRVWT0, FRDRPPI00

**0385**

**Explanation:** The Recovery Data Manager detected a logic error. Reason codes further identify the type of error:

**Reason code****Explanation****001**

## Invalid AWE function code

A Recovery Data Manager thread attempted to process an unknown AWE function request. Register 14 contains the AWE function code.

Modules: FDRRVAM0, FDRRVIR0, FDRRVLM0, FDRRVM00, FDRRVQD0, FDRRVRC0, FDRRVRD0, FDRRVSP1

**002**

## Data set allocation parameter error

**0385**

The Recovery Data Manager detected a data set allocation parameter error. Register 14 contains the address in the FDRRVRA control block.

Modules: FDRRVDA0

**003**

The Recovery Data Manager control pipe routine detected an unknown function request from the Database Recovery Manager.

Modules: FDRRVPR0

**005**

Invalid recovery buffer

The Recovery Data Manager detected an invalid recovery buffer on the unused recovery buffer chain.

Modules: FDRRVGF0, FDRRVIC0, FDRRVLM0, FDRRVQD0, FDRRVRB0

**006**

The Recovery Data Manager detected an unusable storage size for an obtain storage request.

Modules: FDRPPIB0, FDRRVDA0

**009**

SSID error

The Recovery Data Manager detected an invalid number of subsystem identifiers in the SSID list. Register 14 contains the address of the AWE containing the invalid number of SSIDs.

Modules: FDRRVQD0

**00A**

Invalid record detected

The Recovery Data Manager detected an invalid record.

Modules: FDRRVPR0, FDRRVRB0

**00B**

Invalid data buffer detected

The Recovery Data Manager detected a buffer with invalid data.

Modules: FDRRVIA0, FDRRVQD0

**00D**

Image copy restore function error

The Recovery Data Manager image copy restore process detected an invalid function request

Modules: FDRRVID0

**015**

Invalid ECB address

The Recovery Data Manager is unable to get the task ECB address.

Modules: FDRRVLM0

**016**

Log data merge setup or parameter error

The Recovery Data Manager detected one of the following errors when passed a buffer to merge:

- A new merge request has not been received (no position block exists).
- An invalid buffer pointer (0) was detected.
- An invalid (0) merge token was passed.

Modules: FDRRVLM0

**017**

Invalid end of data detected

The Recovery Data Manager detected an end of data notification with at least one SSID still in an active state.

Modules: FDRRVLM0

**018**

Invalid merge identifier

The Recovery Data Manager was passed an invalid merge ID when a buffer was passed to it for processing.

Modules: FDRRVLM

**019**

Invalid merge buffer

The Recovery Data Manager detected a buffer with one of the following errors:

- The subsystem index is greater than the total number of subsystems.
- A request for processing (on behalf of an SSID) was processed before notification was received that it should be part of merge processing.
- The buffer being processed has a Log Sequence Number that is smaller than the previous buffer with the same PRILOG time.
- Data was received for an SSID whose log data already ended.

Modules: FDRRVLM0

**020**

Invalid end of merge

The Recovery Data Manager is passed an invalid merge id for an end of merge for SSID call.

Modules: FDRRVLM0

**Source:** IMS abends

**Module:** FDRPPIB0, FDRRVAM0, FDRRVDA0, FDRRVGF0, FDRRVIA0, FDRRVIC0, FDRRVID0, FDRRVIR0, FDRRVLM , FDRRVLM0, FDRRVLM0, FDRRVPM0, FDRRVPR0, FDRRVQD0, FDRRVRB0, FDRRVRC0, FDRRVRD0, FDRRVSP1

**0388**

**Explanation:** This abend indicates an RSR initialization failure. DFSIINB0 detected an IMS internal error during IMS Transaction Manager initialization of an RSR-capable system. Further initialization cannot continue.

**Analysis:** Register 15 contains the following abend subcodes:

**Subcode****Explanation****X'04'**

USERVAR was not specified in the DFSHSBxx member of IMS.PROCLIB or as an execution parameter override.

**X'08'**

The value specified on the MTOID parameter in the DFSRSRxx member of IMS.PROCLIB was not recognized. Valid values are 1, 2, or 3.

**X'0C'**

Username3 was not specified in the APPLID parameter of the COMM macro, and the APPLID3 parameter was blank. Either username3 or APPLID3 must be specified for the RSR tracker.

**System action:** IMS terminates abnormally

**Programmer response:** Correct the appropriate system parameter.

**Source:** IMS abends

**0390**

**Explanation:** This abend is a standard abend issued by module DFSXLUM0 during IMS initialization. A nonzero return code was returned after DFSBCB GET, IMODULE GETMAIN, DFSCIR, IMODULE LOAD, or IPOST was issued.

**Analysis:** 0390 is a standard abend issued by module DFSXLUM0 during control region initialization. SVRB

## 0391

registers 6, 8, 12, 14, and 15 contain values for diagnosing the error. Register 12 is the base register and register 14 contains the address where the error condition was recognized. Register 15 contains one of the following:

### Codes

#### Explanation

**X'04'**

DFSBCB GET QSAV failed. Register 2 contains the DFSBCB return code.

**X'08'**

IMODULE GETMAIN failed. Register 2 contains the IMODULE return code.

**X'C'**

DFSBCB GET LSAV failed. Register 2 contains the DFSBCB return code.

**X'10'**

DFSPOOL GET failed. Register 2 contains the DFSPOOL return code.

**X'14'**

IMODULE LOAD failed. Register 2 contains the IMODULE return code. Register 6 contains the address of the name of the module being loaded.

**X'18'**

IPOST failed. Register 2 contains the IPOST return code.

**X'1C'**

DFSCIR failed. Register 2 contains the DFSCIR return code.

**X'20'**

The Dynamic Allocation (SVC 99) of the data set SYS1.CSSLIB has failed (the data set might not be catalogued). See register 2 for the SVC 99 return code.

**X'24'**

The open request for data set SYS.CSSLIB has failed. See the accompanying Contents Supervision messages (CSVxxxxl) for details.

### Codes

#### Explanation

**X'28'**

The data set SYS1.CSSLIB is not APF-authorized.

**System action:** IMS terminates abnormally.

**Problem determination:** 4, 33

**Source:** IMS abends

**Module:** DFSXLUM0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

## 0391

**Explanation:** This abend is a standard abend issued by module DFSXRM00 during IMS initialization. A nonzero return code was returned after DFSBCB GET, IMODULE GETMAIN, DFSCIR, or IPOST was issued.

**Analysis:** 0391 is a standard abend issued by module DFSXRM00 during control region initialization. SVRB registers 8, 12, 14, and 15 contain values for diagnosing the error.

Register 15 contains an abend subcode. Register 12 is the program base and register 14 is the address where the error condition was recognized. Register 15 contains the following:

### Codes

#### Explanation

**X'04'**

DFSBCB GET QSAV failed. Register 8 contains the DFSBCB return code.



**X'08'**

IMODULE GETMAIN failed. Register 8 contains the IMODULE return code.

**X'0C'**

DFSBCB GET LSAV failed. Register 8 contains the DFSBCB return code.

**X'10'**

DFSCIR failed. Register 8 contains the DFSCIR return code.

**X'14'**

IPOST failed. Register 8 contains the IPOST return code.

**System action:** IMS terminates abnormally.**Problem determination:** 4, 33**Source:** IMS abends**Module:** DFSXRM00**Related information:**

Chapter 2, "IMS problem determination," on page 13

0392

**Explanation:** This abend is a standard abend issued by module DFSXTMC0 during IMS initialization. A nonzero return code was returned after DFSBCB GET, IMODULE GETMAIN, DFSCIR, IMODULE LOAD, or IPOST was issued.**Analysis:** 0392 is a standard abend issued by module DFSXTMC0 during control region initialization. SVRB registers 8, 12, 14, and 15 contain values for diagnosing the error.

Register 15 contains an abend subcode. For all abend codes, register 12 is the program base and register 14 is the address where the error condition was recognized. The register 15 subcode is as follows:

**Codes****Explanation****X'04'**

DFSBCB GET QSAV failed. Register 8 contains the DFSBCB return code.

**X'08'**

IMODULE GETMAIN failed. Register 8 contains the IMODULE return code.

**X'0C'**

IMODULE LOAD failed. Register 8 contains the IMODULE return code.

**X'10'**

DFSCIR failed. Register 8 contains the DFSCIR return code.

**X'14'**

IPOST failed. Register 8 contains the IPOST return code.

**X'18'**

DFSBCB GET LSAV failed. Register 8 contains the DFSBCB return code.

**System action:** IMS terminates abnormally.**Problem determination:** 4, 33**Source:** IMS abends**Module:** DFSXTMC0**Related information:**

Chapter 2, "IMS problem determination," on page 13

## 0393

**Explanation:** This abend is a standard abend issued by module DFSXALM0 during IMS initialization. A nonzero return code was returned after DFSBCB GET, IMODULE GETMAIN, DFSCIR, or IPOST was issued.

**Analysis:** 0393 is a standard abend issued by module DFSXALM0 during control region initialization. SVRB registers 8, 12, 14, and 15 contain values for diagnosing the error.

Register 15 contains an abend subcode. For all abend codes, register 12 is the program base and register 14 is the address where the error condition was recognized. The register 15 subcode is as follows:

**Codes****Explanation****X'04'**

DFSBCB GET QSAV failed. Register 8 contains the DFSBCB return code.

**System action:** IMS terminates abnormally.

**Problem determination:** 4, 33

**Source:** IMS abends

**Module:** DFSXALM0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

## 0394

**Explanation:** This abend is a standard abend issued by module DFSXALC0 during IMS initialization. A nonzero return code was returned after DFSBCB GET was issued.

**Analysis:** This is a standard abend issued by module DFSXALC0 during control region initialization. SVRB registers 8, 12, 14, and 15 contain values for diagnosing the error.

Register 15 contains an abend subcode. For all abend codes, register 12 is the program base and register 14 is the address where the error condition was recognized. The register 15 subcode is as follows:

**Codes****Explanation****X'04'**

DFSBCB GET QSAV failed. Register 8 contains the DFSBCB return code.

**System action:** IMS terminates abnormally.

**Problem determination:** 4, 33

**Source:** IMS abends

**Module:** DFSXALC0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

## 0396

**Explanation:** This abend is a standard abend issued by module DFSXXCF0 during IMS initialization. A nonzero return code was returned after DFSBCB GET, IMODULE GETMAIN, DFSCIR, or IPOST was issued.

**Analysis:** 0396 is issued by module DFSXXCF0 during control region initialization. SVRB registers 8, 12, 14, and 15 contain values for diagnosing the error.

Register 15 contains an abend subcode. For all abend codes, register 12 is the program base and register 14 is the address where the error condition was recognized. The register 15 subcode is as follows:

**Codes****Explanation**

X'04'

DFSBCB GET QSAV failed. Register 8 contains the DFSBCB return code.

X'08'

DFSBCB GET LSAV failed. Register 8 contains the DFSBCB return code.

X'0C'

DFSCIR failed. Register 8 contains the DFSCIR return code.

X'10'

IPOST failed. Register 8 contains the IPOST return code.

X'14'

IMODULE GETMAIN failed. Register 8 contains the IMODULE return code.

**System action:** IMS terminates abnormally.

**Problem determination:** 4, 33

**Source:** IMS abends

**Module:** DFSXXCF0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

## 0400

**Explanation:** IMS initialization encountered an error processing the Repository section. Refer to the DFS4400E or DFS4401E message issued before the IMS abend to determine the cause of the error.

**System action:** IMS ends abnormally with abend code 0400. A DFS4400E or DFS4401E message is issued before IMS ends abnormally.

**System programmer response:** Check the DFS4400E or DFS4401E message for the reason of the abend. Fix the error and restart IMS.

**Source:** IMS abends

**Related information:**

 [DFS4400E \(Messages and Codes\)](#)

 [DFS4401E \(Messages and Codes\)](#)



---

## Chapter 13. IMS abend codes 0401 - 0500

This topic identifies and explains the following abend codes issued by IMS. Each code is 4 decimal digits.

---

### 0402

**Explanation:** This abend is issued when the intersubsystem interface module encountered a problem. An unauthorized program attempted to use the IMS SVC interface, an invalid call was passed to the SVC interface by IMS, or the IMS control region has terminated.

An abend 0402 might occur after an abend in a message region or a batch message region if the control region has terminated.

This user abend might also occur when an invalid PCB address is passed on a DL/I call. It can be interpreted as a Fast Path PCB when Fast Path is not in the system.

#### **Analysis: For DBFIRC10:**

The following subcodes in register 15 indicate that the Fast Path inter-region communication module (DBFIRC10) detected an error while processing a Fast Path DL/I call.

In the table below, register 8 at entry-to-abend points back to the subroutine that detected the error in DBFIRC10.

Key	Label	Description
Reg1=X'80000192' Reg15=X'140'		DBFIRC10 detected that the pointer to the PST IDENTIFY table (SCDIDTAB) was 0.
Reg1=X'80000192' Reg15=X'144'		DBFIRC10 was unable to find the IDENTIFY table entry for the PST issuing the call.
Reg1=X'80000192' Reg15=X'148'		DBFIRC10 detected that it received control in supervisor state.
Reg1=X'80000192' Reg15=X'152'		DBFIRC10 detected that it was running under a TCB whose key was not 8 (user key).

#### **For DFSDLPR0:**

0402 is a standard abend set at label AB402 in module DFSDLPR0. The following table shows the subcodes and the explanations:

Key	Label	Description
Reg1=X'C0000192'	AB402	
Reg14=BAL	(DFSDLPR0)	The control region subsystem vector table cannot be found.
Reg14=BAL	(DFSDLPR0)	DFSCPY00 PC number is not set.
Reg14=BAL Reg15=PC return code	(DFSDLPR0)	DFSCPY00 PC return code is non-zero.

#### **For DFSRRA00:**

0402 is a standard abend set at label RAAB0402 in module DFSRRA00 on return from DFSISI00. The following table shows the subcodes and the explanations:

Key	Label	Description
Reg1=X'C0000192'	RAAB0402	

## 0403

Key	Label	Description
SSOB+X'4C'='SVC#'	(ISIIDF in DFSISI00)	The identifying region's TYPE 2 SVC number does not match the IMS control region.
SSOB+X'4C'='VER#' Reg4=A(SSOB)	(ISIIDF in DFSISI00)	The identifying region's release number is not the same as that of the IMS control region.
Reg14=BAL Reg15=PC return code	(DFSDLPR)	DFSCPYP00 PC return code is non-zero.

**System action:** If IMS attempted to use the SVC, IMS terminates abnormally. If a user program attempted to use the SVC, the user program terminates abnormally.

**Programmer response:** If a user attempted to use the SVC, contact the IMS system programmer for assistance.

**Problem determination:** 1, 4, 11, or 35

**Source:** IMS abends

**Module:** DBFIRC10, DFSDLPR0, DFSRRA00

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

## 0403

**Explanation:** This abend is issued during batch processing. The OSAM batch interface to the I/O driver encountered an invalid parameter or control block associated with the I/O call. Register 14 in the abend SVRB registers will contain the address of the location that encountered the error.

**Analysis:** This is a standard abend issued by module DFSAOS70, the OSAM BATCH region I/O DRIVER front end. The program status word (PSW) at entry-to-abend points to the instruction in the routine at label SVCERRAB from which the abend (SVC 13) is issued. Register 14 in the abend SVRB registers is the BAL register to the abend routine and contains the address of the location from which control was passed. Comments there indicate which of the following error conditions was detected while parameter lists or control blocks were being validity checked.

- DEB check failed; the DCB points to an invalid DEB.
- The DEB does not contain the global IOSB/IOMA pool address.
- A zero or invalid IOSB sequence number was found in the local IOSB.
- The global IOSB does not point back to the local IOSB.
- The global IOSB is busy from a previous I/O request.
- The request function is zero or invalid.
- The request function specifies FORMAT LOGICAL CYLINDER or FORMAT PHYSICAL EXTENT. These functions are not allowed in the batch environment.
- The DEB pointer to the OSAM section is invalid.
- The extent number is invalid.
- The cylinder or head values in the seek field are invalid.
- The CCW operation codes are invalid.
- The virtual CCW address is invalid.
- The global IOMAs do not point to their respective local IOMAs.
- The global IOMAs are busy from a previous I/O request.
- A zero or invalid IOMA sequence number was found in the local IOMA.

**System action:** IMS is abnormally terminated.

**Programmer response:** Contact the IMS system programmer for assistance.

**Problem determination:** 1, 4, 11

**Source:** IMS abends

**Module:** DFSAOS70

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

**0404**

**Explanation:** This abend is issued by DFSQSPC0 user exit when a request to callable services for working storage fails. Module DFSQSPC0 is the IBM IMS Queue Control Facility for z/OS version of the Queue Space Notification Exit. Register 14 contains the callable services error code (CSPLRTRN) and register 15 contains the callable services return code.

**System action:** IMS terminates abnormally.

**Programmer response:** Contact the system programmer to analyze the return codes.

**Problem determination:** 1, 4

**Source:** IMS abends

**Module:** DFSQSPC0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

**0405**

**Explanation:** The dependent region was placed in a wait state by DFSQSPC0, the IBM IMS Queue Control Facility for z/OS version of the Queue Space Notification Exit. The wait state occurred based on IMS queue usage, which exceeded the user or default limits. The user-defined response in the IMS Queue Control Facility was to abend the application to stop the application from further IMS queue usage.

**System action:** The application terminates abnormally.

**Programmer response:** Analyze the application to determine why it is using excessive queue space, and make the necessary corrections.

**Source:** IMS abends

**Module:** DFSQSPC0

---

**0406**

**Explanation:** The dependent region was placed in a wait state by DFSQSPC0, the IBM IMS Queue Control Facility for z/OS version of the Queue Space Notification Exit. The wait state occurred based on IMS queue usage, which exceeded the user or default limits. While in this wait state, the dependent region was invalidly posted to by a source other than the IMS Queue Control Facility or a /PSTOP or /STOP command. Register 4 contains the post code.

**System action:** The dependent region terminates.

**System programmer response:** This abend occurred due to an internal error. Contact IBM Software Support for assistance.

**Source:** IMS abends

**Module:** DFSQSPC0

---

**0407**

**Explanation:** An entry in the message tree or region tree in DFSILTA0 was deleted twice.

**System action:** DFSILTA0 ends abnormally.

**System programmer response:** This abend occurred because of an internal error. Contact IBM Software Support for assistance.

**Source:** IMS abends

**Module:** DFSILTA0

---

## 0408

**Explanation:** The dependent region was placed in a wait state by DFSQSPC0, the IBM IMS Queue Control Facility for z/OS version of the Queue Space Notification Exit. The wait state occurred based on IMS queue usage, which exceeded the user or default limits. While the dependent region was in a wait state, a /STOP or /PSTOP command was entered against the region. See message DFS0528I for additional information.

**System action:** The application terminates abnormally.

**Programmer response:** Analyze the application to determine why it is using excessive queue space, and make necessary corrections.

**Source:** IMS abends

**Module:** DFSQSPC0

**Related information:**

 DFS0528I

## 0409

**Explanation:** The dependent region was placed in a wait state by DFSQSPC0, the IBM IMS Queue Control Facility for z/OS version of the Queue Space Notification Exit. The wait state occurred based on IMS queue usage, which exceeded the user or default limits. While the dependent region was in this wait state, an invalid IMS Queue Control Facility action was received.

**System action:** The application terminates abnormally.

**System programmer response:** This abend occurred due to an internal error in the IMS Queue Control Facility or IMS. Contact IBM Software Support for assistance.

**Source:** IMS abends

## 0411

**Explanation:** Open failed for a data set noted in message DFS0411I.

**Analysis:** 0411 is a standard abend issued by the log analysis module, DFSILTA0. The program status word (PSW) at entry-to-abend points to the instruction within label OPENWTO from which the abend (SVC 13) is issued.

See the text of message DFS0411I, which accompanies this abend, to determine the name of the data set for which the OPEN failed. A failure opening any one of the following data sets causes the abend: PRINTER, REPORT, LOGIN, HEADING, or LOGOUT.

All 0411 abends issued by this module are the result of a conditional branch to label OPENERR, following a test on the DCB open flags.

Register 4 in the abend SVRB registers is the primary base register.

Key	Label	Description
Reg1=Abend completion code, X'80000198'	FNDSTART	OPEN failed for the data set indicated in message DFS0411I, which accompanies this abend. Register 11 points to the data set in error. The DCBOFLGS field of the REPORT2, the REPORT-FILE, the LOGIN, the REPORT1, and the LOGOUT data sets respectively is tested for a successful OPEN. If OPEN was not successful, a branch is taken to label OPENERR to abend.
Reg11=address of DCB for which OPEN failed	FNDLOG	

**System action:** The log transaction analysis program (DFSILTA0) terminates abnormally.

**Programmer response:** See Message DFS0411I to identify the data set.

**Problem determination:** Check the JCL for the ddname in Message DFS0411I. Correct the JCL, and resubmit the job.

**Source:** IMS abends



**Module:** DFSILTA0

**Related information:**

 DFS0411I

### 0413

**Explanation:** The limit of the internal table of 255 application programs running at once has been exceeded. This is an IMS system error.

**Analysis:** 0413 is a standard abend issued by the log transaction analysis module, DFSILTA0. The program status word (PSW) at entry-to-abend points to the instruction within label P8PGMLP from which the abend (SVC 13) is issued.

DFSILTA0 reads the log records and, when a Type X'08' record (indicating application program scheduling) is detected, a branch is taken to label PROC8 to set up a program table (PGMTBL) entry. Log analysis steps through the table until an empty entry is found. Each entry is 19 bytes long, and the end of the table is indicated by a X'FF' in the high-order entry byte. Register 1 in the abend SVRB registers contains the abend completion code, X'80000191D'. Register 4 is the primary base register. Only type X'08' records for MPPs are used. Any type X'08' records from BMPs are ignored.

Key	Label	Description
Reg6=A(X'08 ' log record)	P8PGMLP	DFSILTA0 attempted to add another entry to the program table, and the table was full. During processing, register 1 points to the start of the table. However, it is later overlaid by abend processing with the completion code.

**System action:** The log transaction analysis program, DFSILTA0, terminates abnormally.

**Programmer response:** Ensure that the log data set used as input to program DFSILTA0 contains valid IMS log records.

**Problem determination:** 1, 2, 3, 4, 8, 25, 35

Save the storage dump of the abend and input log data sets.

**Source:** IMS abends

**Module:** DFSILTA0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

### 0415

**Explanation:** An invalid parameter was detected on the EXEC statement.

**Analysis:** Message DFS0408I or DFS0409I accompanies this standard abend, and identifies the invalid parameter.

**System action:** The log transaction analysis program, DFSILTA0, terminates abnormally.

**Programmer response:** Check the associated message to identify the invalid parameter.

**Problem determination:** Correct the JCL, and resubmit the job.

**Source:** IMS abends

**Module:** DFSILTA0

**Related information:**

 DFS0408I

 DFS0409I

## 0427

**Explanation:** A logical error occurred while processing a VSAM database. Refer to message DFS0427I for a detailed description. If DUMP=YES is specified on the DL/I buffer options statement, the IMS control region ends abnormally when the error is encountered.

If you have a program specification block (PSB) where one PCB has PROCOPT=L and another PCB has PROCOPT=I (or GIRD or A), the results are unpredictable and might lead to this abend.

**Analysis:** 0427 is a pseudoabend that is issued by DFSDVSM0 and DFSDVBH0. 0427 is a standard abend that is issued when the user codes DUMP=YES on the DL/I buffer options statement. Module DFSDVBH0 intercepts the pseudoabend 0427 and changes it to a standard abend.

A pseudoabend dump of the control blocks written on the log and the entry point address of this module (which can be found in the SCD) are needed.

SCDVSAM=DFSDVSM0

To determine why the abend was issued, refer to the DL/I trace table information in the IMS system definition information. Find the entry for this abend from the buffer handler trace table. To do that, locate an entry with the first byte equal to X'AB' and the second and third bytes equal to X'0427'. This entry is formatted as follows:

Disp.	Abend 0427	Trace entry
X'00'	DLTRFCTN	X'AB'
X'02'	ENTRY02	X'0427'
X'04'	ENTRY1 PSTFNCTN	Buffer handler function code.
X'05'	ENTRY1+1 RPLREQ	RPL Request
X'06'	ENTRY1+2 TRACE SEQ NO	Offset to abend
X'08'	ENTRY2	
X'0A'	ENTRY2+2 DSGINDA	
X'0B'	ENTRY2+3 DSGINDB	
X'0C'	ENTRY3 Reg8	RPLI
X'10'	ENTRY4	VSAM argument
X'14'	ENTRY5	VSAM area pointer
X'18'	ENTRY6	VSAM return code
X'19'	ENTRY6+1	VSAM error code
X'1A'	ENTRY6+2	VSAM request option
X'16'	ENTRY7	AMP pointer

There are several things that you need to use within this trace entry to determine where the error was detected: the VSAM return code, the VSAM error code, and the buffer handler function code.

For a description of the VSAM return and error codes, see *z/OS DFSMS Macro Instructions for Data Sets*. For a description of the buffer handler function codes, see the IMS system definition information.

**ENTRY1**

1-byte function code

**ENTRY2**

2-byte displacement into DFSDVSM0

**ENTRY6**

1-byte return code

**ENTRY6+1**

1-byte error code

The BALR was made to set up the pseudoabend from the 2 bytes at ENTRY2, the displacement into DFSDVSM0. Use this value to determine which label to use in the following table.

If an abend dump is needed to diagnose this problem, add the parameter DUMP=YES to the VSAM OPTIONS statement. This action causes an abend to be issued instead of a pseudoabend. The registers and explanations are shown in the following table:

Key	Label	Description
Reg14=BAL	PUT1600	A PUT request returned error code X'6D', which indicates that the VSAM index trap was activated.
Reg14=BAL	PUT4200	An attempt was made to retrieve a logical record that was added to a VSAM database. An unexpected return code was returned by VSAM, which caused the abend to be issued.
Reg14=BAL	PUT6300	A PUT request returned error code X'40', which indicates that there is no placeholder available.
Reg14=BAL	PUT9400	An attempt was made to add a logical record to a VSAM database. An unexpected return code was returned by VSAM, which caused the abend to be issued.
Reg14=BAL	ERSEERR1	An attempt was made to erase a logical record from a VSAM database. An unexpected return code was returned by VSAM, which caused the abend to be issued.
Reg14=BAL	SCHB4200	An attempt was made to search subpool buffers for an RBA in a VSAM database. An unexpected return code was returned by VSAM, which caused the abend to be issued.
Reg14=BAL	GET5020	An invalid local vector index was detected while testing for buffer validity by using coupling facility services. This problem caused the abend to be issued.
Reg14=BAL	GET5030	An invalid VSAM buffer prefix was found while testing for buffer validity by using coupling facility services. This problem caused the abend to be issued.
Reg14=BAL	GET9800	An attempt was made to retrieve a logical record from a VSAM database. An unexpected return code was returned by VSAM, which caused the abend to be issued.
Reg14=BAL	MRKBERR1	An attempt was made to mark a VSAM buffer ALTERED or to release ownership of a VSAM buffer. An unexpected return code was returned by VSAM, which caused the abend to be issued.
Reg14=BAL	WRTB7400	An attempt was made to write buffers to a VSAM database. An unexpected return code was returned by VSAM, which caused the abend to be issued.

**System action:** The application program terminates abnormally.


**Programmer response:** No action is required.

**Problem determination:** 4, 35

**Source:** IMS abends

**Module:** DFSDVSM0, DFSDVBH0

**Related reference:**

 z/OS: VSAM macro return and reason codes

**Related information:**

 DFS0427I (Messages and Codes)

Chapter 2, "IMS problem determination," on page 13

## 0428

**Explanation:** A batch message processing (BMP) step or a Fast Path (IFP) region could not be initiated because the PSB named in the third positional operand of the PARM field on the EXEC control statement was not defined at system definition time.

**Analysis:** 0428 is a pseudoabend issued from module DFSSBMP0.

Key	Label	Description
		This routine tries to find the PSB. Using the PSB name passed with the Subsystem Options Block (SSOB), a DFSCBTS find is done to locate the Program Specification Block Directory (PDIR). If the PDIR is not found, this abend is issued.

**Possible cause:** An incorrect PSB name was specified in the third positional operand of the PARM field on the EXEC control statement.

**System action:** The BMP or IFP region is terminated abnormally.

**Programmer response:** Get a list of PSBs defined at IMS system definition and find the correct name of the one desired. Correct the PSB name in the third positional operand of the PARM field on the EXEC statement, and resubmit the BMP for execution.

**Problem determination:** 2, 3, 10

**Source:** IMS abends

**Module:** DFSSBMP0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

## 0430

**Explanation:** Module DFSDVBI0 determined that the DL/I database buffering services function cannot be initialized. Message DFS0430I is issued and the reason code in the message defines the reason for the failure.

**Analysis:** 0430 is a standard abend issued by DFSDVBI0, the module for DL/I VSAM pool initialization. The program status word (PSW) at entry-to-abend points to the instruction within label ABEND430 from which the abend (SVC 13) is issued. This routine is unconditionally branched to by the routine that detected the error.

Register 12 in the abend SVRB registers is the primary base register; register 11 is a secondary base register. Register 13 contains the address of the save area in the save set.

Before this abend is issued, one or more of the following IMS messages can be issued to indicate the reason for failure: DFS0430I, DFS0431I, DFS0432I, DFS0436I, or DFS0438I. Another possible reason is that the DFSVSAMP data set was not provided.

If a routine within DFSDVBI0 issues a VSAM macro that failed, the return code is in register 15 and also in the accompanying IMS information message. A list follows, showing the VSAM macros issued and their return codes. See *z/OS DFSMS Macro Instructions for Data Sets* for an explanation of the macros and a more detailed description of the return codes.

**Codes****Explanation**

**X'00'**

VSAM completed the request.

**X'04'**

A resource pool already exists in the partition or address space (LSR) or in the system (GSR).

**X'08'**

Insufficient virtual storage space to satisfy the request (GETMAIN or ESTAE failed).

**X'0C'**

Buffers cannot be fixed in real storage (PAGEFIX failed).

**X'10'**

TYPE=GSR is specified, but the program that issued BLDVRP is not in supervisor state with key 0 or 7.

**X'11'**

A GETMAIN failure occurred while processing either DBD statements for the specific subpool/shared-pool ID table, or POOLID statements for the specific shared pool ID table.

**X'14'**

STRNO is less than 1 or greater than 255. Hiperspace™ buffering is specified on a subpool size less than 4K bytes, or is unavailable because of insufficient expanded storage for the subpool size, which is specified as required.

**X'18'**

BUFFERS is specified incorrectly (a *size* or *number* is invalid).

Following are the return codes from the *DLVRP* macro:

**Codes****Explanation****X'00'**

VSAM completed the request.

**X'04'**

There is no resource pool to delete.

**X'08'**

There is not enough virtual-storage space to satisfy the request (GETMAIN or ESTAE failed).

**X'0C'**

There is at least one open data set using the resource pool.

**X'10'**

TYPE=GSR is specified, but the program that issued DLVRP is not in supervisor state with protection key 0 to 7.

If a routine within DFSDVBI0 issues an IMODULE LOAD that has failed, the return code is also be in register 15.

Key	Label	Description
DVBISW $\neg$ = X'40' DVBISW=X'10' Reg14=BAL to WTO	POOLCK05	If VSAM subpools are required, the switch DVBISW is tested to determine if the subpools have been <i>requested</i> . If this switch is not set, but the switch is on indicating VSAM subpools are <i>required</i> , a branch is taken to label ABEND430 to abend. Message DFS0430I (reason code 02) accompanies the abend.
Reg14=BAL to WTO Reg15=nonzero return code from VSAM BLDVRP macro	BLDVP080	The VSAM BLDVRP macro is issued to build the VSAM shared resource pools. Register 15 contains the return code; if other than zero, IMS message DFS0432I is issued with the reason code, and a branch is taken to label ABEND430 to abend. Message DFS0430I (reason code 06) also accompanies the abend.
DVBISW $\neg$ = X'80'	BLDVP100	The DVBISW switch is tested—if the buffer handler pool was <i>not</i> built, and if VSAM subpools were requested but not built, a branch is taken to label MDL0D040, and then to label ABEND430 to abend. If the buffer handler pool was <i>not</i> built, and if VSAM subpools were requested <i>and</i> built, a VSAM DLVRP macro is issued to delete the pools, and a branch is taken to label MDL0D040, then to label ABEND430 to abend. In both cases, IMS message DFS0430I (reason code 09) accompanies the abend.

## 0432

Key	Label	Description
Reg9=address of SCD Reg14=BAL to WTO Reg15=nonzero return code from IMODULE	MDLOD060	An IMODULE LOAD is done for the DL/I buffer handler, DFSDVBH0. The return code is loaded into register 15. If the LOAD was unsuccessful (if register 15 is other than zero), a branch is taken to label MDLOD150, IMS message DFS0438I is issued with the module name and return code, and a branch is taken to label ABEND430 to abend. IMS message DFS0430I (reason code 07) also accompanies the abend.
Reg9=address of SCD Reg14=BAL to WTO Reg15=nonzero return code from IMODULE	MDLOD070	An IMODULE LOAD is done for the DL/I VSAM Interface, DFSDVSM0. The return code is loaded into register 15. If the LOAD was unsuccessful (if register 15 is other than zero), a branch is taken to label MDLOD150, IMS message DFS0438I is issued with the module name and return code, and a branch is taken to label ABEND430 to abend. IMS message DFS0430I (reason code 07) also accompanies the abend.
Reg8=address of DCB Reg14=BAL to WTO	DCBERR	While attempting to read the control statement data set (DFSVSAMP), an I/O error was encountered. IMS message DFS0436I is issued and a branch is taken to label ABEND430 to abend. IMS message DFS0430I (reason code 05) also accompanies the abend.

**System action:** IMS terminates abnormally. One or more messages are issued to indicate the reason for the failure.


**Programmer response:** See message DFS0430I to determine the cause of problems.

**Problem determination:** 1, 2, 3, 18, 19

**Source:** IMS abends

**Module:** DFSDVBI0

**Related reference:**

 z/OS: VSAM macro return and reason codes

**Related information:**

 DFS0430I

Chapter 2, "IMS problem determination," on page 13

## 0432

**Explanation:** A batch message processing (BMP) program step could not be scheduled. The requested PSB was defined as a teleprocessing program, but the BMP step is message-driven, and the parallel option was specified.

**Analysis:** 0432 is a pseudoabend issued from module DFSSBMP0.

Key	Label	Description
	BMPSCHDC	A check is made to see if the PSB is for a batch Type 2 program. If it is not, and the parallel option is specified, the abend is issued.

**Possible cause:** Incorrect program names have been specified in the PARM field on the EXEC control statement.

**System action:** The BMP region is terminated abnormally.

**Programmer response:** Get a list of program names generated at IMS system definition and find the correct name of the program to be executed. Then, correct the program names in the PARM field of the EXEC statement, and resubmit the BMP for execution.

**Source:** IMS abends

**Module:** DFSSBMP0

---

**0436**

**Explanation:** A batch message processing (BMP) program step could not be scheduled because the input symbolic queue named in the fourth positional operand of the PARM field on the EXEC control statement was not defined to system definition.

**Analysis:** 0436 is a pseudoabend detected by DFSSBMP0 and issued by DFSPCC20.

Key	Label	Description
	BMPRSPSB	Either an input symbolic queue name was specified and the subsystem type was DBCTL, or the destination finder (DFSICLF0) was unable to find the SMB. PSTSYMBO contains the input symbolic queue name passed to DFSICLF0.

**Possible cause:** The input symbolic queue specified in the fourth positional operand of the PARM field on the EXEC control statement was not defined at system definition. If the subsystem is DBCTL, message driven BMPs are not allowed and thus the input symbolic queues were not defined.

**For DFSPCC20:**

The IMS scheduler has determined that a BMP step could not be scheduled because an input transaction name was invalid.

**Possible cause:** The input transaction name specified in the fourth positional operand of the PARM field on the EXEC control statement was not defined at system definition time.

**System action:** The BMP region is terminated abnormally.

**Programmer response:** Get a list of input symbolic names defined at IMS system definition, and correct the name. Then, correct the input symbolic queue named in the EXEC statement, and resubmit the BMP for execution.

For Database Control (DBCTL) subsystems, message-driven BMPs are not allowed and you should not specify an input symbolic queue name.

**Source:** IMS abends

**Module:** DFSSBMP0, DFSPCC20

---

**0437**

**Explanation:** IMS resources (transaction, PSBs, and LTERMs) that are not valid for this dependent region are used. A security violation occurred when a dependent region tried to use IMS resources (transaction, PSBs, and LTERMs) that are not protected by Resource Access Security (RAS).

**Analysis:** 0437 is a pseudoabend issued by module DFSASK00, DFSDASP0, or DFSDASS0.

**Possible Cause:** A security violation might have occurred.

**System action:** The dependent region is terminated.

**Programmer response:** Ensure that the dependent regions have the proper access to the IMS resources.

**Problem determination:** 1, 4, 6, 35

**Source:** IMS abend

**Module:** DFSASK00, DFSDASP0, DFSDASS0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

**0438**

**Explanation:** The ODBA user request to schedule the PSB named on the APSB call failed SAF RACROUTE processing.

**Analysis:** The APSB request is unsuccessful. Message DFS2855A is issued if SAF RACROUTE AUTH call was not successful.

**System action:** The APSB request is unsuccessful.

## 0440 • 0442

**Programmer response:** No action is required.

**Problem determination:** 1, 2, 4, 6, 11, 35, 36

**Source:** IMS abends

**Module:** DFSDASP0

**Related information:**

 DFS2855A

Chapter 2, "IMS problem determination," on page 13

---

### 0440

**Explanation:** A batch message processing (BMP) program step could not be initiated because the input symbolic queue named in the fourth positional operand of the PARM field on the EXEC control statement was not a transaction code.

**Analysis:** 0440 is a pseudoabend issued from module DFSSBMP0.

Key	Label	Description
	BMPRSPSB	The input destination is being verified. The destination is not an SMB; therefore, the abend is issued.

**Possible cause:** Input symbolic queue specified in the fourth positional operand of the PARM field on the EXEC control statement was not specified as a transaction code.

**System action:** The BMP region is terminated abnormally.

**Programmer response:** Get a list of transaction codes defined at IMS system definition, and select a valid code. Then, change the input symbolic queue name on the EXEC statement to a transaction code, and resubmit the BMP for execution.

**Source:** IMS abends

**Module:** DFSSBMP0

---

### 0442

**Explanation:** Before writing a VSAM control interval, the Record Definition Field (RDF) and the Control Interval Definition Field (CIDF) were checked and found invalid or inconsistent with the record format required by IMS. Multiple RDFs describing records of different lengths is an example of an inconsistent record format and is also a cause of pseudoabend 0442.

This error can occur if the database was loaded or updated using a database description (DBD) with a specific record length, and later updated or loaded using a changed DBD with a different record length.

**Analysis:** 0442 is a pseudoabend that is issued by module DFSDVSMO. The registers at the time of abend are saved at RPLIERMA + X'40'.

**Possible cause:** See the text for DFS0442A, which accompanies this abend, to get the DBD and DDNAME that caused the error.

Key	Label	Description
	STLEGL00	After a locate logical record by key call is issued, a check is made to see if the CI is bad. If so, abend U0442 is set and a branch to label L9990 is created. In this routine, the return to caller is made.
	S1800	A get to VSAM is issued. After the return from VSAM, the PST flag for a bad CI is checked. If it is on, the abend 0442 is set and a branch to S9990 is made. S9990 resets the caller registers and branches to the caller.
	Y5220	This JRNAD routine was driven because a buffer needed to be written out. A check is made to see if the CI is bad. If so, no abend 0845 is issued and a return to caller is made.



Key	Label	Description
	Y8020	This routine determines if a CI split is in progress. A check is made to see if the CI is a KSDS index CI. If so, the bad CI flag is reset. If the CI is not a KSDS index CI, then a check is made to see if the CI is bad. If so, the bad CI flag is set and the MSGDFS0442A is issued.

**System action:** The application program terminates abnormally.

**Programmer response:** Refer to message DFS0442A to determine the cause of the problem.

**Source:** IMS abends

**Module:** DFSDVSMO

**Related information:**

 DFS0442A

#### 0444

**Explanation:** A batch message processing (BMP) program step could not be scheduled because the output symbolic queue named in the fifth positional operand of the PARM field on the EXEC statement was not defined to system definition.

**Analysis:** 0444 is a pseudoabend detected by DFSSBMP0 and issued by DFSPCC20.

Key	Label	Description
	BMPRSPSB	Either an output symbolic queue name was specified and the subsystem type is DBCTL or the destination finder (DFSICLF0) was unable to find the communication name table (CNT). PSTSYMBO contains the output symbolic queue name passed to DFSICLF0.

**Possible cause:**

- The output symbolic queue specified in the fifth positional operand of the PARM field on the EXEC control statement was not defined at system definition. If the subsystem is DBCTL, message driven BMPs are not allowed and thus output symbolic queues were not defined at system definition.
- A remote LTERM (CNT) was specified in the OUT parameter on the EXEC statement of the BMP JCL, but is not allowed.

**System action:** The BMP region terminates abnormally.

**Programmer response:** Get a list of logical terminal names and transaction codes defined at IMS system definition, and select the right one. Then, correct the output symbolic queue name on the EXEC statement, and resubmit the BMP for execution. You cannot specify a remote LTERM (CNT) on the OUT parameter.

If the subsystem is Database Control (DBCTL), message-driven BMPs are not allowed and you should not specify an output symbolic queue name.

**Problem determination:** None.

**Source:** IMS abends

**Module:** DFSSBMP0

#### 0448

**Explanation:** A batch message processing (BMP) program step could not be initiated because the input transaction code named in the third positional operand of the PARM field on the EXEC control statement specified a remote SMB.

**Analysis:** 0448 is a pseudoabend issued from module DFSSBMP0. Message DFS554A accompanies this abend.

Key	Label	Description
	BMPRSPSB	While verifying the input destination or attempting to reschedule a BMP after intent problems have been resolved, it was found that the SMB is remote. A batch type 2 region cannot schedule an input SMB defined as remote.

**Possible cause:** The transaction code name in the third positional operand of the PARM field on the EXEC control statement is for a remote SMB.

**System action:** The BMP region terminates abnormally.

**Programmer response:** Check the transaction codes defined at IMS system definition. Then, select a transaction code that is not a remote SMB and resubmit the BMP for execution.

**Source:** IMS abends

**Module:** DFSSBMP0

**Related information:**

 DFS554A (Messages and Codes)

## 0451

**Explanation:** A physical input or output error occurred on a database data set. Message DFS0451I or DFS0451A was issued.

When the PSB for the program that terminated abnormally was generated, the IOEROPN parameter was specified on the PSBGEN macro statement. Either IOEROPN=451 was specified and the step is terminating after an I/O error, or IOEROPN=(n,WTOR) was specified and the operator responded ABEND to the DFS0451A message.

**Note:** If n=451, IMS terminates with abend 0451 even if the operator responds CONT to the DFS0451A message.

**Analysis:** Message DFS0451I or DFS0451A is issued, indicating the module detecting the error, the DBD name, and the ddname of the failing data set. This is a pseudoabend unless one of the following is true:

1. If "DUMP=YES" is coded on the DL/I buffer options statement, module DFSDVBH0 intercepts pseudoabend 0451, and changes it to a standard abend at label VBH050.
2. If "IOEROPN=(n,WTOR)" is coded on the PSBGEN statement, and "n"=451, a standard abend 0451 is issued from module DFSDVBH0 at label PURGE090. This happens even though the operator responds "CONT" to message DFS0451A.

**System action:** The application program terminates abnormally with pseudoabend 0451, and IMS continues processing. If DUMP=YES was specified on the DL/I buffer options statement, the application terminates with standard abend 0451.

**Programmer response:** See message DFS0451I or DFS0451A.

**Problem determination:** 1, 2, 3, 5, 8, 11, 17a, 17d, 17g, 20, 35

**Source:** IMS abends

**Module:** DFSDBH40, DFSDISM0, DFSDVBH0, DFSDVSM0

**Related information:**

 DFS0451I

 DFS0451A

Chapter 2, "IMS problem determination," on page 13

## 0452

**Explanation:** A batch message processing (BMP) program step could not be initiated because the transaction named in the fourth positional operand of the PARM field on the EXEC control statement has been stopped or locked by a command or by a prior program failure.

**Analysis:** 0452 is a pseudoabend issued from module DFSSBMP0. Message DFS554A accompanies this abend.

Key	Label	Description
	BMPRSPSB	While verifying the input destination or attempting to reschedule a BMP after intent problems have been resolved, it was found that the SMB is locked or stopped, or is already scheduled on another PST.

**System action:** The BMP region terminates abnormally.

**Programmer response:** Start the transaction again and resubmit the job.

**Source:** IMS abends

**Module:** DFSSBMP0

**Related information:**

 DFS554A (Messages and Codes)

#### 0453

**Explanation:** A BMP program step cannot be initiated because the transaction code specified in the fourth positional operand of the PARM field on the EXEC statement is an LU 6.2-driven dynamically built transaction or an IMS conversational transaction.

**Analysis:** 0453 is a pseudoabend issued by DFSPCC20 and DFSSBMP0.

Key	Label	Description
Reg1=X'800001C5'	BMPRSPSB	A conversational transaction was scheduled in a BMP.

**System action:** The BMP region is terminated abnormally.

**Programmer response:** Ensure that the transaction code specified in the fourth positional operand of the PARM field on the EXEC control statement is not a LU 6.2-driven dynamically built transaction or an IMS conversational transaction.

**Source:** IMS abends

**Module:** DFSSBMP0, DFSPCC20

#### 0454

**Explanation:** One of the following occurred:

- An attempt was made to schedule a dependent region after a /CHE FREEZE or /CHE DUMPQ command was issued.
- A BMP or MPP was started after a shutdown checkpoint had already quiesced and stopped all other active regions.
- A BMP or IFP scheduling detected an IRLM that is not available and scheduling cannot continue.
- A BMP or IFP scheduling that was prestarted on the alternate IMS in an XRF environment has been terminated by a /STOP BACKUP command. Scheduling did not complete.
- BMP or IFP scheduling that was prestarted on the alternate IMS in an XRF environment has been terminated by a /STOP BACKUP command issued on the alternate IMS or /CHE FREEZE command issued on the active IMS and PSDEPAB=YES in the DFSHSBxx member. Scheduling did not complete.

**Analysis:** 0454 is a pseudoabend passed by module DFSASK00 to module DFSPCC20, the module that issues the abend, when PSDEPAB= either specifies or defaults to YES in the DFSHSBxx PROCLIB member.

**For DFSRRA00:**

One of the following occurred:

- An attempt was made to schedule a dependent region after a /CHE FREEZE or /CHE DUMPQ command was issued.
- A BMP or MPP was started after a shutdown checkpoint had already quiesced and stopped all other active regions.

0454 is a pseudoabend passed by module DFSASK00 to module DFSRRA00, which issues the abend.

## 0455 • 0456

**System action:** The dependent region is terminated abnormally.

**Programmer response:** Resubmit the job when IMS is started or restarted.

**Source:** IMS abends

**Module:** DFSPCC20

---

### 0455

**Explanation:** Abend 0455 is issued by DFSSBMP0, which detected a Java application scheduling into a BMP region.

**System action:** The Java application must be executed in a Java region.

**Programmer response:** This psuedoabend is issued to stop scheduling because a Java application cannot schedule in a BMP region. Schedule the Java application in a Java region.

**Source:** IMS abends

**Module:** DFSSBMP

---

### 0456

**Explanation:** A DL/I batch region, a batch message processing (BMP) program step, or a Fast Path (IFP) region could not be initiated. The problem occurred because the PSB named in the third positional operand of the PARM field on the EXEC control statement has been stopped or locked by a command or by a prior program failure.

This abend can also occur when:

- A batch region, BMP, or IFP attempts to reference a PSB that is in NOTINIT status. Some possible causes for NOTINIT status are:
  - The PSB was not in the ACBLIB during initialization.
  - The PSB references a DBD that is not defined to IMS.
  - BLDL failed for the PSB.
- The PSB referenced was defined as DOPT and was found in the first concatenation of ACBLIB.
- A Fast Path PCB referenced a DBD that was not found in the system.
- A main storage database (MSDB) referenced by a Fast Path PCB could not be loaded during IMS initialization because member DBFMSDBX was not found in PROCLIB or that MSDB was not specified in member DBFMSDBX.
- A DEDB randomizing module referenced by a Fast Path PCB could not be loaded during IMS initialization.
- A PSB could not be scheduled because the DMB pool is too small, and the PSB requires a DMB that is exclusively allocated to another PSB.
- The PROCOPT specified is incompatible with the calls being issued.
- You have tried to use ACB members created or updated on an earlier version of IMS on a later version of IMS.

**Analysis:** 0456 is a pseudoabend issued from module DFSSBMP0. Message DFS554A might accompany this abend.

Key	Label	Description
	BMPYPOK SBMP0320 BMPERRBT	This abend is issued when the PSB for a batch type 2 program was found, but the PSB is locked and unavailable.
	BMPERRBT	The PSB failed scheduling because an invalid PSB processing option was specified.
	BMPERRBT	The PSB failed scheduling because an I/O error occurred for a DOPT PCB, and the PSB was not available.
	BMPERRBT	An embedded EOF was found in ACBLIB. This problem typically occurs when compressing the ACBLIB while online IMS is running.

**System action:** The batch region or BMP or IFP region is terminated abnormally. The missing PSB was noted in message DFS830I, which was issued during initialization. DFS563I might also be issued during initialization.

**Programmer response:** Analyze why the PSB is locked or stopped. After the problem that caused the failure has been corrected and the PSB has been unlocked, resubmit the job.

To unlock the PSB for a BMP program or an IFP non-message-driven or a DEDB utility program, issue a /START PROGRAM command, with the PSB name. Then resubmit the job.

**Source:** IMS abends

**Module:** DFSSBMP0

**Related information:**

 DFS554A

#### 0457

**Explanation:** A batch message processing (BMP) region, a Fast Path (IFP) region, a CICS thread, or an ODBA thread was started for a PSB already scheduled.

In a shared-queues environment where Resource Manager (RM) and sysplex serial program management (SSPM) are active, this abend code might be issued if a serial PSB is in use by an IMS region or thread and the control region fails. The serial PSB cannot be scheduled in other members of the IMSplex until the failing control region is restarted.

**Analysis:** 0457 is a pseudoabend that is issued from module DFSSBMP0.

Key	Label	Description
	BMPTYPOK	The PSB is already scheduled in another region, and no parallel scheduling was specified.

**System action:** The BMP region, IFP region, CICS thread, or ODBA thread was terminated abnormally.

**Programmer response:** Determine the reason for the duplicated PSB specification. If this is not a duplication of a current job, resubmit the job when the current job is complete, or correct the PSB with a parallel scheduling option at IMS system definition.

**Source:** IMS abends

**Module:** DFSSBMP0

#### 0458

**Explanation:** A batch message processing (BMP) program step or a Fast Path (IFP) region was not initiated because one of the Fast Path databases used by the PSB named in the third positional operand of the PARM field on the EXEC control statement was stopped or locked by a prior program failure.

If a Fast Path online utility is initiated, the database named in the second positional operand of the PARM field on the EXEC control statement has been stopped or has not been defined as a DEDB in the DBDGEN.

**Analysis:** 0458 is a pseudoabend issued from module DFSSBMP0. Message DFS554A accompanies this abend.

When a FAST PATH online utility region is initiated, this abend indicates that the database named in the EXEC Parm= parameter (that is, the second positional parameter) has been stopped or has not been defined as a DEDB in the DBDGEN.

**System action:** The BMP or IFP region is terminated abnormally.

**Programmer response:** Determine the reason for the Fast Path database being stopped. After the problem that caused the database to be stopped has been corrected, resubmit the job.

**Source:** IMS abends

**Module:** DFSSBMP0

**Related information:**

 DFS554A (Messages and Codes)

#### 0462

**Explanation:** An application program was scheduled in a message region and terminated without successfully issuing a GET UNIQUE for an input message. The application program did successfully process at least one other call.

**Analysis:** 0462 is a standard abend detected by module DFSASK00 in the normal thread termination routine.

## 0474 • 0475

Key	Label	Description
Related log records	Type X'08'	The transaction and program were scheduled.
	Type X'07'	Application termination record: <ul style="list-style-type: none"> <li>• If the field copied from DLRGUMES is greater than zero, a GU was issued by the application program.</li> <li>• If the field copied from DLRGU1 is greater than zero, a GUs was successful. Abend 0462 indicates that this field was equal to zero at termination time.</li> </ul>

**System action:** The application program is abnormally terminated, and the PSB and the SMB are stopped.

**Programmer response:** Determine the problem in the user message processing program, correct it, and resubmit the job.

Correct the application program so that a GET UNIQUE is issued.

**Source:** IMS abends

**Module:** DFSASK00

### 0474

**Explanation:** An application program has been terminated through the use of the /STOP REGION ABDUMP or /STOP REGION CANCEL command.

**Attention:** Abend 0474 can also occur if you issue a /STOP REGION CANCEL command that is preceded by a /STOP REGION ABDUMP command.

**Analysis: For DBFSYN00:**

0474 is a standard abend issued by DBFSYN00 after detecting a /STOP REGION ABDUMP command.

Key	Label	Description
Reg10=address of EPST		The EPSTSTAB flag was set on by DBFICL40 after the operator entered a /STOP REGION ABDUMP command.

**For DBFIRC10, DFSASK00, DFSCPY00, DFSDASC0, DFSDCPY0, DFSTMAD0:**

Abend 0474 is a pseudoabend issued by DBFIRC10, DFSASK00, DFSCPY00, DFSDASC0, DFSDCPY0, and DFSTMAD0 after detecting a /STOP REGION ABDUMP command.

**For DFSDLTR0, DFSUICC0:**

0474 is an abend issued by DFSDLTR0 and DFSUICC0 after detecting a /STOP REGION ABDUMP command.

**For DFSDVBH0:**

0474 is an abend issued by DFSDVBH0 after detecting a /TERMINATE OLREORG OPTION(ABORT) command.

**System action:** The region is abnormally terminated in response to a user request.

| In the case of an MPP where an abendU0474 occurs, the TLIM count will not be incremented. An abendU0474 should  
| not be included in the TLIM count for MPPs.

**Source:** IMS abends

**Module:** DBFSYN00, DBFIRC10, DFSASK00, DFSCPY00, DFSDASC0, DFSDCPY0, DFSDLTR0, DFSDVBH0, DFSTMAD0, DFSUICC0

### 0475

**Explanation:** A Java application program attempted to run as an IMS batch job. IMS batch does not support Java application programs.

**System action:** The batch program terminates abnormally.

**Source:** IMS abends

### 0476

**Explanation:** A DL/I call did not include a valid PCB address. The PCB address in the DL/I call (second parameter or third, if count is the first parameter) is not one of the PCB addresses passed to the application program at its entry point.

Some possible causes for this failure are:

- An address for an AIB was provided but the first eight bytes of the block do not equal 'DFSAIB'.
- The PSB language specified is not the same as that of the application program.
- The call requires a PCB address, but the function is the last entry in the call list.
- The address of the PCB in the call is not one of the PCB addresses passed to the application at its entry point.
- An online program issued an UNLD call. The UNLD call is an internal DL/I call and should never be issued by an application program.

This abend can be the result of an improperly assembled or generated application program.

**Analysis: For DBFIRC10:**

0476 is a pseudoabend issued from module DBFIRC10.

Key	Label	Description
Reg1=address of offset in PCB verify list Reg3=address of EPCB Reg7=address of PCB verify list		The address at the EPCB offset in the EPCB verified that the list did not match the EPCB passed in the call.
Reg1=address of EPCB		The EPCB does not point to the PCB address in the call.

*Possible cause:* The application program might be in error.

**For DFSCDLI0:**

In the DL/I call list, the address for an AIB was provided. However, the AIB is invalid. The first 8 bytes of the block do not equal 'DFSAIB'.

*Possible cause:* The application program might be in error.

**For DFSCPY00:**

0476 is a pseudoabend detected by the interregion copy facility module DFSCPY00, and issued by DFSPCC20. TCBFSA+X'18' is the ECP address and ECP+X'104' is the user call list address. The PCB address is the second word in the list.

**For DFSDLA00:**

0476 is also detected by the DL/I call analyzer, DFSDLA00. To find the applicable registers, use the SNAP dump of the control blocks issued to the log. Locate the save area of the call analyzer by getting the entry point of DFSDLA00 from the SCD (at label SCDDLICT), and searching down the save area sets for that address in register 15. This save area contains the registers as they were when the call analyzer was called.

When an error condition is detected, a conditional branch is taken to label ERR22, from which pseudoabend code, X'100001DC' is moved into the PSTABTRM field of the PST.

Registers 14 through 11 are stored in the save set at PSTSAV15. Register 12 in the save set contains key information to determine the label from which the branch was taken.

Key	Label	Description
Reg8=X'80xxxxxx', where xxx is the address of an entry in the parm list PSTSAV15 Reg12=X'xxxxAA01'	TESTPCB0	Register 8 is loaded and tested. If register 8 contains the end-of-list indicator (high-order bit on), the last parameter in the parameter list was a "function" parameter and no PCB address was passed in the parameter list. The routine then branches to label ERR22 to issue the abend.
Reg3 = pointer to user PCB Reg10=Reg3=0 PSTSAV15 Reg12=X'xxxxAA02'	TESTPCB	Register 10 is loaded from register 3, then tested. If register 10 is zero, a PCB address of zeros was passed from the parameter list. The routine then branches to label ERR22 to issue the abend.
Reg3=user-passed PCB address PSTSAV15 Reg5=address of the last PCB in list Reg12=X'xxxxAA03' Reg14=PCB list address	VALIDCK3	The PCB validity check routine scanned the PCB list and found an entry of X'80xxxxxx', indicating an end-of-list. However, the user-supplied PCB address is not a valid PCB address, so the routine branches to label ERR22 to issue the abend.
PSTSAV15 Reg12=X'xxxxAA04'	PSEUDOCCB	An UNLD call was issued online.

**Possible cause:**

- The PSB language specified was not the same as the application program.
- The parameter count in the DL/I call was invalid.
- An invalid PCB was detected by IMS. For example,
  - The entry point to the application program may not be DLITPLI or DLITCBL. It must be one of the two.
  - The CMPAT parameter may be specified incorrectly. If the application program is designed to use the IOPCB, CMPAT=YES should be specified in the PSBGEN statement to enable the program to run in the DL/I batch region.
  - DCB information may be incorrect on the GSAM input data set DD statement.
- An invalid DL/I function code was detected in the DL/I call parameter list.

**For DFSECP10:**

0476 is also a pseudoabend detected by the interregion copy facility module, DFSCP00, and issued by DFSECP10. A copy of DFSECP is moved into module DFSFSWA0, and is identified by the character string 'ECP'. Module DFSFSWA0 can be located through the CDE entries of the dependent region dump. The call list address passed by the application program is located at X'104' into DFSECP. The PCB address is in this list, and should be corrected in the application program.

This abend can also be detected and set up by the Language Interface Extensions, DFSLIE00 and DFSLIE20. If DFSLIE00 detects an invalid PCB address, or if DFSLIE20 detects an invalid AIB address, it sets up the pseudoabend that is eventually issued by DFSECP10. As above, DFSECP is copied into module DFSFSWA0, which can be located through the CDE entries of the dependent region dump.

**For DFSECP20:**

0476 is also issued from module DFSECP20.

Key	Label	Description
R14=BAL	EC2ABEND	Either an invalid PCB or AIB address was passed in the DL/I call parameter list. DFSLIE00 or DFSLIE20 detects this condition and sets up the pseudoabend. When DFSECP20 detects the abend, it branches to a common routine to issue the abend. R14 points to the instruction following the branch to EC2ABAP0.



**Possible cause:** An application program made a GSAM call, but passed an invalid GSAM PCB address. When DFSPR0X0 detects the abend, an application has issued a DL/I call pointing to a parameter list with a PCB address equal to zero.

**For DFSPR000:**

0476 is issued from module DFSPR000.

Key	Label	Description
Reg2=A(DFSPRPX0) Reg5=A(PSBCODE) in the PSB		The GSAM PCB passed in the user call list did not match one of the GSAM PCBs in the GSAM section of the PSB PCB list. DFSPRPX0 at label PCGPSBL points to the first GSAM PCB in the PSB PCB list. Register 4 points to a one-byte field in the PSB that contains the language flag of the PSB. If this flag is PL1, the PSB PCB list contains pointers to pointers (dope vectors) to the PCB.

**Possible cause:** An application program made a GSAM call, but passed an invalid GSAM PCB address.

**System action:** The application program terminates abnormally.

**Programmer response:** Correct the parameter list and resubmit the job.

- Change the PSB language specified to that of the application program.
- Correct the application program DL/I call statements.

**Problem determination:** 4, 17d

**Source:** IMS abends

**Module:** DBFIRC10, DFSCDLI0, DFSCPY00, DFSDLA00, DFSECP10, DFSECP20, DFSPR000

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

## 0477

**Explanation:** Because the primary area to build FLD blocks was not large enough, an IGETBUF macro was issued to obtain a 8192-byte area for FLD blocks. The area was not available. In batch processing, a GETMAIN obtains the space. In online processing, the space is obtained from the database working pool.

**Analysis:** Abend 0477 is a pseudoabend detected by the DL/I call analyzer, DFSDLAS0. To find the applicable registers, use the SNAP dump of the control blocks issued to the log. Locate the save area of the call analyzer. This can be done by getting the entry point of DFSDLA00 from the SCD (at label SCDDLICT), and searching down the save area sets for that address in register 15. This save area contains the registers as they were when the Call Analyzer was called. The next lower save area contains the registers at entry-to-abend, with the exception that Register 2 contains the abend completion code, X'00001DD'.

For this pseudoabend, the PSTABTRM field in the PST is also set to a X'10' followed by the abend code. The actual abend is issued by one of the program request handlers. The following table shows the register information and explanations:

Key	Label	Description
Reg15=nonzero return code from IGETBUF	GETOVER	Within the routine to obtain the overflow FLD work area is a routine labeled GETOVER. Within the SCD at label SCDSMMGB is the entry point address of the EGETBUF module. The Call Analyzer BALRs to IGETBUF, and register 15 is loaded with a return code. When tested, if register 15 is not zero, indicating that storage was not obtained, the abend is issued.
Reg15=nonzero return code from Buffer Handler and PSTRDCDE ≠ X'04' or X'18'	ABC00477	When the second pass through a STAT call with a GA return code is made, the buffer handler detects an error and the PSTRDCDE, when compared, indicates that the RBN was beyond the data set. The abend is issued.

## 0478 • 0479

**Possible cause:** For batch, the address space was not large enough. For online, the database working pool was too small.

**System action:** The application program is terminated abnormally.

**Programmer response:** For batch processing, provide a larger address space. For online processing, provide a larger database working pool.

**Problem determination:** 13, 17a, 17b

**Source:** IMS abends

**Module:** DFSDLA00, DFSDLAS0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

### 0478

**Explanation:** A DL/I call exceeded the maximum allowable number of qualification statements which make up the segment search arguments on a single DL/I database call. The maximum number of qualification statements for a single DL/I call is 1024.

**Analysis:** 0478 is a pseudoabend detected by DL/I, or one of the DL/I call analyzer modules, DFSDLAS0. Examine the abend dump of the application program to find a current call that has more than 1024 qualification statements.

Each qualification statement included in a segment search argument of a DL/I call is encoded by DL/I into an 8-byte block called an FLD. The maximum space allocated by DL/I for the FLD blocks for a single call is 8192 bytes. Thus a single call can have a maximum of 1024 qualification statements. This abend results when a DL/I call exceeds this limit.

**System action:** The application program terminates abnormally.

**Programmer response:** Modify the application program so that it does not exceed the limit of 1024 qualification statements on a single DL/I call.

**Problem determination:** 11, 17a 17b

**Source:** IMS abends

**Module:** DFSDLAS0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

### 0479

**Explanation:** A nonzero return code was returned from a STAT call to the buffer handler to get buffer pool statistics.

**Analysis:** 0479 is a pseudoabend detected by DFSDLA00, the DL/I call analyzer module. DL/I call analyzer calls the buffer handler with buffer handler function to return buffer handler statistics. On return, return code (register 15) was not zero and PSTRTCDE was not one of the two possible return codes in this case. The two possible return codes are PSTGTDS (summary statistics returned) or PSTNOTFD (no subpool defined).

**System action:** The application program is terminated abnormally.

**Programmer response:** This is probably an IMS internal error.

**Problem determination:** 17a, 17b, 35

**Source:** IMS abends

**Module:** DFSDLA00

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

**0481**

**Explanation:** IMS was unable to build a recoverable in-doubt structure (RIS) to identify and save information about data in the in-doubt state. The system tries to build a RIS whenever a thread terminates abnormally while owning in-doubt data. A subcode in register 15 further defines the problem.

**Analysis:** This abend identifies several different problems that are possible during the RIS build process. Each of the problems is an internal IMS error and is the failure of a request for some IMS service. Check the following registers at entry to abend:

**For DFSDRIS0:**

- R14=the return address to which the requested service returns; it is the address of the instruction immediately following the service macro.
- R5=the address of the RRE block, which is the anchor block of the RIS.
- R6=the address of the AWE that requested the building of the RIS, or the conversion of lock protection to EEQE protection, or the retention of all locks for a failing CCTL.

**For DBFDBLS0:**

- R4=the address of the RPST for this UOR.

**For DBFDRIS0:**

- R2=the address of the RRE block, which is the anchor block of the RIS.
- R4=the address of the RPST for this UOR.
- R14=the return address to which the requested service returns; it is the address of the instruction immediately following the service macro (for subcode 102).

Register 15 contains a subcode identifying the problem. Subcodes 1xx are Fast Path subcodes.

**Subcode****Explanation****001**

Unable to read this log.

**002**

Unable to obtain a block from the IMS storage manager.

**003**

Unable to create an ITASK for this in-doubt.

**004**

Unable to enqueue an RRE block to a SIDX block. An RRE identifies the in-doubt UOW and a SIDX defines the subsystem that owned the work unit.

**005**

Unable to find a DDIR/PDIR.

**006**

Unable to schedule a PSB.

**007**

Unable to obtain an I/O Toleration EEQE block.

**008**

An improperly built/updated RIS was discovered.

**009**

Unable to transfer lock ownership to another block or a lock retention call failed.

**101**

GETMAIN failed to obtain Fast Path IEEQE block.

**102**

Unable to obtain a Fast Path EQEL block from the IMS storage manager.

**System action:** The IMS control region terminates abnormally.

**Problem determination:** 1, 4, 5, 35

## 0482 • 0484

**Source:** IMS abends

**Module:** DFSDRIS0, DBFDLBS0, DBFDRIS0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

### 0482

**Explanation:** A Coordinator Control (CCTL) subsystem issued an INIT call to connect to IMS DB/DC, but the IMS DB/DC system was not specified with the DL/I SAS option required for Database Control (DBCTL) subsystem support.

**System action:** IMS terminated the CCTL connection process.

**System programmer response:** If you want DBCTL support, specify the DL/I SAS option for your IMS DB/DC system.

**Problem determination:** 1, 5, 8, 25, 38

**Source:** IMS abends

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

### 0484

**Explanation:** IMS started but was unable to complete the resynchronization process between a Coordinator Control (CCTL) subsystem and IMS. This process is initiated either when a connection between IMS and a CCTL occurs, or in response to an operator command. A subcode in register 15 further defines the problem.

This abend indicates that a problem occurred while IMS was processing the RESYNC request from the Coordinator Controller (CCTL) or from a command. A RESYNC request is either a commit or an abort of an in-doubt unit of work (UOW).

**Analysis:** This abend identifies several different problems that are possible during the RESYNC process. Check the following registers at entry to abend:

**For DFSDRID0:**

- R14=the return address to which a recently called service routine returned; the abend is the result of a failure to get some service performed.
- R4=the address of the RRE block that is the anchor block of a recoverable in-doubt structure (RIS) that identifies in-doubt data.
- R7=the address of the AWE that initiated this RESYNC process.

**For DBFDLOG0:**

- R6=the address of the parameter list passed to DBRC.

Register 15 contains a subcode identifying the problem. Subcodes 1xx are Fast Path subcodes.

**Subcode**

**Explanation**

**001**

Unable to find the PDIR

**002**

Unable to schedule the PSB

**003**

Unable to authorize a database

**108**

Nonzero return code from DBRC for the ADSLIST request

**112**

Nonzero return code from DBRC for the ADSTAT request

**System action:** The IMS control region terminates abnormally.

**Problem determination:** 1, 4, 5, 36

**Source:** IMS abends

**Module:** DFSDRID0, DBFDLOG0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

#### 0485

**Explanation:** In DFSDASD0, a request for storage for an AWE failed.

**System action:** The dependent region is terminated.

**Problem determination:** 4, 35

**Source:** IMS abends

**Related information:**

Chapter 2, "IMS problem determination," on page 13

#### 0499

**Explanation:**

The maximum insert call count was exceeded and the application received an X'A7' status code. The application then attempted to issue another insert and was terminated abnormally with a 0499.

**Analysis:** 0499 is a standard abend issued by module DFSCPY00 or DBFHSRT0. For more information about the code returned to the application program in the PCB, see the DL/I status code explanation information.

**System action:** The application program is abnormally terminated, and the PSB is stopped.

**Programmer response:** Make sure that your application program recognizes an X'A7' status code, or increase the maximum insert count specified in the TRANSACT macro at system definition.

**Problem determination:** Application dump and IMS system definition listing.

**Source:** IMS abends

**Module:** DFSCPY00, DBFHSRT0

**Related reference:**

 [DL/I status code explanations \(Messages and Codes\)](#)

#### 0500

**Explanation:** IMS was unable to find the DEB pointed to by the 7770-3 DCB by following the TCB DEB chain. This problem was probably caused by an IMS or host system error.

**Analysis:** 0500 is a standard abend that can be issued by the 7770-3 DEB builder for z/OS, module DFSASV20. If the error is detected by this module, the program status word (PSW) at entry-to-abend points to the instruction within label COMABEND from which the abend (SVC 13) is issued. This routine is conditionally branched to by the routine that detected the error.

Register 11 in the abend SVRB registers is the base register. Register 10 contains the 7770-3 DCB address. Register 6 contains the address of the DEB pointed to by the 7770-3 DCB. Register 1 contains the abend completion code, X'800001F4'.

Key	Label	Description
Reg2=0 Reg6=pointer to DEB	NEXTDEB	The TCB DEB chain is searched for the DEB pointed to by the DCB. If the end of the chain (register 2=zero) is reached without a match, the abend is issued.

**System action:** IMS terminates abnormally.

## 0500

**Programmer response:** Contact IBM Software Support for assistance.

**Problem determination:** 1, 6, 10, 11, 35

**Source:** IMS abends

**Module:** DFSASV20

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

## Chapter 14. IMS abend codes 0501 - 0600

This topic identifies and explains the following abend codes issued by IMS. Each code is 4 decimal digits.

---

### 0501

**Explanation:** The device-dependent module for the 3270 remote or 3275 dial terminal detected an undetermined problem.

**Analysis:** 0501 is a standard abend that can be issued by one of three modules: DFSDN130, DFSDN140, or DFSDS060. The program status word (PSW) at entry-to-abend isolates the failure to a particular module. In addition, the console sheets should be checked for any IMS messages related to the failing terminal, and the sense/status bytes for each should be noted.

All 3 modules presently have an unconditional branch instruction within label ABEND501, branching around the 0501 abend. In order to obtain the abend, this branch should be zapped with a NOOP instruction. Or, you can take the branch out and reassemble and relink the module. The line is currently shut down, and message DFS001I-UNDETERMINED ERROR ON 3270, LINE *x*, PTERM *y*- is sent to the master console rather than abnormal termination. Log record X'67', subtype X'01' is written to the log, so that IMS TRACE can trap the control blocks and registers.

**Possible cause:** There may be a user modification to the IMS code, or this could be a terminal hardware failure.

#### For DFSDN130:

This is a standard abend that can be issued by the 3270 remote device dependent module, DFSDN130. If the error is detected by this module, the program status word (PSW) at entry-to-abend points to the instruction within label ABEND501 from which the (SVC 13) abend is issued. This routine is BALed to by the routine that detected the error.

Register 12 in the abend SVRB registers is the primary base register. Register 3 is the secondary base register. Register 14 is used as a BAL register. Register 7 contains the address of the CTB. Register 1 contains the abend completion code, X'800001F5'.

Key	Label	Description
Reg7=address of CTB CTBDCTL2= X'80' Reg14=BAL	RI040	This is the routine to determine the starting address of data for the issuing terminal, once the terminal has been selected. When a READ CONTINUE operation completes, the device-dependent module tests the CTBD2SEL flag to ensure that the inputting terminal CTB is the same CTB that was used to process the input on the previous READ INITIAL. Because CTBs cannot be changed in the middle of a message, the routine BALs to label ABEND501.

#### For DFSDN140:

This is a standard abend that can be issued by the 3270 local device dependent module, DFSDN140. If the error is detected by this module, the program status word (PSW) at entry-to-abend points to the instruction within label ABEND501 from which the abend (SVC 13) is issued. Register 12 in the abend SVRB registers is the base register. Register 14 is used as a BAL register. Register 7 contains the address of the CTB. Register 1 contains the abend completion code, X'800001F5'.

Key	Label	Description
Reg7=address of the CTB CTBFLAG3=CTB3LAST Reg11=address of SCD	RSU003	Within the read setup routine is a routine to determine if we should reread the screen with a larger buffer. The CLBDBUF flag is on in the CLB, but the device-dependent module did not find a CTB with CTBDBUF also set before reaching the last CTB on the line. If CLBDBUF=X'01', at least one CTB should have the CTBDBUF flag set. If not, the routine branches to label ABEND501.

## 0502

Key	Label	Description
Reg7=address of CTB Reg9=address of CLB CLBDCTL=CLBDREAD	RI010	An invalid entry was made for the READ INTERRUPT (DD4 entry). The routine checks for an UNLOCK completion, a READ INITIAL completion, or a READ MODIFIED completion. If it does not detect one of these conditions, the routine BALs to label ABEND501.

### For DFSDS060:

This is a standard abend that can be issued by the 3275 dial device dependent module, DFSDS060. If the error is detected by this module, the program status word (PSW) at entry-to-abend points to the instruction within label ABEND501 from which the abend (SVC 13) is issued.

Register 12 in the abend SVRB registers in the base register. Register 9 contains the address of the CLB. Register 7 contains the address of the CTB. Register 15 contains a return code from either module DFSCSEA0, the bisync switched line error analysis routine, or from module DFSCVET0, the 3270 disconnect determination routine. This return code serves as an index into a branch table to handle the error condition.

DFSCVET0 returns a X'08' in register 15.

This says to dequeue the current output message on the terminal.

DFSCSEA0 returns a X'02' in register 15.

DFSDS060 will then multiply this value by 4 to get the index value of return code X'08'. This says there is a contention situation.

Key	Label	Description
Reg9=address of CLB Reg15=X'08' from DFSCSEA0	CHECK	This subroutine determines if any errors occurred in the I/O operation. A BALR is taken to module DFSCSEA0, which loads a return code into register 15. In this case, a contention situation has developed, and register 15 contains a X'02'. DFSDS060 multiplies this value by 4 (at label CHK040), and the result causes a branch table entry index of X'08'—hence, the branch to label ABEND501 and the abend.
Reg15=X'08' from DFSCVET0	CONNECT2	The routine at CONNECT1 branches to module DFSCVET0, which loads a return code into register 15. In this case, the return code passed is a X'08', indicating that the current output message on the terminal should be dequeued. DFSDS060 uses the return code of X'08' as an index to a branch table, discovers that this is an invalid condition, and branches to label ABEND501 to abend.

**System action:** An abend instruction has been issued to produce a memory dump.

**Programmer response:** Determine the cause of the failure. If the problem was caused by an IMS failure, contact IBM Software Support for assistance. If the problem was caused by user modifications or hardware, correct the problem, and retry the failing operation.

**Problem determination:** 4, 6, 7

**Source:** IMS abends

**Module:** DFSDN130, DFSDN140, DFSDS060

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

## 0502

**Explanation:** The IMS graphics attention handler routine in DFSDN070 has attempted to post a 2260 local CLB, but the DECB was being used by the control region even though the idle flag (CLB2IDLE) was on. This is an IMS system error.

**Analysis:** This is a standard abend issued by module DFSDN070.

**System action:** IMS terminates abnormally. The z/OS system continues with the next job, if any.



**Operator response:** IMS can be restarted using the emergency restart procedures or with a cold start.

**Problem determination:** 4, 35

**Source:** IMS abends

**Module:** DFSDN070

**Related information:**

Chapter 2, "IMS problem determination," on page 13

### 0503

**Explanation:** The CTB you are trying to stop is not on the same line as you are currently dispatched on. The routine that marks PTERMs inoperable has detected the CTB for the PTERM is not on the same line as the communications task. Register 9 points to the CLB for the line and register 7 points to the CTB for the failing PTERM.

**Analysis:** This is a standard abend issued by the communications module, DFSCPINO, that marks terminals inoperable. The program status word (PSW) at entry-to-abend points to the instruction within label ABEND from which the abend (SVC 13) is issued. This routine is conditionally branched to by the routine that detected the error.

Register 12 in the abend SVRB registers is the base register. Each CLB (communication line block, with line status information) has an associated DECB; this block (IECTDECB) is pointed to by register 9. Register 7 contains a pointer to the CTB. Register 1 contains the abend completion code, X'800002F7'.

Key	Label	Description
Reg7=address of CTB Reg9=address of DECB (CLB)	DOIT	The CLB address in Reg9 is compared with the CLB field (CTBCLB) in the CTB pointed to by register 7. If the two addresses are not equal, a branch is taken to label ABEND.

**System action:** IMS terminates abnormally.

**Operator response:** Restart IMS. If the error persists, the DD statement for the failing line should be removed in order to continue processing until the failure is corrected.

**Problem determination:** 5, 11, 35

**Source:** IMS abends

**Module:** DFSCPINO

**Related information:**

Chapter 2, "IMS problem determination," on page 13

### 0504

**Explanation:** The IMS switch line connect/disconnect processor module (DFSICLA0) has detected an error in a get unique (GU) or a dequeue (DEQ) call to the queue manager while attempting to delete messages belonging to an inquiry logical terminal on a switched line. This is an IMS system error.

**Analysis:** This is a standard abend issued by the connect/disconnect processor module for switched lines, DFSICLA0. The program status word (PSW) at entry-to-abend points to the instruction within label RWQERR from which the abend (SVC 13) is detected.

Register 12 in the abend SVRB registers is the base register. Register 11 contains the address of the SCD. Register 1 contains the abend completion code, X'800001F8'. Register 15 contains the return code from the queue manager. DFSICLA0 issues the abend for an unacceptable return code:

- On a GU call, X'00' and X'08' are acceptable return codes; anything else is an error.
- On a DEQ call, X'00' is the only acceptable return code; anything else is an error.

Key	Label	Description
Reg11=address of SCD Reg14=BALR Reg15=error return code from queue manager	DEQMSG	The address of the queue manager's read queue routine is located in the SCD at label SCDIRWQE. DFSICLA0 BALRs to the queue manager to dequeue a call, and register 15 is loaded with the return code. When tested, if register 15 does not contain an acceptable return code of X'00', a branch is taken to label RWQERR to abend.
Reg11=address of SCD Reg14=BALR Reg15=error return code from queue manager	GETMSG	The address of the queue manager's routine is located in the SCD at label SCDIRWQE. DFSICLA0 BALRs to the queue manager to do a GET UNIQUE call, and register 15 is loaded with the return code. When tested, if register 15 does not contain an acceptable return code of X'00' or X'08', the abend is issued.

**System action:** The IMS control region abnormally terminates.

**Problem determination:** 4, 35

The return code from DFSQMGR0 is in register 2.

**Source:** IMS abends

**Module:** DFSICLA0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

## 0505

**Explanation:** IMS attempted to acquire storage for a new 7770-3 DEB in subpool 230, key 5. Virtual storage was not available to satisfy the request.

**Analysis:** This is a standard abend issued by module DFSASV20. Register 11 in the abend SVRB registers is the base register. Register 10 contains the 7770-3 DCB address. Register 6 contains the address of the DEB pointed to by the DCB. Register 2 contains the size of the storage requested.

Key	Label	Description
Reg2=size requested Reg6=pointer to DEB	COUNTED	A GETMAIN SVC has been issued for subpool 230 and the return code is loaded into register 15. If the return code is nonzero, the routine branched to label GETMFAIL to abend.

**System action:** IMS terminates abnormally.

**Operator response:** Increase the region size the next time the IMS control region is started.

**Problem determination:** 1, 4, 8, 11

**Source:** IMS abends

**Module:** DFSASV20

**Related information:**

Chapter 2, "IMS problem determination," on page 13

## 0506

**Explanation:** A logic error was detected by DFSBACK0. One of two types of error occurred:

1. All log data sets were read to EOF in the backward direction without finding a stopping point for the backout of one of the recovery tokens, although a stopping point was found during the forward read. This can occur if the log data sets are on unlabeled tapes, and the operator did not mount the same tapes during the backward read and the forward read.
2. An error occurred in dataspace management.

**Analysis:** Check the contents of register 6.

**If register 6 is zero**

**For a batch IMS input log**, the backward read of the input log should terminate when a type X'41' or X'0605' record is encountered. All log data sets were read in the backward direction to EOF, but a stopping point for the backout of one of the recovery tokens was not found. A stopping point was found during the forward read of the log data sets. Possible reasons for the error are:

- The log data sets are on unlabeled tapes, and the operator did not mount the same tapes during the backward and forward read.
- The input log is a multi-volume DASD data set.

**For an input log from an IMS DB/DC or DBCTL subsystem**, this abend should not occur.

**If register 6 is not zero**

A data space management error occurred. Register 6 contains the address in module DFSBACK0 where the error was detected. Register 15 contains the return code from the data space management module, DFSRVSP0. The return codes from module DFSRVSP0 are:

12	Unable to obtain RVDL block
16	Unable to obtain data space
20	Unable to extend data space
24	Bad data space reference (bad token)
28	Bad length for retrieve
32	Unable to get ALET
36	Not enough auxiliary storage available.

**System action:** IMS terminates abnormally.

**Programmer response:**

**If register 6 is zero**

- If the log data sets are on unlabeled tapes, make sure the operator mounted the same tapes for the forward and backward reads.
- If a model DSCB name was specified for a multi-volume tape data set, run the Database Recovery utility for each modified database, but exclude the Batch Backout output log. Remove the reference to the model DSCB name from the DD statement for the input log, and rerun Batch Backout.

**If register 6 is not zero**

If register 6 is not zero and the code in register 15 is 36, try using the READBACK option. If that does not work, or if there is a different value in register 15, contact IBM Software Support.

**Source:** IMS abends

**Module:** DFSBACK0

0507

**Explanation:** You selected the abend option for the Batch Backout utility. This abend is issued whenever the batch backout utility is executed with the ABEND control statement specified in the SYSIN.

This abend is also issued when the ABENDMSG control statement is specified and message DFS894I, DFS896A, DFS3278A, or DFS3289A is issued.

**Analysis: For message DFS894I:**

Message DFS894I applies to the following table:

Message	Subcode	Key	Label	Description
DFS894I	01	Reg3=0	LOGRSRCH	A variable record has a record length of zero.
	02	2(Reg2)≠0	LOGRSRCH	A variable record with the span bits nonzero was found.
	03	Reg3≠POOLED +1	GOSUB2	For tape input, the BDW plus the buffer start address is greater than POOLEND.
	04	Reg2+Reg3 > BLOCKEND	LOGRSRCH	The variable lengths of records in the buffer are greater than BLOCKEND.

## 0507

If the ABEND control statement is used, the abend occurs after all backout processing has completed. This is before the final return and before the log data sets are closed. This abend is unconditional and is used for only diagnostic purposes.

If the ABENDMSG control statement is used, the abend occurs after issuing the error message and closing the input log, if necessary.

The current log record can be located using the address in the RECPTR field. The registers that are in effect when the error message is issued are stored starting at field REGSAVE using STM register 12, register 11.

*Possible cause:* For this abend and message DFS894I, causes include:

- Improperly closed log data set
- Corrupted log input (bad tape/drive)
- I/O read error
- Bad log records

### For message DFS896A:

Message DFS896A applies to the following table:

Message	Subcode	Key	Label	Description
DFS896A	01	Reg2= 0	DBCTL08	The unit-of-recovery (UOR) token pointed to by GPR02 has the same recovery token value as the TYPE08 non-CICS record pointed to by RECPTR.
	02	TOKFAIL is on	MTCHTOK	The Unit-of-Recovery (UOR) token, pointed to by GPR02, is for the same PST as the non-TYPE08 CICS record pointed to by RECPTR. The TYPE07 termination record, indicating dynamic backout failed, was already found.

*Possible cause:* An incorrect or incomplete log input can cause this abend and message DFS896A.

### For message DFS3278A:

Message DFS3278A applies to the following table:

Message	Subcode	Key	Label	Description
DFS3278A	01	Reg2 + X'21' is less than field SAVETIME.	CASE50BF	The current TYPE5X log record (RECPTR) has a date-time less than the TYPE06 or prior TYPE5X record processed during forward read.
	02	NBLKA or DP00LADR is greater than DPOOLEND.	DREAD020	During the backward read of a DASD log, an attempt was made to process beyond the current buffer.
	03	DDLASTT=TT	DCCHTEST	During the backward read of a DASD log, a missing EOF was detected.
	04	Reg2 + X'14' is less than field SAVETIME.	CASE06BF	A TYPE06 record was found on a batch log and the date-time is less than SAVETIME.

Message	Subcode	Key	Label	Description
	05	LGSEQNR	GOSUB3 BCKSEQNR	The log record sequence number (LGSEQNR) of the last log record that was read during the forward read does not match the log sequence number (BCKSEQNR) of the first record that was read during the backward read for an IMS batch log input data set.

**Possible cause:** Causes for this abend and message DFS3278A include:

- Improperly closed log data set
- Corrupted log input
- Invalid log input
- A JCL error on IMSLOGR input can cause this abend and message DFS3278A.

**For message DFS3289A:**

Message DFS3289A applies to the following table:

Message	Key	Label	Description
DFS3289A	The log sequence number of the record pointed to by Reg2 is not equal to the prior sequence number + 1.		The log record sequence number for two adjacent records did not increase by 1. The message gives the expected sequence number and the sequence number that was found.

**Possible cause:** Causes for this abend and message DFS3289A include:

- Improperly closed log data set
- Corrupted log input
- Invalid log input
- JCL error on IMSLOGR input

**System action:** After processing is complete, the utility terminates abnormally with a dump.

**Source:** IMS abends

**Module:** DFSBACK0

**Related information:**

 DFS894I

 DFS896A

 DFS3278A

 DFS3289A

---

0509

**Explanation:** The message generator module (DFSCLM00) found an invalid SYSID while building a response message for an IMS command.

**Analysis:** This is a standard abend issued by the communications message generator overlay segment processor, DFSCLM00. The program status word (PSW) at entry-to-abend points to the instruction within label ABND509 from which the abend (SVC 13) is issued. This label is branched and linked to by the routine that detected the error.

Register 12 in the abend SVRB registers is the base register. Register 8 points to a scratch pad. Register 7 contains the CTB address. Register 1 contains the abend completion code, X'800001FD'.

## 0511

Key	Label	Description
Reg5=pointer to SID zoned Reg6=SID Reg9=address of CLB	SIDBIN	In the routine to convert the zoned SID to binary, Reg6 is checked to ensure that the SID is only one character in length. If not, the routine branches and links to label ABND509 to abend.

**System action:** IMS terminates abnormally.

**Programmer response:** The problem probably originated in the IMS system code.

**Problem determination:** 4, 35

**Source:** IMS abends

**Module:** DFSCLM00

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

## 0511

**Explanation:** A conversational processing error occurred. Either an invalid buffer request has been specified or a request to free a conversation was made when no conversation was active. This problem is probably an IMS system error.

**Analysis:** This is a standard abend that can be issued by one of the following modules: DFSCONM0, DFSCON00, DFSCON10, DFSCON20. The program status word (PSW) at entry-to-abend isolates the failure to a particular module. This abend results because conversational processing is unable to continue because of an error in one of the conversational control blocks, or because of an invalid pointer that is vital to processing.

Usually, register 5 is used as a BAL register to aid in isolating the failure within that module to a particular descriptive label.

**For DFSCONM0:**

This is a standard abend that can be issued by the conversational processor to insert a SPA from a remote system, module DFSCONM0. If the error is detected in this module, the program status word (PSW) at entry-to-abend points to the instruction within label ABND511 from which the abend (SVC 13) is issued.

There are two different types of errors that can result in abend 0511 from this module:

1. If the CTB pointer is bad, which is the case in the first label below, the routine detecting this error branches and links to label ABND511 to issue the abend.
2. If the information in the CCB is not meaningful, the routine detecting the problem branches conditionally to label ERRCCB, which branches and links to label ABND511 to abend.

Register 12 in the SVRB registers in the base register. Register 6 contains the CCB address. Register 7 contains the link CTB address. Register 10 contains the destination CNT address. Register 5 is used as a BAL register. Register 1 contains the abend completion code, X'800001FF'.

Key	Label	Description
Reg3=CTB pointer from CNT Reg4=CTB pointer	(prior to) CONMOK1	In the routine to determine if the conversation is supposed to be continued, the CTB pointers are checked. Register 3 and register 4 are compared. If the addresses of the registers are unequal, the routine branches and links to label ABND511 to abend.
Reg6=CCB number from SPA Reg8=pointer to packed input SPA Reg11=address of SCD	CONMWASE	In the routine to get and verify the CCB from the input SPA, the CCB number in the SCD (at label SCBCCBN) is compared with the CCB number from the SPA. If the CCB number from the SPA is higher, a branch is taken to label ERRCCB to handle the abend.

Key	Label	Description
Reg4=pointer to CTB (conversational PTERM) CTBFLAG1=X'00' or X'04'	CONMWASE	In the routine to validate a conversation, the CTBFLAG1 field of the CTB pointed to by register 4 is tested: first, to determine, if the PTERM is in conversation mode; if not, a branch is taken to label ERRCCB to handle the abend. The second test is to determine if the conversation is being held (not active on PTERM); if yes, a branch is taken to label ERRCCB to handle the abend.
Reg4=address of CTB Reg6=address of CCB	CONMWASE	In the routine to validate a conversation, a compare is made to determine if the CCB and CTB match. If the two blocks are unequal, a branch is taken to label ERRCCB to handle the abend.

#### For DFSCON00:

This is a standard abend that can be issued by the conversational processor START/CONT/END module DFSCON00. If the error is detected in this module, the program status word (PSW) at entry-to-abend points to the instruction within label ABND from which the abend (SVC 13) is issued. This routine is unconditionally branched to by the routine at label ABND511, which loads the abend code into register 1. Label ABND511 is BALed to by the routine that detected the error.

Register 12 in the abend SVRB registers is the base register. Register 5 is used as a BAL register. Register 1 contains the abend completion code, X'800001FF'.

Key	Label	Description
Reg0	CONTINUE	Register 0=MINUS should contain a 0 (start) or a 1 (continue).
Reg6=CCB	CONTINUE	Register 6 loaded from CTBCCBPT should contain a CCB pointer.
Reg5 $\neq$ 0	LOGERR	Nonzero return code trying to log the start of a conversation.
Reg15 $\neq$ 0	ISRTFLG1	Nonzero queue manager return code trying to do an 'insert prefix' call.
Reg15 $\neq$ 0	ISRTFLG2	Nonzero queue manager return code trying to do a 'put locate' call.
Reg15 $\neq$ 0	FNDBREAD	Nonzero queue manager return code trying to do a reposition call.
Reg15 $\neq$ 0	ISRTISRT	Nonzero queue manager return code trying to insert move message prefix call of APPC prefix.

#### For DFSCON10:

This is a standard abend that can be issued by the conversational processor inserting a SPA, module DFSCON10. If the error is detected in this module, the program status word (PSW) at entry-to-abend will point to the instruction within label ABND from which the abend (SVC 13) is issued. This routine is unconditionally branched to by the routine at label ABND511, which loads the abend code into register 1. Label ABND511 is BALed to by the routine that detected the error.

Register 12 in the abend SVRB registers is the base register. Register 5 is used as a BAL register. Register 1 contains the abend completion code, X'800001FF'.

Key	Label	Description
Reg4=input terminal SID Reg11=address of SCD	VERSID	In the routine to get and verify the SID of the input terminal, if register 4 contains a zero, a branch is taken to label VERSIDER. If it is not a zero, a compare is made between the SID in register 4 and the SID range value in the SCD (at label SCDSIDN). If the input terminal SID is higher than the valid range, a branch is taken to label VERSIDER. Label VERSIDER then branches and links to label ABND511 to handle the abend.
Reg7=address of CTB CTBFLAG1=X'00'	CONLLOC	The CTBFLAG1 field is tested to determine if this PTERM is in conversation mode. If this field is zero, a branch is taken to label CONLERR1, which branches and links to label ABND511 to handle the abend.
Reg8=pointer to SPA from DFSCONG0 Reg9=address of PST Reg15=nonzero return code from DFSCONG0	CONLWASE	The routine in CSECT DFSCONL0 BALRs to module DFSCONG0 to retrieve the SPA from the input message. Register 15 is loaded with, then tested for, the return code. If other than zero, a branch is taken to label CONLESPA. If zero, a compare is made of the SPA and the PSTCCB number. If they are unequal, a branch is taken to label CONLESPA. Label CONLESPA branches and links to label ABND511 to handle the abend.
Reg15=nonzero return code from queue manager DECTYPE=X'9B'	CONLRMT	In the routine to reserve space in the message queue and put in the message segment by a <i>series</i> of calls, the routine BALRs to the queue manager and register 15 is loaded with the return code. When tested, if register 15 is not zero, indicating an invalid length and no segment moved, a branch is taken to label BAL511, which branches and links to label ABND511 to handle the abend.
Reg15=nonzero return code from queue manager DECTYPE=X'9A'	UPPRFSPA	This routine BALRs to the queue manager to insert a prefix, and register 15 is loaded with the return code. When tested, if register 15 is not zero, indicating the message prefix was not updated, the routine branches to label BAL511, which branches and links to label ABND511 to handle the abend.
Reg15=nonzero return code from queue manager DECTYPE=X'9A'	ISRT62PF	

#### For DFSCON20:

This is a standard abend that can be issued by the conversational processor termination module, DFSCON20. If the error is detected in this module, the program status word (PSW) at entry-to-abend will point to the instruction within label ABND511 from which the abend (SVC 13) is issued.

Register 12 in the abend SVRB registers in the base register. Register 5 is used as a BAL register. Register 7 contains the CTB address. Register 11 contains the address of the SCD. Register 1 contains the abend completion code, X'800001FF'. The low-order byte (byte 3) of register 0 can contain a vector value describing the reason DFSCONE0 is called for termination. Below are the acceptable vector values and their meanings:

Code	Meaning
X'00'	Conversational MPP abend.
X'04'	Severe error when processing this conversation.
X'08'	/EXIT local or remote.
X'0C'	/START.
X'10'	SPA for nonactive conversation.
X'14'	Inconsistent definitions recognized.
X'18'	/EXIT and input message still active.
X'1C'	/START and input message still active.



X'20' Conversational MPP did not insert a response to the terminal.

Key	Label	Description
Reg10=work save set SAVVPTR>X'14'	DFSCONX0	A compare is made of the field SAVVPTR to determine if the save vector is valid (less than or equal to the return code of X'14'). If the save vector is invalid, a return code greater than X'14', the routine branches and links to label ABND511 to abend.
Reg6=CCB number VCT=X'00' Reg11=address of SCD	CONXLOC	The CCB number in register 6 is compared with the CCB number range in the SCD. If the value in register 6 exceeds the valid range, the routine branches and links to label ABND511 to abend.
Reg6=address of CCB Reg7=address of CTB VCT=X'00'	CONXLOC3	A compare is made between the CTB address in register 7 and the CTB pointer in the CCB (at label CCBCTBPT) to determine if the destination terminal is the PTERM in conversation. If the two addresses are unequal, the routine branches and links to label ABND511 to abend.
Reg6=CCB pointer from CTB Reg7=address of CTB VCT=X'0C'	CONXICL4	Register 6 is loaded with the CCB pointer from the CTB. When tested, if the value in register 6 is zero or negative, the routine branches and links to label ABND511 to abend.
Reg8 = pointer to SPA VCT=X'00'	CONXRMT	In the routine to check if the conversation is in a remote system, register 8 is tested to determine if the SPA is available. If the value in register 8 is zero, the routine branches and links to label ABND511 to abend.
Reg6=CCB pointer from CTB	DFSCONT0	In the subroutine to restore CCBs to the inactive queue, register 5 is tested. If the address in register 5 is lower than or equal to the pointer in register 6, the routine branches and links to label ABND511 to abend.
Reg4=CTB pointer from CCB Reg4 = Reg7 Reg7=address of CTB	CONT2	The CTB pointer from the CCB (at label CCBCTBPT) is compared with the CTB address in register 7. If the two addresses are unequal, the routine branches and links to label ABND511 to abend.
Reg6=address of CCB CCBFLAG1=zero	CONT2A	The CCBFLAG1 of the CCB is tested to determine if a conversation is active on this CCB. If the flag field is zero, the routine branches and links to label ABND511 to abend.
Reg15=nonzero return code from logical logger	CONT4	The routine to log end of conversation BALRs to the logical logger and loads register 15 with its return code. When register 15 is tested to determine if the logging was successful, if register 15 is not zero, the routine branches to label LOGERR, which branches and links to label ABND511 to abend.
R15=nonzero return code from Q manager	CONXRMT2	This routine BALRs to the queue manager to free the chain of output messages in process. Register 15 is loaded with the return code. When tested, if register 15 is not zero, the routine branches and links to label ABND511 to abend.
Reg15=nonzero return code from Q manager	ENQIT	There was a nonzero return code from the queue manager on a ENQ LIFO request trying to enqueue SPA.
Reg15=nonzero return code from the logger	LOGERR	The logger returned a nonzero return code in register 15 while trying to write a type 12 log record.

## 0513

Key	Label	Description
Reg6=CCB number from SPA Reg8=pointer to packed SPA Reg11=address of SCD	VERACT	In the routine which verifies if a conversation is still active, the CCB number in register 6 is compared with the CCB number range in the SCD (at label SCDCCBN). If the value in register 6 is higher, the routine branches and links to label ABND511 to abend.
Reg4=CTB pointer from CCB CTBFLAG1=zero	VERACT1	The CTBFLAG1 field of the CTB is tested to determine if the PTERM is still in conversation. If the flag is zero, the routine branches and links to label ABND511 to abend.
Reg4=address of CTB CTBFLAG1=X'40'	VERACT2	The CTBFLAG1 field of the CTB is tested to determine if the conversation is being held. If yes, the routine branches and links to label ABND511 to abend.
Reg3 = SID	VERRMT	In the routine which verifies if the SID is valid, register 3 is tested. If zero, the routine branches and links to label ABND511 to abend.
Reg3 = SID Reg11=address of SCD	VERRMT1	The SID number in register 3 is compared with the SID number range in the SCD (at label SCDSIDN) to determine if the SID is in the valid range. If the value in register 3 is higher, the routine branches and links to label ABND511 to abend.
Reg15=nonzero return code from queue manager	ROUTE	This routine BALRs to the queue manager to reserve space in the message queue buffer and puts the message into it. Register 15 is loaded with the nonzero return code. When tested, if register 15 is not zero, indicating the message segment was not moved, the routine branches and links to label ABND511 to abend.
Reg15=nonzero return code from Message Router	ROUTER	This routine BALRs to the message router to enqueue the message, and register 15 is loaded with a return code. When tested, if register 15 is not zero, the routine branches and links to label ABND511 to abend.

**System action:** IMS terminates abnormally.

**Programmer response:** No action is required.

**Problem determination:** 1, 4, 6, 25, 26, 35

**Source:** IMS abends

**Module:** DFSCONM0, DFSCON00, DFSCON10, DFSCON20

**Related information:**

Chapter 2, "IMS problem determination," on page 13

## 0513

**Explanation:** An overlay of one of the Transaction Manager (TM) buffer pools (CIOP, HIOP, or SPAP) or of the Automated Operations buffer pool (AOIP) might have occurred. In Fast Path, a storage overlay of the Fast Path segment compression work area (SEG1 or SEG2) might have occurred. The error might be caused by one of the user's TM exits or edit routines, an access method (such as VTAM), or an error in IMS code. If the overlay is to the AO pool, the error might be caused by the user's TYPE 2 AO exit, DFSAOE00, or an error in the IMS code. The reason code in register 15 provides additional information in diagnosing the cause of this failure.

**Analysis:** This abend is usually a standard abend issued by various modules when an overlay has occurred in the AO buffer pool AOIP. See the description of the error in each module for further diagnostic information when an error is detected while processing for the line, node, MSC link, or dependent region whose buffer is overlaid.

Register 9 contains the address of the communication line block (CLB) associated with the line or node, the logical link block (LLB) of the link, or the partition specification table (PST) of the dependent region. Register 9 is not applicable for 0513 when issued from DFSLIEE0 for input and output to an APPC device.

Register 13 contains the address of the active save area. The complete save area set in the diagnostic area or save area trace in the IMS formatted dump will help you determine the module flow that led up to the abend.

Field SCDSMCON contains the 8-byte constant found at the end of every buffer in the four pools (I/OP, CWAP, HIOP, and LUMP). Since the buffers are validated before and after calling all exit routines, errors detected after calling the exit routine are most likely due to the exit routine itself.

If the exit routine uses standard register saving conventions, the registers passed to the routine will be in the save set pointed to by register 13 at the time of the abend.

Diagnostic information for each module follows:

**For DBFCMP10:**

A potential storage overlay of the Fast Path Segment Compression work area (SEG1 or SEG2) occurred.

This is a pseudoabend, issued by module DBFCMP10 when a potential storage overlay has occurred. Register 1 contains the abend code. The following situations initiate this abend:

1. Movement of data from a user I/O area to a key 7 work area SEG1.
  - Reg2=invalid data length
  - Reg7=address of work area prefix
  - Reg8=address of input data
  - Reg10=A(EPST)
2. Movement of data from a user I/O area to a key 7 work area (SEG2)
  - Reg9=invalid data length
  - Reg7=address of work area prefix
  - Reg8=address of input data
  - Reg10=A(EPST)
3. Overlay of the work area suffix (identifier) after the invoking of the segment compression exit routine
  - Reg7=address of work area suffix
  - Reg10=A(EPST)

**For DFSCFEI0:**

Register 11 points to the communication interface block (CIB). Field CIBWORK points to the user buffer. The first two bytes are the length of the buffer. The overlay constant is at the end of the buffer.

Register 9 points to the communication line block (CLB) of the line or node.

Register 3 points to the system contents directory (SCD). Field SCDSMCON contains the buffer overlay constant.

Register 15 at entry-to-abend contains a subcode. Use this subcode number to determine where the abend was detected and the cause of the problem as shown in the following table:

Key	Label	Description
Reg15=X'A4'	SEGM28	A buffer overlay was detected before calling the user MFS segment edit routine.
Reg15=X'A8'	SEGM29	A buffer overlay was detected after calling the user MFS segment edit routine.

**For DFSCMCP0:**

An MSC channel-to-channel (CTC) link buffer has an invalid length field.

The following registers, in abend SVRB, contain diagnostic information at the time the abend occurs:

Register 3 is the buffer length.

Register 4 is the buffer length calculated from the communication terminal table (CTT).

Register 5 contains the address of the message buffer.

Register 9 contains the address of the logical link block (LLB).

## 0513

Register 15 at entry-to-abend contains a subcode. Use this subcode number to determine where the abend was detected and the cause of the problem.

Key	Label	Description
Reg15=X'84'	SETINPUT	A buffer overlay was detected while preparing to initiate channel-to-channel (CTC) I/O (Multiple Systems Coupling CTC access method only).

### For DFSCMEI0:

Register 4 contains the address of the invalid buffer.

Register 15 at entry-to-abend contains a subcode. Use this subcode number to determine where the abend was detected and the cause of the problem.

Key	Label	Description
Reg15=X'B8' Reg4=address of the invalid buffer	CKBFRTN	A buffer overlay was detected before calling the message control error exit.
Reg15=X'BC' Reg4=address of the invalid buffer	CKBFRTN	A buffer overlay was detected after calling the message control error exit.

### For DFSCMS00:

Register 9 is the address of the logical link block (LLB) of the MSC link.

Register 13 is the address of the current save set.

Registers 3 and 12 are base registers for module DFSCMS00.

The overlaid buffer address is stored in field BUFADDR in DFSCMS00. The first two bytes of the buffer contain the length of the buffer.

Register 11 contains the address of the system contents directory (SCD). Field SCDSMCON contains the buffer overlay constant.

Register 15 at entry-to-abend contains a subcode. Use this subcode number to determine where the abend was detected and the cause of the problem.

Key	Label	Description
Reg15=X'68'	DFSCMS00	A buffer overlay was detected while entering the Multiple Systems Coupling (MSC) analyzer (DFSCMS00 - ITASK CREATE).
Reg15=X'6C'	DFSCIO01	A buffer overlay was detected when the MSC device-dependent module returned to the analyzer to process an input message (ENTRY A01).
Reg15=X'70'	PROSMB	A buffer overlay was detected before calling the user link receive exit routine and the input message received from the link had a transaction destination.
Reg15=X'74'	PROSMB	A buffer overlay was detected after calling the user link receive exit routine and the input message received from the link had a transaction destination.

Key	Label	Description
Reg15=X'78'	PROCNT	A buffer overlay was detected before calling the user link receive exit routine and the input message received from the link was a directed routing message.
Reg15=X'7C'	PROCNT	A buffer overlay was detected after calling the user link receive exit routine and the input message received from the link was a directed routing message.
Reg15=X'80'	RTNDISP	A buffer overlay was detected when the Multiple Systems Coupling (MSC) analyzer exited to the IMS dispatcher.

#### For DFSCFC0:

Register 9 is the communication line block (CLB) address of the line or node that was processing and whose buffer was overlaid.

CLBOUTBF points to the overlaid buffer. The first two bytes of the buffer contain the length of the buffer.

Register 11 points to the system contents directory (SCD). Field SCDSMCON contains the field overlay constant.

Key	Label	Description
Reg15='88'	CALLDD6	A buffer overlay was detected before calling the device-dependent module at entry 6 (DD6M or DD6S).
Reg15='8C'	CALLDD1	A buffer overlay was detected before calling the device-dependent module at entry 1 (DDM1).

#### For DFSCON20:

Register 9 contains one of the following:

- The address of the partition specification table (PST) of the region, if the conversation transaction terminated abnormally in this IMS system.
- The CLB of the terminal that issued the /EXIT command, if the conversation terminated because of the command.
- The logical link block (LLB) of the MSC link that received the scratchpad area (SPA) in error. The conversation transaction terminated abnormally in a remote IMS system (MSC).

Register 7 contains either the address of the communication terminal block (CTB) of the terminal in conversation or 0 if the conversation has already been terminated.

Register 6 contains either the address of the conversational control block (CCB) for the conversation or 0 if the conversation has already been terminated.

Register 10 contains the address of a save set where

- Field SAVGB5 (Reg10+X'30') contains the address of the SPA buffer passed to the exit routine, and
- Field SAVGB9 (Reg10+X'38') is the length of this buffer.

If the caller is DFSCONM0, then the save set that contains the following SPA buffer address and length is located two save sets before the one pointed at by R10 (SAVEID=WORKSA):

- Field SAVGB2 (offset X'24') contains the address of the SPA buffer passed to the exit routine.
- Field SAVGB6 (offset X'34') is the length of this buffer.

Register 11 points to the system contents directory (SCD). Field SCDSMCON contains the buffer overlay constant.

The format of the SPA buffer is:

```
FIELD:  SPA LL|ZZ|CCBID|TRANCODE|SPA DATA|BUFFER CONSTANT|
LENGTH:  2   2   2       8       n       8
```

## 0513

Register 15 at entry-to-abend contains a subcode. Use this subcode number to determine where the abend was detected and the cause of the problem.

Key	Label	Description
Reg15=X'58'	CONE	Before calling the user conversation abnormal termination exit routine, the buffer containing the SPA being passed to the routine was found to be overlaid.
Reg15=X'5C'	CONE10	After calling the user conversation abnormal termination exit routine, the buffer containing the SPA being passed to the routine was found to be overlaid.

### For DFSCRTU0:

Register 2 points to the buffer location where the buffer overlay constant was located. The buffer is USEQDATA parameter area.

Register 9 points to the communication line block (CLB) of the line or node.

Register 15 at entry-to-abend contains a subcode. Use this subcode number to determine where the abend was detected and the cause of the problem, as shown in the following table:

Key	Label	Description
Reg15=X'9C'		A buffer overlay was detected before calling the DFSINSX0 user exit routine.
Reg15=X'A0'		A buffer overlay was detected after calling the DFSINSX0 user exit routine.

### For DFSDMIF0, DFSDMIQ0:

The DFSAOE0 control block, the segment area, and the work area are checked for overlays before and after calling automated operator (AO) exit routine DFSAOE00.

Register 3 is the address of the function specific parameter list. The address of the segment area is at field AOE0SEG. The address of the work area is at field AOE0WRKA.

Key	Label	Description
Reg15=X'C0'	DMIF7630	A buffer overlay was detected before calling AO exit routine DFSAOE00.
Reg15=X'C4'	DMIF7600	A buffer overlay was detected after calling AO exit routine DFSAOE00.

### For DFSICIO0:

Field BUFADDR in DFSICIO0 can contain the address of the invalid buffer. Field RETADDR contains the register 14 address of either the routine that called the buffer overlay check routine (CKBFRTN), or, if the abend was issued because the user edit routine added more than 10 bytes, the routine that detected the error. For fields BUFADDR AND RETADDR, refer to the current assembly listing of module DFSICIO0.

Registers 12 and 6 are DFSICIO0 base registers.

Register 9 is the communication line block (CLB) address of the line or node that was processing and whose buffer was overlaid.

Register 13 contains the current save area address in the save area set. When the error was detected after calling the user edit/exit routine, register 13 is the address of the save area set containing the registers passed to the user routine (if the routine uses standard saving conventions).

Register 11 points to the system contents directory (SCD).

Register 15 at entry-to-abend contains a subcode. Use this subcode number to determine where the abend was detected and the cause of the problem, as shown in the following table:

Key	Label	Description
Reg15=X'04' Reg10=address of the invalid buffer	DFSCIO00	A buffer overlay was detected while entering the communications analyzer (DFSICIO0 - ITASK CREATE).
Reg15=X'08' Reg10=address of the invalid buffer	DFSCIO01	A buffer overlay was detected when the device-dependent module returned to process an input segment (ENTRY A01).
Reg15=X'0C' Reg10=address of the invalid buffer	GOUIEDIT	A buffer overlay was detected before calling the user physical terminal input edit routine.
Reg15=X'10' Reg10=address of the invalid buffer	GOUIEDIT	Either a buffer overlay was detected after calling the user physical terminal input edit routine, or the user increased the length of the message. You cannot increase the length of the LL field, but can decrease the length to any value.
Reg15=X'14' Reg10=address of the invalid buffer	FEINTFMT	A buffer overlay was detected before calling the MFS input edit routine.
Reg15=X'18' Reg10=address of the invalid buffer	FEINTFMT	A buffer overlay was detected after calling the MFS input edit routine.
Reg15=X'1C' Reg10=address of the invalid buffer	DEST2	A buffer overlay was detected before calling the user terminal routing exit routine.
Reg15=X'20' Reg10=address of the invalid buffer	DEST2	A buffer overlay was detected after calling the user terminal routing exit routine.
Reg15=X'24' Reg10=address of the invalid buffer	GOTRANEX	A buffer overlay was detected before calling the user transaction code input edit routine.
Reg15=X'28' Reg10=address of the invalid buffer	GOTRANEX	A buffer overlay was detected after calling the user transaction code input edit routine.
Reg15=X'2C' Reg10=address of the invalid buffer	GOMSGSEX	A buffer overlay was detected before calling the user message switch edit routine.
Reg15=X'30' Reg10=address of the invalid buffer	GOMSGSEX	A buffer overlay was detected after calling the user message switch edit routine.
Reg15=X'34' Reg10=address of the invalid buffer	DFSCIO02	A buffer overlay was detected while calling the access method (ENTRY A02).
Reg15=X'38' Reg10=address of the invalid buffer	DFSCIO05	A buffer overlay was detected while calling the access method (ENTRY A05).

## 0513

Key	Label	Description
Reg15=X'3C' Reg14=address of the invalid buffer	RECEIVE	A buffer overlay was detected while returning a RECANYP buffer to VTAM.
Reg15=X'40' Reg10=address of the invalid buffer	CIOEXIT	A buffer overlay was detected when the communications analyzer exited to the IMS dispatcher.
Reg15=X'44' Reg10=address of the invalid buffer	GOUOEDIT	A buffer overlay was detected before calling the user physical terminal output edit routine.
Reg15=X'48' Reg10=address of the invalid buffer	GOUOEDIT	A buffer overlay was detected after calling the user physical terminal output edit routine.
Reg15=X'4C' Reg10=address of the invalid buffer	GOFES00	A buffer overlay was detected before calling the FES user exit routine.
Reg15=X'50' Reg10=address of the invalid buffer	GOFES00	A buffer overlay was detected after calling the FES user exit routine.
Reg15=X'AC' Reg1=address of the invalid buffer Reg3=maximum message length allowed Reg5=A(RUNEDIT) or A(NONMFS) or A(PROCNT1) Reg14=size of input message segment	USERRC11	The user edit routine expanded the message by more than 10 decimal bytes (19 decimal bytes if the terminal is in preset mode).
Reg15=X'B0' Reg1=address of the invalid buffer Reg3=maximum message length allowed Reg9=address of communication line block (CLB) CLBTEMP4=maximum message length + 10 extra bytes.	POCONT	The user edit routine expanded the message by more than 10 decimal bytes.

### For DFSIIDM0:

The DFSAOE0 control block, the segment area, and the work area are checked for overlays after calling automated operator (AO) exit routine DFSAOE00.

Register 7 is the address of DFSAOE0. The address of the segment area is at field AOE0SEG. The address of the work area is at field AOE0WRKA.



Key	Label	Description
Reg15=X'C4'	IIDM6100	A buffer overlay was detected after calling AO exit routine DFSAOE00.

**For DFSLIEE0:**

Register 2 points to the buffer location where the buffer overlay constant was located.

Register 3 points to the overlaid buffer. The first two bytes of the buffer contain the length of the buffer.

Register 11 points to the system contents directory (SCD). Field SCDSMCON contains the field overlay constant.

Register 15 at entry-to-abend contains a subcode. Use this subcode number to determine where the abend was detected and the cause of the problem.

Key	Label	Description
Reg15=X'00'		A buffer overlay was detected before calling the DFSLUEE0 user exit.
Reg15=X'04'		A buffer overlay was detected after calling the DFSLUEE0 user exit.

**For DFSNCS10:**

Register 2 points to the buffer location where the buffer overlay constant was located. The buffer is either NCLZWORK area or USEQDATA parameter area.

Register 9 points to the Communication Line Block (CLB) of the line or node.

Register 15 at entry-to-abend contains a subcode. Use this subcode number to determine where the abend was detected and the cause of the problem.

Key	Label	Description
Reg15=X'94'		A buffer overlay was detected before calling the DFSSGNX0 user exit routine.
Reg15=X'98'		A buffer overlay was detected after calling the DFSSGNX0 user exit routine.

**For DFSQUEI0:**

The UEHB control block, the copy buffer, and the user buffer are checked for overlays before and after calling automated operator (AO) exit routine DFSAOUE0. (The copy/user buffer may not exist if UEHCPYBF or UEHUBUF=0).

The UEHB address is in register 10. The copy buffer address is contained in field UEHCPYBF in the UEHB, and the user buffer address is contained in field UEHUBUFF. The UEHB and the buffers all have a 4-byte negative prefix containing the length of the buffer (including the prefix).

Register 11 is the address of the system contents directory (SCD). Field SCDSMCON contains the buffer overlay constant.

Register 15 at entry-to-abend contains a subcode. Use this subcode number to determine where the abend was detected and the cause of the problem.

Key	Label	Description
Reg15=X'60'	QUIUEXIT	A buffer overlay was detected before calling AO exit routine.

## 0516

Key	Label	Description
Reg15=X'64'	UEXIT	A buffer overlay was detected after calling AO exit routine.

### For DFSSIST0:

Register 9 points to the communication line block (CLB) of the line or node. Register 11 points to the system contents directory (SCD). Field SCDSMCON contains the buffer overlay constant. Register 15 at the entry to the abend contains a subcode. Use this abend to determine where the abend was detected and the cause of the problem.

Key	Label	Description
Reg15=X'B4'	CALL0300	A buffer overlay of the user message buffer was detected after calling the Greeting Messages exit routine (DFSGMSG0). Also, if the length of the returned user message is larger than allowed this abend code will be used instead of the user message.

**System action:** IMS terminates abnormally or, in the case of a potential overlay of a Fast Path work area, the dependent region is terminated abnormally.

**Problem determination:** 4, 35

**Source:** IMS abends

**Module:** DBFCMP10, DFSCFEI0, DFSCMCP0, DFSCMS00, DFSCOF0, DFSCON20, DFSCRTU0, DFSDMIF0, DFSDMIQ0, DFSICIO0, DFSIIDM0, DFSLIEE0, DFSNCS10, DFSQUEI0, DFSSIST0

### Related information:

Chapter 2, "IMS problem determination," on page 13

---

## 0516

**Explanation:** More than one input and one output logical terminal were in conversation on the same physical terminal.

**Analysis:** This is a standard abend issued by DFSCON00 or DFSICLH0. The program status word (PSW) at entry-to-abend and the registers in the abend SVRB should be used to identify the issuing module and the applicable label below. The abend is issued as a result of branches to label ABND516. Register 12 is the base register for these modules.

### For DFSCON00:

Use register 5 in the abend SVRB to isolate to the specific label as shown in the following table:

Key	Label	Description
Reg5=BAL (to FNDCNT) Reg10=zero	CNUECNT0	A conversation is in process, and a call to handle a new message is given. FNDCNT returns using register 5 plus displacement, indicating CNT not found. This issues the abend.
Reg5=BAL (to FNDCNTNX) Reg6=CCB CCBSAV + 4 = X'00010001' Reg10=zero	CNUECNT9	It is determined at label CNUECNT3 that the input being processed is not from the most current processed conversation, so a search is made of CNTs. If all CNTs are scanned without a hit (indicated by a return using register 5 plus 0 displacement), a check is made to see if more than one input is indicated for this CCB. A not equal condition issues the abend.

### For DFSICLH0:

Use register 4 in the abend SVRB to isolate to the specific label as shown in the following table:

Key	Label	Description
Reg4=BAL (to FNDENT) Reg6=CCB Reg7=CTB Reg10=zero	HLDGIC	This routine processes the HOLD command. A return using register 4 plus 0 displacement from FNDCNT indicates that no active CNTs are associated with this CTB.
Reg4=BAL (to FNDCNTNX) Reg6=CCB CCBSAV + 4 -= X'00010001'	HLDGIC6	This routine verifies that only one input or output CNT is active.
Reg4=BAL (to FNDENT) Reg5=callers return address Reg10=zero	FREIT	This routine determines if an EXIT command can terminate a conversation. The conversational CNT must be found. A return using Reg4 plus 0 displacement indicates a "not found" condition. The abend is issued.

**System action:** The IMS control region terminates abnormally.

**Programmer response:** Ensure that only one input and one output logical terminal on the same physical terminal are concurrently in a conversation. One of them represents the logical input terminal, and the other represents the logical output terminal.

**Problem determination:** 1, 4, 25, 26, 35

**Source:** IMS abends

**Module:** DFSCON00, DFSICLH

**Related information:**

Chapter 2, "IMS problem determination," on page 13

## 0517

**Explanation:** The conversational scratchpad area (SPA) unpack routine (DFSCONU0) encountered an invalid packed SPA and returned a nonzero return code to the caller. In general, SPAs are packed when written to the message queue and logged to the log. They are unpacked when read from the message queue and processed.

The routines that call DFSCONU0 to unpack the SPA, after receiving a nonzero return code, either terminate the control region abnormally or issue a pseudoabend to terminate the message region.

**Analysis:** This is a standard abend issued by three different modules: DFSCON20, DFSCONM0, DFSECP10 (set by DFSCPY00 and others).

Locate the packed SPA and examine it for validity. The format of a packed SPA is: *PPZZBBTRANCODEUUUP...Packed Data...*

The following defines the value of *PPZZBBTRANCODEUUUP*:

*PP* A 2-byte field defining the length of a packed SPA  
*ZZ* The 2-byte Z1 and Z2 fields  
*BB* A 2-byte CCB number  
*TRANCODE*  
 An 8-byte user transaction code  
*UU* A 2-byte field defining the length of an unpacked SPA

The *Packed Data* consists of the following descriptor fields:

**8LLL** Binary zeros of length LLL deleted  
**4LLL** Valid, unpacked data of length LLL follows  
**2LLL** Blanks of length LLL deleted

The program status word (PSW) at entry-to-abend and the registers in the abend supervisor request block (SVRB) should be used to identify the issuing module and the applicable label.

**For DFSCONM0:**

## 0517

The MSC analyzer received a SPA message from a remote conversation transaction and the input terminal is in this system. DFSCONM0 calls DFSCONU0 to unpack the SPA and a nonzero return code is returned.

Key	Label	Description
Reg7=LTB and Reg9=LLB	CALLCONU	Problem determination for this error is the same as for module DFSCON20 label CONDA517.

### For DFSCON20:

When a conversational transaction is terminated abnormally, DFSCON20 is called. DFSCON20 calls DFSCONU0 to unpack the SPA, and DFSCON20 passes the unpacked SPA to the conversational abnormal exit, DFSCONE0. The SPA is unpacked even if no exit routine exists. If DFSCONU0 returns a nonzero return code to DFSCON20, this abend is issued.

Locate the input message (containing the SPA segment) to determine the cause of the abend.

Key	Label	Description
Reg5=BAL Reg7=CTB Reg9=CLB Reg10=CON SAVE work area	CONX2	SAVGB6 in CONSAVE points to a work buffer. At an offset of QTPPCBLT is the packed SPA. Examine the SPA for validity.
Reg5=BAL Reg7=CTB Reg8=packed SPA Reg9=CLB	CONFA517	Check the packed SPA pointed to by register 8 for validity.
Reg5=BAL, Reg7=LTB and Reg9=LLB	CONDA517	This error is signaled by message DFS2146 and the MSC link terminates abnormally the processing and logs type 6701 MSS1/MSS2 records. If the error is not recovered, link is PSTOPPED. CTBUEOB of link LTB points to a SPA segment. If not dumped, CLBINBUF points to an input buffer with the entire or last part of the SPA. CTBPREFIX points to the prefix of the message. This prefix, an MSC segment item, can be used to locate the message (type 01 or 03) on the log of the sending IMS system. Examine the SPA in the sending system to see if the error occurred there (that is, SPA is valid). See message DFS2146 for additional information.

### For DFSECP10:

DFSCPY00 called DFSCONU0 (VS/1) or the z/OS SPA unpack RTN to unpack a SPA into the user I/O area.

The IMS log 01 (C10 dispatch) or 03 (PST dispatch) log records preceding the pseudoabend, put on the IMS log by abend 0517 ABTERM sync point processing, will have the SPA (LLz z with z=40) as the first segment of the input transaction message. The track back through the IMS log to find the source of the bad input message or SPA could be an overlay in message queues by DC.

**System action:** The region in which the error occurred terminates abnormally.

**Problem determination:** 1, 4, 6, 25, 26, 35

**Source:** IMS abends

**Module:** DFSCONM0, DFSCON20, DFSECP10

**Related information:**

 DFS2146 (Messages and Codes)

Chapter 2, "IMS problem determination," on page 13

---

**0519**

**Explanation:** The message router, DFSICLR0, was called with an invalid enqueue request. Register 8 provides a reason code.

**Analysis:** This is a standard abend issued by modules DFSCMT10, DFSCLRM0, DFSICLR0, and DFSCLMO0. The program status word (PSW) at entry-to-abend and the register 5 BAL should be used to isolate to the applicable label below. This abend is a result of branches to a common abend routine at label ABND519. Register 8 contains an indicative reason code. Register 12 is the base register. Register 5 contains the invalid queue number that was computed.

**Codes**
**Explanation**
**X'00'**

The destination is a remote SMB but destination SYSID is local.

**X'01'**

This is an unidentified dispatch.

**X'02'**

The ASIS request on the PST is invalid.

**X'03'**

Invalid request for system routing while in multiple systems.

**X'05'**

The destination is a remote SMB but destination SYSID is invalid.

**Address**

This is the invalid link CLB pointer from the destination SYSID of the remote SMB.

**X'06'**

Invalid return code from DFSUSE func=inuse.

**X'07'**

Invalid return code from DFSUSE func=nouse.

The caller can be determined by locating the save area set prior to the save area with name at entry point of "CLR-LR2...."

Key	Label	Description
Reg5=BAL Reg8=X'02'	PSTDISP	In the subroutine labeled INIT, the type of request is validated. For this entry point, DFSICLR2, the caller has indicated dispatch as a PST. Register 1 should point to the PSTDECB; otherwise, the abend will occur.
Reg8=X'06'		Contains the invalid return code from DFSUSE func=inuse. Register 15 should contain non zero return code.
Reg8=X'07'		Contains the invalid return code from DFSUSE func=nouse. Register 15 should contain non zero return code.

**System action:** The multiple systems VS control region terminates abnormally.

**Problem determination:** 1, 4, 5, 6, 10, 12, 35

**Source:** IMS abends

**Module:** DFSCMT10, DFSCLMR0, DFSICLR0, DFSCLMO0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

---

0525

**Explanation:** One of the following problems occurred:

- The hardware clock was inoperable.
- IMS initialization detected that the GMT offset value is outside the valid range, and the operator replied A (for abort) to message DFS0477A.

**Analysis:**

This is a standard abend issued by DFSTIME0, the timer service routine. At the time of abend, register 15 contains the reason code.

Key	Label	Description
Reg15=X'02'	ABEND	The STCK instruction failed because the hardware clock is inoperable.
Reg15=X'03'	ABEND	DFSTIME0 finds that the UTC offset value from the CVT is out of valid range and the operator replied A (for abort) to message DFS0477A.

**Possible cause:** The causes of this abend include:

- A hardware failure occurred.
- An internal program error occurred.

**System action:** IMS terminates abnormally, or issues message DFS0478A until the problem is corrected.

**Programmer response:** Contact the IBM Support Center for hardware or software support, or follow the instructions in the operator response in message DFS0477A or DFS0478A.

**Source:** IMS abends

**Module:** DFSTIME0

**Related information:**

 DFS0477A

 DFS0478A

---

0545

**Explanation:** During log type 2702 record processing on the alternate, an inconsistent state between the active and the alternate's data set extent information existed at the completion of the pseudo-extend.

**System action:** The IMS alternate system abends.

**Operator response:** Restart the IMS alternate system.

**Source:** IMS abends

**Module:** DFSRELP0

---

0551

**Explanation:** GSAM PCBs were present, but the dependent region was not a batch message processing region (BMP).

**Analysis:** This is a standard abend issued by the MPP/BMP program controller, DFSPCC20. The program status word (PSW) at entry-to-abend will point to the instruction within label PSABEND from which the abend (SVC 13) is issued. It is branched to by the routine at PCAB551 which is, in turn, branched to by the routine at PC09 that detects the error.

Key	Label	Description
Reg2=address of PX  S (DFSPRPX0) dsect Reg10=address of DIRCA2	PC09	The field PDIRSPLT of DIRCA2 is tested to see if GSAM PCBs are present. If yes, the RCTYPE field of the RCPARMS block is tested to see if this is a BMP region. If GSAM PCBs are present and if the region is not a BMP, a branch is taken to PCAB551 to handle the abend.

**Possible cause:** The application programmer may have defined the application program incorrectly as an MPP when it should have been a BMP.

**System action:** The dependent region terminates abnormally.

**Programmer response:** The application was incorrectly defined. Identify it as a BMP.

**Source:** IMS abends

**Module:** DFSPCC20

## 0552

**Explanation:** A conversational MPP terminated abnormally in a multiple systems environment. The SPA cannot be sent to the input terminal system where the conversation is controlled, because no MSNAME exists for the input SYSID. The input SYSID is used for responses.

**Analysis:** This is a standard abend that can be issued by one of two modules: DFSCON10 or DFSCON20. The program status word (PSW) at entry-to-abend will isolate the failure to a particular module.

Register 1 in the abend SVRB registers will contain the abend completion code, X'80000228'.

**Possible cause:** Probable system definition error. Ensure that an MSNAME parameter is supplied for each system while in multiple systems.

### For DFSCON10:

This abend can be issued by the conversational processor inserting a SPA, module DFSCON10. The abend is issued from a common routine at label ABND.

0522 is the result of the system ID (SID) of the input terminal system in the PST either not being defined in this system, or it is invalid. Register 9 in the abend SVRB registers contains the address of the PST. Register 6 contains the CCB address. Register 5 is used as a BAL register. Register 12 is the base register for CSECT DFSCON10, the entry point for a problem program returning an SPA to the queue. Register 12 is the base register for CSECT DFSCONL0, the entry point to update control blocks in relation to the conversation if no SPA was inserted by the application prior to a sync point being reached. DFSCON10 is called if a SPA segment is found by the COMM portion of the user call analyzer, DFSDLA30, from the sync point if the SPA was not inserted.

Key	Label	Description
Reg4<zero Reg11=address of SCD	CONIWASE	While in multiple systems, the routine within CSECT DFSCON10 does a BAL to label VERSID to verify the SID of the input terminal system. Register 4 is loaded with the logical link name block (LNB) pointer in the SID. When tested, if register 4 is negative, the routine branches and links to the label ABND552 to handle the abend.
Reg4<zero Reg8=pointer to SPA Reg11=address of SCD	CONLWASE	While in multiple systems, the routine within CSECT DFSCONL0 does a BAL to label VERSID to verify the SID of the input terminal system. Register 4 is loaded with the logical link name block (LNB) pointer from the SID. When tested, if register 4 is negative, the routine branches and links to label ABND552 to handle the abend.

### For DFSCON20:

## 0553

This abend can be issued by the conversational processor terminal module, DFSCON20. The program status word (PSW) at entry-to-abend will point to the instruction within label ABND552 from which the abend (SVC 13) is issued. This label is conditionally branched to by the routine that detected the error.

The 0552 abend results because the SID for response routing has no LNB (MSNAME) defined in this system. Register 12 in the abend SVRB registers is the base register. Register 5 is used as a BAL register. Register 10 contains a pointer to a vector for communication with DFSCONE0, the conversation abnormal termination exit routine. DFSCOND0 is an entry point in DFSCON20 and is called by the link analyzer (DFSCMS00) when a "returned" ERRSPA message reaches the input system (application abend/inconsistent definition/stopped SMB in processing system). DFSCOND0 is called by the processing system when an ERRSPA message must be returned to the input system (inconsistent definition or stopped SMB detected by link analyzer in *processing* system).

Key	Label	Description
Reg3<zero Reg11=address of SCD	VERRMT2	In the routine which verifies the SID, register 3 is loaded with the logical link name block (LNB) pointer. When tested, if register 3 is negative, a branch is taken to label ABND552 to abend.

**System action:** The IMS control region terminates abnormally.

**Programmer response:** Provide MSNAME in the abending system for input SYSID.

**Problem determination:** 1, 4, 5, 6, 10, 12, 35

**Source:** IMS abends

**Module:** DFSCON10, DFSCON20

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

## 0553

**Explanation:** Module DFSCMM20 posted the link between two main storage-to-main storage IMS systems with a write completion code, but an input/output operation was already in progress. The main storage-to-main storage post handler (DFSCMM20) received a contention posting indication in the ECB(LXB) but no I/O was in progress on the target CLB(LLB). This is an invalid situation, as input/output activity is handled synchronously on the link.

**Analysis:** This is a standard abend issued by DFSCMM20. The data arriving from another IMS machine causes the input area ECB(LXB) to be posted. The dispatcher passes control to DFSCMM20 to determine the validity of the post; in this instance contention posting in the ECB(LXB), but no I/O in progress was indicated on the target CLB(LLB).

The program status word (PSW) at entry-to-abend will point to the instruction with label CMM20BF0 from which the abend (SVC 13) is issued. When the error is detected a conditional branch is made to label CMM20B50, register 12 is loaded with the abend code, and an unconditional branch is made to CMM20BF0 for abend initiation. The following registers in the abend SVRB are relevant to this abend: Register 9=CLB(LLB) pointer, register 10=pointer to the ECB(LXB), and register 8=the base register.

Key	Label	Description
Reg1=X'80000229'	CMM20A00	Contention posting was indicated in the ECB(LXB), but a check of the CLBFLAGS determined that no I/O was in progress.

**Possible cause:** IMS logic error.

**System action:** IMS terminates abnormally.

**Problem determination:** 1, 4, 25, 35

**Source:** IMS abends

**Module:** DFSCMM20

**Related information:**

Chapter 2, "IMS problem determination," on page 13



## 0554

**Explanation:** The main storage-to-main storage post handler module (DFSCMM20) received a read attention indicator on the main storage-to-main storage link. This is an invalid request.

**Analysis:** This is a standard abend issued by DFSCMM20. The ECB(LXB) was posted with a read attention (X'7FD9'). This is an illogical code.

The program status word (PSW) at entry-to-abend will point to the common abend routine CMM20BF0. When the error is detected a conditional branch to label CMM20B60 is taken, register 12 is loaded with the abend code, and a branch is taken to CMM20BF0 to issue the abend. Register 10 in the abend SVRB will point to the ECB(LXB).

Key	Label	Description
Reg1=X'8000022A'	CMM20B01	Read attention is posted in the ECB(LXB). This is an invalid post.
Reg10=A (ECB(LXB))		

**Possible cause:** Probable logic error in the main storage-to-main storage access method (DFSMTMA0).

**For APAR processing:** Save a memory dump from both systems with the access method trace invoked.

**System action:** IMS terminates abnormally.

**Problem determination:** 1, 4, 25, 35

**Source:** IMS abends

**Module:** DFSCMM20

**Related information:**

Chapter 2, "IMS problem determination," on page 13

## 0555

**Explanation:** Module DFSCMM20 posted the main storage-to-main storage link with a post code other than a write or read attention while IMS was preparing to do an input/output operation on that link. Also the CLB(LLB) is currently posted or dispatched.

**Analysis:** This is a standard abend issued by DFSCMM20. The post of the ECB(LXB) was invalid, neither write attention nor contention, and the CLB(LLB) is currently dispatched as indicated by the CLBFLAG3 field. This is a logic error and should only occur if the target CLB(LLB) is not dispatched prior to a valid posting.

The program status word (PSW) at entry-to-abend will point to the common abend routine CCM20BF0. The abend is setup by a branch to CMM20B70 which loads the abend code into register 12 and branches to CMM20BF0. Register 10 in the abend SVRB is the pointer to the ECB(LXB), register 9 is the pointer to the CLB(LLB) and register 8 is the base register. CLBDECB=CLB + 0 and CLBFLAG3=CLB+X'32'.

Key	Label	Description
Reg1=X'8000022B'	CMM20B01	If CLBDECB=X'40' and CLBFLAG3 ≠ X'01', an invalid ECB(LXB) post was detected while the CLB(LLB) was waiting to be run (that is, posted, but not yet dispatched). If CLBFLAG3=X'01', an invalid ECB(LXB) post was detected while the CLB(LLB) was currently dispatched.
Reg9=A(CLB(LLB))		
Reg10=A(ECB(LXB))		

**Possible cause:** IMS logic error.

**For APAR processing:** Save memory dumps from both systems with the access method trace invoked.

**System action:** IMS terminates abnormally.

**Problem determination:** 1, 4, 25, 35

**Source:** IMS abends

**Module:** DFSCMM20

**Related information:**

**0556**

**Explanation:** Module DFSCMM20 posted the main storage-to-main storage link with an error post code because of a previous input/output operation. The main storage-to-main storage post handler (DFSCMM20) received an invalid ECB(LXB) posting, and the target CLB(LLB) has no output I/O pending.

**Analysis:** This is a standard abend issued by DFSCMM20. The program status word (PSW) at entry-to-abend will point to the common abend routine CMM20BF0. When the error is detected a conditional branch to CMM20B80 is taken to load the abend code in register 12 and unconditionally branch to the common abend routine.

The ECB(LXB) was posted with an invalid code; the valid code is write attention/contention (X'E6D9E3'). The CLBFLAG2 field indicates that no I/O is in progress on the target CLB(LLB), pointed to by register 9 in the abend SVRB. CLBFLAG2=CLB+X'31'.

Key	Label	Description
Reg1=X'8000022C' Reg9=CLB(LLB) CLBFLAG2 = X'06'	CMM20B01	The CLBFLAG2 field indicates that no I/O is pending for this CLB(LLB).

**Possible cause:** IMS logic error.

**For APAR processing:** Save memory dumps from both systems with the access method trace invoked.

**System action:** IMS terminates abnormally.

**Programmer response:** Bring up IMS again, and continue to run the job. If the problem persists, contact the IBM Support Center for assistance.

**Problem determination:** 1, 4, 25, 35

**Source:** IMS abends

**Module:** DFSCMM20

**Related information:**

Chapter 2, "IMS problem determination," on page 13

**0557**

**Explanation:** During restart processing of a Multiple Systems Coupling (MSC) system, the LCB/LLB resolution module (DFSCMR00) validates a multiple systems environment. An address for the logical link blocks does exist, but an address for the logical control blocks does not exist in the system contents directory.

**Analysis:** This is a standard abend issued by DFSCMR00. DFSCMR00 first determines that MSC is in the system by checking for an LLB address in the SCD(SCDLLB). The LLBs are scanned to turn off the LCB allocated indicator, then the LCBs are checked to see if they were previously assigned to a LLB. If so, the connection is reestablished by placing the LCB address in the LLB.

This failure is caused by the fact that the SCD did not contain a LCB address (SCDLCB), but the LLB address (SCDLLB) is available.

The caller of this module is DFSRNRE0, register 11 in the abend SVRB will point to the SCD, and register 15 is the base register.

Key	Label	Description
Reg1=X'8000022D' Reg10=zero	CMR0A003	The test to verify the existence of the LCBs Failed (SCDLCB=X'00').

**Possible cause:** Incorrect restart procedures

**System action:** IMS terminates abnormally.

**Problem determination:** 1, 4, 6, 35

**Source:** IMS abends

**Module:** DFSCMR00

**Related information:**

Chapter 2, "IMS problem determination," on page 13

## 0560

**Explanation:** In attempting to acquire a buffer, the message generator received an invalid return code from the queue manager.

**Analysis:** This is a standard abend issued from DFSCLMR0. Use the program status word (PSW) at entry-to-abend and the register 14 BAL in the abend SVRB to determine the applicable label. The queue manager return code in register 15 should be used to identify the cause of the abend. Register 12 is the base register for this module.

DFSQMGR0 return codes are as follows:

### Codes

#### Explanation

**X'00'**

No errors

**X'04'**

Invalid segment length

**X'08'**

No more messages

**X'0C'**

Destination in use for input

For prefix updates, the type 30 log record is helpful in problem analysis.

Key	Label	Description
Reg2=QTPPCB Reg14=BAL (to DFSQMGR0) DECTYPE=X'4A' (DLI caller) DECTYPE=X'8A' (MG caller) DECAREA=length of buffer requested	GTBFGMGR	A request was made to the queue manager to obtain space in a message buffer in which to put a message. An invalid length is returned and the abend results.
Reg2=QTPPCB Reg14=BAL (to DFSQMGR0) DECTYPE=X'49' DECTYPE=X'89'	CLMRCANC	A request was made to cancel an output message. No logging is requested. A nonzero return code results in the abend.
Reg0=Message Area Reg5=QTPPCB Reg6=Message Prefix Reg8=Message Destination Reg9=DECB Reg11=SCD DECTYPE=X'4B'	LU62MP62	A request was made to insert a LU62 message prefix in the queue buffer. A nonzero return code results in the abend.

**System action:** IMS terminates with a memory dump.

**Programmer response:** Examine register 15 in the dump to isolate the queue manager's return code. The return code will identify the problem.

**Problem determination:** 1, 4, 5, 6, 35

## 0561 • 0562

**Source:** IMS abends

**Module:** DFSCLMR0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

### 0561

**Explanation:** The main storage-to-main storage post handler (DFSCMM20) was called to service a post request, but the link number assigned to the link control block at system initialization was 0. The link number must be at least 1.

**Analysis:** This is a standard abend issued by DFSCMM20. The program status word (PSW) at entry-to-abend will point to the common abend routine CMM20BF0. A conditional branch is made to label CMM20B90 to load the abend code into register 12 and branch to the common abend routine. Register 6 in the abend SVRB is a pointer to the LCB.

The post handler is entered with a posted ECB(LXB) in register 1. Link control block (LCB) addressability is established and the link number of the target logical link block (LLB) is determined. Register 9 is loaded with a halfword from the LCBLNR field and tested for a value greater than zero. LCBLNR=LCB+X'0A'.

Key	Label	Description
Reg1=X'80000231' Reg6=A(LCB) LCBLNR=0 Reg9=zero		The link number of the target LLB is zero.

**Possible cause:** Probable system definition problem building the LCB blocks.

**System action:** IMS terminates abnormally.

**Programmer response:** Investigate any application program that might obtain access to the IMS control region to ensure proper addressability throughout the module.

**For APAR processing:** Keep the output of system definition and binder listing.

**Problem determination:** 1, 4, 5, 6, 35

**Source:** IMS abends

**Module:** DFSCMM20

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

### 0562

**Explanation:** A main storage-to-main storage buffer incompatibility occurred. An IMS main storage-to-main storage system has transmitted more data than its counterpart's input buffer can hold. Buffer lengths should be identical.

**Analysis:** This is a standard abend issued by DFSCMM20. The main storage-to-main storage access method (DFSMTMA0) determines that the output buffer size exceeds the input buffer size, indicates the error in the LXB, and issues a post. The main storage-to-main storage post handler (DFSCMM20) recognizes the post code as being invalid and determines that an incompatible buffer length condition exists.

The program status word (PSW) at entry-to-abend will point to the common abend routine CMM20BF0. The post handler determines the type of error, a conditional branch is made to CMM20BA0, and register 12 is loaded with the abend code. Register 10 in the abend SVRB is the pointer to the posted LXB, and register 8 is the base register.

Key	Label	Description
Reg1=X'80000232' Reg10=A(LXB)	CMM20B01	The LXB is posted with a length error (X'41D3C5D5').

**Possible cause:** The input/output buffers are not the same size in the two IMS MSC systems.

**System action:** IMS terminates abnormally.

**Programmer response:** Ensure that main storage-to-main storage input/output buffers have the same length.

**For APAR processing:** Create memory dumps of both systems

**Source:** IMS abends

**Module:** DFSCMM20

### 0563

**Explanation:** During system restart, all the link control blocks (LCB) are reconnected to the logical link blocks (LLB), as they were before the system came down. All link numbers are set relative to 1, but one of the link control block numbers is 0.

**Analysis:** This is a standard abend issued by module DFSCMR00.

Key	Label	Description
Reg1=0233	CMR0C001	LCB1ASS flag is on and LLBLNR field is 0.

**Possible cause:** Error at system definition time.

**System action:** IMS terminates abnormally.

**Programmer response:** Ensure that the link number is correct during restart.

**Problem determination:** 4, 35

**Source:** IMS abends

**Module:** DFSCMR00

**Related information:**

Chapter 2, "IMS problem determination," on page 13

### 0564

**Explanation:** A device-dependent module called the communication analyzer (DFSICIO0) to obtain an extra input/output work buffer, however an extra buffer already exists. This situation occurs because of a system error.

**Analysis:** This is a standard abend issued by the communication analyzer, DFSICIO0. The abend is issued from a common routine at label GXTBUF. The error resulting in the abend is also detected in this routine.

The Common Services entry (DFSCIOC0) is used by selected subroutines to provide analyzer functions which would require multiple user interfaces if not provided. Register 4 contains an entry vector value for the function of this CSECT. In the case of abend 0564, register 4 should contain an X'2C'—function to get an extra work buffer.

Register 12 in the abend SVRB registers is the primary base register; register 6 is the secondary base register. Register 9 contains a pointer to the CLB. Register 1 contains the abend completion code, X'80000234'.

Key	Label	Description
Reg4=X'2C' Reg9=A(CLB)	GXTBUF	In the routine to get an extra work buffer for DDM, the field CLBXTBUF is checked to determine if an extra work buffer already exists. If the field is not zero, the abend is issued.

**System action:** IMS terminates abnormally.

**Problem determination:** 1, 4, 5, 6, 35

**Source:** IMS abends

**Module:** DFSICIO0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

**0566**

**Explanation:** An internal logic error was detected while processing an internal and remote command. A message queue buffer might have been destroyed.

**Analysis:** This is a standard abend issued by the multiple-system remote command controller, DFSCMI00. This module is a multi-CSECT module, with the abend being issued from within CSECT DFSCMI06. The purpose of this CSECT is to SNAP all available blocks to the log.

Register 12 in the abend SVRB registers is the base register. Register 14 is used as a BAL register. Register 2 contains a pointer to the CTB. Register 1 contains the abend completion code, X'80000236'.

Key	Label	Description
Reg5=return address Reg15 != zero	REFETCH	It has been determined that a message must be fetched, and a BAL is taken to the routine at label CALLQ to perform this function. Register 15 is loaded with, and tested for, the return code from the queue manager. If register 15 is not zero, the abend is issued.

**System action:** IMS terminates abnormally.

**Programmer response:** No action is required.

**Problem determination:** 35

**Source:** IMS abends

**Module:** DFSCMI00

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

**0567**

**Explanation:** The communication analyzer detected an I/O error condition in sending the IMS ready message to the system console.

**Analysis:** This is a standard abend issued by DFSCINB0. The error registers are located in the abend SVRB. The program status word (PSW) at entry-to-abend points to the instruction within label IN1BABND from which abend (SVC 13) is issued. The abend completion code is loaded into register 2 and a branch to INBABND is taken. Register 12 is the base register for this module.

Key	Label	Description
Reg10=BAL to DFSIC100 Reg13 + X'C' != 0	INBA01	During communications initialization the Ready message is sent to the system console. The communications analyzer passes the return code back to the caller in the current save area. A nonzero return indicates an I/O error condition.

**System action:** IMS terminates abnormally.

**Programmer response:** No action is required.

**Problem determination:** 23, 35

**Source:** IMS abends

**Module:** DFSCINB0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

**0568**

**Explanation:** The communication initialization module (DFSCINB0) received a nonzero return code from the queue manager in an attempt to acquire a buffer from the queue manager to transmit the IMS Ready message.

**Analysis:** This is a standard abend issued by DFSCINB0. The error registers are located in the abend SVRB.

The program status word (PSW) at entry-to-abend will point to the instruction with label INBABND from which abend (SVC 13) is issued. At label INB4ABND the abend completion code is loaded into register 2. Register 12 is the base register for this module. Register 15 contains the queue manager return code:

**Codes****Explanation****X'00'**

Buffer required.

**X'04'**

Incorrect length specified; no buffer acquired.

Key	Label	Description
Reg9=DECB DECTYPE=X'0A00' (ISRT LOCATE) DECAREA=message length Reg14=BAL to DFSQMGR0 Reg15 = 0	INBQCR	In this routine the queue manager is called to acquire a buffer for the ready message. If any errors are detected, the queue manager passes back the error code in register 15 and abend is issued.

**System action:** IMS terminates abnormally.

**Programmer response:** No action is required.

**Problem determination:** 23, 35

**Source:** IMS abends

**Module:** DFSCINB0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

**0572**

**Explanation:** The length of an inner segment of a multisegment pre-edit message was a negative value.

**Analysis:** This is a standard abend issued by DFSCLMR0. Use register 3 in the abend SVRB to isolate a specific label.

Register 12 is the base register for this module.

Key	Label	Description
Reg3=negative value Reg8=pointer to segment length field of the message	LLZZSSZZ	During processing of a multisegment preedit message, it is found that the segment length field of an inner segment is negative. This results in the ABEND.
Reg3=positive value Reg5=BAL to STATISZ1 Reg8=pointer to the SSZZ field of the inner segment containing a negative value	ADDSSINC	For the same multisegment message described above, this routine increments through the inner segments testing for a negative SS field. The test is accomplished by testing that the high-order bit in the SS field is not on. Should this bit be on, an abend will occur.

**System action:** IMS terminates abnormally.

**Problem determination:** 4, 35

## 0573 • 0577

**Source:** IMS abends

**Module:** DFSCLMR0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

### 0573

**Explanation:** The length of an inner segment of a multisegment message was larger than the total length of the message. One of the length fields was incorrect.

**Analysis:** This is a standard abend issued from DFSCLMR0. The registers in the abend SVRB should be used for problem isolation.

DFSCLMR0 is processing a multisegment preedit message call. Register 12 is the base register for this module.

Key	Label	Description
Reg10 + X'34'=total segment length (WORK10) Reg15=inner segment length	LLZZSSZZ	This code ensures that no inner segment is greater in length than the total segment length. A comparison is made between register 15 and WORK10; should register 15 be greater, an abend is issued.

**System action:** IMS terminates abnormally.

**Problem determination:** 4, 35

**Source:** IMS abends

**Module:** DFSCLMR0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

### 0574

**Explanation:** The Fast DB Recovery region encountered an error. A message describing the error precedes this abend code.

**System action:** The Fast DB Recovery region terminates abnormally.

**Operator response:** See the operator response of the message issued before this abend.

**Source:** IMS abends

---

### 0577

**Explanation:** This abend occurred because input processing was selected for a VTAM terminal, but VTAM support was not generated for DFSICIO0.

**Analysis:** Verify "DFSVTAM COPY" for the global &DFSVTAM=N. If VTAM support is required, add the COMM macro to the IMS system definition.

Key	Label	Description
&DFSVTAM=N	IPCHECK	DFSICIO0 does not generate VTAM support code with &DFSVTAM=N. Instead, it generates the code to issue abedn 0577.

**System action:** IMS terminates abnormally.

**System programmer response:** Verify "DFSVTAM COPY" for the global &DFSVTAM=N; If VTAM support is required, add the COMM macro to the IMS system definition.

**Source:** IMS abends



**Module:** DFSICIO0

### 0578

**Explanation:** This abend occurred because the CTT device type (CTTDEVIC) was either in binary zeros or the value was less than the system console (X'33"). CTTDEVIC is picked up and examined for each defined terminal during Transaction Manager initialization (control TCB). IMS terminal device types begin with number 1.

Both of the above conditions are obvious errors and are probably caused by one of the following conditions:

1. The stage 2 assembly of DFSCLL0x had an error.
2. The link edit of DFSCLL0x had an error.

**Note:** The *x* value in the module name indicates the nucleus suffix.

**Analysis:** This is a standard abend issued by module DFSIINB0. Ensure that the stage 2 assembly of DFSCLL0x produces acceptable output and that the subsequent bind occurs without errors.

Key	Label	Description
Reg7=A(CTG)	OPN010	The first word in the CTB is the address of CTT which is incorrect. The first byte of CTT is the device type (CTTDEVIC).

**System action:** IMS terminates initialization processing.

**Operator response:** Ensure that the stage 2 assembly of DFSCLL0x produced acceptable output and that the subsequent bind occurred without errors. Notify the appropriate system programmer to correct the IMS problem.

**Problem determination:** 1, 4, 6, 35

**Source:** IMS abends

**Module:** DFSIINB0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

### 0579

**Explanation:** The control region initialization was unable to successfully open any line groups. DFSIINB0 has detected an internal IMS error during IMS Transaction Manager initialization. Further initialization cannot continue.

**Analysis:** This is a standard abend issued by module DFSIINB0. The registers at entry-to-abend contain the following information:

Register 8 = CIT that contains the invalid device-type code.

Register 15 = The invalid device-type code.

**System action:** IMS terminates abnormally.

**Programmer response:** Ensure the correct IMS control blocks were linked into the IMS nucleus.

**Problem determination:** 1, 4, 6, 35

**Source:** IMS abends

**Module:** DFSIINB0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

### 0580

**Explanation:** Either DFSIINB0 detected an IMS internal error during IMS Transaction Manager initialization, or DFSDINB0 detected an IMS internal error during initialization of a Database Control (DBCTL) subsystem. Initialization cannot continue.

**Analysis:** For DFSIINB0:

## 0581

An IMODULE GETMAIN failed to obtain CSA storage for the CSACLB (ECB) for DFSCMTI0; this problem is an IMS internal error.

0580 is a standard abend issued by module DFSIINB0. Register 3 contains the length of the CSA to be obtained by GETMAIN. Register 15 contains one of the following return codes:

### Codes

#### Explanation

#### X'04'

Invalid function or option.

#### X'08'

Execution of z/OS was not consistent with assembler z/OS for SVC routine.

#### X'14'

Insufficient storage in subpool, or the requested length was zero.

#### X'18'

Program check or other internal error.

### For DFSDINB0:

An IMODULE GETMAIN failed during DB Control (DBCTL) initialization.

Message DFS697I is issued along with this abend. The EPLOC field in the message contains the characters DFSDWBUF to indicate that IMODULE GETMAIN failed for DWBUF storage. Register 4 in the abend SVRB contains the size of the DWBUF area requested. The message also displays the IMODULE return code.

**System action:** IMS terminates abnormally.

**Programmer response:** No action is required.

**Problem determination:** 1, 4, 6, 35

**Source:** IMS abends

**Module:** DFSIINB0, DFSDINB0

**Related information:**

 DFS697I

Chapter 2, "IMS problem determination," on page 13

---

## 0581

**Explanation:** DFSIINB0 or DFSIINV0 detected an internal IMS error during IMS Transaction Manager initialization. Initialization cannot continue. The destination-find module (DFSICLF0) was unable to locate a CNT/LNB/RCNT, defined for a VTAM/MSC terminal.

**Analysis:** This is a standard abend issued by modules DFSIIMS0, DFSIINB0, DFSIINU0, and DFSIINV0.

### For DFSIINB0:

The destination-find module (DFSICLF0) was unable to locate a system control block DFSMTCNT-CNT or DFSRMCNT-CNT; this is an internal IMS error.

Register 2 contains a pointer to the destination to be located.

### For DFSIINV0:

The destination find module (DFSICLF0) was unable to locate a system control block (CNT) related to a set of VTAM terminal blocks (VTCB). Register 2 points to the destination that could not be located.

**Possible cause:** The IMS module containing the CNT blocks (DFSVNUCx) was changed while the IMS module containing the VTCB blocks was not changed, or vice versa.

Alternatively, a required destination could not be found, possibly due to a JCLIN error.

**System action:** IMS terminates abnormally.

**Programmer response:** Be sure the correct IMS control blocks were linked into the IMS nucleus. If the blocks were linked correctly, contact IBM Software Support for help in determining the problem.

**Problem determination:** 1, 4, 6, 35

**Source:** IMS abends

**Module:** DFSIIMS0, DFSIINB0, DFSIINU0, DFSIINV0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

## 0582

**Explanation:** DFSIINB0 detected an internal IMS error during IMS Transaction Manager initialization. Initialization cannot continue. The DFSBCB operation was unable to obtain storage for a VTAM Receive Any buffer. This is an internal IMS error.

**Analysis:** This is a standard abend issued by module DFSIINB0. For a description of the DFSBCB return codes, see the IMS system services return codes information.

**System action:** IMS terminates abnormally.

**Programmer response:** Check the following:


- The number of VTAM Receive Any buffers specified in the RECA keyword of the EXEC parameter
- The number and size of the VTAM Receive Any buffers in the RECANY parameter of the IMS system definition COMM macro.

**Problem determination:** 1, 4, 6, 10, 35

**Source:** IMS abends

**Module:** DFSIINB0

**Related reference:**

 [IMS system services return codes \(Messages and Codes\)](#)

**Related information:**

Chapter 2, "IMS problem determination," on page 13

## 0583

**Explanation:** System definition was performed without VTAM; however, the SCD control block indicates that VTAM control blocks exist.

**Analysis:** This is a standard abend issued by module DFSIINB0.

**Possible cause:** The IMS/VTAM control blocks might be bound with a non-IMS/VTAM nucleus.

**System action:** IMS terminates abnormally.

**Programmer response:** Ensure the correct SCD was linked into the IMS nucleus.

**Problem determination:** 1, 4, 6, 35

**Source:** IMS abends

**Module:** DFSIINB0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

## 0584

**Explanation:** An error situation was detected by the Automated Operator Interface Message Generator. Register 5 contains the address in the module where the error was detected. Register 2 contains a code indicating the reason for the abnormal termination.

**Analysis:** 0584 is a standard abend issued by the AOI message generator module, DFSCLMA0. Register 5 points to

## 0585

the code in the module where the error was detected. Register 2 contains an error code indicating the reason for the abend, as follows:

### Codes

#### Explanation

#### X'04'

The message generator was called to enqueue a message for a remote destination which was not an LTERM.

#### X'08'

A nonzero return code was received from the queue manager on a PUT LOCATE call. Register 15 contains the queue manager return code.

#### X'0C'

A nonzero return code was received from the queue manager on a CHANGE PREFIX call. Register 15 contains the queue manager return code.

#### X'10'

A nonzero return code was received from the queue manager on an ENQUEUE call. Register 15 contains the queue manager return code.

#### X'14'

Message generator was called with an invalid destination parameter.

**System action:** The IMS control region is abnormally terminated.

**Operator response:** Perform an emergency restart.

**Problem determination:** 1, 4, 5, 6, 3

**Source:** IMS abends

**Module:** DFSCLMA0

## 0585

**Explanation:** An error occurred during initialization of the XRF-capable IMS system. The subcode in register 15 indicates the specific problem.

**Analysis:** The subcode in register 15 indicates the specific problem.

Key	Label	Description
Reg15=X'01'	HOTI01	The HSBID parameter value is invalid.
Reg15=X'02'	HOTI65	The XRF system link with Intersystem Communication (ISC) interface was defined at IMS system definition, but a valid communication line block (CLB) does not exist. This is an internal IMS error.

**Possible cause:** The causes include:

- If register 15 is X'01', the HSBID parameter in the EXEC statement of the control region execution procedure is invalid.
- If register 15 is X'02', an internal IMS error occurred. Contact IBM Software Support for assistance.

**System action:** The control region is terminated abnormally.

**Programmer response:** If register 15 = 1, correct the HSBID parameter value in the EXEC statement of the control region execution procedure. If register 15 = 2, contact IBM Software Support for assistance.

**Problem determination:** 1, 2, 3, 4, 6, 35

**Source:** IMS abends

**Module:** DFSIINB0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

## 0586

**Explanation:** One of the following errors occurred during initialization of an XRF-capable IMS system:

- The system detected an invalid value on the HSBID parameter on the EXEC statement of the control region. Only HSBID=1, HSBID=2, or HSBID=null is allowed. HSBID=null deactivates the XRF capability.
- The DFSHSBxx member of the procedure library could not be opened.

**Analysis:** This is a standard abend issued by DFSHINT0 during XRF initialization. An invalid HSBID parameter value was detected on the EXEC statement, or the DFSHSBxx procedure library member could not be opened.

Key	Label	Description
Reg15=X'01'	ABND5861	The HSBID parameter value on the EXEC statement is not 1, 2, or blank. FRBHOTID, pointed to by register 10 plus X'118', contains the invalid value.
Reg15=X'02'	ABND5862	The DFSHSBxx procedure library member could not be opened.

**Possible cause:** The causes include:

- An invalid parameter was specified.
- An invalid HSBMBR parameter was specified.

**System action:** The control region is terminated abnormally.

**Programmer response:** For the first error, correct the HSBID parameter; for the second error, correct the HSBMBR parameter.

**Problem determination:** 1, 2, 3, 4, 6, 35

**Source:** IMS abends

**Module:** DFSHINT0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

## 0587

**Explanation:** Fast Path emergency restart could not complete because a request for storage failed or a lock request failed. A return code of X'08' - X'18' indicates that the abend was issued by the Fast Path emergency restart routine (DBFERST0) due to a nonzero return code from the DEDB forward recovery routine (DBFERS10) . A return code of X'1C', X'20', X'24', or X'28' indicates that the abend was issued by the DEDB emergency restart update log record processor (DBFERDB0) .

**Analysis:** Register 15 contains one of the following return codes:

**Codes****Explanation****X'08'**

Unable to get a Quick Save Area.

**X'0C'**

Unable to GETMAIN temporary storage.

**X'10'**

Unable to create any Forward Recovery ITASKs because of insufficient storage.

**X'14'**

Unable to create SRB because of insufficient storage.

**X'18'**

Unable to acquire a CI lock from DBFMGXC0. The table below tells how to determine why the lock request failed.

Key	Label	Description
Reg5=A(DMHR) Reg8=A(DMAC) Reg15=X'18' Reg15=X'1C' Reg15=X'24'	ERSTDBCL	DBFMGXC0 was unable to get a CI lock and received a nonzero return code from the Fast Path Lock Request Handler (DBFLRH00).  The IRLM passes back a return code and a reason code that explains why the lock request was rejected. You can find these codes in the following way:  Locate the RESTART PST pointed to by SCDRSTEB. Location PSTLRXRC contains the return code from the IRLM, and location PSTLRXFB contains the reason code from the IRLM. IRLM codes are explained in IRLM messages and codes.

**X'1C'**

Unable to get storage for a buffer during processing of a 5947 log record.

**X'20'**

Unable to get storage needed in an extended recovery facility (XRF) alternate region or a Fast Database Recovery region (FDBR region) to process a 5950 private buffer log record. In the previous log record processing, only half the number of buffers could be obtained.

**X'24'**

Unable to get storage needed to process a 5950 private buffer log record.

**X'28'**

An extended recovery facility (XRF) alternate region or a Fast Database Recovery (FDBR) region is unable to get a private buffer for a shared VSO area.

**System action:** The control region terminates abnormally.


**Programmer response:** For the return codes and their meanings, see the Analysis section.

**Problem determination:** 1, 4, 6, 35

**Source:** IMS abends

**Module:** DBFERST0, DBFERDB0

**Related tasks:**

 IRLM return and reason codes

**Related information:**

Chapter 2, "IMS problem determination," on page 13

## 0588

**Explanation:** An attempt to load and link to user-written, dependent-region, pre-initialization routines failed.

**System action:** The dependent region terminates abnormally.

**Programmer response:** This abend is preceded by message DFS3848 or DFS3849. In either case, a BLDL macro instruction detected the problem. BLDL searches the link library and the job or step library. Either the modules listed in the error message was not found in any of these libraries or an I/O error occurred while searching one of these libraries.

**Problem determination:** 1, 2, 3, 6, 11, 35

**Source:** IMS abends

**Related information:**

 DFS3848 (Messages and Codes)

 DFS3849 (Messages and Codes)

Chapter 2, "IMS problem determination," on page 13

**0589**

**Explanation:** An internal system error occurred. Either a request to I/O toleration services for an extended error queue element (EEQE) control block failed, or a request to get or release an asynchronous work element (AWE) failed.

**Analysis:** For DFSHREQ0:

A FIND BLOCK request to CBT Services failed.

This is a standard abend issued from module DFSHREQ0 by an XRF alternate system during takeover. DFSHREQ0 creates I/O Tolerant EEQEs for all buffers tracked by the XRF alternate system. If a FIND BLOCK request to CBT Services for a DDIR (DMB Directory) fails, this abend is issued. The FIND BLOCK request failed because of incorrect search parameters (usually the DDIR name). The DDIR name is obtained from the Buffer Tracking tables. Make sure that the IMS log is available for problem analysis.

Key	Label	Description
Reg1=A(parms) Reg4=A(BTTE) Reg7=A(DDIR) Reg15=V(code)	EEQE00	Address of the parameter work area for CBT Services. Address of the Buffer Tracking Table. Address of the DDIR name being searched for. The return code from CBT Services.

**Possible cause:** The causes include:

- The XRF active system generated an incorrect log record.
- The XRF alternate system incorrectly processed a log record.

**For DFSHCI00:**

A GET or RELEASE control block request (DFSBCB FUNC=GET/REL) for an asynchronous work element (AWE) from the AWE pool failed.

Register 14 points to label HCIGAWA for a GET AWE, or to label HCIRAWA for a RELEASE AWE. Register 15 contains the error code returned by DFSBCB00.

**Possible cause:** Insufficient storage was available in CSA to satisfy the request.

**System action:** The control region is terminated abnormally.

**Programmer response:** For the return codes and their meanings, see Analysis above.

**Problem determination:** 1, 2, 3, 4, 6, 15, 35

**Source:** IMS abends

**Module:** DFSHREQ0, DFSHCI00

**Related information:**

Chapter 2, "IMS problem determination," on page 13

**0590**

**Explanation:** VTAM or the IRLM failed in the XRF complex where the IMS alternate system was running. Either terminated abnormally on the central processor complex (CPC).

**Analysis:** This is issued when either an internal VTAM error drives the TPEND exit routine, or IRLM terminates abnormally on the CPC where the alternate system is running.

Key	Label	Description
	HTRMTERM	The XRF alternate system terminates with abend 0590. No dump is taken.

**System action:** The control region terminates abnormally.

**Programmer response:** Correct any errors that caused VTAM or IRLM to fail; then restart the alternate IMS system.

**Problem determination:** 1, 2, 3, 4, 6, 35

## 0591 • 0592

**Source:** IMS abends

**Module:** DFSHTRM0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

### 0591

**Explanation:** An error occurred while attempting to enqueue or dequeue a Fast Path resource. The Fast Path Forward Recovery Processor (DBFERS20) was either unable to obtain or unable to release a CI lock during Forward Recovery processing of the CI. This problem is caused by an internal IMS error.

**Analysis:** 0591 is a standard abend issued by the Fast Path Forward Recovery module, DBFERS20. Look at register 15 to determine whether a lock request or a lock release failed. Look at register 4 to determine the IRLM return code and reason code.

See the IRLM request return and reason codes for an explanation of the IRLM codes.

Key	Label	Description
Reg2=DMAC address Reg4=X'bbbbcccc' Reg7=XCRB address Reg15=X'0001aaaa'	ERS2LOCK	X'0001' indicates that a lock request failed. X'aaaa'= return code from DBFMGXC0 X'bbbb'= IRLM reason code X'cccc'= IRLM return code The DMAC contains information about the DEDB Area. The XCRB contains information about the CI involved in the lock request.
Reg2=DMAC address Reg4=X'bbbbcccc' Reg7=XCRB address Reg15=X'0002dddd'	ERS2RLSE	X'0002' indicates that a lock release failed. X'dddd'= return code from DBFLRH00 X'bbbb'= IRLM reason code X'cccc'= IRLM return code The DMAC contains information about the DEDB Area. The XCRB contains information about the CI involved in the lock request.

**System action:** The control region terminates abnormally.

**Problem determination:** 1, 2, 3, 4, 6, 35

**Source:** IMS abends

**Module:** DBFERS20

**Related tasks:**

 IRLM messages and codes

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

### 0592

**Explanation:** IMS was unable to get a data entry database (DEDB) area lock. A Fast Path XRF module was unable to obtain an Area lock while preparing to refresh the DMAC from the second CI. This is an internal IMS error.

**Analysis:** DBFMGXC0 was unable to get a CI lock, and received a nonzero return code from the Fast Path Lock Request Handler (DBFLRH00).

The IRLM passes back a return code and a reason code that explain why the lock request was rejected. You can find these codes in the following way:

Locate the RESTART PST, pointed to by SCDRSTEB. Location PSTLRXRC contains the return code from the IRLM, and location PSTLRXFB contains the reason code from the IRLM.

**System action:** The control region terminates abnormally.


**Problem determination:** 1, 2, 3, 4, 6, 35

**Source:** IMS abends



**Module:** DBFEACL0, DBFHCI0, DBFENIS0, DBFE2CI0, DBFERS10

**Related tasks:**

 IRLM return and reason codes

**Related information:**

Chapter 2, "IMS problem determination," on page 13

0593

**Explanation:** A database block failure occurred during restart.

**Analysis: For DFSRBLB0:**

A database block failure occurred during IMS restart processing.

This is a standard abend code issued by DFSRBLB0. It occurs during IMS restart End-of-File processing. Register 14 contains the address from which the abend was detected.

Register 15 contains one of the following abend subcodes:

**Codes**

**Explanation**

**X'01'**

DFSBCB GET failure

**X'03'**

DSFCWU error

**X'04'**

SCHEDULE PSB FAILURE

**X'05'**

DDIR not found error

**X'06'**

Transaction code not found and no type 07 log record exists

**X'07'**

IMODULE GETSTOR for global storage for a work area to build the type 0408 log record failed.

**For DFSRDBP0:**

A database block failure occurred during restart processing.

This is a standard abend issued by DFSRDBP0. It only occurs in the processing of X'07', X'08', X'47', or X'5607' log records during restart, XRF tracking, or Fast Database Recovery (FDBR) tracking. Register 12 is the base register.

Register 15 contains one of the following return codes:

**Codes**

**Explanation**

**X'01'**

DFSBCB GET block error

**X'02'**

DFSCBTS ENQ/DEQ error

**X'03'**

DSFCWU error

**X'04'**

Block mover error

**X'05'**

PDIR not found

X'06'

Insufficient pool space for PSB Schedule

Key	Label	Description
Reg2=A(X'47' log record) Reg15=X'01'	P470100 (CLNP020)	DFSBCBGET was unable to obtain an RRE or a SIDX.
Reg2=A(X'47' log record) Reg15=X'02'	P470130 (CLNP042)	DFSCBTS ENQ or DEQ call to DFSSIDX0 failed.
Reg2=A(log record) Reg14=A(CWU return) Reg15=X'03'	GETDPST (CLNP020)	Create work unit failed in X'07'/X'08'/X'47' log record processing.
Reg2=A(PDIR) Reg14=A(return SCHED PSB) Reg15=X'04'	SCHDPSB	This return code applies only to an XRF alternate system. An error was returned from the block mover. The PSTSCHDF field at X'385' in the PST contains one of these values: <b>X'01'</b> PSB is stopped or locked. <b>X'02'</b> Database stopped or locked. <b>X'03'</b> I/O error in reading PSB/DMB.DMB not found, or prior DMB error (DDIRBAD set). <b>X'04'</b> Intent conflict. <b>X'05'</b> PSBW/DMB pool too small to hold PSB/DMB. <b>X'06'</b> EPCB/DMB/PSB/PSBW pool storage is temporarily not available or DFSDBAU could not get needed space. Refer to PSTCODE1 for more information. <b>X'07'</b> Invalid PCB PROCOPT L or LS. <b>X'08'</b> FP buffer page fix error. <b>X'09'</b> PCB processing intent not compatible with database access. <b>X'0A'</b> Database authorization failure. <b>X'0B'</b> Database not available. <b>X'0C'</b> Waiting for I/O prevention (BMP with GSAM after takeover).
Reg2=A(log record) Reg11=find PDIR RC Reg14=A(return find PDIR) Reg15=X'05'	P080020 (P470050)	The PDIR corresponding to the PSBNAME in the log record was not found.
Reg2=A(log record) Reg14=(return SCHED PSB) Reg15=X'06'	SCHDPSB	This return code applies only to an XRF alternate system. A pool space error has occurred. Byte X'B02' in the PST contains one of these values: <b>X'40'</b> Intent conflict <b>X'20'</b> PST on DMB wait queue <b>X'08'</b> Type 3 batch region <b>X'04'</b> PST on PSB wait queue <b>X'x2'</b> If X'02', then PSB pool shortage. If X'12', then PSB work pool shortage. DSF992I or DFS993I, issued before the abend, specifies the pool space shortage type. <b>X'00'</b> Possible error return from Intent List load or DMB load and Relocate. Check PSTSCHDF at byte X'345' in the PST to determine the cause. For example, if PSTSCHDF contains X'05', the DMB pool is too small.

**For DFSRESP0:**

A database or restart block failure occurred during restart.

This is a standard abend issued by DFSRESP0. It only occurs in the processing of X'4027' and X'4028' type log records during restart, due to a DFSBCB/DFSCBTS problem.

#### Codes

##### Explanation

#### X'08'

The block was not found during requested SIDX function.

Key	Description
Reg2=A(X'4027' or X'4028') log record Reg15=X'08'	A DFSBCB/DFSCBTS error occurred for a SIDX FUNC=FOUND or GET, BLK=EQEL, DDIR, EQEL, or RRE request.

#### For DFSRST00:

A database block failure occurred during restart processing.

This is a standard abend issued by DFSRST00. It occurs only in the processing of log records during restart or XRF tracking. Register 14 contains the address where the abend is detected. Register 15 contains one of the following abend subcodes:

#### Codes

##### Explanation

#### X'01'

DFSBCBGET was unable to obtain an EQEL, RPST, or SIDX in subroutine GETRPST.

#### X'02'

DFSCBTS ENQ failed in subroutine GETRPST or DEQ failed in subroutine RELRPST.

**System action:** IMS terminates abnormally.

**Programmer response:** A code in register 15 at the time of abend identifies the cause of the failure.

Take appropriate action according to the error code indicated above. CSA fragmentation might cause codes 3 and 4. In this case, re-IPL and rerun the job.

**Source:** IMS abends

**Module:** DFSRBLB0, DFSRDBP0, DFSRESP0, DFSRST00

#### 0594

**Explanation:** An internal system error occurred. A request to Storage Management Services for an RPST control block failed. The requester can issue an abend.

**Analysis:** This is a standard abend issued by an XRF/non-XRF system during Error Restart, or by an XRF alternate system during the tracking phase. A service request to the RPST Block Management module failed, and the requester issued 0594. This is an internal system error. Get the following information for IBM problem analysis:

1. The place where the RPST Service request failed. (Use the PSW at entry-to-abend to determine the abending module and the instruction from which the abend was issued. This will isolate the place where the RPST Service request failed.)
2. A copy of the log record being processed. (See register 2.) Log record types are X'07/08/37/47/50/51/52'.
3. A copy of the caller's parameter work area pointed to by register 1. This area is 40 (X'28') bytes long.
4. The abend subcode in register 15.

Key	Label	Description
Reg1=A(parms)		Address of the caller's parameter work area for RPST Block Management service.
		Address of the caller's log record

Key	Label	Description
Reg14=A(return)		Address of the return to the caller and the return address from the DFSCBTS call, or where the call branched to the abend routine.
Reg15=V(subcode) or 0 if the error was not the result of a nonzero return from a DFSCBTS call		The abend subcode returned to the caller.

**Possible cause:** The causes include:

- The XRF active system generated an incorrect log record.
- The XRF alternate system incorrectly processed a log record.

**System action:** The control region terminates abnormally.

**Programmer response:** Check the return codes and their meanings.

**Problem determination:** 1, 2, 3, 4, 6, 15, 35

**Source:** IMS abends

**Related information:**

Chapter 2, "IMS problem determination," on page 13

#### 0595

**Explanation:** A GETMAIN request for main storage from subpool 231 failed.

**Analysis:** This is a standard abend issued by the alternate system during XRF synchronization.

Key	Label	Description
Reg1=V(abend code) Reg15=V(return code)		The user abend code The return code from the GETMAIN request

**Possible cause:** The REGION parameter on the JOB statement for the IMS control region JCL needs to be increased.

**System action:** The control region on the alternate system terminates abnormally.

**Programmer response:** Check the return codes and their meanings.

**Problem determination:** 1, 2, 3, 4, 6, 35

**Source:** IMS abends

**Module:** DFSHDAI0, DFSHPTK0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

#### 0596

**Explanation:** An internal system error occurred. This abend is issued by all IMS buffer tracking modules in an XRF complex or in a Fast Database Recovery region.

**Analysis:** This is a standard abend issued by an XRF alternate system during the tracking or takeover phase. Ensure that the IMS log is available for problem analysis. Record types are X'07/08/27/37/50/51/52/53'.

Register 1 contains the user abend code 0596 (X'254'). Register 15 contains a subcode.

Key	Label	Description
Reg15=X'04'	RD53	Unable to locate a Buffer Tracking Pool
Reg15=X'08'	CHECKAG	Unable to locate a buffer in the Buffer Tracking Primary Pool

Key	Label	Description
Reg15=X'0C'	MATCHR	Unable to locate a buffer in the Buffer Tracking Overflow Pool
Reg15=X'10'	BLDWKA	Unable to allocate storage for the Buffer Tracking Pool

**Possible cause:** The causes include:

- The XRF active system generated an incorrect log record.
- The XRF alternate system incorrectly processed a log record.

**System action:** The control region on the alternate system terminates abnormally.

**Programmer response:** Check the return codes and their meanings.

**Problem determination:** 1, 2, 3, 4, 6, 15, 35

**Source:** IMS abends

**Module:** DFSHPTK0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

## 0597

**Explanation:** An internal system error occurred. This abend is issued by all IMS lock tracking modules in an XRF complex.

**Analysis:** This is a standard abend issued by an XRF alternate system during the tracking or takeover phase. Ensure that the IMS log is available for problem analysis. Record types are X'07/08/27/37/50/51/52/53'.

Register 1 contains user abend code 0597 (X'255') from either the Lock Tracking or Lock Reacquire failure. Register 15 contains the subcode.

**For DFSHLTK0:**

Key	Label	Description
Reg15=X'01'	HLTK0200	Lock Tracking failure —There was no lock tracking data within the log record.
Reg15=X'03'	TRLK1500	Lock Tracking failure —Attempting to delete a lock tracking entry which was not acquired.
Reg15=X'04'	TRLK1700	Lock Tracking failure —Attempting to delete a lock tracking entry which was not acquired.
Reg15=X'05'	DELLWKAE	Lock Tracking failure —Unable to locate a pool header for this lock tracking or hash table entry.
Reg15=X'06'	GETP0500	Lock Tracking failure —Unable to IMODULE GETMAIN a lock tracking pool. Reg5 contains the return code.
Reg15=X'07'	DELL0200	Lock Tracking failure —Unable to IMODULE DELETE a lock tracking pool. Reg5 contains the return code.

**For DFSHRAL0:**

Key	Label	Description
Reg15=X'05'	HRAL0420	Lock Reacquire failure —Unable to find a DDIR from the lock track entry.
Reg15=X'06'	HRAL0502	Lock Reacquire failure —Unable to reacquire a RIDX lock from the IRLM.
Reg15=X'07'	HRAL0521	Lock Reacquire failure —Unable to reacquire a SEGX lock from the IRLM.

## 0598

Key	Label	Description
Reg15=X'08'	HRAL0511	Lock Reacquire failure —Unable to reacquire a SEGL lock from the IRLM.
Reg15=X'09'	HRAL0541	Lock Reacquire failure —Unable to reacquire a BIDP lock from the IRLM.
Reg15=X'0A'	HRAL0551	Lock Reacquire failure —Unable to reacquire a BIDX lock from the IRLM.
Reg15=X'0B'	HRAL0531	Lock Reacquire failure —Unable to reacquire a BIDL lock from the IRLM.
Reg15=X'0C'	HRAL0531	Lock Reacquire failure —Unable to reacquire a XIDP lock from the IRLM.

**Possible cause:** The causes include:

- The XRF active system generated an incorrect log record.
- The XRF alternate system incorrectly processed a log record.

**System action:** The control region on the alternate system terminates abnormally.

**Programmer response:** Check the return codes and their meanings.

**Problem determination:** 1, 2, 3, 4, 6, 15, 35

**Source:** IMS abends

**Module:** DFSHLTK0, DFSHRAL0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

## 0598

**Explanation:** An inconsistency in the system data sets was found during the synchronization phase of an extended recovery facility (XRF) complex or during emergency restart of an alternate system. Register 3 contains a character string indicating which system data set caused this problem. The possible data sets in error are: the restart data sets (RDS), the MODSTATs, the QBLKs, the LGMSG-Qs, the SHMSG-Qs, and the MSDBs. For more information, refer to message DFS3889A.

**Analysis: For DFSRST00:**

The DSNAMES and VOLIDs of the system data sets in the active and alternate systems are inconsistent for XRF synchronization or for restart.

0598 is a standard abend issued by DFSRST00 because of an inconsistent DDNAME during XRF synchronization or restart of a non-XRF system. The DSNAMES and VOLIDs in the active system are passed to the alternate system in the '4001' checkpoint log record, and are compared with the DSNAMES and VOLIDs in the alternate system. If the DSNAMES or the VOLIDs of the system data sets in the active and alternate systems are inconsistent, this abend is issued.

Message DFS3889A, including a return code (RC=), is sent to the master terminal before the alternate system terminates. Register 2 contains the address of the log record, register 7 contains the address of the fast restart block (FRB). Register 15 contains the following return codes:

### Codes

#### Explanation

#### RC=04

During a restart of the active system, the DSNAMES or VOLIDs used were different. They must be the same unless a COLDSTART or a BLDQ option is specified.

#### RC=08

During an emergency restart, the number of data sets for the short and long message data sets must be consistent. If a change is required, either a COLD start or the BLDQ option is required.

**RC=0C**

During an emergency restart in an XRF environment, the QUEUE Manager Data Sets (qblks-shmsg-lgmsg) for the backup system are the same. They must be different than the System Data Sets used by the active system.

**RC=10**

During an emergency restart in an XRF environment, the system data sets for the backup system online change or restart data sets were different. They must be the same as the active system.

Key	Label	Description
Reg3=C'MODS' Reg8=address of ONLINE CHANGE work area MSWA	MODBAD CHECKSDS	The DSNAMES and VOLIDS MODSTAT or MODSTAT2 data sets in the active and alternate systems are not the same.
Reg3=C'RDS' REG4=A(DCBEXT) Reg5=A(DCB) REG8=A(UCB)	RDSBAD CHECKSDS	The DSNAMES and VOLIDS of the RDS or RDS2 data sets in the active and alternate systems are not the same.
REG3=C'QBLK' REG4=A(DCBEXT) REG5=A(DCB)	QBLKBAD CHKSYS10	The DSNAMES and VOLIDS of the QBLKs data sets on the active and alternate systems are the same during XRF synchronization or are not the same during a restart.
REG3=C'SMSG' REG4=A(DCBEXT) REG5=A(DCB)	SMSGBAD CHKSYS20	The DSNAMES and VOLIDS of the SHMSG data sets on the active and alternate systems are the same during XRF synchronization or are not the same during a restart.
REG3=C'LMSG' REG4=A(DCBEXT) REG5=A(DCB)	LMSGBAD CHKSYS30	The DSNAMES and VOLIDS of the LGMSG data sets on the active and alternate systems are the same during XRF synchronization or are not the same during a restart.

**For DBFNRST0:**

An MSDB data set name or VOLID is inconsistent for restart or XRF synchronization.

During processing of 4080 log record (begin Fast Path checkpoint) by restart or XRF synchronization, a logged MSDB data set name or VOLID was not found in the restarting/backup system. Message DFS3889A is issued before the abend. Register 8 points to the 4080 log record. Register 11 points to the ESCD. Field ESCDMDSN is a pointer to a block containing the DSNAMES and VOLIDS of the MSDB checkpoint data sets; this block is mapped in macro DBFMDSN.

Key	Label	Description
Reg3=C'MSDB'	NRST4080	One of these problems occurred: <ol style="list-style-type: none"> <li>1. During normal restart, emergency restart, or XRF backup synchronization, a logged MSDB DBD name was not found in the system.</li> <li>2. During XRF synchronization, a discrepancy was found between an MSDB VOLID on the active and alternate systems.</li> </ol>

**Possible cause:** The causes include:

- IMS was started with an incorrect procedure.
- The data set was migrated to a different volume.
- An internal program logic error occurred.

**System action:** The control region terminates abnormally.

**Programmer response:** Correct the control region JCL. See Allocating XRF data sets (System Definition) for data set name and volume requirements for an XRF-capable IMS system.

**Problem determination:** 1, 2, 3, 4, 6, 35

**Source:** IMS abends

## 0599 • 0600

**Module:** DFSRST00, DBFNRSST0

**Related information:**

 DFS3889A (Messages and Codes)

Chapter 2, “IMS problem determination,” on page 13

---

### 0599

**Explanation:** A page-fix or page-free error occurred during checkpoint of an XRF-capable IMS active system or in the synchronization phase of an XRF complex. The IMSAUTH PGFIX/PGFREE function passed a return code in register 15. For a description of these return codes, see the IMS system services return codes information.

**Analysis:** 0599 is a standard abend issued by DFSRCP00 or DFSRLP00. The IMSAUTH macro called DFSV4200 for a page fix or a page free service. A nonzero return code was passed back to the caller in register 15.

Key	Label	Description
Reg14=next sequential instruction of BAL R14,ABND599 following the IMSAUTH call Reg15=nonzero return code	ABND599	The page fix/free request failed.

**Possible cause:** An internal program logic or interface error occurred.


**System action:** The control region terminates abnormally.

**Problem determination:** 1, 2, 3, 4, 6, 35

**Source:** IMS abends

**Module:** DFSRCP00, DFSRLP00

**Related reference:**

 IMS system services return codes (Messages and Codes)

**Related information:**

Chapter 2, “IMS problem determination,” on page 13

---

### 0600

**Explanation:** An error occurred in a DIAGNOSE-0 instruction while running an XRF-capable IMS system under VM.

**Analysis:** This is a standard abend issued by DFSRCP00 and DFSRLP00 when either an XRF-capable IMS system is running under VM and a checkpoint is taken, or the X'4001' log record is processed by the alternate system. When running under VM, IMS issues a DIAGNOSE O instruction to obtain the VM USERID. A comparison is made to determine if the active and alternate IMS system are running under the same USERID.

Key	Label	Description
Reg14=address of the next sequential instruction of a branch and link (BAL) instruction	ABEND600 ABND600	DFSRCP00 DFSRLP00 The DIAGNOSE O work area, pointed to by register 2 (DFSRCP00) or register 8 (DFSRLP00), does not begin with the character string 'VM', or the USERID was not obtained. The DIAGNOSE O instruction, DC X'83100000', is executed with interrupts disabled.

**Possible cause:** An internal program logic error occurred.

**System action:** The control region terminates abnormally.

**Programmer response:** If the attempt to rerun your XRF-capable IMS system fails, contact IBM Software Support for assistance.

**Problem determination:** 1, 2, 3, 4, 6, 35



**Source:** IMS abends

**Module:** DFSRCP00, DFSRLP00

**Related information:**

Chapter 2, "IMS problem determination," on page 13



---

## Chapter 15. IMS abend codes 0601 - 0700

This topic identifies and explains the following abend codes issued by IMS. Each code is 4 decimal digits.

---

### 0601

**Explanation:** A failure occurred during a request for getting or freeing storage in the address space private area or common service area (CSA) using IMODULE services.

**Analysis:** This abend is a standard abend issued by DBFDBFM0, DBFERS20, DBFTOPU0, DFSIDC00, DFSTODI0, and DFSTOPR0 when temporary storage could not be acquired or freed using IMODULE GETMAIN/ DELETE. The program status word (PSW) at entry-to-abend points to the instruction from which the abend (SVC 13) is issued.

Register 15 contains a return code. For an explanation of IMODULE return codes, refer to the IMS system services return codes information.

**System action:** IMS terminates abnormally.


**Programmer response:** Increasing the region size or the size of CSA might be necessary. Register 15 contains the IMODULE return code.

**Problem determination:** 1, 2, 3, 4, 6, 35

**Source:** IMS abends

**Module:** DBFDBFM0, DBFERS20, DBFTOPU0, DFSIDC00, DFSTODI0, DFSTOPR0

**Related reference:**

 [IMS system services return codes \(Messages and Codes\)](#)

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

### 0602

**Explanation:** A logic error occurred in the I/O toleration function of IMS. This abend is preceded by message DFS0612I, which contains the register 14 and register 15 return codes that explain the unexpected condition.

**Analysis:** For DBFTOFNO, DBFTOPUO:

This is a standard abend that can be issued by DBFTOPUO. When this abend is issued, the program status word (PSW) at entry-to-abend points to the instruction that the abend (SVC 13) is issued from.

Register 12 in the abend SVRB registers is the base register for this module. Register 15 contains a return code.

Key	Label	Description
Reg15=X'61'	ABNOSRB	Get FP SRB failed
Reg15=X'62'	ABNOSRBR	Free FP SRB failed
Reg15=X'63'	ABNPOST	No PST was passed
Reg15=X'64'	ABNEPSTR	Free EPST failed
Reg15=X'65'	ABNOLCHG	Lock change failed
Reg15=X'66'	ABNLKREL	Lock release failed (RO = IRLM rc)
Reg15=X'67'	ABNOEPST	Get EPST failed
Reg15=X'68'	ABNOBUFF	Buffer address is zero
Reg15=X'69'	ABNODATA	Buffer data is invalid

For DFSTOBHO, DFSTORSO:

## 0602

This is a standard abend that can be issued by DFSTORSO. When this abend is issued, the program status word (PSW) at entry- to-abend points to the instruction from which the abend (svc 13) is issued. Message DFS0612I precedes the abend and is written to the master console.

Register 12 in the abend SVRB registers is the base register for this module. Register 15 contains a return code.

Key	Label	Description
Reg15=X'22'	ABND602	Nonzero RC from the buffer handler routine
Reg15=X'31'	DOMSG	EEQE not found
Reg15=X'32'	DOMSG	ddir not found

### For DFSTODI0:

This is a standard abend that can be issued by DFSTODI0. When this abend is issued, the program status word (PSW) at entry-to-abend points to the instruction from which the abend (svc 13) is issued. Message DFS0612I precedes the abend and is written to the master console.

Register 12 in the abend SVRB registers is the base register for this module. Register 15 contains a return code.

Key	Label	Description
Reg15=X'41'	MSG612	Nonzero RC from DBRC creating an EEQE
Reg15=X'42'	MSG612	Nonzero RC from DBRC deleting an EEQE
Reg15=X'43'	MSG612	Nonzero RC from DBRC

### For DFSTOPRO:

This is a standard abend that can be issued by DFSTOPRO. When this abend is issued, the program status word (PSW) at entry- to-abend points to the instruction from which the abend (svc 13) is issued. Message DFS0612I precedes the abend and is written to the master console.

Register 12 in the abend SVRB registers is the base register for this module. Register 15 contains a return code.

Key	Label	Description
Reg15=X'10'	ABND602	DDIR/DMAC not found
Reg15=X'11'	ABND602	EEQE not found for deletion
Reg15=X'12'	ABND602	Buffer length was not specified for an EEQE create
Reg15=X'13'	ABND602	DBRC change EEQE error
Reg15=X'14'	ABND602	DBCR change EEQE error during Batch backout
Reg15=X'15'	ABND602	Invalid EEQE found
Reg15=X'1F'	ABNOEPST	EEQE COPY found an invalid EEQE

**System action:** The control region terminates abnormally.

**Programmer response:** See message DFS0612I for an explanation of the return codes.

**Problem determination:** 1, 2, 3, 4, 6, 35

**Source:** IMS abends

**Module:** DBFTOFNO, DBFTOPUO, DFSTOBHO, DFSTODIO, DFSTOPRO, DFSTORSO

**Related information:**

 DFS0612I

Chapter 2, "IMS problem determination," on page 13

## 0603

**Explanation:** An error occurred when the alternate IMS system in the XRF complex loaded main storage databases (MSDBs) from the system log data set. A return code in register 15 indicates the reason for this error.

**Analysis:** This is a standard abend issued by DBFHLOD0 or DFSRLP00 while loading the MSDBs from the IMS system log. When a CHECKPOINT DUMPQ is taken, the MSDB data is written as X'407x' log records to the IMS system log. During XRF synchronization, DFSRLP00 reads these records and passes them to DBFHLOD0. At the time of the abend, register 14 points to the next sequential instruction after a BAL R14,ABEND instruction, and register 15 contains a reason code. Register 2 points to the log record being processed.

Key	Label	Description
Reg8+X'40'= address of FRBCHKNO Reg10=A(ESCD) Reg15=X'01'	ABEND	The checkpoint ID passed in log record X'4070' does not match the checkpoint ID in log records X'4071', X'4071', X'4072', X'4073', X'4074', or X'4079'. FRBCHKNO contains the checkpoint ID.
Reg1=A(log record entry) Reg6=ECNT entry length Reg8+X'28'=FRBECNT =CURRENT ECNT address Reg10=A(ESCD) Reg15=X'02'	HLOD4071	The extended communications node table (ECNT) name, passed in the X'4071' log record, was not found in the system.
Reg15=X'03' ESCDDDBFA=Reg10+X'224' =X'00000000'	HLOD4073	A X'4073' log record was passed to DBFHLOD0, but no page-fixed MSDBs were defined.
Reg7=accumulated length. Reg10+X'228'=ESCDDDBFL Reg15=X'04'	HLOD4079	The total length of page-fixed MSDB data does not match the length in ESCDDDBFL.
Reg15=X'05' ESCDDDBPA=Reg10+X'22C' =X'00000000'	HLOD4074	A X'4074' log record was passed to DBFHLOD0, but no pageable MSDBs were defined.
Reg7=accumulated length Reg10+X'230'=ESCDDDBPL Reg15=X'06'	HLOD4079	The total length of pageable MSDB data does not match the length in ESCDDDBPL.
Reg1=A(log record entry) Reg 5=A(header entry) Reg 15=X'07' Reg15=X'08'	HLOD4072	The MSDB name, HEADER length, or segment length passed in the X'4072' log record does not match the name, length, or segment length of the corresponding MSDB HEADER entry. Register 15 is set to X'08' if the header spans X'4072' log records.
Reg2=A(X'4070' log record) Reg10+X'2C'=ESCDCNNO =number of ECNTs Reg10+X'32'=ESCDCnln =length of ECNT Reg15=X'09'	HLOD4070	The total number and length of checkpointed ECNTs does not match the number and length of the ECNTs in this system.
Reg14+X'42'= FRBMSDFL=X'80' Reg14+X'42' ^= X'C0' Reg15=X'0A'	L4099S10	MSDB checkpoint records (X'407x') were found during the synchronization phase of the XRF complex, but no X'4079' record was processed. DFSRLP00 makes this check at the end of the X'4099' process.

**Possible cause:** The causes include:

- An internal program logic error occurred.
- The active and the alternate system are not the same system.

## 0604 • 0606

**System action:** The control region terminates abnormally.

**Programmer response:** Correct any errors; then rerun the alternate system.

**Problem determination:** 1, 2, 3, 4, 6, 35

**Source:** IMS abends

**Module:** DBFHLOD0, DFSRLP00

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

### 0604

**Explanation:** The /SWITCH SYSTEM FORCE (ACTIVE) or /SWITCH SYSTEM ACTIVE command was entered from either the active or the alternate system in an XRF complex to request a takeover by the alternate system. Abend code 0604 is expected for the active system in response to this command. The alternate system takes over.

**System action:** The active system's control region terminates.

**Programmer response:** None required.

**Source:** IMS abends

**Module:** DFSICA20, DFSTRM00

---

### 0605

**Explanation:** An unexpected condition occurred while attempting to release the scheduling serialization latch or the ACBLIB read serialization latch. The abend occurred because the latch is not owned, or the routine trying to release the latch is not the owner of the latch. This is an IMS system error. Abend 0605 is issued by module DFSBML00 or DFSDBLR0. The program status word (PSW) at entry-to-abend will isolate the failure to the specific module.

**Analysis: For DFSBML00:**

An unexpected condition has occurred in the attempted release of the scheduling serialization latch. The abend occurred because the latch is not owned, or the releaser is not the owner of the latch. This is an IMS system error.

**For DFSBML00, DFSDBLR0:**

This is a standard abend issued by either DFSBML00 or DFSDBLR0. The program status word (PSW) at entry-to-abend should be used to isolate the failure to a particular module.

**System action:** IMS terminates abnormally.

**Problem determination:** 4

**Source:** IMS abends

**Module:** DFSBML00, DFSDBLR0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

### 0606

**Explanation:** An unexpected condition occurred in the IMS IRLM suspend exit. This might be an IMS system error, or an unauthorized application program might be attempting to use the IMS cross-memory option.

**Analysis:** This is a standard abend issued by module DFSSUSX0. The program status word (PSW) at entry-to-abend points to the instruction in the routine at label ABND0606 from which the abend (SVC 13) is issued. This routine is branched to from various locations in DFSSUSX0 when an error is detected. Register 4 is the BAL register to the abend routine, and contains the address of the location from which control was passed.

**System action:** IMS terminates abnormally.

**Problem determination:** 4, 35

**Source:** IMS abends

**Module:** DFSSUSX0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

**0608**

**Explanation:** While the /TRACE SET ON TRAP 2 command was enabled, IMS detected an overwrite of the MFS blocks, or the MOD and DOF failed block verification.

*For DFSCFE80, DFSCFE90, DFSCFEI0:* While the /TRACE SET ON TRAP 2 command was enabled, IMS detected an overwrite of the MFS blocks.

*For DFSCFE00:* While the /TRACE SET ON TRAP 2 command was enabled, IMS verified some of the MOD and DOF parameters. A MOD or DOF failed verification.

**Analysis:** The registers contain the following information:

**Register 11**

Contains the address of the CIB. The CIB indicates the formats that need to be recompiled.

**Register 14**

Contains the address within the error detecting module. This address can be used to determine which module subroutine detected the error.

**Register 15**

Contains the storage address that exceeded the buffer boundary, or an error code from block verification.

Key	Label	Description
CIB+X'00'='		Name of MID or MOD
CIB+X'0C'='		Name of DIF or DOF

**R15 error codes:**

**Codes**

**Explanation**

- 1 DOF work buffer size (DOFSWKSZ) is negative.
- 2 DOF first buffer offset (DOFSBOWF) is negative.
- 3 MOD field data length (MODFLNG) is negative.
- 4 MOD literal offset (MODFLIT) is zero or negative.
- 5 MOD syslit vector (MODFVECT) is greater than X'0020'.
- 6 MOD syslit vector (MODFVECT) is not divisible by 4.
- 7 MOD linkage entry offset (MODDOFL) is negative.
- 8 DOF FDE offset linkage (DOFSSIZE) is negative.

**DOF build type 0 (3270):**

**Codes**

**Explanation**

- 9 DOF line buffer data size (DOFSLBDS) is zero or negative.
- A DOF end of FDE series (DOF1EOF) before FDE type 7 found
- B DOF series link (DOFFLINK) is zero or negative.
- C DOF index value (DOFFLAG1, bits 4-7) is greater than 8.
- D DOF data/literal length (DOFFLNG) is negative.
- E DOF literal offset (DOFFLIT) is zero or negative.

**DOF build type 1 (non-3270):**

## 0611

### Codes

#### Explanation

- F DOF line buffer data size (DOFSLBDS) is zero or negative.
- 10 DOF index value (DOFFLAG1, bits 4-7) is greater than 8.
- 11 DOF line buffer offset (DOFFBOFF) is negative.
- 12 DOF data/literal length (DOFFLNG1) is negative.
- 13 DOF literal offset (DOFFLIT1) is zero or negative.
- 14 DOF physical page link (DOFFLINK) is zero or negative.

**System action:** Recompile the Format or Test Format libraries.

**Problem determination:** If an overwrite occurred, register 14 contains the location in the module where the overwrite was detected. If the MOD or DOF failed block verification, register 15 contains a debugging code.

**Source:** IMS abends

**Module:** DFSCFE80, DFSCFE90, DFSCFEI0, DFSCFEO0

---

## 0611

**Explanation:** During processing of a device output format (DOF) field descriptor element (FDE), Message Format Service (MFS) detected that an invalid request was made. An invalid DOFFTYPE was found in DOFFLAG1 by one of the MFS output build modules (DFSCFE80 or DFSCFE90). This might be because the DOF:

- Was not built properly.
- Was overlaid in storage.
- Was modified improperly.

**Analysis:** This is a standard abend issued from the MFS output build module (DFSCFE80 for 3270s, or DFSCFE90 for non-3270 devices).

The program status word (PSW) at entry-to-abend and the registers in the abend SVRB can be used to isolate a label. Register 11 contains the address of the message output descriptor (MOD) name in use when the error occurred (the CIB). The abend from both modules is the result of an unconditional branch from label PROCFDE to label ABEND611.

Key	Label	Description
Reg2=action value Reg8=DOFFDE dsect address Reg15=DFSCFE80 base DOFFLAG1=DOFFDETYPE	DOSERIES (PROCFDE)	The proper physical page has been verified, so the DOF FDE series can be processed. The abend is issued because the action requested, in register 2, is invalid for the device.
Reg2=action value Reg3=CURSTBL DOFFLAG1=X'80' Reg8=DOFFDE dsect address Reg15=DFSCFE80 base	STEPFDE (PROCFDE)	This routine is processing an FDE for either a 6-byte or a 4-byte entry. The abend is issued because the action requested, in register 2, is invalid for the device.
Reg1=DOFFLAG1 Reg8=DOFFSZ1 Reg12=DFSCFE90 base	FDEDONE (PROCFDE)	Processing is finished for the current FDE, and a check is made for a DOF literal. If there is none, the next FDE is to be processed. This abend is issued because bit 5 is on for build type 1 in DOFFLAG1, and that action is invalid for the device.
Reg1=DOFFLAG1 Reg8=DOFFLAG Reg12=DFSCFE90 base	FDEDONE (PROCFDE1)	Processing is finished for the current FDE, and a check is made for a DOF literal. If there is one, go process it. The abend is issued because bit 5 is on for build type 1 in DOFFLAG 1, and that action is invalid for the device.



Key	Label	Description
Reg1=DOFFLAG1 Reg8=FDE address Reg12=DFSCFE90 base	PPLINK (PROCFDE1)	The physical page link FDE has been found, and it is to be processed. The abend is issued because bit 5 is on for build type 1 in DOFFLAG1 and that bit is invalid for the device.

**Possible cause:** The DOFFLAG1 might have been modified.

**System action:** IMS terminates abnormally.

**Programmer response:** None.

**Problem determination:** 11, 16, 35

Register 11 contains the address of the message output descriptor (MOD) name in use when the error occurred.

**Source:** IMS abends

**Module:** DFSCFE80, DFSCFE90

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

## 0616

**Explanation:** A permanent I/O error occurred on the last remaining IMS log data set. If dual logging is in effect for IMS batch, this will occur when both of the log data sets are damaged. For IMS online, this will occur when the last remaining OLDS has a permanent I/O error or does not have enough space to complete a checkpoint freeze.

**Analysis:** This is a standard abend that can be issued by the physical logger - buffer post process routine, DFSFDLBO. When this abend is issued, the program status word (PSW) at entry-to-abend will point to the instruction from which the abend (SVC 13) is issued.

Register 12 in the abend SVRB registers is the base register for this module. Register 11 and register 10 contain the address of the system contents directory, SCD and the log control directory, LCD, respectively. Register 14 contains the address from which the failure was detected.

Key	Label	Description
Reg8=A(LDSET) Reg9=A(LBUFFER)	WRTE1500	The SLDS data set, IEFORDER (IEFRDER2 if dual), has an I/O error. The buffer prefix, LBUFFER, contains the BSAM DECB and LDSET contains the status flags.

### For DFSFDLS0:

For batch, this abend indicates there was an error closing the log. For online, either the maximum block count exceeded 4 294 967 295, or no error-free OLDS (or OLDS pair) exists and the current OLDS (or OLDS pair) is full.

This abend can be issued by the physical logger setup routine, DFSFDLS0. When this abend is issued, the program status word (PSW) at entry-to-abend will point to the instruction from which the abend (SVC 13) is issued.

Register 12 in the abend SVRB registers in the base register for this module. Register 11 and register 10 contain the addresses of the system contents directory (SCD) and the log control directory (LCD), respectively. Register 14 contains the address from which the failure was detected.

Key	Label	Description
Reg8=A(LDSET)	SETU0700	When the next usable OLDS was requested, the request was ignored because there remained no error-free OLDS and the current OLDS was full.
Reg8=A(LDSET)	OPEN1400	The log block count exceeded its maximum value (4 294 967 295).
Reg8=A(LDSET)	EOVSTART EOVL1300	After filling up the last OLDS, the EOV routine detected that there was no error-free OLDS remaining.
Reg8=A(LDSET)	EOVL1550	The log block count exceeded its maximum value (4 294 967 295).

Key	Label	Description
Reg8=A(LDSET)	CLOSEBAD	A CLOSE error was encountered on the batch log data set.

**System action:** IMS terminates abnormally.

**Programmer response:** Allocate new log data sets to replace the ones which have encountered I/O errors. It might be necessary to use the Log Recovery Utility to create log data sets free from write or close errors. When sufficient log data sets are available, emergency restart IMS to resume processing.

**Source:** IMS abends

**Module:** DFSFDLBO

## 0622

**Explanation:** IMS attempted to initialize a non-control region with a protect key in the range 0-7. The CTL and CTX online control regions in z/OS are the only regions that can run under keys 0-7. For details, refer to the procedures discussion in Procedures used in IMS environments (System Definition).

**Analysis:** This is a standard abend issued by DFSRRA40 (online) or DFSRRA70 (batch). Message DFS0622I is written to the IMS master console before the abend.

A DL/I, dependent, or utility region was started with the program DFSRRC00 (region controller) specified to run under keys 0 to 7. The noncontrol region so specified terminates abnormally with a memory dump.

Key	Label	Description
	RAMP COM RADSVC1	The TCB protect key field (TCBPKF) is tested for a key in the range 0 to 7. A noncontrol region must run in a protect key range of 8 to 15.

**Possible cause:** Program name DFSRRC00 incorrectly resides in the z/OS Program Properties Table. This table (module IEFSD060) is located in the system LPALIB.

**System action:** IMS terminates the region that issued this abend.

**Programmer response:** Correct the JCL and resubmit the job. Check with the z/OS system programmer to make sure that program DFSRRC00 is not specified to run under keys 0-7.

**Problem determination:** 1, 2, 3, 4, 8, 35

**Source:** IMS abends

**Module:** DFSRRA40, DFSRRA70

**Related information:**

 DFS0622I

Chapter 2, "IMS problem determination," on page 13

## 0623

**Explanation:** IMS attempted to initialize an online control region in z/OS with a protect key in the range of 8-15; a key in the range of 0-7 is required.

**Analysis:** This is a standard abend issued by DFSRRA00. Message DFS0623I precedes the abend and is issued to the master console.

An online control region must execute in the protect key range of 0 to 7. The control region, whether CTL or CTX, will terminate with a memory dump.

Key	Label	Description
	RACTX/ RACTL	Within these two labels the TCB protect key field (TCBPKF) is tested to ensure the key is in the range 0 to 7. If the test fails a branch to label RACTXLAB is taken, the message is issued, and abend results.

**Possible cause:** The causes include:

- The program DFSMVRC0 is not included in the z/OS Program Properties Table, or the protect key of the program is incorrect. The table is member name IEFSD060, and is located in the system's LPALIB.
- Some data sets in JOBLIB are not authorized.

**System action:** IMS terminates the region that issued this abend.

**Programmer response:** Correct the JCL and resubmit the job. Check with the z/OS system programmer to make sure that DFSMVRC0 is specified to run under keys 0-7, and that all JOBLIB data sets are APF-authorized.

**Problem determination:** 1, 2, 3, 4, 8, 35

**Source:** IMS abends

**Module:** DFSRRA00

**Related information:**

 DFS0623I

Chapter 2, "IMS problem determination," on page 13

## 0624

**Explanation:** IMS received a nonzero return code from a DFSBCB request for AWE storage. The low-order 16 bits in register 15 contain the DFSBCB return code; the high-order 16 bits in register 15 might contain qualifying information.

**Analysis:** 0624 is a standard abend issued by various IMS modules when a nonzero return code is returned from DFSBCB after issuing a DFSBCB FUNC=GET call for AWE storage. Register 15 contains the return code from DFSBCB. For an explanation of the DFSBCB return codes, see the IMS system services return codes information.


**System action:** IMS terminates abnormally with a memory dump.

**Programmer response:** Take appropriate action based on the codes in register 15.

**Problem determination:** 4, 11, 35

**Source:** IMS abends

**Related reference:**

 IMS system services return codes (Messages and Codes)

**Related information:**

Chapter 2, "IMS problem determination," on page 13

## 0630

**Explanation:** The initialization module DFSXCB00 issues this abend code for any error that is encountered during processing, when one of the following situations occurs during IMS initialization:

1. An IMODULE load failed in one of the following modules: DFSSPF00, DFSCBT10, DFSBC000, or DFSBCB60.
2. Initialization for one of the pools failed in QSAV, BXQE, or AWE.
3. The DFSCBTS FIND function failed to locate an IMS control block defined in the IMS control block table (DFSCBT10).
4. A storage request could not be fulfilled by the IMS storage manager (DFSSTM00).
5. An IMODULE GETMAIN request failed.

**Analysis:** This abend is a standard abend issued by DFSXCB00.

Register 14 and register 15 contain the required diagnostic information. Register 14 contains the address in the module where the abend condition was detected. Register 15 contains indicators and return codes that pinpoint the exact cause of the abend. This register contains a 4-byte return code consisting of a 1 to 3 character EBCDIC failure indicator followed by a hexadecimal return code. The possible contents of register 15 are first summarized and then described in more detail as follows:

'F' 00 00 *nn*

DFSCBTS FUNC=ALTER or FUNC=FIND for CBTE failed. The return code is *nn*. DFSCBT10 does not include the requested control block definitions.

## 0630

'G' 00 00nn

IMODULE GETMAIN failed for DFSCBTHD. The IMODULE return code is *nn*.

'T' nnnnnnn

IMODULE GETMAIN failed for DFSXCB01. The IMODULE return code is *nnnnnnn*.

'LBC' nn

IMODULE LOAD failed for DFSBC000. The IMODULE return code is *nn*.

'LCB' nn

IMODULE LOAD failed for DFSCBT10. The IMODULE return code is *nn*.

'LC5' nn

IMODULE LOAD failed for DFSCBT50. The IMODULE return code is *nn*.

'LSC'nn

IMODULE LOAD failed for DFSSTC00. The IMODULE return code is *nn*.

'LSP' nn

IMODULE LOAD failed for DFSSPF00. The IMODULE return code is *nn*.

'L60' nn

IMODULE LOAD failed for DFSBCB60. The IMODULE return code is *nn*.

'Q' nnnnnnn

DFSBCB GET failed for QSAV. The DFSBCB return code is *nnnnnnn*.

'SM'nnnn

DFSSTM00 was called to obtain an IPAGE. IMODULE GETMAIN failed attempting to get the requested storage. Register 8 contains the CBTE address. The IMODULE return code is *nnnnn*.

'S' 00 00 nn

During control block table entry initialization the CBTE was not found for one or more of these blocks: QSAV, BXQE, AWE. The value *nn* is the number of blocks that do not have a CBTE defined in DFSCBT10.

1. IMS IMODULE load failed in one of the following modules: DFSSPF00, DFSCBT10, DFSBC000, or DFSBCB60. Register 15 contains the following:

Bytes	Description
-------	-------------

1	L - IMODULE LOAD failed
---	-------------------------

2 and 3	CB - load failed for DFSCBT10 at label HDRSTART. BC - load failed for DFSBC000 at label BCB80. SP - load for DFSSPF00 failed at label BCB80 60 - load failed for DFSBCB60 at label HDREXIT.
---------	---

4	IMODULE return code
---	---------------------

2. QSAV, BXQE, AWE pool failure. A probable bind error. DFSCBT10 does not include one of the above control block definitions. DFSCBT10 assembly had an error but the output was still bound into IMS. (Label BCB07)

Register 15 contains the following:

Bytes	Description
-------	-------------

1	S - Scan failed for one of the above blocks. The table (DFSCBT10) must be examined by the system programmer to determine the block ID.
---	--

2	00
---	----

3	00
---	----

4	00
---	----

3. DFSCBTS FIND failure. The internal control block FIND function was unable to locate a required control block. A probable bind error. DFSCBT10 does not include one of the above control block definitions. DFSCBT10 assembly had an error but the output was still bound into IMS. (Label BCB80)

Register 15 contains the following:

Bytes	Description
-------	-------------

1	F - DFSCBTS FIND failure.
---	---------------------------

2	00
---	----

3	00
---	----

4	08 - DFSSCBT0 return code. The block could not be located.
---	--

Register 7 contains the address of the control block type that could not be found in the control block table (DFSCBT10).

4. A storage request failure. A request for storage (DFSSTM00) was not fulfilled. Increase the region size on the JCL. (Label GETBLK30)

Register 15 contains the following:

**Bytes Description**

**1 and 2** SM - DFSSTM00 storage request failure.

**3 and 4** RC - 2-byte return code set by DFSSTM00 denoting the reason for the failure. Most likely it indicates the region size that must be increased by at least 8K bytes. You must refer to the prolog for module DFSSTM00 for additional information concerning the return codes.

5. IMODULE GETMAIN request failed. An internal request for storage failed. (Label HDRSTART)

Register 15 contains the following:

**Bytes Description**

**1** G - IMODULE GETMAIN request failed.

**2** 00

**3** 00

**4** RC - IMODULE return code.

6. IMODULE load request failed for DFSCBT50.

Register 15 contains the following:

**Bytes Description**

**1** L - IMODULE load failed.

**2 and 3** C5 - Load failed for DFSCBT50.

**4** IMODULE return code.

7. IMODULE GETMAIN request failed for DFSXCB01

Register 15 contains the following:

**Bytes Description**

**1** I - IMODULE GETMAIN failed for DFSXCB01

**2** 00

**3** 00

**4** IMODULE return code.

8. DFSBCB GET failed for QSAV block. This can only occur in a batch region.

Register 15 contains the following:

**Bytes Description**

**1** Q - DFSBCB GET failed for QSAV block.

**2** 00

**3 and 4** 2-byte return code for DFSBCB00. Most likely it indicates that the region size must be increased by at least 8K bytes.

**System action:** IMS terminates abnormally.

**Operator response:** Notify the system programmer.

**Programmer response:** Determine the cause of failure as indicated by register 15, correct the problem, and resubmit the job. For the meanings of the IMODULE and DFSBCB return codes, see the IMS system services return codes information.

If the abend occurred because of insufficient storage, increase private or CSA storage allocations as follows:


- To increase the size of private storage, increase the region size in the JCL.
- To increase the size of CSA, re-IPL z/OS, specifying a larger CSA allocation.

**Problem determination:** 1, 4, 5, 6, 7, 11, 35

**Source:** IMS abends

**Module:** DFSXCB00

**Related reference:**

 IMS system services return codes (Messages and Codes)

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

**0631**

**Explanation:** A call was made to the IMS CREATE ITASK Processor using a DFSCIR macro to initialize an ECB. DFSCIR returned with a nonzero return code in register 15, indicating that the request could not be processed.

**Analysis:** This is a standard abend issued by DFSIINB0 or DFSIINV0. The DFSCIR macro returned with a return code in register 15; the code has the following meaning:

**Codes****Explanation****X'08'**

Unable to allocate a SAP.

**X'0C'**

Save area offset was not specified.

**X'10'**

Unable to allocate a QMGR work area.

**X'14'**

Routine address was not specified.

**X'18'**

Unable to locate DSPWRK.

**X'1C'**

An invalid function was specified.

In addition, the following registers are useful in diagnosing this problem:

**Register****Meaning****Register 7**

Contains the address of the routine to receive control when the ECB is posted.

**Register 9**

Contains the address of the ECB to be initialized.

**Register 10**

Contains the address of the routine that called the DFSCIR macro.

**System action:** IMS terminates initialization processing with this abend.

**Problem determination:** 1, 4, 6, 10, 11, 35

**Source:** IMS abends

**Module:** DFSIINB0, DFSIINV0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

**0632**

**Explanation:** The EXEC statement PARM field contained too many positional parameters.

**Analysis:** This is a standard abend issued by DFSRRA20. Message DFS632I is sent to the IMS master console prior to the abend.

**Possible cause:** Verify that the PARM field has the correct number of positional parameters.

**System action:** The region that issued this abend is terminated.

**Programmer response:** Correct the JCL and resubmit the job.

**Problem determination:** 1, 2, 3, 4

**Source:** IMS abends

**Module:** DFSRRA20

**Related information:**

 DFS632I (Messages and Codes)

Chapter 2, “IMS problem determination,” on page 13

### 0633

**Explanation:** The startup procedure for an IMS DL/I DBB region specified DBRC=C, but the MBR parameter did not specify DFSBBO00. DBRC=C is valid only for IMS Batch Backout.

**Analysis:** This is a standard abend issued by DFSRRA00. Message DFS0633I is sent to the z/OS console before to this abend.

Key	Label	Description
	RAIR	DBRC=C and RCPGM do not equal Batch Backout (DFSBBO00).

**System action:** IMS terminates abnormally.

**Programmer response:** Correct either the DBRC parameter or the MBR parameter.

**Problem determination:** 1, 2, 3, 4, 8

**Source:** IMS abends

**Module:** DFSRRA70

**Related information:**

 DFS0633I

Chapter 2, “IMS problem determination,” on page 13

### 0634

**Explanation:** You have one or more leading blanks in a positional parameter in the PARM field of the EXEC statement.

**Analysis:** This is a standard abend issued by DFSRRA20. Message DFS634I is written to the IMS master console prior to the abend.

**Possible cause:** Verify that the positional parameters in the PARM field do not have leading blanks.

**System action:** The region that issued this abend is terminated.

**Programmer response:** Correct the JCL and resubmit the job.

**Problem determination:** 1, 2, 3, 4, 35

**Source:** IMS abends

**Module:** DFSRRA20

**Related information:**

 DFS634I (Messages and Codes)

Chapter 2, “IMS problem determination,” on page 13

### 0636

**Explanation:** The last fixed-length parameter value in the PARM field of the EXEC statement was too short.

**Analysis:** This is a standard abend issued by DFSRRA30 (control region), DFSRRA40 (dependent region), or DFSRRA70 (batch region), the EXEC statement PARM analysis modules. Message DFS636I is written to the IMS master console prior to the abend.

This abend is detected in module DFSRRA20. DFSRRA30, DFSRRA40, or DFSRRA70 BALs to DFSRRA20 to perform the character string analysis of the PARM field. DFSRRA20, using the parameter list passed in register 1, scans the PARM field for errors. If an error is detected, a message code is passed back to the caller in register 15, and an abend is issued. The return code is developed by a branch table within module DFSRRA20.

## 0638 • 0640

Register 14, at time of abend, contains an address that indicates which module and routine called DFSRRA20.

Key	Label	Description
	PMSCN3	The last fixed-length parameter is too short.

**Possible cause:** Verify that each field of the PARM is the correct length.

**System action:** The region that issued this abend is terminated.

**Programmer response:** Correct the JCL and resubmit the job.

**For APAR processing:** Retain the region dump and input JCL.

**Source:** IMS abends

**Module:** DFSRRA30, DFSRRA40, DFSRRA70

**Related information:**

 DFS636I (Messages and Codes)

---

### 0638

**Explanation:** A comma was found within the extent of a fixed-length parameter.

**Analysis:** 0638 is a standard abend issued by DFSRRA30 (control region), DFSRRA40 (dependent region), or DFSRRA70 (batch region), the EXEC statement PARM analysis modules. Message DFS638I is written to the IMS master console prior to the abend.

This abend is detected in module DFSRRA20. DFSRRA30, DFSRRA40, or DFSRRA70 BALs to DFSRRA20 to perform the character string analysis of the PARM field. DFSRRA20, using the parameter list passed in register 1, scans the PARM field for errors. If an error is detected, a message code is passed back to the caller in register 15 and an abend is issued. The return code is developed by a branch table within module DFSRRA20.

Register 14, at time of abend, contains an address that indicates which module and routine called DFSRRA20.

Key	Label	Description
	PMSCN5	An embedded comma was detected within a fixed-length parameter.

**Possible cause:** JCL error on the EXEC statement, PARM field. Refer to the IMS system definition information for the correct format.

**For APAR processing:** Region dump, and input JCL.

**System action:** IMS terminates abnormally.

**Programmer response:** Correct the JCL, and resubmit the job.

**Problem determination:** 1, 2, 3, 4

**Source:** IMS abends

**Module:** DFSRRA30, DFSRRA40, DFSRRA70

**Related information:**

 DFS638I (Messages and Codes)

Chapter 2, "IMS problem determination," on page 13

---

### 0640

**Explanation:** You omitted a required parameter for the kind of execution specified by the first 3 characters of the EXEC statement PARM field.

**Analysis:** This is a standard abend issued by DFSRRA30 (control region), DFSRRA40 (dependent region), or DFSRRA70 (batch region), the EXEC statement PARM analysis modules. Message DFS640I is written to the IMS master console prior to the abend.



This abend is detected in module DFSRRA20, DFSRRA30, DFSRRA40, or DFSRRA70 BALs to DFSRRA20 to perform the character string analysis of the PARM field. DFSRRA20, using the parameter list passed in register 1, scans the PARM field for errors. If an error is detected, a message code is passed back to the caller in register 15, and an abend is issued. The return code is developed by a branch table within module DFSRRA20.

Register 14, at time of abend, contains an address that indicates which module and routine called DFSRRA20.

Key	Label	Description
	PMSCN8/ PMSCN13	The input parameter list indicates that required fields exist for this PARM field, but the scan operation determines that the required positional parameter is missing.

**Possible cause:** Omission of those required positional parameters on the EXEC statement PARM field. Refer to the IMS sstem definition information for the required parameters for the different regions specified as the first three characters.

**For APAR processing:** Region dump and input JCL.

**System action:** The region that issued this abend is terminated.

**Programmer response:** Correct the JCL, and resubmit the job.

**Problem determination:** 1, 2, 3, 4

**Source:** IMS abends

**Module:** DFSRRA30, DFSRRA40, DFSRRA70

**Related information:**

 DFS640I (Messages and Codes)

Chapter 2, "IMS problem determination," on page 13

## 0641

**Explanation:** The Database Recovery utility DFSURDB0 was executed in an invalid region type. The batch region type specified in the first subparameter of the //EXEC PARM field must be UDR.

**Analysis:** This is a standard abend issued by DFSRRA80 (batch region) when it detects that the region parameter in the first subparameter of the //EXEC PARM field was specified as ULU when the database utility specified in the second subparameter was DFSURDB0. The Database Recovery utility DFSURDB0 must be run with the region type UDR specified.

**Possible cause:** The region parameter in the PARM field of the EXEC statement is incorrect.

**System action:** IMS terminates abnormally.

**Programmer response:** Correct the JCL and resubmit the job.

**Source:** IMS abends

**Module:** DFSRRA80

## 0642

**Explanation:** A parameter value, such as the program name in a DL/I execution, exceeds the maximum allowable length.

**Analysis:** This is a standard abend issued by DFSRRA30 (control region), DFSRRA40 (dependent region), or DFSRRA70 (batch region), the EXEC statement PARM analysis modules. Message DFS642I is written to the IMS master console prior to the abend.

This abend is detected in module DFSRRA20, DFSRRA30, DFSRRA40, or DFSRRA70 BALs to DFSRRA20 to perform the character string analysis of the PARM field. DFSRRA20, using the parameter list passed in register 1, scans the PARM field for errors. If an error is detected, a message code is passed back to the caller in register 15, and an abend is issued. The return code is developed by a branch table within module DFSRRA20.

## 0643 • 0644

Register 14, at time of abend, contains an address that indicates which module and routine called DFSRRA20.

Key	Label	Description
	PMSCN9	The length of a parameter exceeds the maximum allowable.

**Possible cause:** JCL error, such as the program name in a DL/I execution greater than eight characters.

**For APAR processing:** Region dump and input JCL.

**System action:** The region that issued this abend terminates.

**Programmer response:** Correct the JCL and resubmit the job, or reissue the START command specifying parameters with the correct length.

**Problem determination:** 1, 2, 3, 4

**Source:** IMS abends

**Module:** DFSRRA30, DFSRRA40, DFSRRA70

**Related information:**

 DFS642I

Chapter 2, "IMS problem determination," on page 13

---

### 0643

**Explanation:** A non-numeric value was specified for a numeric parameter.

**Analysis:** This is a standard abend issued by DFSRRA30 (control region), DFSRRA40 (dependent region), DFSRRA70 (batch region), the EXEC statement PARM analysis modules. Message DFS643I is written to the IMS master console prior to the abend.

This abend is detected in module DFSRRA20. DFSRRA30, DFSRRA40, or DFSRRA70 BALs to DFSRRA20 to perform the character string analysis of the PARM field. DFSRRA20, using the parameter list passed in register 1, scans the PARM field for errors. If an error is detected, a message code is passed back to the caller in register 15, and an abend is issued. The return code is developed by a branch table within module DFSRRA20.

Register 14, at time of abend, contains an address that indicates which module and routine called DFSRRA20.

Key	Label	Description
	PMSCN52	A nonnumeric value was specified for a numeric value.

**For APAR processing:** Region memory dump and input JCL.

**System action:** The region that issued this abend terminates.

**System programmer response:** Correct the EXEC statement parameter in the JCL, and resubmit the job.

**Problem determination:** 1, 2, 3, 4, 8

**Source:** IMS abends

**Module:** DFSRRA30, DFSRRA40, DFSRRA70

**Related information:**

 DFS643I

Chapter 2, "IMS problem determination," on page 13

---

### 0644

**Explanation:** The internal destination list was invalid. This is an internal IMS error.

**Analysis:** 0644 is a standard abend issued by DFSRRA30 (control region), DFSRRA40 (dependent region), DFSRRA70 (batch region), the EXEC statement PARM analysis modules. Message DFS644I is written to the IMS master console prior to the abend.

The destination list is built by DFSRRA00 and is passed to DFSRRA20 as an input parameter by DFSRRA30, DFSRRA40, or DFSRRA70. The list is addressed by register 8 in DFSRRA20 and can be seen by zapping the instruction "BZ PRMC7" (in DFSRRA20) to force a program check.

This abend is detected in module DFSRRA20. DFSRRA30, DFSRRA40, or DFSRRA70 BALs to DFSRRA20 to perform the character string analysis of the PARM field. DFSRRA20, using the parameter list passed in register 1, scans the PARM field for errors. If an error is detected, a message code is passed back to the caller in register 15, and an abend is issued. The return code is developed by a branch table within module DFSRRA20.

Register 14, at time of abend, contains an address that indicates which module and routine called DFSRRA20.

Key	Label	Description
	PMSCN3	The length specification (DESLNT) within the destination list is zero. This is invalid.

**Possible cause:** Internal logic error.

**For APAR processing:** Abend dump with the program check trap installed in DFSRRA20.

**System action:** The region that issued this abend is terminated.

**Programmer response:** Check for user modifications to module DFSRRA00.

**Problem determination:** 1, 2, 3, 4, 35

Also, get an assembly listing of DFSRRA00 as used on site.

**Source:** IMS abends

**Module:** DFSRRA30, DFSRRA40, DFSRRA70

**Related information:**

 DFS644I

Chapter 2, "IMS problem determination," on page 13

## 0646

**Explanation:** The SPIE option in the PARM field of the EXEC statement was invalid.

**Analysis:** This is a standard abend issued by DFSRRA00. Message DFS646I is sent to the IMS master console prior to the abend. The IMS region that has the error is terminated with a dump.

The SPIE option is valid for the following regions:

- standalone batch
- message processing
- batch message processing

Refer to IMS system definition information for an explanation of the correct SPIE option settings, in the information about procedures.

Key	Label	Description
	RADLNMVM	This routine is entered for a standalone batch region. A BAL is taken to label RAOPTN where the SPIE option is checked; if the value is greater than X'01', a branch to label RCRC1 is taken, message DFS646I is issued and abend results.
	RAMSG/RACMP	These routines are entered to process a message processing region. The same logic as described above is executed.
	RABMP/RABMS	These routines are entered while processing parameters for a batch message region. The logic to validate the SPIE option is the same as in the batch only region.

**System action:** The region that issued this abend is terminated.

**Programmer response:** Correct the JCL, and resubmit the job.

**Problem determination:** 1, 2, 3, 4

## 0648 • 0652

**Source:** IMS abends

**Module:** DFSRRA00

**Related information:**

 DFS646I (Messages and Codes)

Chapter 2, "IMS problem determination," on page 13

---

### 0648

**Explanation:** The validity check option in the PARM field of the EXEC statement was invalid.

**Analysis:** This is a standard abend issued by DFSRRA00. Message DFS648I is issued to the IMS master console before the abend. The region from which the abend was issued terminates with a memory dump.

The validity check option specified whether (1) or (0) the addresses in the user call list should be validity checked. This option is a symbolic field (*&TEST*) for a DLI or BMP region, and a positional parameter for an MSG region.

Key	Label	Description
	RADLNMVM	This routine is entered for a stand-alone batch region. A BAL is taken to label RAOPTR where the validity check option is checked; a value greater than X'01' results in a branch to label RCRC2 where register 1 is loaded with the message code. A branch to RAMSGAB results in the proper message and eventual abend at label RAABND.
	RAMSG/RACMP	These routines are entered to process a message processing region. The logic as described above is exactly the same for this routine.
	RABMP/RABMS	These routines are entered while processing the parameters for a batch message region. The validity check logic is the same as that for the stand-alone batch region.

**Possible cause:** JCL error in the PARM field of the EXEC statement.

**System action:** The region that issued this abend is terminated.

**Programmer response:** Correct the JCL, and resubmit the job.

**Problem determination:** 1, 2, 3, 4

**Source:** IMS abends

**Module:** DFSRRA00

**Related information:**

 DFS648I

Chapter 2, "IMS problem determination," on page 13

---

### 0652

**Explanation:** The PARM field was omitted. All IMS EXECUTE statements must contain a PARM field.

**Analysis:** 0652 is a standard abend issued by the EXEC statement PARM analysis program, DFSRRA00. Message DFS652I is issued to the IMS master console before the abend of the affected region.

Upon initial entry to DFSRRA00, the PARM field of the EXEC statement is validated. All IMS EXEC statements *must* contain a PARM field. Refer to *IMS Version 13 System Utilities* or the IMS system definition information for examples of the PARM field format.

Key	Label	Description
	RASTART	During the initial scan of the EXEC statement, the PARM field is determined to be invalid. A branch is taken to RCRC3 to test for a zero length. If a zero length is indicated, a branch to RCRC31 is taken to issue the message and the abend.

*Possible cause:* JCL error

**System action:** The region that issued this abend is terminated.

**Programmer response:** Correct the JCL, and resubmit the job.

**Problem determination:** 1, 2, 3, 4

**Source:** IMS abends

**Module:** DFSRRA00

**Related information:**

 DFS652I (Messages and Codes)

Chapter 2, "IMS problem determination," on page 13

#### 0654

**Explanation:** The PARM field specified in the EXEC statement was fewer than three characters in length.

For control region types, the first two parameters of the EXEC statement are required. They must both be exactly 3 characters in length. If this is not true, then this abend occurs. For other region types, the first parameter in the PARM field of the EXEC statement is required. It must be exactly 3 characters in length. If this is not true, then this abend occurs.

See the (IMS system definition information for a description of parameters used in the PARM field of the EXEC statement for all region types.

**Analysis:** This is a standard abend issued by the EXEC statement PARM analysis program, DFSRRA00. Message DFS654I is issued to the IMS master console prior to the abend of the affected region.

The first three characters of the PARM field identifies the type of region that is to be started. This field must exist and must be exactly three characters in length.

Key	Label	Description
	RASTART	During the initial scan of EXEC statement, the PARM field is determined to be in error. A branch is taken to label RCRC3 to test for a zero length (omitted). This test fails and the PARM-too-short exit routine is taken.

*Possible cause:* JCL specification error.

**System action:** The region that issued this abend terminates.

**Programmer response:** Correct the JCL, and resubmit the job.

**Problem determination:** 1, 2, 3, 4

**Source:** IMS abends

**Module:** DFSRRA00

**Related information:**

 DFS654I (Messages and Codes)

Chapter 2, "IMS problem determination," on page 13

#### 0657

**Explanation:** A GSAM initialization error occurred. During region initialization, the GSAM routines encountered an error condition such as:

- Invalid PCB processing options
- Invalid DBD or PCB
- A system error
- An inconsistency between the PSB definition in the ACBLIB and PSBLIB
- A failure to obtain storage.

## 0657

- The PSTNR or LCRESEQN of the failing PST is binary zero. LCRESEQN can be located in LCRE via the pointer PSTCRE.

**Analysis:** 0657 is a standard abend that the GSAM initialization module issues as a result of its error handling routine. Message DFS0657I is issued before this abend. The program status word (PSW) points to the instruction within the module from which the abend is issued. Register 14 is the BAL return register to the error discoverer. Register 2 contains the error reason code. This error corresponds to the label name at the failing location.

This abend can be the result of adding GSAM PCBs to the PSB without doing both an ACBGEN and an Online Change. For an online application program to use a PSB that has GSAM PCBs defined, that PSB must be the same in both ACBLIB and PSBLIB. If GSAM PCBs are added to the PSB in PSBLIB without doing an ACBGEN and an Online Change, unpredictable results can occur when an application is scheduled with that PSB.

### Codes

#### Explanation

#### C4C900xx

DFSZDI00 detected the error.

#### C4C920xx

DFSZDI20 detected the error.

Key	Label	Description
R2=C4C90001	C4C90001	<p>Error opening IMS DD statement.</p> <p><b>Explanation:</b> Opening of an IMS DD statement for PSBLIB and DBDLIB failed.</p> <p><b>Operator Response:</b> Ensure that the step contains an IMS DD statement for PSBLIB and DBDLIB and that it names the proper libraries.</p>
R2=C4C90002	C4C90002	<p>Inconsistent PSBs.</p> <p><b>Explanation:</b> The PSB in ACBLIB indicates that the PSB contains a GSAM PCB, but the PSB in PSBLIB does not contain any GSAM PCBs.</p> <p><b>Operator Response:</b> Correct the appropriate PSB and rerun ACBGEN or PSBGEN.</p>
R2=C4C90003	C4C90003	<p>Additional PCB denied.</p> <p><b>Explanation:</b> An error was encountered in a BMP region. The GSAM PCBs were being added to the PCB list. This is an internal IMS error.</p> <p><b>Operator Response:</b> Rerun the job. If the situation persists, contact the IMS system programmer.</p>
R2=C4C90004	C4C90004	<p>Failure to obtain storage. The amount of storage requested can be found in register 6.</p> <p><b>Operator Response:</b> Increase the maximum virtual storage that is available for the region, and rerun the application.</p>
R2=C4C92001	C4C92001	<p>Invalid DBD (not a GSAM DBD).</p> <p><b>Explanation:</b> The GSAM PCB names a DBD that is not a GSAM DBD. Message DFS0657I contains the DBD name. Register 8 also points to the DBD name.</p> <p><b>Operator Response:</b> Correct the appropriate PSB or DBD and rerun PSBGEN, DBDGEN, or ACBGEN.</p>

Key	Label	Description
R2=C4C92002	C4C92002	<p>R15 contains a nonzero return code from the RDJFCB macro.</p> <p><b>Explanation:</b> This could indicate that the DD statement is not present in the step. However, a prior DEVTYPE macro indicated that the DD statement is present.</p> <p><b>Operator Response:</b> Provide the appropriate DD statement for the GSAM database, if required, and rerun the job.</p>
R2=C4C92003	C4C92003	<p>On return from the RDJFCB macro, the ARLAREA (address of the ARA area obtained by RDJFCB) is zero.</p> <p>In this case, ARLCODE (described below) is probably = 08.</p> <p><b>Operator Response:</b> See appropriate ARLCODE (shown below in C4C9004).</p>
R2=C4C92004	C4C92004	<p>On return from the RDJFCB macro, ARLCODE at REG5 + X'1C' is nonzero.</p> <p><b>Explanation:</b></p> <p><b>If ARLCODE = 4</b> The ARL was not initialized properly prior to issuing the RDJFCB macro.</p> <p><b>If ARLCODE = 8</b> There was insufficient virtual storage available to return the ARA area. (The ARA area is approximately 200 bytes per data set included in the DD statement.)</p> <p><b>Operator Response:</b> Increase maximum virtual storage available for the region and rerun the step.</p>
R2=C4C92005	C4C92005	<p>A GSAM or BSAM DD statement has greater than 255 data sets in the concatenation.</p>

**System action:** IMS terminates abnormally.

**Programmer response:** Correct the problem and rerun.

**Problem determination:** 11, 35

**Source:** IMS abends

**Module:** DFSZDI00, DFSZDI20

**Related information:**

 DFS0657I

Chapter 2, "IMS problem determination," on page 13

0658

**Explanation:** The program name parameter was omitted from the PARM field.

**Analysis:** This is a standard abend issued by module DFSRRA00.

**Possible cause:** JCL specification error

**System action:** The region that issued this abend is terminated.

**Programmer response:** Correct the JCL, and resubmit the job.

**Problem determination:** 1, 2, 3, 4

## 0662 • 0670

**Source:** IMS abends

**Module:** DFSRRA00

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

### 0662

**Explanation:** The first positional parameter in the PARM field was invalid.

**Analysis:** This is a standard abend issued by the EXEC statement PARM analysis program, DFSRRA00. Message DFS662I is sent to the IMS master console prior to the abend of the affected region.

The first positional parameter (positions 1 through 3 of the PARM field) must contain a valid 3-character field indicating the type of region that is to be started. The valid 3-character fields are: CMP, MSG, DLI, BMP, BMS, CTL, CTX, DBB, UPB, RST, or PRL. If 'UST' is selected or if the field does not match any of those stated, the abend will occur.

The CIC region is not valid in a DCCTL environment and causes this abend to be issued.

Key	Label	Description
	RATBS	The table at label RATBID is used as a vector to the valid routines. The first three characters from the PARM field are compared to the valid entries. If a match is not found, a branch is taken to label RAABE to initialize the message and abend code.

**Possible cause:** JCL specification error

**System action:** The region that issued this abend is terminated.

**Programmer response:** Correct the JCL, and resubmit the job.

**Problem determination:** 1, 2, 3, 4

**Source:** IMS abends

**Module:** DFSRRA00

**Related information:**

 DFS662I

Chapter 2, "IMS problem determination," on page 13

---

### 0670

**Explanation:** A failure occurred in the DFSYPSI0 or DFSYTIB0 OTMA module to process an OTMA message.

For the DFSYPSI0 module, the dynamic storage that is needed is greater than the pre-allocated DYNP storage saved in TPIPE, and IMS ran out of the LUMC storage to process OTMA messages.

For the DFSYTIB0 module, IMS ran out of both DYNP and LUMC storage for the request to process the OTMA message.

**Analysis:** DFSYPROC in DFSYPSI0 or DFSYTIB0 invoked the DFSYMAGL macro to get the LUMC storage for module entry. However, IMS ran out of the LUMC storage. Because LUMC storage is needed for most of the OTMA modules in IMS, an IMS abend occurred to prevent further problems in the system.

**System action:** IMS terminates abnormally.

**Programmer response:** Make sure that enough LUMC storage is available in the system, and then restart the IMS to process OTMA messages.

**Problem determination:** None.

**Source:** IMS abends

**Module:** DFSYPSI0

---



## 0684

**Explanation:** IMS type 2 SVC initialization failed. Some possible reasons for the failure are:

- An attempt was made to start a second copy of the IMS control region with the same IMSID as one already running. This can also occur if IMS SVC termination failed in a previous run of IMS with the same IMSID.
- SDFSRESL is not authorized or the IMS SVC module is being loaded from an unauthorized library.
- The region size is too small.

**Analysis:** This abend is issued by the subroutine SVCRTNE in module DFSRRA00 when the SVC initialization (DFSVCI INITSVC request) encounters an error. For SVC initialization failures, message DFS684I is routed to the system console before the termination of the control region.

This abend is issued by DFSXBAT0 when batch initialization (DFSVCI INITBAT request) encounters an error.

DFSVCI function requests are processed by the type 2 SVC module DFSVC200 and the SVC initialization/termination module DFSVCI00. When an error is detected during processing in these modules, message DFS686W is issued containing the function code and return code from the DFSVCI request.

Message DFS686W, which is issued along with this abend, provides additional diagnostic information.

**For DFSRRA00:**

Key	Label	Description
	SVCABEND	SVC initialization failed for control or batch region.

**For DFSXBAT0:**

Key	Label	Description
Reg14	CLBIC	Register 14 points to the subroutine that detected the error at entry-to-abend (macro DFSBLC).
Reg15		SVC return code at entry-to-abend.

**Possible cause:** For the reasons described above and for one of the following reasons:

- The IMSID specified on the IMSCTRL macro during system definition, on the EXEC statement of the JCL, or in the IMS execution parameters for member DFSPRRG0 or DFSPRRD0 is not unique. Another executing region has the same IMSID.
- The IMS execution library is not authorized.
- The command recognition character (CRC) specified during system definition, on the IMS execution parameters for member DFSPRRG0 or DFSPRRD0, or on the JCL EXEC statement is not unique. Another executing IMS region has the same CRC.
- DFSMRCL0 (IMS Resource Cleanup) might not be installed properly. If message DFS627I or DFS627W is not received at IMS termination, then DFSMRCL0 is not gaining control to clean up IMS resources.

**For APAR processing:** Abend dump

**System action:** The region that issued the ABEND is terminated.

**Programmer response:** Do one of the following:

- Correct the duplicated IMSID and rerun the job. If the IMS SVC termination failed in a previous run with the same IMSID, either re-IPL the operating system, or run with a different IMSID.
- Make sure SDFSRESL is authorized, or include the DFSRESLB DD statement that points to an authorized library containing the IMS SVC modules.
- Increase the region size.

For additional information, refer to message DFS686W.

**Problem determination:** 1, 2, 11

**Source:** IMS abends

**Module:** DFSRRA00, DFSXBAT0

**Related information:**

 [DFS684I \(Messages and Codes\)](#)

 [DFS686W \(Messages and Codes\)](#)

Chapter 2, “IMS problem determination,” on page 13

## 0688

**Explanation:** The operator entered CANCEL or C in response to message DFS690A.

**Analysis:** This is a standard abend issued by the EXEC statement PARM analysis program, DFSRRA00. Message DFS688I is sent to the IMS master console prior to the abend of the dependent region.

DFSRRA00 determines that the control region is not active while processing the parameter analysis for a dependent region. One of two conditions will result in this abend:

1. The cancel option (OPT=C) was specified on the dependent region EXEC statement.
2. The operator entered CANCEL or C in response to the error message DFS690A.

The operator has the option to specify WAIT to message DFS690A if the online control program is expected to be active shortly.

Key	Label	Description
	RAENQ1	The operator entered a response of 'C' or 'CANCEL' to message DFS690A. A branch is taken to label RAENQ6 to issue the message and abend.

**System action:** The region that issued this abend is terminated.

**Source:** IMS abends

**Module:** DFSRRA00

**Related information:**

 [DFS688I](#)

 [DFS690A](#)

## 0689

**Explanation:** The DBRC region or DL/I subordinate address space region abends with this code when the control region is not active during its initialization. The IMSID of the IMS issuing the START DBRC PROC command does not correspond with the IMSID in the DBRC procedure. Message DFS689I is issued before the abnormal termination. It is likely that the control region is in the process of terminating abnormally or the control region does not exist.

**Analysis:** 0689 is a standard abend issued by the EXEC statement PARM analysis program, DFSRRA00. Message DFS689I is issued to the IMS master console before the abend of the DBRC or DL/I subordinate address space region.

DFSRRA00 determines that the control region is not active while processing the parameter analysis for the DBRC or DL/I subordinate address space region. This situation is caused by one of the following conditions:

1. The DBRC or DL/I subordinate address space region was started issuing a START command or by a job, and the IMS control region is not active.
2. The IMS control region has abnormally terminated since issuing the START command for the DBRC or DL/I subordinate address space region.
3. The IMSID might be defined to a non-IMS SSID. Ensure the IMSID has not been defined for use by a non-IMS SSID.

Key	Label	Description
	RAENQWT1	A branch is taken to label RAMSGAB to issue the message and abend.

**System action:** The region that issued this abend is terminated.

**Programmer response:** Analyze the control region dump to determine the cause of the abnormal termination and resubmit the job.

**Problem determination:** 1, 2, 4, 6, 35

**Source:** IMS abends

**Module:** DFSRRA00

**Related information:**

 [DFS689I \(Messages and Codes\)](#)

Chapter 2, "IMS problem determination," on page 13



---

## Chapter 16. IMS abend codes 0701 - 0750

This topic identifies and explains the following abend codes issued by IMS. Each code is 4 decimal digits.

---

### 0701

**Explanation:** IMS attempted to enqueue using either an invalid queue control block or queue element.

**Analysis:** This is a standard abend issued by DFSIIEN0. The program status word (PSW) at entry-to-abend will point to the instruction within label ABEND from which the abend (SVC 13) is issued. The registers stored in the abend SVRB can prove useful in determining the enqueue label.

**Important:** The IMS format dump diagnostic area shows the calling module ID as the module that abended, instead of DFSIIEN0.

DFSIIEN0 enqueues queue elements (QEs) on IMS queue control blocks (QCBs). This module processes queue elements for the common system queues. The input registers are as follows: Register 1=QCB, register 2=QEs forward pointer (QE+4), register 12=base register, register 14=return address of the calling routine.

Key	Label	Description
Reg2=QE address + 4 QESQCB bit is on	SETQE01	This test determines if the address passed in register 2 is a QE. If the high-order bit is on, the input block is not a QE.
Reg1=QCB address QCBSQCB bit is off	SETQE02	This test determines if the address passed in register 1 is a QCB. If the high-order bit (QCBSQCB) is off, the input block is not a QCB.
Reg1=QCB address Reg2=address of QE + 4 Reg3=address of QE QCBLIFO bit = ON (QUEUE LIFO) Reg6=zero Reg10=positive address	ENQLIFO	This routine attempts to enqueue QE LIFO on QCB. Register 6 should contain the address of the first QE to be used.
Reg1=QCB address Reg2=address of QE + 4 Reg3=address of QE Reg6=zero Reg10=negative address	ENQPRTY	The enqueue of the QE is to be performed in a specified priority, but the dequeue pointer (register 6 first QE to use) is zero.

**Possible cause:** The calling module, pointed to by register 14, passed invalid parameters in register 1 and register 2.

**System action:** The IMS control region is abnormally terminated.

**Operator response:** Perform an emergency restart.

**Problem determination:** 1, 4, 5, 6, 35

**Source:** IMS abends

**Module:** DFSIIEN0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

## 0702

**Explanation:** IMS attempted to dequeue using either an invalid queue control block or queue element.

**Analysis:** 0702 is a standard abend issued by DFSIIDE0. The registers in the abend SVRB are those that were current when the abend occurred. Use register 14 to determine the caller of DFSIIDE0. Register 1 at entry points to either the forward chain pointer of a queue element (QE + 4) or a queue control block (QCB). The status bit is the high-order bit and identifies the control block as either a QE (bit off), or QCB (bit on).

The following cause for the abend results in a conditional branch to label ABEND.

Key	Label	Description
Reg1=address of QE + 4 Reg3=address of QE Reg4=0	DEQQE	Register 4 should contain the address of the queue element's back pointer. A logic error exists if the control block indicates that the pointer is present, but no address is found.

**Operator response:** Perform an emergency restart.

**Problem determination:** 1, 4, 5, 6, 35

**Source:** IMS abends

**Module:** DFSIIDE0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

## 0704

**Explanation:** An ICREATE call was issued to obtain a buffer. The buffer could not be allocated because a buffer already exists with the same name

**Analysis:** For abend 0704, find label ICSTART. Several instructions later there is a branch to ICDUBUF where the abend code is loaded into register 1, and a branch is taken to ABNDEXIT from where the abend is issued.

Key	Label	Description
Reg1=abend code	ICSTART	Subroutine FINDZIB found an existing ZIB with the same name. Register 2 contains the 4-byte name of the pool the user is trying to create. Register 6 contains the address of the Zone Initialization Block (ZIB) with the same name.

**Possible cause:** The caller called DFSISMN0 to create a pool for the second time.

**System action:** The IMS control region is abnormally terminated.

**Operator response:** Perform an emergency restart.

**Problem determination:** 1, 4, 5, 6, 35

**Source:** IMS abends

**Module:** DFSISMN0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

## 0705

**Explanation:** An IDESTROY call was issued to free a buffer, but a buffer with the name specified could not be found.

**Analysis:** For abend 0705, find label DFSIDEST. Several instructions later you will find a branch to IDNOBUF, which loads the abend code into register 1 and branches to ABNDEXIT where the abend is issued.

Key	Label	Description
Reg1=abend code	DFSIDEST	Subroutine FINDZIB could not find a zib with the name specified. Register 2 contains the 4-byte name of the pool the user is trying to return.

**Possible cause:** The caller called DFSISMN0 to destroy a pool that was never created.

**System action:** The IMS control region is abnormally terminated.

**Operator response:** Perform an emergency restart.

**Problem determination:** 1, 4, 5, 6, 35

**Source:** IMS abends

**Module:** DFSISMN0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

## 0707

**Explanation:** IMS attempted to get a buffer (IGETBUF) from a nonexistent pool. Register 2 at abend contains the erroneous pool name.

**Analysis:** For a 0707 abend, find label IGSTART. A few instructions later there is a branch to IGNOPOOL, which loads register 1 with the abend code and branches to ABNDEXIT where the abend is issued.

Key	Label	Description
Reg1=abend code	IGSTART	Subroutine FINDPOOL issued a DFSSPOOL FIND call to locate the specified pool in the HASH table. A nonzero return code was returned from the DFSSPOOL call. The specified pool could not be found. Register 2 contains the 4-byte name of the pool from which the user was trying to get a buffer.

**Possible cause:** You called DFSISMN0 to get a buffer from a pool that was never created.

**System action:** The IMS control region is abnormally terminated.

**Operator response:** Perform an emergency restart.

**Problem determination:** 1, 4, 5, 6, 35

**Source:** IMS abends

**Module:** DFSISMN0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

## 0708

**Explanation:** IMS attempted to free a buffer (IFREEBUF) to a nonexistent pool name. Register 2 at abend contains the erroneous pool name.

**Analysis:** For abend 0708, find label DFSIFBUF. After issuing several instructions, a branch is made to label IFNOPOOL. IFNOPOOL loads register 1 with the abend code and branches to ABNDEXIT where the abend is issued.

Key	Label	Description
Reg1=abend code	DFSIFBUF	Subroutine FINDPOOL issued a DFSSPOOL FIND call to locate the specified pool in the HASH table. A nonzero return code was returned from the DFSSPOOL call. The specified pool could not be found. Register 2 has the 4-byte name of the pool to which the user was trying to free a buffer.

**Possible cause:** The caller called DFSISMN0 to free a buffer from a nonexistent pool.

**System action:** The IMS control region is abnormally terminated.

**Operator response:** Perform an emergency restart.

**Problem determination:** 1, 4, 5, 6, 35

**Source:** IMS abends

**Module:** DFSISMN0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

## 0709

**Explanation:** IMS attempted to obtain a buffer (IGETBUF) from a storage pool. The length of the buffer requested was greater than the maximum buffer size allowed. R2 at abend contains the pool name. R3 contains the requested size.

**Possible cause:** The storage pool is not large enough to satisfy the request.

**System action:** Depending on the storage pool involved, ABENDU0709 might result in one of the following conditions:

- The dependent region is abnormally terminated.
- The IMS control region is abnormally terminated.

**Operator response:** Specify a larger pool size at the next IMS start.

If the pool involved is the EPCB pool, this abend is typically accompanied by message DFS0673E. Locate the error message on the master console for more information. Specify a larger EPCB pool size at the next IMS start, or lower the EPCB size requirement for the dependent region.

**Source:** IMS abends

**Module:** DBFDBPV0, DFSISMN0

**Related information:**

 DFS0673E (Messages and Codes)

Chapter 2, "IMS problem determination," on page 13

## 0710

**Explanation:** The save area chain pointed to by register 13 was followed to locate the first save area. This save area did not contain a pointer to the save area prefix (SAP). Either the save area chain has been destroyed, or the SAP pointer in the first save area has been overlaid.

**Analysis:** This is a standard abend issued from one of two locations within DFSISMN0. At label INOSAP within the main routine, DFSISMN0 loads the abend code into register 1 and branches to ABNDEXIT where the abend is issued.

Within the DFSIZBR0 subroutine abend 0710 is issued from the label GSAB710. At the time of abend, R13 should contain the address of the current save area. Use this address to trace back through the save areas to determine if an overlay of the save set or SAP address in the top save area has occurred. This can be helpful to determine the module flow, and possibly a caller that has overlaid the save sets.

**System action:** IMS terminates abnormally with a memory dump. If the abend occurs under the ESS TCB, the ESS TCB terminates with a memory dump and allows the control region to keep running.



**Programmer response:** This problem is an internal IMS error.

**Problem determination:** 4, 11, 35

**Source:** IMS abends

**Module:** DFSISMN0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

## 0711

**Explanation:** A system error was detected while processing in an z/OS Resource Recovery Services (RRS) environment or the z/OS UNIX System Services (z/OS UNIX) environment.

You might receive a DFS0613I message, which indicates that an abend 0113 has occurred due to a 0711. This situation can occur when RRS terminates and an IMS dependent region ends with an ABEND058 while in Fast Path sync point processing. The ABEND058 is converted to a 0711 RC2D for diagnostic purposes. To verify this situation, follow these steps:

1. Find the dependent region name listed in message DFS0613I.
2. Look for a previous message DFS629I indicating an ABEND058 for that dependent region
3. Use the abend 0113 dump to find the DPST for that dependent region. Flag SAPX1AFP is on and EPSTW1SY=1.
4. An IRC START entry in the IMS Scheduler trace indicates an abend 0058 completion code for that dependent region.

When 0711 is issued in an RRS environment, IMS might write log record X'67D0' subtype X'0607', where X'06' is D0SUBTYP and X'07' is D0ERRCOD. Subtype X'0607' means that the IMS DFSRRSI SWITCH\_CONTEXT service failed. The RC= value (4-byte field D06RC) contains the return code of the RRS service that failed. Fields D0SUBTYP, D0ERRCOD, and D06RC are defined in the DFS67D0 mapping macro.

When 0711 is issued in a z/OS UNIX environment, IMS might write log record X'67D0' subtype X'0608', where X'06' is D0SUBTYP and X'08' is D0ERRCOD. Subtype X'0608' means that a z/OS UNIX service such as BPX1TLS (z/OS UNIX task level security) failed. The RC= value (4-byte field D06RC) contains the return and reason codes of the z/OS UNIX service that failed. The higher 2 bytes contain the z/OS UNIX return code and the lower 2 bytes contain the z/OS UNIX reason code.

**Analysis:** Abend 0711 diagnosis can require gathering information about the RRS address space, its data spaces, the IMS control region, and the MVS Logger address space. The following SLIP trap can be used to obtain this information:

```
SLIP SET,C=U0711,JOBLIST=(ctljname,rrsjname,IXGLOGR),
SDATA=(CSA,PSA,RGN,SQA,SUM,TRT,GRSQ,LPA,ALLNUC),
ID=nnnn,DSPNAME=('rrsjname'*),END
```

Where:

*ctljname*

Control region job name

*rrsjname*

RRS job name

*nnnn*

Optional trap identifier

**For DFSASK00, DFSCPY00, DBFHGU10, DFSPCC20, DFSECP10, DFSDCPY0:**

0711 a pseudoabend when issued by one of these modules. IMS writes log record X'67D0', subtype X'07', to indicate which module wrote the record, which RRS service failed, and the return code from that service.

If a symptom dump is produced indicating that the abend was issued by DFSPCC20, then DFSASK00 sets up abend 0711. If the symptom dump indicates that DFSECP10 issued the abend, the abend was set up by DFSCPY00.

If RRS reason code 0701 is received, it might be because RRS became available, but IMS has not yet completed registration with RRS. Abend 0711 ceases when registration is complete, which is indicated by message DFS0653I.

**DFSDCPY0:**

## 0711

Reason code	Explanation
-------------	-------------

<b>X'35'</b>	The inter-region communication (IRC) request could not be completed because it was made in an ODBA environment and the unit of recovery is RRS local transaction.
--------------	---

**For DFSRRC10, DFSRRSB0, DFSRRSI0, DFSTMS00, DFSAERG0:**

0711 is a standard abend when issued by one of these modules. A severe error occurred while preparing for the processing of either protected conversations or synchronous APPC/OTMA shared queues transactions with RRS. Register 15 contains a reason code. All of the following reason codes result in the abnormal termination of IMS except X'12', X'13', X'14', X'15', X'16', X'1D', X'1E', X'1F', X'20', X'21', X'26' (only batch abnormally terminates), X'32' and X'33'.

**DFSRRC10:**

Reason code	Explanation
-------------	-------------

<b>X'2D'</b>	The dependent region was processing a call to ATRCMIT or ATRBACK when RRS terminated, resulting in an ABEND058. The ABEND058 is converted to a 0711 RC2D. Register 1 reflects the ABEND058 completion code.
--------------	---

**DFSRRSB0:**

Reason code	Explanation
-------------	-------------

<b>X'23'</b>	IMS failed to register as a resource manager with RRS. Register 2 contains the return code from the CRGGRM service.
--------------	---

<b>X'24'</b>	IMS was unable to switch the private context to another batch TCB. Register 2 contains the return code from the CTXSWCH service.
--------------	--

<b>X'26'</b>	IMS was unable to create a private context. Register 2 contains the return code from the CTXBEGC service.
--------------	---

<b>X'2B'</b>	IMS was unable to load all the RRS callable stubs.
--------------	--

**DFSRRSI0:**

Reason code	Explanation
-------------	-------------

<b>X'00'</b>	The caller requested an invalid DFSRRSI function.
--------------	---

<b>X'01'</b>	IMS failed to obtain storage for DFSRRSIB. Register 2 contains the return code from IMODULE GETMAIN.
--------------	--

<b>X'03'</b>	Retrieval of the IMS and RRS log names failed. Register 2 contains the return code from the ATRIRLN service.
--------------	--

<b>X'04'</b>	The length of the IMS log name returned by RRS is invalid. Register 2 contains the length of the log name.
--------------	--

<b>X'05'</b>	The IMS log name returned by RRS is invalid. Register 2 contains the address of the log name.
--------------	---

<b>X'06'</b>	The RRS Commit exit failed before IMS committed. If the UR is full function only, this abend is preceded by message ATR306I and message DFS0693I. The operator must issue the /CHANGE UOR COMMIT command to
--------------	---

commit the URID specified in message DFS0693I. If the UR is pure Fast Path or mixed mode, this abend is preceded by message DFS0698W with reason code 0000 in the message text. It also causes an abend 0113.

**X'07'**

The LXRES call failed to obtain a system LX value. Register 2 contains the return code from the LXRES service.

**X'08'**

IMS was unable to set the IMS log name with RRS. Register 2 contains the return code from the ATRISLN service.

**X'09'**

IMS was unable to begin the restart process with RRS. Register 2 contains the return code from the ATRIBRS service.

**X'0A'**

IMS was unable to retrieve the next incomplete interest. Register 2 contains the return code from the ATRIRNI service.

**X'0B'**

IMS was unable to respond to the retrieved interest. Register 2 contains the return code from the ATRIRRI service. Register 4 contains the IMS response code.

**X'0C'**

The ETCRE call failed to create the entry table for the IMS resource manager exit routines. Register 2 contains the return code from the ETCRE service. Register 3 contains the address of the entry table definition.

**X'0D'**

The ETCON call failed to connect the entry table to the linkage table of every address space. Register 2 contains the return code from the ETCON service.

**X'0E'**

IMS was unable to establish its resource manager exit routines with RRS context services. Register 2 contains the return code from the CRGSEIF service.

**X'0F'**

IMS was unable to establish its resource manager exit routines with RRS resource recovery services. Register 2 contains the return code from the CRGSEIF service.

**X'10'**

IMS was unable to post the deferred unit of recovery with RRS. Register 2 contains the return code from the ATRPDUE service.

**X'13'**

IMS issued the ATRBACK call, but the call failed. The associated program and transaction will not be placed in a stopped status. Register 2 contains the return code from the ATRBACK service.

**X'14'**

IMS issued the ATRCMIT call, but the call failed. Therefore, the application was abended to force backout processing. The associated program and transaction will not be placed in a stopped status. Register 2 contains the return code from the ATRCMIT service.

**X'15'**

LCURXITN contained an invalid RRS UR exit number. Register 2 contains the invalid exit number.

**X'16'**

Expression of protected interest in the next UR failed. Register 2 contains the return code from the ATRSROI service.

**X'17'**

IMS failed to enqueue the RRE on the SIDX. Register 2 contains the return code from the DFSCBTS service.

**X'18'**

IMS failed to obtain an RRE block. Register 2 contains the return code from the DFSBCB service.

**X'19'**

IMS failed to obtain a QSAV block. Register 2 contains the return code from the DFSBCB service.

**X'1A'**

IMS failed to obtain an AWE block. Register 2 contains the return code from the DFSBCB service.

## 0711

### X'1B'

The IMS notification exit failed to set the resource manager exit routines with either RRS context services. Register 2 contains the return code from the CRGSEIF service.

### X'1C'

RRS did not invoke the COMMIT exit or BACKOUT exit for a restart expression of interest. Register 2 contains the exit number.

### X'1F'

When attempting to switch the context (CTXSWCH) to the dependent region TCB, an existing context was disassociated. This problem is a system error.

### X'22'

An internal table containing active protected conversation tasks has been exhausted.

### X'27'

IMS was unable to express protected interest in the unit of recovery with RRS. Register 2 contains the return code from the ATREINT service.

**Note:** With APPC/OTMA SMQ enablement active, an IMS back-end system might receive a DFS554A message with REASON=03A30027 in the message text if the IMS front-end system is no longer available. This situation occurs for the remaining transactions in the shared queues. A review of the IMS front-end and the IMS log records is needed to determine this condition.

### X'28'

There is an incompatible unit of recovery status between IMS and RRS.

### X'29'

The unit of recovery interest token is no longer valid; RRS might have terminated and was restarted while IMS work was in progress. Register 2 contains the return code from the ATRRURD service.

### X'2C'

An expression of interest in the current context failed. Register 2 contains the return code from the CTXEINT service.

### X'2D'

The dependent region was processing a call to ATRCMIT or ATRBACK when RRS terminated.

### X'31'

IMS was unable to obtain an XID when one was required.

### DFSTMS00:

#### Reason code

#### Explanation

### X'02'

An application that is processing a cascaded transaction on a back-end IMS system has issued a program switch after issuing an APPC outbound protected conversation. The request to commit the UR on the front-end IMS failed.

### X'11'

IMS was unable to end the restart process with RRS. Register 2 contains the return code from the ATRIERS service.

### X'12'

IMS was unable to determine the sync point coordinator. Register 2 contains the return code from the ATRREIC service.

### X'1D'

The application is involved in a protected conversation and attempted to perform a program-to-program switch, which is invalid.

### X'1E'

The application is involved in a protected conversation or synchronous APPC/OTMA shared queues transaction. During phase 1 sync point processing, IMS is notified by either an LU 6.2 device or a front-end shared queues

IMS to terminate the sync point for the application. For a CM1 OTMA transaction, the client terminates at application commit time by replying ROLLBACK at receipt of transaction response (in place of PREPARE); Commit phase 1 is not started.

IMS Connect takes that ROLLBACK request and sends an AWE to RRS with AWE\_FUNC = AWERBAUR (Rollback request). RRS as resource coordinator tells IMS not to proceed with sync point so that IMS issues an 0711 RC1E abend to ensure that proper ROLLBACK takes place.

This return code can also occur if the front-end RRS is canceled and restarted for synchronous APPC/OTMA transactions that are processed by the back-end IMS.

**X'21'**

Dissociation of the private context from the current TCB failed. Register 2 contains the return code from the CTXSWCH service.

**X'2A'**

The IMS dependent region pseudoabended because an asynchronous terminate call was received for the unit of work. This problem error can also occur if either a /STOP REGION *xx* ABDUMP command was issued for a region in WAIT-RRS/OTMA PC status, or RRS is restarted at the back-end IMS for synchronous APPC/OTMA transactions.

**X'32'**

The IMS dependent region pseudoabended because a DFSRRSI FUNC=SET\_SIDE\_INFORMATION call failed during sync point processing. This problem is an IMS internal or RRS error. Analyze the pseudodump (67FF log records).

**X'34'**

An application that is processing a cascaded transaction on a back-end IMS system has issued a ROLB call after issuing an APPC outbound protected conversation. The application is terminated abnormally to avoid data integrity errors and the original transaction is rescheduled.

**X'35'**

After issuing an APPC outbound protected conversation, an application that was processing a cascaded transaction on a back-end IMS system has been rolled back (ROLB) because a deadlock occurred. The deadlock condition would have issued a U0777 abend of the application, or returned an FD or BC status code to the application. The application is terminated abnormally to avoid data integrity errors, and the original transaction is rescheduled.

**DFSAERG0:****Reason code****Explanation****X'33'**

ODBA could not successfully validate the application call. Register 5 contains the return code from the CTXRDIA service or other diagnostic information. Register 14 contains the address where the error was detected.

**For DFSPCJM0, DFSPCJB0, DFSPCC20, DFSECP10, or DFSECP20:**

0711 is a standard abend when issued by one of these modules. A severe error occurred during the processing of a z/OS UNIX call (such as BPX1TLS) from dependent region scheduling.

IMS writes log record X'67D0', subtype X'0608', where X'06' is D0SUBTYP and X'08' is D0ERRCOD. Subtype X'0608' means that a z/OS UNIX call failed. Message DFS554A is issued to report this error with the z/OS UNIX return and reason codes in the REASON= message text.

**For DFSTMAS0:**

0711 is a pseudoabend when issued by module DFSTMAS0. A severe error occurred during the processing of a z/OS UNIX call (such as BPX1TLS) from dependent region scheduling.

IMS writes log record X'67D0', subtype X'0608', where X'06' is D0SUBTYP and X'08' is D0ERRCOD. Subtype X'0608' means that a z/OS UNIX call failed. Message DFS554A is issued to report this error with the z/OS UNIX return and reason codes in the REASON= message text.

## 0712

**System action:** If this abend occurs during restart processing, IMS terminates abnormally. If it occurs during dependent region processing, the dependent region terminates abnormally.

**System programmer response:** If this problem is an internal error, such as an RRS/MVS service failure, first make sure that RRS/MVS was active on your system. Then contact IBM Software Support for assistance.

**Source:** IMS abends

**Module:** DFSASK00, DFSCPY00, DBFHGU10, DFSPCC20, DFSECP10, DFSRRC10, DFSRRSI0, DFSTMS00, DFSRRSB0, DFSAERG0, DFSDCPY0, DFSPCJM0, DFSPCJB0, DFSECP20, DFSTMAS0

**Related reference:**

[z/OS: z/OS UNIX System Services return codes](#)

**Related information:**

[DFS0613I \(Messages and Codes\)](#)

[DFS629I \(Messages and Codes\)](#)

[IMS abend 0113](#)

---

## 0712

**Explanation:** The IMS transaction enqueue or dequeue service encountered a system error. The reason code in register 15 indicates the error.

**Analysis:** For *DFSIMBE0*:

This is a standard abend issued by module DFSIMBE0.

Key	Label	Description
Reg15=X'04' Reg14=BAL Reg2=Latch Manager return code Reg3=A(TCT)		A request to obtain the TCTB latch in exclusive mode failed.
Reg15=X'08' Reg14=BAL Reg2=Latch Manager return code Reg3=A(TCT)		A request to release the TCTB latch in exclusive mode failed.
Reg15=X'0C' Reg14=BAL Reg3=A(TCT)		The pointer (TCTPRIOR) to the highest priority transaction within a class was invalid.
Reg15=X'10' Reg14=BAL Reg3=A(TCT)		The highest priority level within a class was exceeded when trying to find a higher priority transaction after which to enqueue the new transaction.
Reg15=X'14' Reg14=BAL Reg3=A(TCT)		The pointer (TCTPRIOR) to the highest priority transaction within a class was zero.

For *DFSIMBD0*:

This is a standard abend issued by module DFSIMBD0.

Key	Label	Description
Reg15=X'04' Reg14=BAL		An invalid function code was provided by the caller.
Reg15=X'08' Reg14=BAL Reg10=Latch Manager return code Reg9=A(TCT)		A request to obtain the TCTB latch in exclusive mode failed.
Reg15=X'0C' Reg14=BAL Reg9=A(TCT)		The address of the last enqueued transaction on a Transaction Class Table (TCT) was zero.

Key	Label	Description
Reg15=X'10' Reg10=Latch Manager return code Reg9=A(TCT)	Reg14=BAL	A request to release the TCTB latch in exclusive mode failed.

**System action:** The IMS control region is abnormally terminated.

**Operator response:** Perform an emergency restart.

**System programmer response:** A return code in register 15 identifies the cause of the error.

**Problem determination:** 1, 4, 5, 6, 25, 35

**Source:** IMS abends

**Module:** DFSIMBE0, DFSIMBD0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

### 0713

**Explanation:** The IMS MPP scheduler or the IMS transaction enqueue service encountered a system error. The reason code in register 15 indicates the error.

**Analysis:** For DFSSMSC0:

This is a standard abend issued by module DFSSMSC0.

Key	Label	Description
Reg15=X'04' Reg14=BAL		An invalid function code was provided by the caller.
Reg15=X'08' Reg14=BAL Reg7=A(TCT) Reg2=Latch Manager return code		A request to release the TCTB latch in exclusive mode failed.
Reg15=X'0C' Reg14=BAL Reg5=A(PDIR) Reg2=Latch Manager return code		A request to release the PDRB latch in exclusive mode failed.
Reg15=X'10' Reg14=BAL Reg2=Latch Manager return code		A request to release the APSB latch in shared mode failed.
Reg15=X'14' Reg14=BAL Reg7=A(TCT) Reg2=Latch Manager return code		A request to obtain the TCTB latch in exclusive mode failed.
Reg15=X'18' Reg14=BAL Reg2=Latch Manager return code		A request to obtain the SCHD latch in shared mode failed.
Reg15=X'1C' Reg14=BAL Reg2=Latch Manager return code		A request to release the SCHD latch in ANY mode failed.
Reg15=X'20' Reg14=BAL Reg2=Latch Manager return code		A request to obtain the APSB latch in shared mode failed.

## 0714

Key	Label	Description
Reg15=X'24' Reg14=BAL Reg5=A(PDIR) Reg2=Latch Manager return code		A request to obtain the PDRB latch in exclusive mode failed.
Reg15=X'28' Reg14=BAL Reg9=A(PST) Reg8=SUBQ number Reg7=Return Code from the Scheduler Subqueue Enqueue service		A request to enqueue a region (PST) on one of the Scheduler Subqueues failed.
Reg15=X'2C' Reg14=BAL Reg9=A(PST) Reg8=SUBQ number Reg7=Return Code from the Scheduler Subqueue Dequeue service		Request to dequeue a region (PST) from one of the Scheduler Subqueues failed.

### For DFSIMBE0:

This is a standard abend issued by module DFSIMBE0.

Key	Label	Description
Reg15=X'04' Reg14=BAL Reg10=A(Current <sup>®</sup> ITASK) Reg6=A(PST) Reg5=A(SQE) Reg4=A(SMB) Reg3=SUBQ number Reg2=A(SUBQ)		After finding a region (PST) to process the newly enqueued transaction (SMB), the region's Scheduler Subqueue Element (SQE) indicates that the region is on the wrong Scheduler Subqueue (that is, SQPSTSQNReg3).

**System action:** The IMS control region terminates abnormally.

**Operator response:** Perform an emergency restart; this should correct the situation.

**System programmer response:** A return code in register 15 identifies the cause of the error.

**Problem determination:** 1, 4, 6, 11, 15, 35

**Source:** IMS abends

**Module:** DFSSMSC0, DFSIMBE0

### Related information:

Chapter 2, "IMS problem determination," on page 13

---

## 0714

**Explanation:** A system error occurred when the IMS BMP region scheduler or the IMS APSB/DPSB call processor attempted to enqueue or dequeue a region on a scheduler subqueue. The reason code in register 15 indicates the error.

### Analysis: For DFSSBMP0:

This is a standard abend issued by module DFSSBMP0.



Key	Label	Description
Reg15=X'04' Reg9=A(PST) Reg2=IMS destination finder service Return code		The IMS destination finder service failed when processing a request to find the transaction specified on the IN= parameter.
Reg15=X'08' Reg9=A(PST) Reg2=IMS destination finder service Return code		The IMS destination finder service failed when processing a request to find the transaction or LTERM specified on the OUT= parameter.
Reg15=X'0C' Reg10=A(SAP) Reg9=A(PST) Reg3=SUBQ number Reg2=IMS scheduler subqueue enqueue service Return code		A request to enqueue the BMP region on one of the scheduler subqueues failed.
Reg15=X'10' Reg10=A(SAP) Reg9=A(PST) Reg3=SUBQ number Reg2=IMS scheduler subqueue dequeue service Return code		A request to dequeue the BMP region from one of the scheduler subqueues failed.
Reg15=X'14' Reg14=Return address of caller of subroutine REL PDRBL Reg9=A(PST) Reg7=A(PDIR) Reg2=Latch manager Return code		A request to release the PDRB latch in exclusive mode failed.
Reg15=X'18' Reg14=Return address of caller of subroutine REL PDRBL Reg9=A(PST) Reg7=A(PDIR) Reg2=Latch manager Return code		A request to release the APSB latch in shared mode failed.

**For DFSTMAD0:**

This is a standard abend issued by module DFSTMAD0.

Key	Label	Description
Reg15=X'04' Reg9=A(PST) Reg3=IMS scheduler subqueue enqueue service return code Reg2=A(SAP)		A request to enqueue a CPI-C application region on scheduler subqueue 4 failed.

## 0716

Key	Label	Description
Reg15=X'08' Reg9=A(PST) Reg3=IMS scheduler subqueue dequeue service return code Reg2=A(SAP)		A request to dequeue a CPI-C application region from scheduler subqueue 4 failed.

**System action:** The IMS control region terminates abnormally.

**Operator response:** Perform an emergency restart; this should correct the situation.

**System programmer response:** Register 15 contains a reason code identifying the cause of the error.

**Problem determination:** 1, 4, 6, 11, 15, 35

**Source:** IMS abends

**Module:** DFSSBMP0, DFSTMAD0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

## 0716

**Explanation:** During initialization of an IMS control region, a NO BUFFERS condition was detected.

**Analysis:** 0716 is a standard abend issued by the message queue buffer initialization module, DFSIINQ0, when an error is encountered during processing DFSIINQ0 is invoked using a BAL from DFSIINS0 to initialize the queue management buffer pool. This abend is issued because the size of the QPOOL is smaller than the size of one message buffer. The QPOOL size is located in the SCD at label SCDQPOOS. The buffer size is equal to the DCB block length of the long MSG queue data set (DCBBUFL).

The program status word (PSW) at entry-to-abend points to the instruction within the routine labeled NOBUFFS from which the abend (SVC 13) is issued. Register 11 in the abend SVRB points to the SCD, register 12 is the base register, and register 7 contains the DCB block size of the long MSG queue data set. Register 8 contains the following abend subcodes:

**Subcode (hex)**  
**Explanation**

### X'04'

IMODULE GETMAIN failed while allocating private storage for the queue buffer address list, DFSPQBFA.

- Register 3 contains the requested storage length.
- Register 15 contains the IMODULE return code.

For an explanation of IMODULE return codes, refer to the IMS system services return codes information. If the IMODULE failure was caused by a storage allocation, increase the region size parameter in the JCL and rerun the job.

### X'08'

The number of buffers required for the QBUF pool can be specified on the EXEC parameters or during system definition. If the number specified is greater than 3, the size of the QBUF pool is calculated by using the number of required buffers multiplied by the buffer size. If the number of buffers specified is three or less, the QBUF pool is allocated using an internal default size. If the default size is not large enough to hold at least one queue manager buffer, this abend is issued. Register 5 contains the computed length of a queue manager buffer. This problem is an internal logic error.

### X'10'

During processing of the user provided DD statements, the DCB DDNAME was found to contain all blanks. Register 3 contains the address of the DCB being processed.

### X'14'

During processing of the user provided DD statements, the primary DCB DDNAME contains eight characters. The primary ddname must be less than eight characters. Register 3 contains the address of the DCB being processed.

**X'18'**

IMODULE GETMAIN failed while allocating private storage for the optional DCB DD cards. Register 15 contains the IMODULE return code. For explanations of IMODULE return codes, see the IMS system services return codes information. If the IMODULE failure was caused by a storage allocation failure, increase the region size parameter in the JCL and rerun the job.

**X'0C'**

After the QBUF pool was allocated, a test was made to ensure that the pool was large enough to hold at least one queue manager buffer. During this validation, the QBUF pool size was found to be insufficient. Register 5 contains the computed length of a queue manager buffer.

**X'1C'**

DFSCBTS FUNC=ALTER failed while setting the block size and number for the QMBA block. Register 15 contains the DFSCBTS return code.

Key	Label	Description
Reg1=X'800002CC' Reg6=negative value	(prior to) LOOP	The size of the QPOOL (register 1) is divided by the buffer length of the Long Message data set DCB (DCBBUFL (register 5)). The result must be a positive value (register 6) or abend occurs.

**Possible cause:** Incorrect system definition specification of the MSGQUEUE buffer operand (size 3) is incorrect.

**System action:** IMS terminates abnormally.

**Programmer response:** Take the following actions:

**Subcode (hex)****Action****X'04'**

If the IMODULE failure was caused by a storage allocation failure, increase the region size parameter in the JCL and rerun the job.

**X'08'**

This is an internal logic error. Refer to the problem determination.

**X'10'**

This is an internal logic error. Refer to the problem determination.

**X'14'**

This is an internal logic error. Refer to the problem determination.

**X'18'**

If the IMODULE failure was due to a storage allocation failure, increase the region size parameter in the JCL and rerun the job.

**X'0C'**

This is an internal logic error. Refer to the problem determination.

**X'1C'**


This is an internal logic error. Refer to the problem determination.

**Problem determination:** 1, 4, 11, 34

**Source:** IMS abends

**Module:** DFSIINQ0

**Related reference:**

 IMS system services return codes (Messages and Codes)

**Related information:**

Chapter 2, "IMS problem determination," on page 13

## 0717

**Explanation:** During initialization, DFSIINS0 was called to perform pool initialization. An error occurred while trying to establish pool sizes as specified or implied by the parameter field of the IMS control region EXEC statement. This error can also occur if insufficient CSA or private storage is available during an IMODULE GETMAIN for buffer pools in DFSDOBIO.

In a batch environment using OSAM, the buffer pools are located in private storage. You should ensure that the region is large enough to accommodate the buffers that were requested.

**Analysis:**

Abend 0717 is a standard abend issued by modules DFSIINS0 and DFSDOBIO.

Module DFSDOBIO, the OSAM buffer pool initialization module, issues an IMODULE GETMAIN call to get storage for the buffer pools. If a nonzero return code is returned from DFSMODU0, abend 0717 is issued. At the time of abend, register 10 contains the SCD address, register 2 contains the total size of the buffer pool. The return code from the IMODULE GETMAIN request is in register 15.

When this abend is issued by module DFSIINS0, it is a general abend indicating that an error was found during processing. During IMS initialization, DFSIINS0 was called to perform storage pool initialization. Register 11 contains the SCD address and register 8 contains one of the following abend subcodes:

**Subcode (hex)****Explanation****X'04'**

A DFSBCB GET request was made to get QSAV storage in order to have enough save areas for modules called by DFSIINS0. A nonzero return code was received from DFSBCB. Register 15 contains the DFSBCB return code.

**X'08'**

An IMODULE GETMAIN request was made to get storage for DFSZIB00, which is the primary allocation of ZIB and FAQE blocks. A nonzero return code was received from IMODULE.

- Register 2 contains the address of the IMODULE parameter list.
- Register 3 contains the requested storage length.
- Register 15 contains the IMODULE return code.

For a description of these codes in register 15, see the IMS system services return codes information.

**X'0C'**

An IMODULE GETMAIN request was made to get extended CSA storage for a storage pool header block, PHDRxxxx, where xxxx is the storage pool name. A nonzero return code was received from IMODULE.

- Register 2 contains the address of the IMODULE parameter list.
- Register 3 contains the requested storage length.
- Register 4 contains the address of the storage pool header CDE name.
- Register 15 contains the IMODULE return code.

For a description of these codes in register 15, see the IMS system services return codes information.

**X'10'**

An IMODULE GETMAIN request was made to get storage for a storage pool, DFSPxxxx, where xxxx is the storage pool name. A nonzero return code was received from IMODULE.

- Register 2 contains the address of the IMODULE parameter list.
- Register 3 contains the requested storage length.
- Register 4 contains the address of the storage pool CDE name.
- Register 15 contains the IMODULE return code.

For a description of these codes in register 15, see the IMS system services return codes information.

**X'14'**

All storage pools manipulated by the storage pool managers using the IGETBUF/IFREEBUF or DFSPPOOL macros must have a unique four-character storage pool name. The pool name is used to find the storage pool header in the storage manager hash table. DFSIINS0 was attempting to add a new storage pool header address to the hash table, but the pool name was a duplicate.

- Register 4 contains the storage pool name.
- Register 9 contains the address of the storage pool header.

| **X'18'**  
 | An IMODULE LOAD request for DFS1MP00 was made. A nonzero return code was received from IMODULE.  
 | Register 15 contains the return code.

| **X'1C'**  
 | A STORAGE OBTAIN request was made to obtain 31-bit storage backed by large (1 MB) pages. A nonzero return  
 | code was received from the storage service. Register 15 contains the return code.

| **X'20'**  
 | The total amount of storage in pools backed by 1 MB pageable pages exceeds what will fit in a 2 GB 31-bit  
 | address space (X'7FFFFFFF').

**System action:** IMS terminates abnormally.

**Programmer response:** To recover from a failure in the DFS modules, take the following actions:

**DFSIINS0**

See the IMS system services return codes information for explanations of IMODULE and DFSBCB return codes.

**DFSMODU0**

Specify a larger CSA, a larger region size, or both; then rerun the job.

**DFSISMN0**

Correct any error in the pool size specification on the PARM field on the JCL EXEC statement; then rerun the job.

**DFSDOBI0**

Either:


- Specify a larger CSA size, region size, or both.
- Decrease the number of subpools defined, decrease the size of the buffer pool, or both. Then rerun the job.

**Problem determination:** 1, 4, 11, 34

**Source:** IMS abends

**Module:** DFSIINS0, DFSDOBI0

**Related reference:**

 IMS system services return codes (Messages and Codes)

**Related information:**

Chapter 2, "IMS problem determination," on page 13

**0718**

**Explanation:** An error occurred during IMS initialization. Either a required module could not be found or a permanent I/O error was encountered while searching a PDS directory.

**Analysis:** This is a standard abend issued by one of these modules: DFSAINB0, DFSPLDR0, DFSPCC30, DFSPLPP0, DFSXBAT0, DFSXCTL0, DFSXDINB0, DFSXLIC0, DFSXSL10, DFSXSTM0, and DFSXTRA0. The program status word (PSW) at entry-to-abend should be used to determine the failing module.

DFSPLDR0 is started using a BAL from DFSRRA00, which then loads DFSPLDT0 and BALs to DFSPLPP0, the module preload processor. DFSPLPP0 performs the BLDL processing of those modules specified by PROCLIB members DFSIIN10, DFSIIN20, and those modules specified by the user in member DFSMPLxx. When the BLDL processing is completed, DFSPLPP0 passes a list back to DFSPLDR0, which then performs the IMODULE LOAD function.

The work area DSECT, DFSPLDT0, addressed using register 11, is used as a common area for DFSPLPP0 and DFSPLDR0.

DFSXLIC0 is the first module to get control under the Common Services ITASK. The Common Services ITASK is created by DFSXSTM0.

DFSPCC30 is the application program controller for batch systems. When the batch program request handler module failed during loading, DFSPCC30 issued abend 0718.

**For DFSAINB0:**

Abend 0718 is issued by DFSAINB0 CSECT LOADXITS when the attempted load of DFSCCMD0 fails and ICMD

## 0718

security is implemented using RACF (or equivalent), the Command Authorization exit routine, or both. No error messages are issued before this abend code is issued.

Key	Label	Description
Reg15=Return code	LOADXITS	The SVC for IMODULE LOAD of DFSCCMD0 returned a nonzero return code. The load failed.
Reg15=Return code	LOADSECX	The SVC for IMODULE LOAD of IRRSPW00 returned a non-zero return code. The load failed.

### *Possible causes:*

- The required module is not on PDS.
- The AOIS parameter in the DBC, DCC, or IMS procedure is incorrect for the installation.
- The IRRSPW00 module provided by RACF is not included in LPA, in a library in LINKLIST, or in one of the IMS.SDFSRESL concatenated data sets.

### **For DFSPLDR0:**

This is issued by DFSPLDR0 when the attempted load of the required modules failed for one of the following reasons:

1. z/OS failed.
2. the IMS module manager, DFSMODU0, rejected the load request.

Message DFS677I or DFS678I is issued to the system console before the abend, along with message DFS679I. DFS677I indicates an I/O error, and DFS678I indicates that a module was not found. Message DFS679I contains the module names together with the associated IMODULE LOAD return code.

The program status word (PSW) at entry-to-abend points to the instruction within label ABNDBADL from which the abend (SVC 13) was issued. Register 12 in the abend SVRB is the base register.

Key	Label	Description
Reg1=X'800002CE'	LLPCHECK	A required IMS module failed to load.

### *Possible cause:* The causes include:

1. Hardware failure
2. PDS directory damaged
3. Required modules not on the PDS

### **For DFSPLPP0:**

This abend is issued by DFSPLPP0 because BLDL processing encountered a permanent I/O error condition while scanning the PDS directory. Message DFS677I lists the modules that encountered the I/O error. This message is sent to the IMS master console before the abend.

The program status word (PSW) at entry-to-abend points to the instruction within label BLAR from which the abend (SVC 13) is issued. Register 15 in the abend SVRB contains the error return code from the BLDL macro instruction, and register 12 is the base register. The possible BLDL return codes are:

#### **Code    Meaning**

- X'04'    One or more entries in the list could not be filled; the list supplied might be invalid.
- X'08'    A permanent I/O error was detected.

Register 15 should be used to isolate to the specific label in the following table:

Key	Label	Description
Reg15>X'08'	BLDLOOP	The return from the BLDL macro instruction was out of the specified range.
Reg15=X'08'	BLDLNXT	A permanent I/O error condition was detected.

*Possible cause:* Hardware failure or PDS directory damaged.

**For DFSXBAT0:**

This is issued by DFSXBAT0 when an IMODULE LOAD fails for one of the following modules: DFSBSCD, DFSPCCC0, DFSRDSH0, DFSBNUC0, DFSKBDP0, DFSCST00, DFSCSS00, DFSCNS00, DFSPCC30, DFSBACM0, DFSFXC10, DFSRDBC0, DFSSDL90, DFS3DF10.

The program status word (PSW) at entry-to-abend points to the instruction within label ABEND from which the abend (SVC 13) was issued. Register 12 in the abend SVRB is the base register.

Key	Label	Description
Reg1=X'800002CE'	ABEND4	An IMODULE LOAD failed for a required IMS module.
Reg14=BAL register		Register 14 points to the location in module DFSXBAT0 that detected the error.
Reg15=return code		Register 15 contains the return code from IMODULE LOAD.

**For DFSXCTL0:** This is a standard abend issued by DFSXCTL0.

Key	Label	Description
Reg15=return code	ABEND3	The IMODULE LOAD of DFSXCIC0 or DFSKDP10 returned a nonzero return code. The load failed. Register 14 points to the location in module DFSXCTL0 that detected the error.

**For DFSXLIC0:** This is a standard abend issued by DFSXLIC0.

Key	Label	Description
Reg15=return code	CALLXRIC	SVC For IMODULE LOAD or DFSPCCC0 returned a nonzero return code. The load failed.

**For DFSDINB0:** An IMODULE LOAD failed during DB Control (DBCTL) initialization. Message DFS697I is issued along with this abend. The message contains the name of the module that IMODULE was unable to load and the return code from IMODULE.

**For DFSXSL10:** An IMODULE LOAD or IMODULE GETMAIN in DFSXSL10 failed during IMS initialization.

This is issued by DFSXSL10 when IMODULE LOAD fails for one of the following modules: DFSSINP0, DFSOCM10, or IMODULE GETMAIN fails.

The program status word (PSW) at entry-to-abend points to the instruction from which the abend (SVC 13) is issued. Register 15 in the abend SVRB contains the error return code. Register 12 is the base register.

Key	Description
Reg14=A(caller) IMODULE LOAD DFSOCM10 Reg15=1	An IMODULE LOAD failed for DFSOCM10.

## 0719

Key	Description
Reg14=A(caller) IMODULE GETMAIN Reg15=2	An IMODULE GETMAIN failed.
Reg14=A(caller) IMODULE LOAD DFSSINP0 Reg15=3	An IMODULE LOAD failed for DFSSINP0.

### For DFSXSTM0:

This is issued by DFSXSTM0 when IMODULE LOAD fails for one of the following modules: DFSFSTM0, DFSXRPS0, DFSHINT0, DFSFDOT0, DFSXLIC0, DFSCNS00. Reg14 should be the return register from the routine that had the load failure, and Reg15 should contain the return code from the IMODULE LOAD.

Key	Label	Description
Reg14=BAL register		Register 14 points to the location in module DFSXSTM0 that detected the error.
Reg15=return code		Register 15 contains the return code from IMODULE LOAD.

**For DFSXTRAO:** The system was unable to load needed external trace modules or control blocks. The external trace TCB was abended, allowing OLDS external tracing only. Register 14 contains the BAL REG pointing to the routine that had the error. Register 15 contains the return code from IMODULE.

**System action:** IMS terminates abnormally.

**Programmer response:** If this abend was issued by either DFSPLPP0 or DFSPLDR0, the modules for which the I/O error occurred are listed in message DFS677I, DFS678I, or DFS679I. Refer to the appropriate message for the information necessary to correct the problem.

**Source:** IMS abends

**Module:** DFSAINB0, DFSPLDR0, DFSPCC30, DFSPLPP0, DFSXBAT0, DFSXCTL0, DFSDINB0, DFSXLIC0, DFSXSL10, DFSXSTM0, DFSXTRA0

### Related information:

[DFS677I \(Messages and Codes\)](#)

[DFS678I \(Messages and Codes\)](#)

[DFS679I \(Messages and Codes\)](#)

---

## 0719

**Explanation:** The control region initialization was unable to successfully open any line groups. DFSIINB0 has detected an internal IMS error during IMS Transaction Manager initialization. Initialization cannot continue.

For a switched device, the terminal type identification exceeds the valid range.

**Analysis:** This is a standard abend issued by module DFSIINB0. The registers at entry-to-abend contain the following information:

### Register 8

CTT that contains the invalid device type

### Register 15

The invalid device type.

**System action:** IMS terminates abnormally.

**Programmer response:** Ensure the correct control blocks were linked into the IMS nucleus.

**Problem determination:** 1, 4, 6, 34

**Source:** IMS abends



**Module:** DFSIINB0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

**0720**

**Explanation:** IMS initialization was unable to page fix storage. Register 14 contains the location that issued the abend. Register 15 contains the return code from the IMSAUTH SVC PGFIX function. For a description of the IMSAUTH PGFIX return codes, see the IMS system services return codes information.

IMS also issues this abend for other reasons.

**Analysis:** The program status word (PSW) on entry-to-abend can be used to determine which module issued the abend. The registers in the abend SVRB should be used for problem isolation. The dump of control blocks written to the system is needed.

The abend is issued from module DFSMINI0 for reasons other than page fix errors. Refer to the individual write-up for DFSMINI0 for diagnostic assistance.

In either module, the IMS SVC IMSAUTH was just issued. The IMS SVC (DFSV4200) returned an error code in register 15.

**For DFSMINI0:**

The registers in the abend SVRB should be used for problem isolation.

Key	Label	Description
Reg15=X'01'	INREP01	Unable to load a dispatcher module (DFSREP00 or DFSDSPX0).
Reg15=X'02'	INIDSP0	Unable to load dispatcher module DFSIDSP0.
Reg15=X'03' Reg14=BAL		IMODULE GETMAIN failure to acquire storage.
Reg15=X'04'		"Reserve an authorization index" failed.
Reg15=X'05'		"Set an authorization index" failed.
Reg15=X'06'		"Reserve a linkage table index" failed.
Reg15=X'07'		"Create an entry table" failed.
Reg15=X'08'		An entry table connect request failed.
Reg15=X'09'		Unable to load DFSCPY00.
Reg15=X'0A'		Unable to load PC router routine DFSPCR00.

For all of these subcodes, register 10 contains the return code from the service that detected the error, and register 14 contains the address in DFSMINI0 where the error was detected.

**For DFSXIOB0:**

IMS was unable to page fix the OSAM I/O module (DFS AOS70) during OSAM I/O pool initialization.

The registers in the abend SVRB should be used for problem diagnosis. The program status word (PSW) at entry-to-abend will point to the instruction within label RETURN5 from which abend (SVC 13) is issued.

Key	Label	Description
Reg1=abend code Reg14=offset from table at RET1A Reg15=return code from IMSAUTH page fix call	RETURN5	An attempt was made to page fix the OSAM I/O module and a nonzero return code was returned by IMSAUTH.

**Possible cause:** An incorrect parameter list or incorrect module addresses were used.

## 0721 • 0722


**System action:** IMS terminates abnormally.

**Problem determination:** 1, 4, 6, 34

**Source:** IMS abends

**Module:** DFSMINI0, DFSXIOB0

**Related reference:**

 [IMS system services return codes \(Messages and Codes\)](#)

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

### 0721

**Explanation:** An IFREEBUF call was issued to free a buffer address that does not exist.

**Analysis:** For abend 0721, find label IFSCAN. Several instructions after it you find instruction BZ IFNOBUF. IFNOBUF sets the abend code in register 1 and branches to ABNDEXIT where the abend is issued.

Key	Label	Description
Reg1=abend code Reg2=pool name Reg3=A(buffer to be freed) Reg6=pool header Reg6 + X'2C'=first FAQE	IFSCAN	A scan through the FAQE chain for the pool in question failed to locate a buffer address to match register 3. Scan the queue to verify that it is correct.

**Possible cause:** DFSISMN0 was called to free a buffer, but the buffer is not in the pool. Program may be trying to free the buffer for the second time.

**System action:** The IMS control region is abnormally terminated.

**Operator response:** Perform an emergency restart.

**Problem determination:** 1, 4, 5, 6, 34

**Source:** IMS abends

**Module:** DFSISMN0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

### 0722

**Explanation:** An improper post of an ECB waited on by DFSISMN0 has been detected.

**Analysis:** 0722 is a standard abend issued from one of two locations within DFSISMN0.

Within the main routine following WAIT30, you will find the instruction BNE BADPOST. BADPOST sets the error code in register 1 and branches to ABNDEXIT where the abend is issued.

Within the DFSIZBR0 subroutine, following GSTGABND, you will find the instruction BAL R14, GSAB722. GSAB722 issues the 0722 abend.

Key	Label	Description
Reg1=abend code Reg15=expected post code	WAIT30	The ECB being WAITed on was posted by a routine other than storage management. Register 9 contains the ECB address.
Reg1=abend code Reg15=expected post code		The ECB being ISERWAITed on was posted by a routine other than storage management. Register 9 contains the ECB address.

**System action:** The IMS control region terminates abnormally.

**Operator response:** Perform an emergency restart.

**Problem determination:** 1, 4, 5, 6, 34

**Source:** IMS abends

**Module:** DFSISMN0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

0723

**Explanation:** IMS was unable to locate the subsystem control table (SSCT) during OSAM initialization.

**Analysis:** Use the registers in the abend SVRB for problem diagnosis. The program status word (PSW) at entry-to-abend points to the instruction with label RETURN6 from which the abend (SVC 13) is issued. This abend is a common abend for errors encountered during I/O pool initialization processing.

Key	Label	Description
Reg1=abend code Reg14=offset from table at RET1A Reg15=FUNCTION CODE in left half Reg15=return code in right half	RETURN6	An invalid condition occurred during I/O pool initialization. Register 15 contains descriptive codes.
Reg15=0001 0004	RETURN6	OSAM I/O pool not allocated on DW boundary.
Reg15=0002 00xx	RETURN6	Unable to locate OSAM I/O module return code. <i>xx</i> is set by IMODULE.
Reg15=0003 0004	RETURN6	Located I/O module failed verification check.
Reg15=0004 00xx	RETURN6	IMODULE GETMAIN failed for page fix list or work area. Return code <i>xx</i> is set by IMODULE.
Reg15=0006 00xx	RETURN6	OSAM I/O pool allocation failed. Return code <i>xx</i> is set by DFSSTM00.
Reg15=0007 00xx	RETURN6	IMODULE DELETE failed for work area. Return code <i>xx</i> is set by IMODULE.
Reg15=0008 0004	RETURN6	Nonzero condition code set by LRA while translating CCW.
Reg15=0009 0004	RETURN6	An invalid pool type was passed to OSAM for formatting.

**Possible cause:** Two possible reasons for this abend are:

1. The resource cleanup module (DFSMRCL0) is not installed correctly.
2. The wrong level of DFSMRCL0 is being used. If multiple levels of IMS are run on the same operating system, use the copy of DFSMRCL0 for the highest level.

**System action:** IMS terminates abnormally.

**System programmer response:** Make sure that the resource cleanup module (DFSMRLCO) has been installed correctly as described in *IMS Version 13 Installation*. If you are running multiple levels of IMS on the same operating system, then make sure that DFSMRCL0 is from the highest level IMS system that you are running.

**Problem determination:** 1, 4, 6

**Source:** IMS abends

**Module:** DFSXDCC0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

**0725**

**Explanation:** IMS encountered insufficient storage while trying to obtain additional ZIBs and FAQEs, or an error was encountered in the execution of DFSMODU0. The return code is in register 15

**Analysis:** Either IMS encountered insufficient storage while trying to obtain additional ZIBs and FAQEs, or an error was encountered in the execution of DFSMODU0. The return code from the DFS0CSS call is in Register 15.

**System action:** The IMS control region terminates abnormally.

**Programmer response:** Re-IPL z/OS, specifying a larger CSA allocation; rerun the job.

**Problem determination:** 1, 4, 5, 6, 34

**Source:** IMS abends

**Module:** DFSISMN0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

**0728**

**Explanation:** While attempting to make the IMS batch region nonswappable, the IMSAUTH SVC encountered a problem.

**Analysis:** This is a standard abend issued by module DFSDLN00.

The return code is set in register 15. For a description of the IMSAUTH NOSWAP return codes, see the IMS system services return codes information.

**System action:** The IMS batch region terminates abnormally.


**Operator response:** Contact the IMS system programmer.

**Problem determination:** 1, 4, 8, 34

**Source:** IMS abends

**Module:** DFSDLN00

**Related reference:**

 [IMS system services return codes \(Messages and Codes\)](#)

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

**0729**

**Explanation:** During a DFSPPOOL GET request, module DFSSPM40 was called to allocate a block from an existing storage pool. Unrecoverable internal errors were detected during processing.

This abend is issued from both DFSSPM40 and DFSSPM50. During a DFSPPOOL GET request, module DFSSPM40 is called to allocate a buffer from an existing storage pool. During a DFSPPOOL REL request, module DFSSPM50 is called to release a buffer. If DFSSPM40 is the issuer of the abend, then R0, R1, and R15 used by this logic (just before the abend) are saved in the save area set following DFSSPM40. If an unrecoverable internal error is detected by either module, an abend 0729 abend is issued.

**Analysis:** This is a general abend issued by module DFSSPM40 and DFSSPM50. Register 15 contains the abend subcode:

**Codes****Explanation****X'04'**

When obtaining a buffer, the caller's task was put in a wait state. When the task was posted, the storage manager determined that the post code was invalid.

**X'08'**

While processing a GET request, the storage manager found a block with a free buffer and allocated that buffer

by updating the bit map in the block header. Before returning the address of the buffer to the caller, the storage manager either verifies that the buffer resides in the block. If it doesn't, this abend is issued. This error may have been caused by a storage overlay.

**X'10'**

The address in the first word of the buffer prefix does not point to a block header. Either an overlay has occurred or a bad address was passed to the storage manager.

**X'14'**

A buffer is not being released to the correct pool. An overlay may have occurred.

**X'18'**

The buffer being released is not currently allocated. An overlay may have occurred.

**X'C'**

The first word of the buffer prefix contains zeroes. Either an overlay occurred or a bad address was passed to the Storage Manager.

**X'1C'**

The address of the buffer being released is not on a buffer boundary within the block.

**System action:** IMS terminates abnormally.

**Problem determination:** 1, 4, 6, 11, 34

**Source:** IMS abends

**Module:** DFSSPM40, DFSSPM50

**Related information:**

Chapter 2, "IMS problem determination," on page 13

**0730**

**Explanation:** During IMS initialization, module DFSXSPM0 was called to perform storage pool manager initialization.

**Analysis:** This is a general abend issued by module DFSXSPM0 when an error is found during processing. For explanations of IMODULE return codes, see the IMS system services return codes information. Register 8 contains the abend subcode:

**Codes****Explanation****X'04'**

An IMODULE GETMAIN request was made to get extended CSA storage for the storage pool manager hash table, SPMHSHTB. A nonzero return code was received from IMODULE. Register 2 contains the address of the IMODULE parameter list. Register 3 contains the requested storage length. Register 15 contains the IMODULE return code.

**X'08'**

An IMODULE GETMAIN request was made to get extended CSA storage for the storage pool manager bitmap lookup table, SPMBMTAB. A nonzero return code was received from IMODULE. Register 2 contains the address of the IMODULE parameter list. Register 3 contains the requested storage length. Register 15 contains the IMODULE return code.

**X'0C'**

An IMODULE LOAD request was made to load the storage pool definitions, DFSSPM10, into extended CSA. A nonzero return code was received from IMODULE. Register 2 contains the address of the IMODULE parameter list. Register 15 contains the IMODULE return code.

**X'10'**

An IMODULE LOAD request was made to load composite load module DFSSPM40 which contains the storage pool manager routine. A nonzero return code was received from IMODULE. Register 2 contains the address of the IMODULE parameter list. Register 15 contains the IMODULE return code.

**X'14'**

An IMSVS.PROCLIB member was specified using the SPM= suffix on the control region JCL but the member was not found.

**X'18'**

An internal error occurred while processing IMSVS.PROCLIB FPL= statements. DFSXSPM0 encountered more valid pool names than it had temporary pool areas defined. Only fixed type pools contain buffer definitions which can be overridden by the PROCLIB statements. All fixed pool definitions reside in DFSSPM10. The amount of storage reserved for temporary definitions in DFSXSPM0 must be enough to allow all fixed pools to be processed.


**System action:** IMS terminates abnormally.

**Problem determination:** 1, 4, 11, 34

**Source:** IMS abends

**Module:** DFSXSPM0

**Related reference:**

 [IMS system services return codes \(Messages and Codes\)](#)

**Related information:**

Chapter 2, "IMS problem determination," on page 13

**0731**

**Explanation:** The storage pool manager was called for a DFSPPOOL ALLOC function to allocate one or more of the storage pools defined in module DFSSPM10. During the storage pool allocation process, one or more errors were found. The storage manager returned with a nonzero return code causing the caller to terminate processing.

**Analysis:** Register 2 contains the DFSPPOOL parameter list address which points to one or more pool allocation entries. Each entry is mapped by the POOLALOC DSECT which is generated by DFSPPOOL GENLISTD. The last entry in the list is one word, X'FFFFFFFF'. Byte 2 of each entry contains a 2-byte return code which indicates whether the pool specified in the entry was allocated. The high byte of the return code, if nonzero, is the return code from the IMODULE GETMAIN service. Register 15 contains the largest DFSPPOOL return code encountered during processing for all entries. For explanations of DFSPPOOL and IMODULE GETMAIN return codes, refer to the IMS system services return codes information.

**System action:** IMS terminates abnormally.

**Programmer response:** Register 2 contains the DFSPPOOL parameter list address, which points to one or more pool allocation entries. Each entry is mapped by the POOLALOC DSECT generated by DFSPPOOL GENLISTD. The last entry in the list is the word, X'FFFFFFFF'.


Each entry at byte 2 contains a 2-byte return code which indicates whether the specified pool in the entry was allocated. The high byte of the return code, if it is nonzero, is the return code from the IMODULE GETMAIN service. The low byte of the return code is from the DFSPPOOL ALLOC service.

Register 15 contains the largest DFSPPOOL return code found during processing for all entries.

**Source:** IMS abends

**Module:** DBFINI20, DFSIINB0, DFSIING0, DFSIINQ0, DFSXESI0

**Related reference:**

 [IMS system services return codes \(Messages and Codes\)](#)

**0732**

**Explanation:** The storage pool manager was called for a DFSPPOOL GET function to obtain a buffer from a previously allocated storage pool. An error was found during the buffer allocation process. A nonzero return code from the storage manager caused the caller to terminate processing. Register 15 contains the DFSPPOOL GET return code.

**Analysis:** Register 15 contains the DFSPPOOL GET return code. For explanations of the DFSPPOOL return codes, refer to the IMS system services return codes information. If the return code indicates there was a problem with the upper limit, make sure that the user specified upper limit was not in error. If the return code is greater than X'20', the problem is an internal system error.


**System action:** IMS terminates abnormally.

**Programmer response:** If the return code indicates that there was a problem with the upper limit, make sure the upper limit that the user specified was not in error. If the return code is greater than X'28', the problem is an internal IMS error.

**Problem determination:** 1, 4, 11, 34

**Source:** IMS abends

**Related reference:**

 [IMS system services return codes \(Messages and Codes\)](#)

**Related information:**

Chapter 2, "IMS problem determination," on page 13

## 0735

**Explanation:** This abend is a general abend issued by modules DFSICIO0 or DFSASLT0 when an unrecoverable internal error is detected during processing (that is, when the CLB ITASK is running but the running flag is not on).

This abend is also issued by module DFSCMS00 when MSCWA\_CMS10\_SAV is nonzero. It indicates that DFSCMS00 did not return to the caller and instead is attempting to return to the dispatcher.

**Analysis:** PSW should tell where the abend occurs in DFSICIO0, DFSASLT0, and DFSCMS00. IMS dump and logs are needed for problem diagnosis.

**System action:** IMS terminates abnormally.

**Problem determination:** 1, 4, 5, 6, 11, 36

**Source:** IMS abends

**Module:** DFSCMI00, DFSCMS00, DFSICIO0, DFSASLT0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

## 0736

**Explanation:** There was insufficient storage to obtain the restart work area during IMS initialization.

**Analysis:** This is a standard abend issued by DFSTMEI0. The program status word (PSW) at entry-to-abend will point to the abend routine which has a label of ABEND. Register 15 will contain the return code from the IMODULE GETMAIN routine and register 0 will contain the address pointer to the constant DFSSTPWA and should be used to isolate the correct label. These registers can be found in the abend SVRB.

Key	Label	Description
Reg0=A (DFSSTPWA) Reg1=X'800002E0'	IMOS0009	The IMODULE GETMAIN for the STOP region work area failed because of a nonzero return code in register 15. The IMS system is terminated abnormally.

**Possible cause:** Region size for the IMS control region is too small.

**System action:** The IMS control region is terminated.

**Programmer response:** Increase the region size for the IMS control region, and restart IMS.

**Problem determination:** 8

**Source:** IMS abends

**Module:** DFSTMEI0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

## 0737

**Explanation:** An internal IMS error occurred while attempting to start a database.

**System action:** IMS terminates abnormally.

**Programmer response:** One of the following codes in register 15, at the time of the abend, identifies the cause of the failure:

**Code (dec)****Explanation**

- 1 A GETMAIN for working storage failed. The size requested is in register 8 at abend time.
- 2 The control region has knowledge of a backout required for the database being started, but DBRC does not know the necessary backout.
- 3 An attempt to obtain an internal RRE block failed.
- 4 The database being started required a backout. However, this database was one of several databases originally changed under a particular recovery scope. After performing the backout, the blocks for the other databases could not be located for updating system information.
- 5 The control region successfully issued a backout for the database being started, but the release of the LCRE storage failed.

Take appropriate action according to the indicated return code.

**Source:** IMS abends

---

## 0738

**Explanation:** When the IMS control region completed a service for a message region, the return to the message region failed because the region was canceled by z/OS.

**Analysis:** Abend 0738 can be a pseudoabend or a standard abend issued when an ISWITCH fails because the region is no longer there. The ISWITCH might have been either from a dependent region to the control region, or from the control region to a dependent region.

**Possible cause:** The control blocks were usable, but now the control blocks are overlaid or broken.

**System action:** The PST is freed for reuse. (The message region has already terminated as a result of MVS CANCEL.)

**Problem determination:** 1

**Source:** IMS abends

**Module:** DFSASK00, DFSCP00, DFSUIC00

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

## 0739

**Explanation:** There was insufficient storage in an IMS message region to obtain the dependent region inter-region communication area.

**System action:** The message region is terminated.

**Programmer response:** Increase the size of the message region, and resubmit the job.

**Problem determination:** 8

**Source:** IMS abends

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---



**0740**

**Explanation:** The partner product user exit returned a non zero return code to module DFSRDY00. The return code is in R15.

**System action:** IMS terminates abnormally.

**Programmer response:** Correct the problem encountered by the partner product user exit, DFSPPEU0.

**Source:** IMS abends

**0741**

**Explanation:** Multiple Systems Coupling was unable to complete initialization.

**Analysis:** 0741 is a standard abend issued by DFSIIMS0. The program status word (PSW) at entry-to-abend points the instruction within label ABEND from which the abend (SVC 13) is issued. Register 14 in the abend SVRB should be used as the key to the specific label.

Register 0 and register 1 are saved, before the termination, at label ABENDSAV. Register 12 is the base register for this module, and register 11 is the pointer to the SCD.

Key	Label	Description
Reg14=BAL Reg9=BAL	GETLXBS	The channel to channel (CTC) link must be established; the IMODULE LOAD failed for DFSCMC40. Register 15 contains the return code passed back by the IMS SVC (DFSMODU0).
Reg14=BAL	IMOS0005	The IMODULE locate for DFSCMC40 failed. Register 15 contains the error return code. In ABENDSAV, register 0 contains the size of DFSCMC40 and register 1 its address.
Reg14=BAL	IMOS0007	The IMODULE GETMAIN for storage for the CTC LXBS failed, and the abend is issued. Register 15 has the error return code.
Reg14=BAL	(no label)	The IMS SVC was run to page fix the CTC LXBS. The page fix failed with an error return code in register 15.
Reg14=BAL	IMOS0010	The IMODULE delete for module DFSCMC40 failed. Register 15 contains the error code and register 0 in ABENDSAV points to the z/OS prefix save area (PSA).
Reg14=BAL	TRYMTM	For a main-storage to main-storage link the access method module must be loaded. The IMODULE LOAD for DFSMTMA0 failed, resulting in abend. Register 15 contains the return code.
Reg14=BAL	IMOS0014	The IMODULE GETMAIN for the LXBS and I/O buffers failed. Register 15 contains the error return code.
Reg14=BAL	BLDP1	The CTT is tested to determine the type of link so that the LXBS can be initialized. An unknown link type results in failure and abend. Register 15 contains the address pointer to the CTT.
Reg14=BAL	SIDLOOP1	It has been determined that a SID exists, but it was not a local SID. Abend results. Register 1 in ABENDSAV points to the SID.

Abend 0741 can be issued from module DFSCMMP0 at label MS0ABEND when a GETMAIN fails. The IMODULE GETMAIN for storage for the CTC I/O buffers failed, and the abend code is issued. Register 15 contains the error return code.

**Possible cause:** The causes include:

- Insufficient amount of CSA to build MSC control blocks.
- Loading or deleting of the MSC post handlers (DFSCMC20-DFSCMM20) failed because of an invalid request to SVC4.
- No local SIDs for this system (probable system definition bug).

**For APAR processing:** Abend memory dump, console sheet

**System action:** IMS terminates abnormally.

## 0742

**Programmer response:** Register 14 minus 4 is the address of the instruction that determined the error. Ensure that the modules to be loaded are in SDFSRESL. Ensure that z/OS includes sufficient CSA at IPL time.

**Problem determination:** 1, 4, 35

**Source:** IMS abends

**Module:** DFSIIMS0, DFSCMMP0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

## 0742

**Explanation:** An error was detected in the maintenance of the DL/I task identification, which governs the reuse of freed space in a database during IMS online execution.

**Analysis: For DFSDLA00:**

An error was detected in the maintenance of the DL/I task identification that governs the reuse of freed space in a database during IMS online execution. One of three conditions was encountered:

- During checkpoint, a new date was detected, but a valid DL/I task identification could not be generated. The cause could be an error in the time SVC, or the date set in the processor. The condition can also occur in an IMS system that has run continuously for more than 658 days: the portion of the task identification containing the date information has overflowed.
- An error in the range of active identifications was detected. For example, the low identification was higher than the next-available identification.
- The identification of a dependent region outside the known range (of active identifications) was detected.

This is a standard abend issued by module DFSDLA00. The fields involved are:

**SCDLOWID:**

Identification of the oldest active task.

**SCDNAVID:**

The next task identification to be assigned.

**PSTTSKID:**

The identification of this task.

*Possible cause:* IMS system error

**For DFSRDBC0:**

While a backout was being performed as part of an XRF takeover, one of the following conditions occurred:

- An error in the range of active identifications was detected. For example, the low identification was higher than the next-available identification.
- The identification of a dependent region outside the known range (of active identifications) was detected.

When DFSRDBC0 issues this standard abend, register 12 in the abend SVRB registers is the base register. Register 9 points to the PST, and register 11 points to the SCD. Fields SCDLOWID, SCDNAVID, and PSTTSKID, explained below, are also helpful in problem diagnosis.

**SCDLOWID:**

Identification of the oldest active task.

**SCDNAVID:**

The next task identification to be assigned.

**PSTTSKID:**

The identification of this task.

**For DFSDHD00:**

DFSDHD00 issues this abend on return from an enqueue request for the TASKID of the PST if there is:

- A nonzero return code and
- No pseudoabend currently scheduled

Field PSTTSKID contains the identification of this task.

**System action:** For conditions 1, 2, and 4, the IMS control region terminates abnormally. For condition 3, the application in the dependent region terminates abnormally.

**Operator response:** For conditions 1, 2, and 4, perform an emergency restart. For condition 3, issue a /START command to start the program and transaction affected, and, if the region was a BMP, restart the region.

**Programmer response:** Identify the problem and rerun.

**Source:** IMS abends

**Module:** DFSDLA00, DFSRDBC0, DFSDHD00

#### 0743

**Explanation:** The active region counter in the SCD is negative. The number of active MPP regions (SCDMPPAC) or active BMP regions (SCDBMPAC) became negative during the termination processing of that type of region. The data address must be addressable by 32 bits.

**Analysis:** This is a standard abend issued by DFSSABN0. DFSSABN0 is called when the TERMINATE THREAD call is issued for a dependent region to perform necessary PST cleanup functions.

Register 2 in the abend SVRB contains a negative number representing the number of active regions. Register 12 is the base register and register 11 is the pointer to the SCD.

Key	Label	Description
Reg2=negative value Reg11=SCD address Reg12=base register	SABN0600	Either the number of active MPP regions or active BMP regions became negative during the termination processing of a dependent region.

**System action:** IMS terminates abnormally.

**Source:** IMS abends

**Module:** DFSSABN0

#### 0744

**Explanation:** Module DFSSMIC0 was called invalidly. The call function in the RCSCHD field of the relocated RCPARMS image in the PST must be INIT or LOAD. This problem is an internal system error.

**System action:** The IMS control region terminates abnormally.

**Operator response:** Perform an emergency restart.

**Problem determination:** 1, 4, 5, 6, 25, 35

**Source:** IMS abend

**Related information:**

Chapter 2, "IMS problem determination," on page 13

#### 0745

**Explanation:** The IMS scheduler discovered an invalid function (not CREATE THREAD) in the SSOB during scheduling of a batch message processing (BMP) region.

**Analysis:** The function passed was not CREATE-THREAD. Register 6 in the abend SVRB should contain a call function in EBCDIC format. Register 12 is the base register.

Key	Label	Description
Reg1=X'800002E9' Reg8=a(SSOB) Reg9=a(PST)	GOON	The function passed in the field SSOBFUNC is not CREATE-THREAD.

**Possible cause:** The SSOB passed by DFSASK00 is invalid.

**System action:** The program terminates abnormally.

## 0746 • 0748

**Problem determination:** 1, 5, 11, 12

**Source:** IMS abends

**Module:** DFSSBMP0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

### 0746

**Explanation:** Module DFSDLA00 detected a failure on a CLOSE call during batch termination or checkpoint.

**Analysis:** Check the return codes and information from the IMS OPEN/CLOSE error messages. Also check the return codes and information from any system error messages, if present.

Key	Label	Description
Bit PSTOCBAD is on in byte PSTFNCTN	UNCLCOM	DFSDLA00 initiates abend 0746 if PSTOCBAD is on and another abend has not been initiated.

**System action:** The application terminates abnormally.

**Programmer response:** Check the system error messages, the IMS error messages issued, or both for the CLOSE failure.

**Source:** IMS abends

**Module:** DFSDLA00

---

### 0747

**Explanation:** A second CREATE-THREAD was issued for the same TCB without issuing a TERM-THREAD.

**Analysis:** 0747 is a pseudoabend issued by module DFSASK00.

**Possible cause:** This is probably an IMS system error.

**System action:** The dependent region is terminated.

**Problem determination:** 4, 35

**Source:** IMS abends

**Module:** DFSASK00

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

### 0748

**Explanation:** The IMODULE macro encountered an error while trying to perform an IMODULE LOAD request for module DFSCST00.

**Analysis:** 0748 is a standard abend issued from module DFSCST00. The registers in the abend SVRB are current at time of abend. The low-order half of register 15 contains the return code. The high-order half of register 15 contains the reason code loaded by DFSCST00.

Key	Label	Description
Reg15 = X'000400xx'	Term	IMODULE LOAD failed for DFSTRM00.
Reg15 = X'000800xx'	DBCMD5	IMODULE LOAD failed for DFSDBDR0.

**Attention:** *xx* is equal to the return code passed from the IMODULE macro.

**System action:** IMS terminates abnormally.

**Programmer response:** The IMODULE LOAD macro returned a nonzero return code in register 15. For an


explanation of the return code, see the IMODULE LOAD return codes in the IMS system services return codes information.

**Problem determination:** 1, 4, 12, 35

**Source:** IMS abends

**Module:** DFSCST00

**Related reference:**

 [IMS system services return codes \(Messages and Codes\)](#)

**Related information:**

Chapter 2, "IMS problem determination," on page 13

## 0749

**Explanation:** When a user abend code 0 has been issued by an application program, this abend appears. Additional information can be obtained from your system console log. If terminate thread processing is done by the end-of-task or end-of-storage instead of the IMS STAE processing, abend 0749 can occur. The IMS (E)STAE either did not get control or terminated abnormally before completing TERMINATE-THREAD processing.

**Analysis:** This is a standard abend issued by modules DFSASK00 and DFSPCC20.

**System action:** IMS uses abend 0749 as the completion code for IMS messages and log records.

**Programmer response:** If abend 0749 completion is desired, use abend code 0. If abend 0749 is not desired, do not use abend code 0.

**Problem determination:** 1, 5, 6, 11

**Source:** IMS abends

**Module:** DFSPCC20, DFSASK00, DFSSDA20, DFSSABN0, DFSBCK00

**Related information:**

Chapter 2, "IMS problem determination," on page 13



---

## Chapter 17. IMS abend codes 0751 - 0800

This topic identifies and explains the following abend codes issued by IMS. Each code is 4 decimal digits.

---

### 0756

**Explanation:** In an XRF environment, an OPEN failure was detected for a local message queue data set, or DFSQDOC0 failed to load.

IMS also issues this abend if the size of the local message queue is too small. This occurs when the number of records in any of the local message queue data sets (IMS.LGMSGL, IMS.SHMSGL, and IMS.QBLKSL) is less than the number of records specified in the SHUTDWN parameter in the MSGQUEUE macro defined during system definition.

**Analysis:** 0756 is a standard abend issued by module DFSCST00 for one of two causes.

If the IMODULE LOAD failed for DFSQDOC0, the IMODULE macro places a return code in register 15. For an explanation of the return code, see the IMODULE LOAD return codes in the IMS system services return codes information.

If an OPEN failure occurred for one or more of the local message queue data sets, DFSQDOC0 places a return code in register 15.

Register 3 points to the following information used by DFSCNS00:

Reg3+X'00'=address of AWE (X'30' bytes)  
Reg3+X'0C'=01 FUNC=OPEN  
Reg3+X'30'=DCB open list. (Each of the local message queue data sets has one entry in the list in the following format.)

- +0 DCB address
- +4 Reserved
- +5 Format option=01
- +6 Access is OSAM=01
- +7 Return code from OSAM\*

For a description of the OSAM return codes, see message DFS986A.

Key	Label	Description
Reg5=BAL to ABENDU0756 Reg15=Return code from IMODULE LOAD for DFSQDOC0	CST0003	DFSQDOC0 failed to load.
Reg3=address of AWE containing open list passed to DFSCNS00 Reg5=BALR to ABENDU0756 Reg15=Return code from DFSCNS00	ABENDU0756	One of the local message queue data sets failed to open.

**System action:** IMS terminates abnormally.

**Programmer response:** This is probably a user error. Check LRECL and BLOCKSIZE definitions for the local message queue data sets: IMS.LGMSGL, IMS.SHMSGL, and IMS.QBLKSL.


**Problem determination:** 4, 35

**Source:** IMS abends

## 0757

**Module:** DFSCST00

**Related reference:**

 IMS system services return codes (Messages and Codes)

**Related information:**

 DFS986A

Chapter 2, "IMS problem determination," on page 13

---

### 0757

**Explanation:** An error condition was detected in the IMS system message Queue Manager component. It was probably caused by a logic error in one of the Queue Manager callers, by a logic error within one of the Queue Manager routines, or by an operations error relating to an emergency restart situation. The following operational errors can lead to this abend:

- Warm starting after changing the defined system
- Attempting an emergency restart with the wrong, or a missing, log data set
- Using an improperly terminated log data set for an emergency restart

**Analysis:** Abend 0757 is a standard abend issued by the Queue Manager from any of the modules listed below. Use the address in the program status word (PSW) at entry-to-abend to determine which module detected the error condition. Register 15 in the abend SVRB has a subcode to help determine which module issued 0757. Register 6 normally contains an address that indicates approximately where the error was detected. The contents of registers 15, 0, and 1 at the time the error was detected are saved in the QSAPWKAD area (fields QSAPAB15, QSAPABR0, and QSAPABR1) before the abend is issued. The subcodes are cross-referenced in the next section and then described in detail in the subsection for each module.

**For DSECT usage:**

The following macros can be viewed to find the offset within a DSECT to a particular label as referenced within a module: QSAPWKAD, ISCD, ICLI, QLOGMSGP, IDLI, QBALSTD, QLOGMSGP, IAPS, OSAMDECB, QLOGRECS, and DFSQMGR.

**Subcode cross-reference:**

#### Subcode

	Modules
000	DFSQDQ00 DFSQEQ00 DFSQGU00 DFSQIS00 DFSQMGR0 DFSQNP00 DFSQRH00 DFSQRL00 DFSQXF00
001	DFSQMGR0
002	DFSQMGR0
003	DFSQC070
004	DFSQEQ00 DFSQGU00
005	DFSQEQ00
006	DFSQIS00
007	DFSQDQ00
008	DFSQDQ00 DFSQEQ00 DFSQMGR0
009	DFSQGU00
00A	DFSQIS00
00B	DFSQMGR0
00C	DFSQIS00
00D	DFSQNP00
00E	DFSQDQ00
00F	DFSQIS00 DFSQGU00
010	DFSQIS00 DFSQEQ00
011	DFSQEQ00
012	DFSQEQ00
013	DFSQEQ00
014	DFSQEQ00
015	DFSQEQ00
016	DFSQC060
017	DFSQNP00
018	DFSQGU00 DFSQNP00
019	DFSQGU00 DFSQNP00



01A	DFSQNP00
01B	DFSQNP00
01C	DFSQMGR0
01D	DFSQGU00
01E	DFSQGU00
01F	DFSQGU00
020	DFSQMGR0
021	DFSQGU00
022	DFSQGU00
023	DFSQGU00
024	DFSQGU00
025	DFSQGU00
026	DFSQGU00
027	DFSQXF00
028	DFSQXF00
029	DFSQXF00
02A	DFSQXF00
02B	DFSQEQ00 DFSQMGR0
02C	DFSQEQ00
02D	DFSQGU00
02E	DFSQRL00
02F	DFSQRL00
030	DFSQC020
031	DFSQC020
032	DFSQC020
033	DFSQEQ00
034	DFSQEQ00
035	DFSQEQ00 DFSQMGR0
036	DFSQDQ00
037	DFSQDQ00
038	DFSQDQ00
039	DFSQDQ00
03A	DFSQDQ00
03B	DFSQDQ00
03C	DFSQDQ00
03D	DFSQDQ00 DFSQMGR0 DFSQGU00
03E	DFSQIS00
03F	DFSQC040
040	DFSQGU00
041	DFSQGU00 DFSQEQ00
042	DFSQGU00
043	DFSQGU00
044	DFSQGU00
045	DFSQDQ00
046	DFSQEQ00 DFSQIS00
047	DFSQIS00
048	DFSQGU00 DFSQMGR0
049	DFSQGU00
04A	DFSQRL00 DFSQXF00
04B	DFSQMGR0
04C	DFSQC010 DFSQC020 DFSQC050 DFSQEQ00 DFSQGU00 DFSQIS00 DFSQMGR0 DFSQXF00
04D	DFSQC030 DFSQC060 DFSQC080 DFSQMGR0 DFSQXF00
04E	DFSQGU00
04F	DFSQGU00
050	DFSQDQ00 DFSQEQ00 DFSQGU00 DFSQMGR0 DFSQRL00 DFSQXF00
051	DFSQRL00
052	DFSQC030
053	DFSQC030
054	DFSQDQ00
055	DFSQEQ00
056	DFSQEQ00

**0757**

057	DFSQDQ00
058	DFSQGU00
059	DFSQNP00
05A	DFSQNP00
05B	DFSQC010
05C	DFSQC010
05D	DFSQC010
05E	DFSQNP00
05F	DFSQXF00
060	DFSQDQ00, DFSQEQ00, DFSQGU00, DFSQIS00, DFSQRL00
061	DFSQGU00
063	DFSQGU00
064	DFSQXF00
065	DFSQRL00
066	DFSQXF00
067	DFSQDQ00
101	DFSQBFM0
102	DFSQBFM0
104	DFSQBFM0
105	DFSQBFM0
205	DFSQRST0, DFSQRSQ0
206	DFSQRST0
209	DFSQRST0, DFSQRSQ0
20A	DFSQRST0, DFSQRSQ0
20B	DFSQRST0, DFSQRSQ0
20C	DFSQRST0, DFSQRSQ0
20D	DFSQRST0, DFSQRSQ0
20E	DFSQRST0
210	DFSQRST0
211	DFSQRST0
212	DFSQRST0
213	DFSQRST0
214	DFSQRST0
215	DFSQRST0, DFSQRSQ0
216	DFSQRST0
217	DFSQRST0
218	DFSQRST0
219	DFSQRST0
21A	DFSQRST0
21B	DFSQRST0
21C	DFSQRST0, DFSQRSQ0
21D	DFSQRST0
21E	DFSQRST0
21F	DFSQRST0, DFSQRSQ0
301	DFSQLOG0
306	DFSQLOG0
309	DFSQLOG0
30A	DFSQLOG0
30B	DFSQLOG0
30C	DFSQLOG0
30D	DFSQLOG0
30E	DFSQLOG0
30F	DFSQLOG0
310	DFSQLOG0
311	DFSQLOG0
312	DFSQLOG0
313	DFSQLOG0
314	DFSQLOG0
315	DFSQLOG0
316	DFSQLOG0
801	DFSQFIX0

802 DFSQFIX0  
 803 DFSQFIX0  
 804 DFSQFIX0  
 805 DFSQFIX0  
 806 DFSQFIX0  
 807 DFSQFIX0  
 901 DFSQRH00

**For DFSQC010:**

DFSQC010 assigns a relative record number to a message during an insert request. Register 5 points to the QSAPWKAD area.

SC	Key Data	Label	Description
04C	Reg15=X'4C' QSAPAB15=DFSBCB return code	ASSIGN70 ASSIGN90	An attempt to obtain an MSGP type message block failed.
05B	Reg15=X'5B' Reg11=A(SCD) SCDFRB=0 SCDSTOP1 bit: SCDSTQUE off	ABENDQU	During a continuation insert call, the Queue manager normal message queues were unavailable and IMS was not XRF capable.
05C	Reg15=X'5C' QSAPCFUN bit: DCDLI on	NWASSIGN	During a continuation insert or ENQUEUE call, the caller was DL/I or RESTART.
05D	Reg15=X'5D' QSAPAB15=A(CNT) CNTFLG2 bits: CNTF2CNT off OR CNTF2LQU off	NWASGN20	During a continuation insert or ENQUEUE call, the local QPOOL was not in use when trying to assign a DRRN.

**For DFSQC020:**

The DFSQC020 module contains three entry points (DFSQC020, DFSQC021, DFSQC022), two of which (DFSQC020, DFSQC021) issue reason codes. DFSQC020 frees a record (DRRN) in a queue buffer. DFSQC021 frees a chain of messages and produces the necessary "free" log records to allow for an emergency restart. Register 5 points to the QSAPWKAD area.

SC	Key Data	Label	Description
030	Reg15=X'30' Reg10=A(first of message) MSGFLAGS bit: MSGFFRST off	FREEM10	The record returned was not the first record of the message to be freed.
031	Reg15=X'31' Reg4=A(record DRRN that was freed) Reg10=A(returned message) MSGMDRRN=QSAPFRFR	FREEM30	While freeing the second through last DRRN of a message, the message DRRN of the returned message (MSGMDRRN) was not equal to the DRRN of the first record (QSAPFRFR).
031	Reg15=X'31' Reg4=DRRN of freed record Reg10=A(returned message) MSGFLAGS bit: MSGFFRST on	FREEM33	While freeing the second through last DRRN of a message, the message returned was the first record of the message.
032	Reg15=X'32' QSAPFRLM=QSAPFRFR or 0 QSAPFRLM0=0	FREEM60	An invalid termination to the message chain was detected after freeing each message.
04C	Reg15=X'4C' QSAPAB15=DFSBCB return code	QSMFRE25	An attempt to obtain an MSGP type message block failed.

**For DFSQC030:**

The DFSQC030 module contains four entry points (DFSQC030, DFSQC031, DFSQC032, DFSQC033), all of which issue reason codes. DFSQC030 gets the Queue Manager generic latch. If the latch cannot be obtained, the Latch Manager terminates abnormally. DFSQC031 releases the Queue Manager latch. DFSQC032 gets the QBFM steal generic latch. DFSQC033 releases the QBFM steal generic latch. Register 5 points to the QSAPWKAD area.

SC	Key Data	Label	Description
04D	Reg15=X'4D' QSAPAB15=DFSCLM return code	QC310070	An attempt to release the Queue Manager latch failed with QSAPAB15 containing the Latch Manager return code.
		QC310080	
		QC310090	
		QC330005	
		QC330010	
052	Reg15=X'52' Reg4=A(latch header)	QC33RCT	An invalid latch header address was detected.
		QC300005	
053	Reg15=X'53' Reg8=latch mode	QC310005	An invalid latch mode was requested. Valid modes are SHR or EXCL.
		QC300020	
		QC320005	

**For DFSQC040:**

DFSQC040 completes the processing of the prior inserted message segment.

SC	Key Data	Label	Description
03F	Reg15=X'3F' Reg10=A(message)	ENDIT20	The previous caller exceeded the size of the LRECL in the destination DCB.

**For DFSQC050:**

DFSQC050 moves the completed message to queue buffers in a non-shared-queues environment or to shared queues and logs it. Register 5 points to the QSAPWKAD area.

SC	Key Data	Label	Description
04C	Reg15=X'4C' QSAPAB15=DFSBCB return code	QCSH0900	An attempt to obtain a UOWE type block failed.
		QCSH6010	
		QLD0LOG	An attempt to obtain a QMBA type block failed.

**For DFSQC060:**

The DFSQC060 module contains two entry points (DFSQC060, DFSQC061), each of which issue reason codes. DFSQC060 provides serialization of the Queue Manager on a destination address. DFSQC061 removes the serialization provided by DFSQC060. Register 5 points to the QSAPWKAD area.

SC	Key Data	Label	Description
016	Reg15=X'16' Reg1=0	QXCDEQ20	There was no QCB to be released.
04D	Reg15=X'4D' QSAPAB15=DFSCLM return code	QXCENQ3	An attempt to release the DC system latch obtained by the Queue Manager failed.
		QXCDEQ	

**For DFSQC070:**

The DFSQC070 module contains two entry points (DFSQC070, DFSQC071), one of which (DFSQC071) issues a reason code. DFSQC071 Queue Manager decrement QPOOL no-purge counts processor. It decrements the QPOOL count of in-progress operations that must be completed before the checkpoint logging of the X'4001' record.

SC	Key Data	Label	Description
003	Reg15=X'03' Reg0<0 Reg0=QPNOPRG-1 Reg1=QPNOPRG Reg3=A(QPOOL)	QC710900	The count of activities for which the purge had to wait went negative.

**For DFSQC080:**

DFSQC080 is the Queue Manager SMB cleanup processor. It moves messages from the SMB suspend queue to its normal queue. Register 5 points to the QSAPWKAD area.

SC	Key Data	Label	Description
04D	Reg15=X'4D' QSAPAB15=DFSCLM return code	CLNQ0220	An attempt to release the DC system latch obtained by the Queue Manager failed.

**For DFSQBFM0:**

DFSQBFM0 accesses message queue records in the Queue Manager buffer pool. DECTYPE describes the type of request:

X'00' locate by DRRN  
 X'04' release (unalter)  
 X'08' locate and alter  
 X'0C' purge (write all altered blocks)  
 X'10' translate address to DRRN  
 X'14' decrement buffer intent count

SC	Key Data	Label	Description
101	Reg15=X'101' Reg9=A(DECBB) DECBBRN=Req type code	QBMLCT QBMLCTA QBMRDSU QBMSNID2	The 'D' portion of the DRRN (byte 0 of DECBBRN) does not equal X'00', X'04', X'08', or the in-storage X'88'.
101	Reg15=X'101' Reg1=0 Reg1=DECBBRN & X'00FFFFFF' Reg9=A(DECBB)	QBMLCTB QBMSNID3	After stripping the 'D' from the DRRN, the remaining 'RRN' portion was found to be zero.
102	Reg15=X'102' Reg3=(DECBTYP-1) & X'0000001F' Reg3>X'14' Reg9=A(DECBB)	QBMTTYP	An invalid type request code was detected. A list of the valid type request codes can be found in the DFSQBFM0 analysis section.
104	Reg15=X'104' Reg3=(QBSTAT1 & X'0000FF00')/256-1 Reg3<0 Reg4=A(QBFFR) Reg14=QBSTAT1	QBINA104	A negative value was encountered while attempting to decrement the intent count.
105	Reg15=X'105' Reg2=DECBBRN Reg4=A(QBFFR)= A(input buffer) NOT(QBIDL<=DECBBRN<=QBIDH)	QBINT050 QBINT070	The input DRRN was not included in the specified buffer range.

**For DFSQDQ00:**

DFSQDQ00 is the Queue Manager DEQUEUE/SAVE/DELETE command processor. The QSAPXFUN field in the QSAPWKAD area pointed to by register 5 contains the function code:

X'04' dequeue request

X'05' save request  
X'07' delete request

SC	Key Data	Label	Description
000	Reg15=X'00' Reg8=A(QTPPCB) QTPRRN=0 or DRRN	DEQFRST DEQCONT	An invalid function was requested. Refer to the table following Chart 1 for a list of the valid function codes. If QTPRRN is zero then the call was the first of a series of calls. If it is nonzero then the call was a continuation of a series of calls.
007	Reg15=X'07' QSAPDQOP bit: QLDQOFNM off	DEQ40	A broken queue was found. There was no message following the first message on the queue.
007	Reg15=X'07' QSAPDQOP bit: QLDQOFNM off	DEQ44	A broken queue was found. A message that was neither the first nor the last on the queue was not followed by another.
007	Reg15=X'07' QSAPDQOP bit: QLDQOFNM on	DEQ46	A broken queue was found. A subsequent message was found following the last message on the queue.
008	Reg15=X'08'	DEQ86	The suspend queue was found to be broken. While searching for the message to be removed, the next message on the chain was not found.
008	Reg15=X'08' Reg10=A(message) MSGFLAGS bit: MSGFFRST off	DEQ12G DEQ183 DEQ50 DEQ86A DQMSC280	The queue was found to be broken when attempting to locate the next message in the chain. The message pointed to by register 10 is not the first record of the message.
008	Reg15=X'08' Reg7=A(QDEST) Reg10=A(message) MSGODSTN=QDNAME	DEQ50A DEQ86B	The queue was found to be broken when attempting to locate the next message in the chain. The log record destination name and the queue manager destination name did not match.
00E	Reg15=X'0E' Reg8=A(QTPPCB) QTPFLG1 bits: DCALR not off	DELETE	The caller of DELETE was not communications.
036	Reg15=X'36' QSAPAB15=message DRRN	DEL20	The DRRN of the message passed to delete was invalid.
037	Reg15=X'37' Reg10=A(message) MSGFLAGS bit: MSGFFRST off	DELETE20	This is not the first record of a message to be deleted.
038	Reg15=X'38' Reg10=A(message) MSGFLAGS bits: MSGFQNR not on	DELETE30	The QNR was not properly set-up. It should be all ones when deleting a message.
039	Reg15=X'39' Reg7=A(QDEST) Reg8=A(QTPPCB) QTPDST=A(destination) QTPQNR<1 OR QTPQNR>5 QDFLG2 bits: QDF2SMB not on	DELETE52 DELETE54	The destination was not an SMB and the QNR was not in the range of 1 through 5 inclusive.
03A	Reg15=X'3A' Reg7=A(QDEST) QDFLG1 bit: QDF1ERRL on	DELETE60	A queue was locked due to a read I/O error.

SC	Key Data	Label	Description
03B	Reg15=X'3B' Reg7=A(destination) Reg8=A(QTPPCB) QTPFLG1 bit: DCDLI on QDFLG2 bits: QDF2CNT on	DEQ12	The destination CNT was not a system CNT for a DL/I caller.
03B	Reg15=X'3B' Reg7=A(destination) Reg8=A(QTPPCB) QSAPSV4=DRRN set by DFSQGU00 QSAPSV4~=QTPRRN QTPFLG1 bit: DCDLI on QDFLG2 bits: QDF2CNT not on AND QDF2MDEL on	DEQ12E	While testing for an SMB destination for a DL/I caller, the DRRN set by DFSQGU00 was not the same as the DRRN being processed.
03B	Reg15=X'3B' Reg7=A(destination) Reg8=A(QTPPCB) Reg9=A(PST) PSTQIMSG=DRRN to be freed PSTQIMSG~=QTPRRN QTPFLG1 bit: DCDLI on QDFLG2 bits: QDF2CNT not on AND QDF2MDEL off	DEQ12F	While testing for an SMB destination for a DL/I caller, the DRRN that was to be freed was not the same as the DRRN being processed.
03B	Reg15=X'3B' Reg7=A(destination) Reg8=A(QTPPCB) Reg10=A(message) QTPFLG1 bit: DCDLI on QDFLG2 bits: QDF2CNT not on MSGDFLG3 bit: MSGF3ZLN off MSGFLAGS bit: MSGFNRQU off	DEQ12H	The message for the SMB destination was not an APPC zero length message.
03C	Reg15=X'3C' Reg8=A(QTPPCB) QTPFRRN~=QTPRRN	DEQQFIVE	The message to be removed was not a single message for DEQ, SAVE, or DELETE from the back-up queue.
03D	Reg15=X'3D'	DEQ150	The message to be deleted was not found during a DELETE call. The message could have been on the wrong queue or not on the queue at all. Check all references to DEQOFFQ.
045	Reg15=X'45' Reg7=A(QDEST) Reg10=A(message) QDFLG2 bits: QDF2CNT on	DEQ13B DQMSC100	During a DEQUEUE call the RACF segment item was not in the prefix although the prefix flag indicates that it exists.
045	Reg15=X'45' Reg7=A(QDest) Reg10=A(Message)	DQMSC460	During a DEQUEUE call the system segment item was not in the prefix although the prefix flag indicates that it exists.

SC	Key Data	Label	Description
050	Reg15=X'50' QSAPAB15=DFSBCB return code	DEQRETRN	An attempt to release the Queue Manager message buffer work area (QMBA) failed.
054	Reg15=X'54' Reg4=number of buffer intents	RETURN	Buffer intent was not released.
057	Reg15=X'57' Reg8=A(QTPPCB) QTPDST=A(destination)	DELETE50	The destination address was not fullword aligned.
060	Reg15=X'060' Reg2=A(CCB) Reg5=A(QSAP) Reg10=A(message)	DEQ19A	The expected system extension segment, which contains the UTC time stamp, was not found.
067	Reg15=X'67' Reg7=A(QDest) Reg10=A(Message)	DQMSC160 DQMSC180 DQMSC200	The destination type or name did not match the destination type or name in the message.

**For DFSQEQ00:**

DFSQEQ00 is the Queue Manager ENQUEUE/REENQUEUE command processor. The QSAPXFUN field in the QSAPWKAD area pointed to by register 5 contains the function code:

X'0C' enqueue FIFO  
X'0D' enqueue LIFO  
X'0E' reenqueue FIFO  
X'0F' reenqueue LIFO  
X'2F' reenqueue LIFO from SMB suspend queue to SMB normal queue  
X'1C' conditional enqueue FIFO-TDEST  
X'1D' conditional enqueue LIFO-TDEST

SC	Key Data	Label	Description
000	Reg15=X'00' Reg8=A(QTPPCB) QTPRRN=0 or DRRN	QEQFIRST QEQCONT	An illegal function was requested. Refer to the table following Chart 1 for a list of valid function codes. If QTPRRN is zero, the call was the first of a series of calls. If nonzero, the call was a continuation of a series of calls.
004	Reg15=X'04' Reg8=A(QTPPCB) Reg10=A(message) MSGMDRRN=QTPFRRN	ENQ0100	The PCB was found to be incorrect for an ENQUEUE call. The message DRRN did not equal the DRRN of the first buffer of the message.
005	Reg15=X'05' Reg8=A(QTPPCB)	CKDS0010 CKDS0020	The destination was not on a fullword boundary or it was locked due to a read I/O.
008	Reg15=X'08' Reg6=A(error detected) Reg7=A(QDEST) Reg8=A(QTPPCB) Reg9=A(DECB) Reg10=A(message) Reg11=A(QSCD)	SQENQ100 SQENQ150 SQENQ200	The message returned by DFSQBFM0 (DRRN obtained from QTPFRRN) is not the correct message.
011	Reg15=X'11' Reg7=A(QDEST) QDFLG2 bits: QDF2SMB not on AND QDF2CNT not on	CKDS0040	The destination was not an SMB or a CNT.
012	Reg15=X'12' Reg8=A(QTPPCB) QTPQNR<1 OR QTPQNR>5	CKDS0060 CKDS0100	An invalid QNR was specified during an ENQUEUE call.



SC	Key Data	Label	Description
013	Reg15=X'13' Reg7=A(QDEST) QSAPABR1=A(prior message) MSGODSTN=QDNAME or MSGMDRRNQSA=PNQPM	CHNT0500 CHNT0530	A broken queue was found during a FIFO ENQUEUE request. Either the prior message destination did not equal the input destination or the DRRN in the prior message was invalid.
014	Reg15=X'14' Reg10=A(message) MSGFLAGS bits: MSGFQNR not off	ENQ0448	An invalid QNR was found for a normal enqueue. The QNR should have been zeros in the message.
014	Reg15=X'14' Reg10=A(message) MSGFLAGS bits: MSGFQNR not on	RENQ0448	An invalid QNR was found for a normal reenqueue. The QNR should have been ones.
015	Reg15=X'15' Reg8=A(QTPPCB) QTPFLG1=QMRNQFCM	CKDS0110	The caller requesting the REENQUEUE should have been a simple communications call.
02B	Reg15=X'2B' QSAPABR0=A(DEQ chain) QSAPABR1=A(ENQ chain)	GFTD0260 GFTD0400 GFPD0400 GFPD0405	Both the dequeue chain and the enqueue chain should have been zero, but one of them was nonzero.
02B	Reg15=X'2B' QSAPABR0=A(ENQ chain) QSAPABR1=A(DEQ chain)	GFTD0410 GFTD0415 GFPD0410 GFPD0415	Both the dequeue chain and the enqueue chain should have been nonzero, but one of them was zero.
02C	Reg15=X'2C' Reg8=A(QTPPCB) Reg10=A(message) MSGMDRRN=QTPFRRN	ENQ0444 RENQ0444	The message accessed to be enqueued/reenqueued was not pointing to itself.
02C	Reg15=X'2C' Reg10=A(message) MSGFLAGS bit: MSGFFRST off	ENQ0446 RENQ0446	The message accessed to be enqueued/reenqueued was not the first record of a message.
033	Reg15=X'33' Reg7=A(QDEST) QDFLG1 bit: QDF1BKR on	RENQ0150	During a REENQUEUE to a backup queue call, there was no message although QDFLG1 indicated that there was.
033	Reg15=X'33' Reg7=A(QDEST) QDFLG1 bit:5=0 QDF1BKR off	RENQ0160	During a REENQUEUE to a backup queue call, there was a message although QDFLG1 indicated that there was none.
034	Reg15=X'34' Reg8=A(QTPPCB) QTPFLG1 bits: QLNQCREQ on OR DCDLI off	GFTD0090 GFTD0100	A REENQUEUE call was made to a queue other than the backup queue, or DL/I was not the caller of the ENQUEUE request.
035	Reg15=X'35' Reg7=A(QDEST)	GFPD0320	A broken queue was found in the CNT chain. When no QBLK is assigned then there should not be any messages on any of the queues.

SC	Key Data	Label	Description
041	Reg15=X'41' Reg6=A(error detected) Reg7=A(QDEST) Reg8=A(QTPPCB) Reg9=A(DECBC) Reg10=A(message) Reg11=A(QSCD)	SQENQ450	The message pointed to by register 10 does not contain the required transaction management router segment item X'8C'.
046	Reg15=X'46' Reg10=A(message) MSGCFLG2 bit: MSGC2MSC off	CONV4600	During an ENQUEUE call, the MSC segment item was not in the prefix.
046	Reg15=X'46' Reg10=A(message) MSGCFLG2 bit: MSGC2MSC on	CONV46ER	During an ENQUEUE call, the MSC segment item was not in the prefix although the prefix flag indicated its presence.
04C	Reg15=X'4C' Reg5=A(QSAPWKAD) Reg6=A(error detected) Reg7=A(QDEST) Reg8=A(QTPPCB) Reg9=A(DECBC) Reg11=A(QSCD)	SQENQ300	A DFSBCB request for a UOWE block failed. The return code is in QSAPAB15. Refer to BCBRC in the FSBCB macro for return codes.
050	Reg15=X'50' QSAPAB15=DFSBCB return code	RETORD10	An attempt to release the Queue Manager message buffer work area (QMBA) failed.
055	Reg15=X'55' Reg8=A(QTPPCB) QTPFLG1 bit: QLNQCTMP on	REENQ	A temporary destination is invalid during a REENQUEUE function.
056	Reg15=X'56' Reg8=A(QTPPCB) QTPRRN=0 QTPFLG1 bit: DCDLI off	RENQ0050	A communications REENQUEUE call is invalid if a current call is active.
060	Reg15=X'060' Reg5=A(log record) Reg10=A(message)	ENQ0490	The expected system extension segment, which contains the UTC time stamp, was not found.
060	Reg15=X'060' Reg8=A(QTPPCB) Reg9=A(CLB) Reg10=A(message)	RENQ0513	The expected system extension segment, which contains the UTC time stamp, was not found.
060	Reg15=X'060' Reg3=A(CCB) Reg5=A(QSAP) Reg10=A(message)	CONV4625 CONV4820	The expected system extension segment, which contains the UTC time stamp, was not found.

**For DFSQFIX0:**

DFSQFIX0 ensures that the queue data sets and chains are valid. It inspects the queues and removes or corrects any invalid records. Register 14 points to the QSAPWKAD area. In most cases, a 0757 abend in DFSQFIX0 during a restart of IMS requires an /ERE COLDCOMM or a cold-start of IMS to recover.

SC	Key Data	Label	Description
801	Reg15=X'801' Reg10=A(message) QSAPAB15=CHKRCD return code	MCHNCK4	A unrecoverable message was detected while checking the record of a message.

SC	Key Data	Label	Description
802	Reg15=X'802' Reg10=A(message) MSGCFLG2 bit: MSGC2APP on	NOAPPC	While checking a message record, the APPC system segment was not found although the flag indicated its presence.
803	Reg15=X'803' QSAPAB15=SCDLSCD value	LOGDST10	When attempting to log an X'3C' log record, the block was not an SMB, CNT, LNB, QAB, or TIB block.
804	Reg15=X'804' QSAPAB15=DFSLUMIF return code	LOGDST26	Unable to obtain an LUP token when attempting to log an X'3C' log record for an APPC TIB block.
805	Reg15=X'805' Reg10=A(message) QSAPAB15=DFSLUMIF return code	CHKRCD1E	The destinations did not match when checking an APPC message record.
806	Reg15=X'806' QSAPAB15=DFSLUMIF return code	LOGDST46	Unable to obtain the LU, SIDE, and TP names when attempting to log an X'3C' log record for an APPC QAB block.
807	Reg15=X'807' Reg2=A(CCB) Reg10=A(message)	MCHNCK1C	The expected system extension segment, which contains the UTC time stamp, was not found.
807	Reg15=X'807' Reg7=A(CCB) Reg10=A(message)	CKCCB3	The expected system extension segment, which contains the UTC time stamp, was not found.

**For DFSQGU00:**

DFSQGU00 is the Queue Manager GET command processor. It processes two types of GET commands: UNIQUE, NEXT. The QSAPXFUN field in the QSAPWKAD area pointed to by register 5 contains the function code:

X'02' get unique

X'03' get next

SC	Key Data	Label	Description
000	Reg15=X'00' Reg8=A(QTPPCB) QTPRRN=0 or DRRN	QGUFIRST QGUCONT_10 QMGGNT DLIGU	An illegal function was requested. Refer to the table following Chart 1 for a list of the valid function codes. If QTPRRN is zero, the call was the first of a series of calls. If nonzero, the call was a continuation of a series of calls. For a continuation call, QTPQMBA cannot be zero.
004	Reg15=X'04' Reg10=A(message) MSGMDRRN=QTPFRRN	BYP62ENQ	The PCB was found to be incorrect for an ENQUEUE call. The message DRRN did not equal the DRRN of the first buffer of the message.
009	Reg15=X'09' Reg8=A(QTPPCB) QTPFLG3 bit: QTPQCOMP on QTPFLG1 bit: DCDLI off	QGUAB09	During a GU continuation call, the caller was not DL/I or RESTART.
00F	Reg10=A(message) Reg8=A(QTPPCB)	BSHR1270 BSHR1290 BSHR1280	The message chain is broken.
018	Reg15=X'18'	QMGGN10	The segment length was invalid for an input record during a GN call.

SC	Key Data	Label	Description
018	Reg15=X'18' Reg10=A(message) MSGFLAGS bit: MSGFLAST on	GUGNSEG1	During a GN call, the segment length was invalid for an input record.
019	Reg15=X'19' Reg10=A(message) MSGFLAGS bit: MSGFFRST on	GUGN100 MGN0070	While trying to get the next record during a GN call, the record that was returned was the first record.
01D	Reg15=X'1D'	GU11 GU20 GU61 BY60520	The destination address was not fullword aligned.
01E	Reg15=X'1E' Reg7=A(QDEST) SMBFLAG2 bit: SMB2RMT off	GU12	Communications issued a GU to a local SMB.
01F	Reg15=X'1F' Reg7=A(QDEST)	GU14A	An invalid QNR number was supplied for a GU call.
021	Reg15=X'21' Reg7=A(QDEST) Reg8=A(QTPPCB) QTPFLG1 bit: DCDLI on QTPFLG2 bits: QDF2SMB not on	GU61A	During a GU call, the destination was not the address of an SMB.
021	Reg15=X'21' Reg7=A(QDEST) SMBFLAG2 bit: SMB2RMT on	GU61B	During a GU call, the destination SMB was remote.
022	Reg15=X'22' Reg4=A(buffer prefix) Reg8=A(QTPPCB) Reg10=A(message) QTPFLG1 bit: DCDLI off MSGFLAGS bit: MSGFFRST off	GU21	A broken queue was found. The next message did not start with the first record of the chain during a GU call.
023	Reg15=X'23' Reg10=A(message) MSGFLAGS bit: MSGFFRST off	GU65	A broken queue was found during a DL/I, RESTART, or GU call. When the message was accessed to log the GU call, the first record to be copied was not the first record of the message.
024	Reg15=X'24' Reg10=A(message)	GUCSWEND GU10_1D GU10_3D GU67	The chain pointers for a message were broken or failed to end correctly.
025	Reg15=X'25' Reg7=A(QDEST) Reg8=A(QTPPCB) Reg10=A(message) MSGODSTN=QDNAME QTPOFST bit: QLGUFOPD on	GU81 RELOC_40 RELOCABD	The destination was on the wrong queue.

SC	Key Data	Label	Description
025	Reg15=X'25' Reg7=A(QDest) Reg8=A(QTPPCB) Reg10=A(Message) MSGODSTN=QDNAME	MGUCM460	Destination name does not match name in the message.
026	Reg15=X'26' Reg10=A(message) Reg8=A(QTPPCB)	GU82 GU83 MGUCM500 MBUCM520	The PCB and message QNRs were not equal.
02D	Reg15=X'2D' Reg10=A(message) MSGFLAGS bit: MSGFFRST off	GU18 GU23 SHR0020 SHRA100 SHRA120 SHRLKCLN MGUSQ200	During a GU call, the message that was returned was not the first record.
03D	Reg7=A(QDEST)	SHR4120 SHR0007	During a GU call with LOCAL=yes option, a QBLK was not assigned.
040	Reg15=X'40'	QMGGNT05 MGN0030	A GN call was issued after the prior GU call failed to find a message.
041	Reg15=X'41' Reg10=A(message)	GU30A1 GU32A GU32C GU64_A1 GU72_005 GU72G GU72I GU79_A GU79A GU80AB42 DEL0185 DEL0200 MGUCM100 MGUCM160 MGUCM420	During a GU call, one of the following segment items was missing from the message prefix: APPC segment, TMR segment, conversation prefix segment, the extended prefix segment, the WLM prefix segment, or the security segment.
043	Reg15=X'43' Reg10=A(message)	GU66AB43 GU68AB43 BYP60310 GU75AB43 GU75AE43 SHRLK055	During a DL/I GU call, the TMR segment item was not in the prefix.
044	Reg15=X'44' Reg10=A(message)	GU80AB44 MGUCM300	During a GU call, the RACF segment item was not in the prefix although the prefix flag indicated its presence.
048	Reg15=X'48' Reg10=A(message)	GU80AB48 MGUCM380	During a GU call, the ISC segment item was not in the prefix although the prefix flag indicated its presence.
049	Reg15=X'49' Reg8=A(QTPPCB) QTPFLG3 bit: QTPQCOMP on	QMGGUF	During the first GU call of a series, the message needed to be completed and logged before the GU call could be processed.

SC	Key Data	Label	Description
04C	Reg15=X'4C' QSAPAB15=DFSBCB return code	DFSQGU09 QGU09_300 PREFUPDT SHR0050 SHR0110 REC_BUF REC_BUF200 QLD0LOG CQS_DEL DEL0170 SHRLK045	An attempt to obtain a QMBA, a UOWE, or a QLST failed.
04E	Reg15=X'4E' QSAPIFL1 bit: QSAP1PRM off	GU30B GU64A GU74B GUGNEXIT MGUCM660	The parameter list interface was not used to communicate with the Queue Manager.
04F	Reg15=X'4F'	GU16	A GU call was issued without a destination and the supplied DRRN in the QSAPAREA field was invalid.
050		QGU09_200 QGU09_400	An attempt to release the Queue Manager message buffer work area (QMBA) failed during a GU call.
058	Reg15=X'50' QSAPAB15=DFSBCB return code	QGUCONT1 QMGGUT DLIGU010 BYP60500	The destination address was zero during a GU call.
060	Reg15=X'58' Reg8=A(QTPPCB) QTPDST=0	GU790CA	The expected system extension segment, which contains the UTC time stamp, was not found.
061	Reg15=X'061' Reg6=A(ABEND) Reg7=A(QDEST) Reg8=A(QTPPCB) Reg9=A(DECB) Reg10=A(message) Reg11=A(QSCD)	SHR1150 SHR1250 BSHR1235 BSHR1250	The message record on CQS is larger than the size of the Large Message Queue buffer.
063	Reg15=X'063' Reg6=A(ABEND) Reg7=A(QDEST) Reg8=A(QTPPCB) Reg9=A(DECB) Reg11=A(QSCD)	QGUCONT4 BYPRTN BYP000 CONTRTN BYP60320 GU10_1A GU61F SHRQRTN_DC GU60 GUISRTN BYPQRTN MGUSQ100	The return code value returned from a called routine is invalid.

**For DFSQIS00:**

DFSQIS00 is the Queue Manager INSERT command processor. It processes four types of INSERT commands: LOCATE, PREFIX, MOVE, MOVE(spannable). The QSAPXFUN field in the QSAPWKAD area pointed to by register 5 contains the function code:

X'0A' insert locate  
X'0B' insert move  
X'15' error exit - message reroute  
X'1A' insert prefix (update a field in the prefix)  
X'1B' insert move (spannable call)

SC	Key Data	Label	Description
000	Reg15=X'00' Reg8=A(QTPPCB) QTPRRN=0 or DRRN	QIS1000 QISCONT	An illegal function was requested. Refer to the table following Chart 1 for a list of the valid function codes. If QTPRRN is zero, the call was the first of a series of calls. If nonzero, the call was a continuation of a series of calls.
006	Reg15=X'06' Reg1=a (message text) MSGXFLG1 bit: MSGX1FST off	ISMFI0J	Is not the first segment of the message text.
006	Reg15=X'06' Reg3=Needed record size Reg4=Extended prefix length	ISIEPHAL	Message prefix is too large to fit in a message record.
006	Reg15=X'06' Reg3=Length of remaining message text Reg8=A(QTPPCB) QTPFLG1 bit: FSPC on	ISMF60	The previous caller exceeded the destination DCB LRECL.
006	Reg15=X'06' Reg3=Length of remaining message text Reg10=A(message)	ISM30	During an INSERT-MOVE call, the previous caller exceeded the destination DCB LRECL.
00A	Reg15=X'0A' Reg11=A(SCD) SCDSSSTYP bit: SCDSSDBC on	PREFO20	During an INSERT-PREFIX call, an UPDATE FORMAT-NAME was requested from the DBCTL system. This request is invalid in the DBCTL system because there is no MFS.
00C	Reg15=X'0C' QSAPPFUN bits: QMGOPIL not on	ISMCONTS	The call that was being processed was an INSERT-MOVE SPANNABLE, but the previous call was NOT an INSERT-LOCATE.
00F	Reg15=X'0F' Reg8=A(QTPPCB) Reg10=A(message) QTPRRN=new DRRN MSGFLG1 bit: MSGFLAST on	ISICMP	Invalid continued message segment. The last DRRN of the message was encountered.
00F	Reg15=X'0F' Reg8=A(QTPPCB) Reg10=A(message) QTPRRN=new DRRN MSGRDRRN=old DRRN	ISICMP10	Invalid continued message segment. The record DRRN does not equal the old DRRN.
00F	Reg15=X'0F' Reg8=A(QTPPCB) Reg10=A(message) QTPRRN=new DRRN MSGRDRRN=DRRN of 1st buffer	ISICMP20	Invalid continued message segment. The message DRRN does not equal the DRRN of the first buffer (QTPFRRN).
010	Reg15=X'10' Reg8=A(QTPPCB) QTPRRN=0	QIS0900	An illegal INSERT-PREFIX request was made.

SC	Key Data	Label	Description
03E	Reg15=X'3E' Reg11=A(SCD) SCDPRDEF bit: SCDPDMUL off	PREF021	An UPDATE MSC FLAGS was requested during an INSERT-PREFIX call, but the MSC was not in the system.
046	Reg15=X'46'	PREF021C	Invalid system segment for MSC during processing of the UPDATE MSC FLAGS request during an INSERT-PREFIX call.
046	Reg15=X'46'	PREF022A	Invalid system segment for MSC during processing of the UPDATE RESPONSE MODE INDICATOR request during an INSERT-PREFIX call.
047	Reg15=X'47' Reg10=A(message) MSGCFLG2 bit: MSGC2APP off	PREF029	Invalid system segment for APPC (segment item does not exist) during processing of the REPLACE APPC MESSAGE PREFIX request during an INSERT-PREFIX call.
047	Reg15=X'47' Reg10=A(message) MSGCFLG2 bit: MSGC2APP on	PREF029B	Invalid system segment for APPC during processing of the REPLACE APPC MESSAGE PREFIX request during an INSERT-PREFIX call.
047	Reg15=X'47' QSAPABR2=new APPC length	PREF029C	Invalid system segment for APPC (new APPC length does not equal the segment item length) during processing of the REPLACE APPC MESSAGE PREFIX request during an INSERT-PREFIX call.
04C	Reg15=X'4C' QSAPAB15=DFSBCB return code	ISLF10 ISMF10	An attempt to obtain a Queue Manager message buffer work area (QMBA) failed.
060	Reg15=X'060' Reg5=A(QSAP) Reg10=A(message)	PREF028D	The expected system extension segment, which contains the UTC time stamp, was not found.

**For DFSQLOG0:**

DFSQLOG0 exists as part of the IMS nucleus and is logically an extension of the Queue Manager module. It performs two functions:

1. It builds message prefixes and logs message records as messages are inserted to the IMS message queue (01 and 03 records).
2. It builds and logs the other log records as needed by DFSQMGR0.

R0 on entry contains the log code:

X'00' first record of message  
X'01' non-first record of message  
X'02' reject DL/I input  
X'03' transfer with QBLK  
X'04' transfer without QBLK (no logging done)  
X'05' DL/I message GU  
X'06' non-DL/I message GU  
X'07' free with more chain to come  
X'08' free with no more message segment  
X'09' release input without any input msg (no logging done)  
X'0A' release input message  
X'0B' release output message (free output queue)  
X'0C' enqueue or re-enqueue of message  
X'0D' DEQ/SAVE/DELETE from queue  
X'0E' DEQ/SAVE/DELETE from queue 5  
X'0F' cancel message



**X'10'** message prefix information  
**X'11'** release output msg from TDEST at RELB from ROLS  
**X'12'** update message DRRN at RELB from ROLS  
**X'13'** update QBLK at RELB from ROLS  
**X'14'** end of release input message (REL/ROLB or ROLS)  
**X'15'** end of REL/ROLS

SC	Key Data	Label	Description
301	Reg15=X'301' Reg1=log code * 2	QLOG0100	An invalid log code was detected. Refer to the DFSQLOG0 analysis section for a list of the valid codes.
306	Reg15=X'306' Reg14=CTBCNTPT=A(source CNT) CNTFLG2 bits: CNTF2CNT not on	MSG235	Invalid CNT offset for input terminal CTBCNTPT.
309	Reg15=X'309' Reg8=A(QTPPCB) QTPDST=0	MSG320 MSG520A1 MSG522	The output destination offset was not set.
30A	Reg15=X'30A' Reg1=A(QDEST) QDFLG2 bit: QDF2PRM off	MSG523	The message was not for a permanent destination.
30B	Reg15=X'30B' Reg1=A(QDEST) QDFLG2 bits: QDF2CNT not on	MSG520A	The destination CNT offset was invalid.
30B	Reg15=X'30B' Reg1=A(QDEST) QDFLG2 bits: QDF2SMB not on	MSG525	The destination SMB offset was invalid.
30C	Reg15=X'30C' Reg4=A(log record) MSGFLAGS bit: MSGFFRST off	MSG528	The record to be logged was not the first segment of the message when trying to build the prefix.
30C	Reg15=X'30C' Reg3=A(message) MSGFLAGS bit: MSGFFRST off	DEQ90 XFER90	The record returned was not the first record of the message to be dequeued or transferred.
30D	Reg15=X'30D' Reg4=A(log record) MSGFLAGS bit: MSGFFRST on	LMSGB	The record to be logged was the first record of the message when trying to build the prefix.
30E	Reg15=X'30E' Reg4=A(log record) MSGRDRRN or MSGMDRRN not equal to X'04' or X'08'	MSG600 MSG612	Either the record DRRN or the message DRRN was not equal to X'04' or X'08'.
30F	Reg15=X'30F' Reg4=A(log record) Reg5=A(QSAPWKAD) QSAPAB15=ILOG return code	MSG6216 QLOG93	The logical logger was unable to log either the short part of a record of a unrecoverable message or a complete record.
310	Reg15=X'310' Reg7=A(QDEST) QDFLG2 bits: QDF2SMB not on AND QDF2CNT not on	GU3	The destination was neither a CNT nor an SMB during a GU call.

SC	Key Data	Label	Description
310	Reg15=X'310' Reg4=A(QLGURCD) Reg11=A(SCD) QLGUFLGS bit: QLGUFDLI off SCDPRDEF bit: SCDPDMUL off	GU5	During a GU call with a communications SMB destination, the MSC was not in the system.
311	Reg15=X'311' Reg4=A(log record) MSGFLAGS bits: MSGFQNR not off	MSG614	A message queue number (QNR) was unexpectedly found in the message prefix (low order four bits of MSGFLAGS). DFSQLOG0 makes this check before setting the field from QTPQNR. The field should have been zero.
312	Reg15=X'312' Reg5=A(QSAPWKAD) QSAPAB15=A(message)	MSG199 MSG15A MSGI05	The MSC segment item was not in the message prefix while trying to build the prefix.
312	Reg15=X'312' Reg5=A(QSAPWKAD) QSAPAB15=A(message)	QLOG3010	The APPC segment item was not in the message prefix while trying to add the APPC prefix to a log record.
313	Reg15=X'313' Reg5=A(QSAPWKAD) QSAPAB15=DFSPPOOL return code	QLOG3020	Unable to obtain storage from HIOP to complete the log record.
314	Reg15=X'314' Reg3=A(message) MSGCFLG2 bit: MSGC2APP off	XFER91	The transaction was not an APPC transaction when trying to transfer a message.
315	Reg15=X'315' Reg5=A(QSAPWKAD) QSAPXFRB=A(destination QBLK)	XFERDQLP	While trying to transfer a message for an APPC transaction, there were no messages on any of QBLK five queues.
316	Reg15=X'316' Reg4=A(message)	LMSGA	The expected system extension segment, which contains the UTC time stamp, was not found.

**For DFSQMGR0:**

DFSQMGR0 determines what function has been requested and calls the appropriate module. Refer to the table following Chart 1 for a list of the valid function codes. Register 5 points to the QSAPWKAD area.

SC	Key Data	Label	Description
000	Reg15=X'00' Reg8=A(QTPPCB) QTPFLG1 bit: DCDLI on MSGDFLG3 bit: MSGF3NOE off	CANCI025	An invalid cancel input call was made by DL/I.
001	Reg15=X'01' QSAPPFUN bit: FOUT on QTPFLG3 bit: QTPQCOMP off	QMGENT45	The prior call was input and the current call is output. QSAPXFUN contains the requested function without the caller-ID and QSAPCFUN includes the caller-ID.
001	Reg15=X'01' QSAPPFUN bit: FOUT off QTPFLG1 bit: QMRNQFDL off	QMGENT4A	The prior call was output and the current call is input. QSAPXFUN contains the requested function without the caller-ID and QSAPCFUN includes the caller-ID.

SC	Key Data	Label	Description
002	Reg15=X'02' Reg8=A(QTPPCB) QTPFLG1 bit: FDN on	RPOS0500	A reposition call was made without a current message.
00B	Reg15=X'0B' Reg10=A(message) Reg11=A(SCD) SCDSSTYP bit: SCDSSDBC on	PREFI	Format services are not in the (DBCTL) system when the format name is to be retrieved.
01C	Reg15=X'1C' QTPFLG1 bit: DCDLI off	QMREJECT	The caller must be DL/I during a REJECT call.
020	Reg15=X'20' Reg8=A(QTPPCB) QTPFLG3 bit: QTPQMBP off	QMGENT1	The Queue Manager message buffer work area (QMBA) is not available.
02B	Reg15=X'2B' Reg10=A(QBLK) Reg0=Deq chain Reg1=Enq chain	MCINS100 MCINS120	Both the dequeue chain and the enqueue chain should have been zero, but one of them was not zero.
02B	Reg15=X'2B' Reg10=A(QBLK) Reg0=Deq chain Reg1=Enq chain	MCINS140 MCINS150	Both the dequeue chain and the enqueue chain should have been nonzero, but one of them was zero.
035	Reg15=X'35' Reg7=A(QDEST)	MCINS060	A broken queue was found in the CNT chain. When no QBLK is assigned then there should not be any messages on any of the queues.
03D	Reg15=X'3D' Reg6=A(ABEND) Reg7=A(QDEST) Reg8=A(QTPPCB) Reg9=A(DECB) Reg11=A(QSCD)	CANCI110 CANCI200	No QBLK record was assigned to the CNT for a CANCEL INPUT call.
048	Reg15=X'48' Reg10=A(message)	RPOSERR	During a GU call, the ISC segment item was not in the prefix even though the prefix flag indicated its presence.
04B	Reg15=X'4B' QSAPAB15=DFSQC080 return code	QMGENT38	IMS branches to the cleanup routine (DFSQC080). If cleanup was unsuccessful then a nonzero return code is placed in register 15.
04C	Reg15=X'4C' Reg3=Function code Reg8=A(QTPPCB) QSAPAB15=DFSBCB return code QSAPCFUN=Caller ID + Function code QTPRRN=0 or DRRN	QMGENTC	An attempt to reobtain the Queue Manager message buffer work area (QMBA) failed. If QTPRRN is zero, the call is the first in a series of calls. If nonzero, the call is a continuation of a series of calls.  This failure usually results from specifying an inadequate amount of virtual storage in the IMS control region "REGION=" JCL parameter.
04D	Reg15=X'4D' QSAPAB15=DFSCLM return code	QMGRATE	An attempt to release the QLOG acquired DC system latch failed.
050	Reg15=X'50' QSAPAB15=DFSBCB return code	RELQMBA	An attempt to release the Queue Manager message buffer work area (QMBA) failed.

SC	Key Data	Label	Description
008	Reg15=X'08' Reg10=A(Message) MSGFLAGS bit: MSGFFRST off	MCI0120	The record return was not the first record of the message to be canceled.

**For DFSQNP00:**

DFSQNP00 is the Queue Manager NOTE and POINT command processor. The QSAPXFUN field in the QSAPWKAD area pointed to by register 5 contains the function code:

X'1F' note

X'3F' point

SC	Key Data	Label	Description
000	Reg15=X'00'	QMGNP005	An illegal function was requested. Refer to the table following Chart 1 for a list of the valid function codes.
00D	Reg15=X'0D' Reg8=A(QTPPCB) QTPRRN=0	QMGNP010	A NOTE/POINT function request is invalid if the call is not a continuation call.
017	Reg15=X'17' QSAPCSV1=previous function code	QMGNP020	A NOTE/POINT function was requested with an invalid prior call. A NOTE/POINT call must follow a GU, GN, or DL/I ROLB call.
018	Reg15=X'18' Reg10=A(message) MSGFLAGS bit: MSGFLAST on	GUGNSEG1	The segment length was invalid for an input record.
019	Reg15=X'19' Reg10=A(message) MSGFLAGS bit: MSGFFRST on	GMGGNT4	The record returned was the first record while trying to get the next record.
01A	Reg15=X'1A' Reg8=A(QTPPCB) Reg10=A(message) QTPFRRN-=MSGMDRRN MSGFLAGS bit: MSGFFRST off	POINT10	During a POINT call, the record returned was neither the first record of the message nor part of the same message.
01A	Reg15=X'1A' Reg8=A(QTPPCB) Reg9=A(DECB) Reg10=A(message) QTPFRRN-=DECBRBN MSGFLAGS bit: MSGFFRST on	POINT20	During a POINT call, the record returned was the first record of the message but the record was not part of the same message.
01B	Reg15=X'1B' Reg10=A(message) MSGFLAGS bit: MSGFFRST off	POINT55	During a DL/I POINT call to reread the current message, the first record of a message was not obtained.
059	Reg15=X'59' Reg10=A(message)	POINT30	During a POINT call, the supplied record offset was beyond the record.
05A	Reg15=X'5A' Reg10=A(message)	POINT31	During a POINT call, the LL at the offset into the message being processed was not greater than four.
05E	Reg15=X'5E' Reg10=A(message)	POINT32	During a POINT call, the offset plus the LL pointed beyond the record.

**For DFSQRH00:**

DFSQRH00 is the new parameter list interface. The parameter list is described by the dsect DFSQMGR.

SC	Key Data	Label	Description
000	Reg15=X'000' Reg4=A(DFSQMGR) QMGRFUNC=function code	QRH24000 QRH25000 QRH26000	The function requested is invalid. Refer to the table following Chart 1 for a list of the valid function codes. The function code can be found in the parameter list pointed to by register 4. The DSECT that maps this parameter list is DFSQMGR.
901	Reg15=X'901' Reg5=A(QSAPWKAD) QSAPABR0=APPC prefix len QSAPABR1=A(APPC prefix)	QRH11000 QRH12000 QRH27000 QRH28000	The APPC prefix length is zero or negative. The length is found in the first halfword of the prefix pointed to by QSAPABR1.

**For DFSQRL00:**

DFSQRL00 is the Queue Manager RELEASE command processor. The QSAPXFUN field in the QSAPWKAD area pointed to by register 5 contains the function code:

X'16' release

SC	Key Data	Label	Description
000	Reg15=X'00'	REL10	An illegal function was requested. Refer to the table following Chart 1 for a list of the valid function codes.
02E	Reg15=X'2E' Reg7=A(QDEST) Reg10=A(message) MSGODSTN=QDNAME	REL190	The destination during a RELEASE call was invalid.
02F	Reg15=X'2F' Reg7=A(QDEST) QDFLG1 bit: QDF1LQ1 on	REL215	The SMB was not empty after the message was released.
02F	Reg15=X'2F' Reg7=A(QDEST) QDFLG1 bit: QDF1LQ1 off	REL230	The SMB should have been flagged as having something in it since a prior message was present.
04A	Reg15=X'4A' Reg10=A(QBLK) QSAPAB15=DFSFNDDST return code	REL337 REL634	The destination specified in the QBLKDSTN field was not found.
050	Reg15=X'50' QSAPAB15=DFSBCB return code	REL1020	An attempt to release the Queue Manager message buffer work area (QMBA) failed.
051	Reg15=X'51' Reg8=A(QTPPCB) QTPFLG3 bit: QTPQMBP off	REL175 REL280	The Queue Manager message buffer work area (QMBA) was not available.
060	Reg15=X'060' Reg2=A(CTB) Reg5=A(QSAP) Reg7=A(CNT) Reg10=A(log record)	REL340	The expected system extension segment, which contains the UTC time stamp, was not found.
065	Reg15=X'65'	REL1120	RRE address was not passed by the caller when processing a Protected Conversation message.

**For DFSQRSQ0:**

The DFSQRSQ0 module handles ERE restart of IMS queues in a shared queue environment. ERESTART is the only caller of this module. DECTYPE describes the call type or the function to be performed:

X'00' BUILDQ processing  
 X'01' ERE work area initialization  
 X'02' ERE processing of Queue Manager log records  
 X'03' ERE cleanup processing

SC	Key Data	Label	Description
205	Reg15=X'205' Reg5=A(log record) Reg3=A(message)	UPDMSG00 UPDMSG01	One of the following segment items was missing from the message prefix: segment extension or TMR segment.
206	Reg15=X'206' Reg5=message record length Reg5>DCBLRECL	BLDMSG0B	The message record length was greater than the logical record length.
209	Reg15=X'209' Reg9=A(DECB) DECBTYPE=X'01'	BLDWKA10	There was not enough space for at least two entries in the work area.
20A	Reg15=X'20A' Reg9=A(DECB) Reg11=A(SCD) DECBTYPE=X'02', X'03', or X'04'	QRSQERE	The queue restart work area was not initialized for ERE processing.
20B	Reg15=X'20B' Reg9=A(DECB) Reg11=A(SCD) DECBTYPE=X'01' SCDQRWKA=0	BLKWKA	There was no queue restart work area.
20C	Reg15=X'20C' Reg8=A(RWKA)	FRWKABND	The work area chain pointer address was zero when trying to free a work area.
20D	Reg15=X'20D' Reg5=A(log record) Reg9=A(DECB) DECBTYPE=X'02' MSGLCODE≠X'01', X'03', or X'07' and NOT(X'30' ≤MSGLCODE≤X'3F')	QLOGREC QLOGRECA PPLOOPA PPCLN30	The log code was neither X'01', X'03', X'07', nor was it a Queue Manager log record (X'30'-X'3F').
215	Reg15=X'215' Reg5=A(message) MSGFLAGS bit: MSGFFRST on MSGCFLG2 bit: MSGC2APP on	QLOGRINB	When attempting to insert an X'01' or X'03' APPC message, the QAB/TIB was not found or created, and the destination was not an SMB.
21C	Reg15=X'21C' Reg7=log record length Reg5=A(log record)	BLDPFX0B UPDPFX12	The allowable message length was exceeded.
21F	Reg15=X'21F' Reg5=A(log record)	QLOGRINA	There was no APPC system segment header during BUILDQ processing although the flag indicated that it existed.

**For DFSQRST0:**

The DFSQRST0 module handles the queue log records for emergency restart and performs final cleanup at the end of ERESTART. ERESTART is the only caller of this module. DECTYPE describes the call type or the function to be performed:

X'00' BUILDQ processing  
 X'01' ERE work area initialization  
 X'02' ERE processing of Queue Manager log records

X'03' ERE cleanup processing

SC	Key Data	Label	Description
205	Reg15=X'205' Reg3=A(message) Reg8=A(log record)	PMSGI20	The expected system extension segment, which contains the UTC time stamp, was not found.
205	Reg15=X'205' Reg5=A(log record) Reg14=A(message)	PENQ33A	The expected system extension segment, which contains the UTC time stamp, was not found.
205	Reg15=X'205' Reg1=A(CCB) Reg14=A(message)	PENQ34C	The expected system extension segment, which contains the UTC time stamp, was not found.
206	Reg15=X'206' Reg5=message record length Reg5>DCBLRECL	PBUILDQP	The message record length was greater than the logical record length.
209	Reg15=X'209' Reg9=A(DECB) DECBTYPE=X'01'	BLDWKA11	There was not enough space for at least two entries in the work area.
20A	Reg15=X'20A' Reg9=A(DECB) Reg11=A(SCD) DECBTYPE=X'02' or X'03'	QRSTERE	The queue restart work area was not initialized for ERE processing.
20B	Reg15=X'20B' Reg9=A(DECB) Reg11=A(SCD) DECBTYPE=X'01' SCDQRWKA=0	BLDWKA	There was no queue restart work area.
20C	Reg15=X'20C' Reg8=A(RWKA)	FRWKABND	The work area chain pointer address was zero when trying to free a work area.
20D	Reg15=X'20D' Reg5=A(log record) Reg9=A(DECB) DECBTYPE=X'02' MSGLCODE $\neq$ X'01', X'03', or X'07' and NOT(X'30' $\leq$ MSGLCODE $\leq$ X'3F')	QRSTERE1 QRSTERE2	The log code was neither X'01', X'03', X'07', nor was it a Queue Manager log record (X'30'-X'3F').
20E	Reg15=X'20E' Reg9=A(DECB) DECBTYPE $\neq$ 0, 1, 2, or 3	CLNERE	An invalid function code was detected. Refer to the DFSQRST0 analysis section for a list of the valid function codes.
210	Reg15=X'210' Reg9=A(DECB) DECBBRN $\neq$ 0, 4, or 8	PBUILDQG PISRT38 PENQ42	An invalid DRRN was detected in DFSQRST0. The 'D' portion of the DRRN does not equal X'00', X'04', or X'08', the RRN portion of the DRRN is zero, or the RRN portion of the DRRN is outside the high range of the data set. The RRN outside the high range is usually caused by decreasing the data set size before an emergency restart.
211	Reg15=X'211' Reg7=A(RWKID) Reg8=A(RWKA)	NGWKAPCB	The ID of the caller was found in the work area chain for a condition when the ID must not be in use.
212	Reg15=X'212' Reg8=A(RWKA)	ACWKAPCB ACWKAXXX	A different status series was found in the work areas.
213	Reg15=X'213' Reg8=A(RWKA) RWKMINOR $\neq$ X'00'	PXFER18 PRELI88	An old work area was found with the same ID for XFER or RELI but the minor status did not match.

SC	Key Data	Label	Description
213	Reg15=X'213' Reg8=A(RWKA) RWKMINOR≠X'01' or X'04'	PRELO80 PRELO90	An old work area was found with the same ID for RELO but the minor status did not match.
214	Reg15=X'214' Reg5=A(QLFNMRCD) QLFRFUNC=logged function	PFREE70	An invalid logged function code was detected while processing an X'33' log record without a work area.
215	Reg15=X'215' Reg5=A(message) MSGFLAGS bit: MSGFFRST on MSGCFLG2 bit: MSGC2APP on	PISRT34 PISRT80B	When attempting to insert an X'01' or X'03' APPC message, the QAB/TIB was not found or created, and the destination was not an SMB.
216	Reg15=X'216' Reg5=A(QLGURCD)	PGU10	For label PISRT80B, Reg14=A(message) While processing an X'31' log record, the destination was not found.
216	Reg15=X'216' Reg5=A(QLENQRCD)	PENQ20 PENQ31B	While processing an X'35' log record, the destination was not found.
216	Reg15=X'216' Reg5=A(QLXFRRCD)	PXFER27	While processing an X'37' log record, the destination was not found.
216	Reg15=X'216' Reg5=A(QLRIRCD)	PRELI10	While processing an X'38' log record, the destination was not found.
216	Reg15=X'216' Reg5=A(QLFRERCD)	FENDC20	While processing an X'33' log record, the destination was not found.
216	Reg15=X'216' Reg5=A(QLMCRCD)	CLNC08 CLNDCUR	While processing an X'3E' log record during a C or D cleanup series, the destination was not found.
216	Reg15=X'216' Reg5=A(QLDQSRCD)	PDEQABND	While processing an X'36' log record, the destination was not found.
217	Reg15=X'217' Reg8=A(RWKA) QLFRFLGS bit: QLFRFEND off	PFREE44	During free log record processing, the series was X'0A' (ISRT) which is invalid.
217	Reg15=X'217' Reg8=A(RWKA) QLFRFLGS bit: QLFRFEND on	PFREE50	During free log record processing for a final record on a chain, the series was X'0A' which is invalid.
218	Reg15=X'218' Reg8=A(RWKA) RWKMAJOR>X'10'	PFREE40	The series for this 'FREE' log record was greater than the highest series allowed.
219	Reg15=X'219' Reg5=A(QLMCRCD) MSGFLAGS bit: MSGFFRST on MSGCFLG2 bit: MSGC2APP on	CLNC04	When attempting to update the message DRRN of an X'3E' log record during a C cleanup series, the QAB/TIB was not found or created and the destination was not an SMB.
21A	Reg15=X'21A' Reg3=A(message prefix)	PMSGI110	When attempting to modify the prefix of an X'30' log record, the QAB/TIB was not found or created.
21B	Reg15=X'21B' Reg14=A(message) Reg7=A(SMB)	CLNR24	Chaining to the same message twice. The message chain has been corrupted. Perform an /ERE BUILDQ to rebuild the message queues.
21C	Reg15=X'21C' Reg3=log record length Reg5=A(log record)	PISRT38	The allowable message length was exceeded.



SC	Key Data	Label	Description
21D	Reg15=X'21D' Reg5=A(QLENQRCD) or A(QLDQSRCD)	PENQ60 PDEQ30A	While processing an X'35' or X'36' log record, the OTMA Tpipe could not be located.
21E	Reg15=X'21E' Reg5=A(QLMCRCD)	CLND10A FDST00	While processing an X'3E' log record during a D cleanup series or when calling FINDDEST, the block was not an SMB, CNT, LNB, QAB, or TIB block.
21F	Reg15=X'21F' Reg4=A(log record)	PBUILDQC PISRTA1 PISRT80A	There was no APPC system segment header during BUILDQ processing although the flag indicated that it existed.

**For DFSQXF00:**

DFSQXF00 is the Queue Manager TRANSFER command processor. The QSAPXFUN field in the QSAPWKAD area pointed to by register 5 contains the function code:

X'1E' transfer

SC	Key Data	Label	Description
000	Reg15=X'00'	DFSQXF00	An illegal function was requested. Refer to the table following Chart 1 for a list of the valid function codes.
027	Reg15=X'27' Reg7=A(QDEST)	XFR21	A queue was found to be empty, but the DRRNs of the Queue Manager destination were not zeros as they should be for an empty queue.
028	Reg15=X'28' Reg9=A(DECB) Reg10=A(last message) MSGMDRRN=DECBRBN	XFR24	Last message toward printer was already chained to this DECBRBN.
029	Reg15=X'29' Reg10=A(QBLK)	XFR32	When moving a pointer from the temporary destination queue to CNT, the pointer was not zero when the flag indicated that there were messages in that queue.
02A	Reg15=X'2A' Reg9=A(DECB) Reg10=A(message) MSGMDRRN=DECBRBN	XFR34	The message to be moved should not have been chained forward.
04A	Reg15=X'4A' Reg10=A(QBLK) QSAPAB15=DFSFNDDST return code	XFR11A	The destination specified in the QBLKDSTN field was not found.
04C	Reg15=X'4C' QSAPAB15=DFSBCB return code	CQSDEL DEL040 GETQMBA GETQMBS PROTCONV PUTR200 QLD0LOG SHR215	An attempt to obtain a UOWE, QMBA, QMBS, or QLST work area failed.
04D	Reg15=X'4D' QSAPAB15=DFSCLM return code QSAPNQF2 bit: QSAPLATE on	XFR46	An attempt to release the QLOG acquired DC system latch failed.
050	Reg15=X'50' QSAPAB15=DFSBCB return code	GQMBA_300 XFR80	An attempt to release the Queue Manager message buffer work area (QMBA) failed.

SC	Key Data	Label	Description
05F	Reg15=X'5F' QSAPAB15=DFSUSE return code	XFR26	A DFSUSE FUNC=INUSE call failed with a return code other than X'1C'.
064	Reg15=X'64' Reg8=A(QTPPCB) Reg10=A(message)	SHR100	The message accessed for the CQS PUT was not the first record in the message.
066	Reg15=X'66' Reg4=A(message)	BSTQ1000 BSTQ1500 BSTQ2000 BSTQ3000 BSTQLOOP BSTQ5000 BSTQ6500	Invalid message returned from ILOG Function Read call when re-building Stage Queue entries for Serial Tran after a CQSPUT error.

**CHART 1:** Dectype Format as Used by the Message Queue Manager:

**xx.. ....** Defines caller identity  
**00.. ....** Caller is communications  
**10.. ....** Caller is DL/I (application program request)  
**01.. ....** Caller is the message generator  
**..x. ....** Special modifier for note/point and insert move calls  
**..0. ....** If note/point, indicates note  
**..1. ....** If note/point, indicates point  
**..1. ....** If message generator insert move, indicates AOI call  
**.... x...** Generally used to denote input (process a message that is already in the queue) or output (actions to place a message into the queue)  
**.... 0...** Input operation class  
**.... 1...** Output operation class  
**.... .x..** Normally used to terminate a series of calls to the queue manager by placing a completed message into, or removing a message from, the queue  
**.... .0..** The series of calls is not complete  
**.... .1..** The series of calls is complete  
**..xx xxxxx**  
 These bits are used to completely define the requested function

Function	Name	Description
..00 0000	Get prefix	Read specified prefix information
..00 0001	Cancel input	Cancel the input request
..00 0010	Get unique	Get first segment of the next message
..00 0011	Get next	Get next segment of the current message
..00 0100	Dequeue	Dequeue all messages read in this series of calls
..00 0101	Save	Save the current message
..00 0110	Reject	Reject the current message from the specified temporary destination
..00 0111	Delete	Delete the specified message from the queue
..00 1000	Cancel output	Cancel the current output message and log the text as canceled
..00 1001	Cancel output	Cancel the current output message but do not log the text
..00 1010	Insert locate	Reserve contiguous space in an output message
..x0 1011	Insert move	Insert the message segment into an output message
..00 1100	Enqueue	Enqueue the completed message on its permanent destination (FIFO)
..00 1101	Enqueue	Enqueue the completed message on its permanent destination (LIFO)
..00 1110	Reenqueue	Reenqueue a saved message on its permanent destination (FIFO)
..00 1111	Reenqueue	Reenqueue a saved message on its permanent destination (LIFO)
..01 0000	Reposition	Reposition to the last segment in the current message
..01 0001	Command input	start AOI processing for the specified command
..01 0010	AOI SYMSG	start AOI processing for the specified message

Function	Name	Description
..01 0011	Abort	Terminate AOI processing for the current series of calls
..01 0100	AOI term	Terminate AOI processing for the current series of calls
..01 0101	Error exit	MSC error message reroute
..01 0110	Release	Release all messages from the specified temporary destination
..11 0110	Release	ROLS function
..01 0111	Reserved	
..01 1000	Reserved	
..01 1001	Reserved	
..01 1010	Insert prefix	Insert additional information into the message prefix of the current output message
..01 1011	Put spanned	Insert the message segment into an output message spanning queue records
..01 1100	Enqueue (Temp)	Enqueue the completed message FIFO on the specified temporary destination with the intent that it will be transferred to its permanent destination at a later time
..01 1101	Enqueue (Temp)	Enqueue the completed message LIFO on the specified temporary destination with the intent that it will be transferred to its permanent destination at a later time
..01 1110	Transfer	Transfer all messages from the specified temporary destination to their permanent destination
..x1 1111	Note/Point	(Note) Remember message position in a series of input calls or (Point) return to 'noted' message position in a series of input calls

**System action:** The system terminates abnormally.

**Programmer response:** First eliminate the possibility that an operational error could have caused the abend. If this is not the cause of the problem, then use the subcode value in register 15 at the time of the abend and refer to the Analysis above for a complete description of the problem.

**Problem determination:** 1, 2, 3, 4, 6, 8, 11, 35

Save the log data sets from the last DUMPQ/BUILDQ or from the last cold start.

**Source:** IMS abends

**Module:** DFSQC010, DFSQC020, DFSQC030, DFSQC040, DFSQC060, DFSQC070, DFSQC080, DFSQBFM0, DFSQDQ00, DFSQE000, DFSQFIX0, DFSQGU00, DFSQIS00, DFSQLOG0, DFSQMGR0, DFSQNP00, DFSQRH00, DFSQRL00, DFSQRSQ0, DFSQRST0, DFSQXF00

**Related information:**

Chapter 2, "IMS problem determination," on page 13

## 0758

**Explanation:** A message queue data set has overflowed before the system could be shut down with an internal checkpoint dump queue.

For IMS running with Shared Queues, abend 0758 is issued from DFSQC010 because all the available device relative record numbers (DRRN) have been exhausted.

**Analysis:** 0758 is a standard abend. The program status word (PSW) at entry to abend points to the failing module.

**Possible cause:** This abend is typically caused by a looping dependent region program that is doing an INSERT-LOCATE, INSERT-MOVE, or INSERT-MOVE-SPANNABLE.

- If the problem is INSERT-LOCATE:
  - Reg5+X'168'= QSAPXFUN = X'0A'
  - Reg9+X'0C'= DECBAREA = Length of the text
- If the problem is INSERT-MOVE:
  - Reg5+X'168'= QSAPXFUN = X'0B'
  - Reg9+X'0C'= DECBAREA = Address of the text
- If the problem is INSERT-MOVE-SPANNABLE:
  - Reg5+X'168'= QSAPXFUN = X'15'

## 0759

- Reg9+X'0C'= DECBAREA = Address of the text

For shared queues, this abend is typically caused by setting QBUFMAX. If you set QBUFMAX too low, all the available DRRNs can be exhausted. The recommendation is not to specify a value for the QBUFMAX parameter so that then the number of available buffers in the queue pool is unlimited.

For more information about the QBUFMAX parameter, see the IMS system definition information.

*For APAR processing:* ABEND DUMP, console sheet, copy of system log.

Key	Label	Description
Reg6=Bal	ASSIGN65	
Reg3=RRN	ASSIGN65	If the RRN in register 3 is larger than the highest RRN of the data set, the abend is using BAL to QMGNOSPC.  The highest RRN is at X'0C' off the address at Reg7+X'74'.
REG2='QBLK' or 'MSG' or 'LMSG'	ASSIGN65	Register 2 contains the identity of the data set in error.

If register 2 contains 'MSG' or 'LMSG' and you have defined multiple data sets for that message queue, verify that the data sets are allocated correctly. See the information about IMS.QBLKS, IMS.SHMSG/1-9,IMS.LGMSG/1-9 in Execution data sets (Installation) and the information about message queue data set allocation in IMS Queue Manager (System Administration).

**System action:** IMS terminates abnormally.

**Programmer response:** The space allocation for one of the queue data sets listed below must be increased and the system emergency restarted with a build queue. The number of shutdown records can be increased by performing an IMS system definition. The contents of register 14 at abend point to the DCB of the full queue data set.

IMS.QBLKS  
IMS.SHMSG  
IMS.LGMSG

Often this error is related to a loop in an application program that creates a large amount of message data. If so, the output from the looping application can be identified by examining the log data set.

**Problem determination:** None.

**Source:** IMS abends

**Module:** DFSQC010

---

## 0759

**Explanation:** Unrecoverable error in queue manager attempting to read or write message queue data sets. The I/O operation is retried three times. The abend is issued if retry is not successful.

**Analysis:** 0759 is a standard abend. The registers in the abend SVRB should be examined to determine where the error was detected in DFSQBFM0.

### Register 3

#### Contents

**Byte 1** OSAM operation code from DECBTYPE

**Byte 2** I/O post code

**Byte 3** status code from DECBSTAT

**Byte 4** contains the DCB ID, where

X'00' is the Queue-Blocks data set

X'04' is the Short message queue data set

X'08' is the Long message queue data set

**Register 9**

Points to the OSAM DECB in error

**Register 2**

Contains the label name/error code in DFSQBFM0 where the error was detected

Key	Label	Description
DECB/=X'7F' (Reg9) FUNC=aa (Reg3 byte 1) STATUS=bb,cc (Reg3 byte 2,3) DCBid (Reg3 byte 4)	D8C2C601 (Reg2)	An I/O operation completed with an IOS POST code indicating an I/O error. Message DFS0762I precedes the abend.
DECB/=X'7F' (Reg9) FUNC=aa (Reg3 byte 1) STATUS=bb,cc (Reg3 byte 2,3) DCBid (Reg3 byte 4)	D8C2C602 (Reg2)	An attempt to do I/O resulted with an IOS POST code indicating an I/O error. An OSAM IOSB is not available to issue message DFS0762I. Register 3, bytes 1, 2, and 3 contain the OSAM operation code, the IOS POST code, and the DECB status code as defined in message DFS0762I. Register 3, byte 4 contains the DCBid on which the error occurred. Possible Cause: Reg 3, byte 2=X'10'Status=DECBIOSBE There is no more room in CSA storage to allocate an IPAGE for additional IOSBs.
DECB=X'7F' (Reg9) FUNC=aa (Reg3 byte 1) STATUS=bb,cc (Reg3 byte 2,3) DCBid (Reg3 byte 4) Address of the read-in area = Reg4+X'40'	D8C2C603 (Reg2)	IMS puts a X'FF'in the first byte of the read-in area before issuing the read request. After the read has been POSTed complete, a check is made to verify that the X'FF'has been overlaid. A X'FF'in the first byte indicates that the I/O requested completed successfully (X'7F'POST) but no data transfer was done. This condition is considered an I/O error. Possible Cause: An I/O error occurred without an indication. There has been no data transfer between the I/O device and the buffer storage.

**System action:** IMS terminates abnormally.**Operator response:** Attempt to restart IMS with the BUILDQ and FORMAT options.**Source:** IMS abends**Module:** DFSQBFM0**Related information:**
 DFS0762I
**0760****Explanation:** One of two error conditions was detected:

- A message queue data set requires formatting.
- The number of records reserved for a message queue data set for IMS shutdown is greater than the number of records in the data set.

**Analysis:** This is a standard abend issued by DFSQDOC0. The DFSCNS macro called DFSAOS10 to open one of the above system data sets. Return code X'18' was passed back to DFSQDOC0.

Key	Label	Description
Register 7 contains the DCB address. DFS986A, issued just prior to abend 0760, contains the name of the data set in error.	OSM60BAD	The number of records reserved for a message queue data set for IMS shutdown is greater than the number of records in the data set.

**System action:** IMS terminates abnormally.

## 0762

**Programmer response:** Do the following:

1. Format one of the data sets listed below by restarting IMS using the FORMAT keyword parameter. Message DFS986A, issued just prior to abend 0760, contains the name of the data set requiring formatting. Register 7, at the end of the abend, contains the DCB address.
2. Increase the space allocation for one of the system data sets listed below. Emergency restart the system with a build queue (/ERESTART BUILDQ FORMAT), remembering to FORMAT those data sets that were reallocated. You can alter the number of shutdown records by performing an IMS system definition. Message DFS986A, issued just prior to abend 0760, contains the name of the data set causing the error. Register 7, at the time of abend, contains the address of the DCB in error. The system data sets are as follows:

IMS.QBLKS  
IMS.SHMSG  
IMS.LGMSG  
IMS.SPA

This error is caused by starting IMS with an unformatted system data set (above) or by reallocating one of these system data sets with a quantity that is too small.

**Source:** IMS abends

**Module:** DFSQDOC0

**Related information:**

 DFS986A

---

## 0762

**Explanation:** A critical system error was detected during IMS cross-memory processing.

**Analysis:** This abend is issued by module DFSDSPX0 when it detects severe internal errors when processing a cross-memory function. Register 14 in the abend registers point to the address within DFSDSPX0 where the error was detected. Register 15 contains one of the following reason codes:

### Codes

#### Explanation

**X'01'**

DFSDSPX0 was called with an invalid function code.

**X'02'**

DFSDSPX0 detected an invalid primary address space number while trying to return from cross-memory mode to home mode. The home primary address space number saved in an IMS internal control block does not match the true home address space number. A storage overlay or control block pointer error may have occurred.

**X'03'**

DFSDSPX0 detected an invalid secondary address space number while trying to return from cross-memory mode to home mode. The home secondary address space number saved in an IMS internal control block does not match the true home address space number. A storage overlay or control block pointer error may have occurred.

**X'04'**

DFSDSPX0 detected an invalid target address space index while trying to switch to cross-memory mode.

**X'05'**

Validation failed for XMCA cross-memory control block. The eye-catcher in the block did not equal "XMCA".

**X'06'**

Validation failed for XMCI cross-memory control block. The eye-catcher in the block did not equal "XMCI".

**X'07'**

Validation failed for XMCI cross-memory control block. The SAP pointer in the XMCI did not equal the current SAP address.

**X'08'**

An error occurred trying to enter or exit cross-memory mode. The high byte of register 15 will contain one of the following values:

**X'01'** Target primary address space number was invalid.

**X'02'** Target secondary address space number was invalid.

**X'09'**

DFSDSPX0 detected an invalid primary address space number when called to restore the cross-memory mode of an ITASK.

**Possible cause:** All of these reason codes denote IMS internal errors.

**System action:** IMS terminates abnormally.

**Operator response:** Perform an emergency restart.

**Problem determination:** 1, 3, 4, 6, 33, 34

**Source:** IMS abends

**Module:** DFSDSPX0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

**0763**

**Explanation:** The IMS system message queue manager was waiting to be posted by a queue manager for another IMS subtask, but the DECB was posted by another part of the IMS system. This is an IMS system error.

**Analysis:** This is a standard abend issued by DFSQCP00, DFSQMGR0, or DFSQBFM0 so the registers in the abend SVRB are the ones to use to determine where the error condition was detected.

Use the program status word (PSW) at entry-to-abend to determine which module issued the abend and the register 14 BAL to isolate to a specific label.

Key	Label	Description
Reg14≠BAL	DFSQCP20	On return from IWAIT, a check is made to ensure that the post code was from the queue manager. The expected post code is X'63'. If not, the abend is issued, with register 9 pointing to the DECB in error.
Reg14=BAL	INCNPG0	On return from IWAIT a check is made to ensure that the post code was from queue manager. The expected post code is X'63'. If not, this abend is issued by branching to QMGRPERR. Register 9 points to the DECB in error.
Reg14=BAL	QXCENQ4	In this routine, serialization of a destination address is provided. Register 7 contains the address to be serialized, and abend occurs if, after the IWAIT for destination ownership, the post code is not from the queue manager. Register 9 points to the DECB in error.
Reg14=BAL	QBMWTI	On return from IWAIT, a check is made to ensure that the post code was from the queue manager. If not, this abend is issued. Register 9 points to the DECB in error.
Reg14≠BAL	QBMIOER	On return from the check I/O routine, the post code in the DECB (register 9) is checked for a valid IOS post code. If post code is invalid, the abend is issued by branching to QMGPERR.

**System action:** IMS terminates abnormally.

**Problem determination:** 1, 2, 4, 6, 9, 35

**Source:** IMS abends

**Module:** DFSQBFM0, DFSQCP00, DFSQMGR0

**0764**

**Explanation:** The OSAM access method was unable to convert an RBN to a full-disk address in the form MBBCCHHR during WRTQUED processing. OSAM posted I/O ECB with an abnormal I/O ending code which does not equal (X'7F'). There are two cases causing the abend 0764. They are described as follows:

## 0765

1. If R15 = 0, it indicates that OSAM is unable to convert an RNB to a full disk address in the form MBBCCHHR during a WRTQUED redrive processing.
2. IF R15 = X'FFFFFFFF', it indicates that the user specified 'DUMPIO=YES' in DFSVSMxx to ask abend 764 for any abnormal I/O ending.

**Analysis:** This is a standard abend issued by DFSA0S80. The error registers are stored in the abend SVRB and the program status word (PSW) at entry-to-abend will point to the instruction within label AB0764 from which the abend (SVC 13) was issued.

At the time of failure, register 5 contains the DECB address, register 6 contains the SCD address, register 4 contains the IOB address and register 12 is the base register.

Key	Label	Description
Reg1=X'800002FC'	INTERR	The DECB was posted with a X'41' (abnormal completion). During redrive processing, the OSAM access method found an invalid RBN. The RBN could not be converted to a full disk address. The invalid RBN is located in field IBFCBLK of the current buffer prefix. The current buffer address is located in the IOB.

**Possible cause:** Probable hardware malfunction in the control unit or channel.

**System action:** IMS terminates abnormally.

**Operator response:** Contact the IMS system programmer.

**Programmer response:** This is an internal IMS error.

**Problem determination:** 1, 2, 4, 35

**Source:** IMS abends

**Module:** DFSAOS80

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

## 0765

**Explanation:** The OSAM access method attempted to translate a virtual address to a real address. The translation failed. As all pages were page fixed before attempting translation, this problem is an internal IMS error. In an z/OS environment, the failure could have been caused by a bad return code from the page fix routine. The DECBSTAT field in the DECB contains an error status code other than an X'7F'.

**Analysis:** This is a standard abend issued by DFSAOS80. The error registers are in the abend SVRB. The program status word (PSW) at entry-to-abend points to the abend (SVC 13) which was issued at lab el XLERROR.

Register 12 is the base register, register 5 contains the DECB address, register 4 contains the IOSB address, and register 1 contains the abend code, X'800002FD'.

The status field of the DECB (DECB STAT=18) contains a specific error code.

Key	Label	Description
DECBSTAT=X'1F'	SOMERR	A page fix error was detected.
DECBSTAT=X'20'	SOMERR	A CCW translate error was detected.
DECBSTAT=X'03'	SOMERR	An IDAL translate error was detected.

**System action:** IMS terminates abnormally.

**Problem determination:** 1, 2, 4, 35

**Source:** IMS abends

**Module:** DFSAOS80

**Related information:**

Chapter 2, "IMS problem determination," on page 13



---

**0766**

**Explanation:** A critical system problem was detected during IMS ITASK dispatching.

**Analysis:** This is a standard abend issued by DFSIDSP0, the IMS ITASK Dispatcher. The abend is the result of errors in the ECB selection, ITASK creation and termination.

Register 8 contains the address of the ECB prefix for the ECB in question for all reason codes except X'07'.

Register 15 in the abend SVRB contains one of the following reason codes:

**Codes****Explanation****X'01'**

The next ECB on the ready queue to be dispatched is chained to itself.

**X'02'**

A dynamic SAP ECB is posted to a nondynamic SAP TCB.

**X'03'**

The next ECB in the ready queue to be dispatched has an invalid chain pointer. The pointer should be the address of the next ECB on the queue, or zero; instead, it is the dispatcher code X'0FC4E2D7'.

**X'05'**

An IMS ITASK terminated with an invalid IMS save area. The first word of the save area pointed to by R13 did not point to the ITASK SAP. Either R13 is not pointing to the ITASK's original save area, or the SAP pointer has been overlaid.

**X'07'**

The specified ITASK termination is not under an IMS TCB.

**X'08'**

An ECB being redispached has never IWAITed before.

**System action:** IMS terminates abnormally.

**Operator response:** Perform an emergency restart.

**Problem determination:** 1, 4, 5, 6, 35

**Source:** IMS abends

**Module:** DFSIDSP0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

**0767**

**Explanation:** A critical systems problem was detected while the IMS dispatcher was processing an IWAIT or ISERWAIT call.

**Analysis:** This is a standard abend issued by DFSIDSP0, the IMS ITASK dispatcher. Errors that are detected in the ISERWAIT or IWAIT routine branch to issue SVC 13.

Register 8 contains the address of the ECB prefix for the ECB in question. Register 14 contains the return address of the IWAIT or ISERWAIT caller.

Register 15 in the abend SVRB contains the following reason code:

**Codes****Explanation****X'01'**

The IWAIT caller's ECB has no active ITASK.

**X'02'**

On an IWAIT or ISERWAIT call, the ECB passed to the dispatcher was not the currently dispatched ECB.

## 0768

### X'03'

On an IWAIT call with TYPE=IXCTL specified, the ECB prefix of the IWAIT call's target ECB was invalid. Register 5 contains a return code indicating the reason for the invalid prefix.

Code	Meaning
------	---------

X'04'	The ECB is on a dispatcher queue.
-------	-----------------------------------

X'0C'	The ECB ITASK is already active.
-------	----------------------------------

X'10'	The ECB is assigned to an IMS TCB that is different from the one issuing the IWAIT call.
-------	--

Register 3 contains the address of the target ECB. Register 7 contains the address of the current ECB.

**System action:** IMS terminates abnormally.

**Operator response:** Perform an emergency restart.

**Problem determination:** 1, 4, 5, 6, 35

**Source:** IMS abends

**Module:** DFSIDSP0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

## 0768

**Explanation:** A system problem was detected while the IMS dispatcher was processing an ISWITCH call.

**Analysis:** This is a standard abend issued by modules DFSIDSP0 and DFSREP00.

For all of the reason codes listed below, except X'09' through X'0B', register 8 points to the ECB prefix of the ECB (ITASK) in question. Register 10 points to the ECB's SAP. For all codes except X'01', register 4 points to the dispatcher work area of the IMS TCB currently executing.

Register 15 in the abend SVRB contains one of the following reason codes:

### Codes

Explanation
-------------

#### X'01'

The ISWITCH caller is not an active ITASK.

#### X'02'

Either flags in the caller ITASK's SAP indicate an ISWITCH was already in progress for an ISWITCH TO= call, or the ITASK is a dependent region trying to ISWITCH to a different TCB while running under the IMS control TCB.

#### X'03'

Flags in the caller ITASK's SAP indicate an ISWITCH was already in progress for an ISWITCH TO=RET call.

#### X'04'

On an ISWITCH TO=DEP call, the target TCB of the ISWITCH was a dependent region, but it was not the home dependent region of the ITASK.

#### X'05'

On an ISWITCH call, the ECB passed to the ISWITCH routine was not the currently dispatched ECB. This subcode can be issued out of either the DFSISWIT subroutine in DFSIDSP0 (normal TCB ISWITCH), or the DFSKXMSW subroutine in DFSREP00 (cross-memory ISWITCH).

#### X'06'

An ECB being redispached after an ISWITCH has an invalid post code. The code must be either T0 or RET.

#### X'07'

An ECB was being redispached as if it had ISWITCHed to a new TCB, but flags in its SAP indicate it has not ISWITCHed.

#### X'08'

A dependent region shutdown ITASK was running under a TCB other than the IMS control region TCB.

**X'09'**

Module DFSDSPX0 was called to perform an unstack function, but the ITASK stack was empty (nothing to unstack), or the current stack entry index was invalid (beyond the end of the stack).

**X'0A'**

Module DFSDSPX0 was called to perform a stack function, but the ITASK stack was full (no room to add a new entry).

**X'0B'**

Module DFSDSPX0 was called to perform an unstack function, but the current stack index indicated an entry that was invalid. Either the index was out of range or the indicated stack entry was not active (contained no valid data).

**Possible cause:** Internal systems error

**System action:** IMS terminates abnormally.

**Operator response:** Perform an emergency restart.

**Problem determination:** 1, 4, 5, 6, 35

**Source:** IMS abends

**Module:** DFSIDSP0, DFSREP00

**Related information:**

Chapter 2, "IMS problem determination," on page 13

**0769**

**Explanation:** A critical system problem was detected in a dispatching subroutine while trying to initialize dispatcher control blocks.

**Analysis:** This is a standard abend issued by DFSIDSP0, the IMS events ITASK dispatcher, or by DFSREP00 for errors in IPOST, IXCTL, and INITECB.

Register 15 in the abend SVRB contains one of the following reason codes:

**Codes****Explanation****X'01'**

On an INITECB call, the event control block (ECB) being initialized was either on a chain, or was already waiting. Register 8 points to the ECB prefix.

**X'02'**

On an INITECB call, the ECB being initialized was already posted and on a queue. Register 8 points to the ECB prefix.

**X'03'**

On an INITECB call, the ECB being initialized was posted. The dispatcher tried to put the ECB on the posted queue, but the TCB to which the ECB was assigned was suspended. Register 8 points to the ECB; register 4 points to the suspended TCB.

**X'07'**

On an IXCTL call, the ECB being transferred to was already an active ITASK. Register 8 points to the current ECB prefix; register 3 points to the target ECB prefix.

**X'08'**

On an IXCTL call, the ECB being transferred to failed an UNINIT call. Register 9 contains the UNINIT return code. If R9=X'04', the ECB prefix's chain field was invalid. If R9=X'10', the ECB was assigned to a different TCB than the one currently running. Register 8 points to the current ECB prefix; register 3 points to the target ECB prefix.

**X'09'**

A DFSKPXT call (z/OS branch entry POST) was made in cross-memory mode when the ECB being posted was in a nonpost exit routine wait state. Register 3 points to the ECB.

## 0770

**X'0A'**

An obsolete SCP post routine was called.

**System action:** IMS terminates abnormally.

**Operator response:** Perform an emergency restart.

**Problem determination:** 1, 4, 5, 6, 35

**Source:** IMS abends

**Module:** DFSIDSP0, DFSREP00

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

## 0770

**Explanation:** A systems error was detected while a dependent region was attempting to initialize an ITASK, attempting to OPEN or CLOSE a dependent region.

**Analysis:** This is a standard abend issued by DFSIDSP0, the IMS events ITASK dispatcher, or DFSREP00.

Register 15 in the abend SVRB contains one of the following reason codes:

### Codes

#### Explanation

**X'01'**

An ECB prefix error; the SAP address was not set.

**X'02'**

An ECB prefix error; the dispatcher work area address was not set or was not a valid TCB for dependent region open.

**X'03'**

No TCB table entry was found in the current TCB's dispatcher work area.

**X'04'**

Enqueue of recovery element for dependent region failed.

**X'05'**

Invalid or recursive shutdown call.

**X'06'**

The dependent region abend cleanup routine was called from a TCB other than the control region.

**X'07'**

On a call to the dependent region abend cleanup routine, the current ITASK was not associated with a dependent region.

**X'08'**

DEP signoff DFSBCB get quick save failed.

**X'09'**

DEP signoff DFSBCB get AWE failed.

**X'0A'**

DEP signoff save area is not in the ITASK set.

**X'0B'**

DFSBCB get for LSO signon AWE failed.

**System action:** The dependent region terminates abnormally.

**Operator response:** Shut down the control region with a memory dump. Restart the control region, and then restart the dependent region.

**Problem determination:** 1, 5, 6 (save the control and dependent region dumps), and 35

**Source:** IMS abends

**Module:** DFSIDSP0, DFSREP00

**Related information:**

Chapter 2, "IMS problem determination," on page 13

**0773**

**Explanation:** The application received a read or open error from the buffer handler during an ISRT or DLET call that required single call backout to remove updates made during the call. Single call backout failed for one of these reasons:

- No disk log was specified in batch for dynamic backout.
- Positions in other PCBs in the PSB were adjusted.
- Backout failed

**Analysis:** This is a pseudoabend issued by DFSDDLCO or DFSDDLCO when DFSDA00 is unable to perform a single-call backout. A SNAP of the control blocks may be necessary to diagnose the problem. For batch regions, ensure that a disk log with dynamic backout= Y is specified. For all regions, abend 0773 can occur because position in other PCBs in the PSB were altered (PSTDLSB=PSTDLSB7) or DFSRDBC0 returns a nonzero code. In the latter case, DFSRDBC0 will SNAP the blocks before DFSDLA00 (return code in register 7).

**System action:** The application program terminates abnormally.

**Programmer response:** Correct the cause of the read error or single call backout failure.

**Problem determination:** 1, 17

**Source:** IMS abends

**Module:** DFSDDLCO, DFSDDLCO

**Related information:**

Chapter 2, "IMS problem determination," on page 13

**0774**

**Explanation:** A PST that was waiting for a lock for a resource from module DFSPLEX0 or DFSPLEXA0, was posted out of a wait state with a post code that was not X'60' or X'6F'. A X'60' post code is used when the PST is posted as the result of a deadlock. A X'6F' post code initiates the PST when it has been granted the lock for which it has been waiting.

**Analysis:** Register 10 in the abend dump contains the address of the PST. The first word of the PST is the DECB that contains the invalid post code.

If the memory dump is from an IMS control region, the Dispatcher trace should be scanned to find the X'06' entry that indicates the post. This entry contains the address where the post was initiated.

If the memory dump is from a CICS region, the CICS trace should be scanned for an entry for the post. If help is needed in deciphering the CICS trace entry, the CICS Support Group should be contacted.

**System action:** IMS terminates abnormally.

**Programmer response:** If the memory dump is from an IMS control region, scan the Dispatcher trace to find the X'06' entry that indicates the post. This entry contains the address where the post was initiated.

If the memory dump is from a CICS region, scan the CICS trace for an entry for the post.

**Source:** IMS abends

**Module:** DFSPLEX0

**0775**

**Explanation:** This abend is issued when IMS cannot acquire any more ENQ/DEQ blocks or the amount of storage that was specified by the PIMAX parameter has been exhausted. It can be either a standard abend or a pseudoabend.

**Case 1** A standard abend is issued at label XC10ABND. Use the registers at entry-to-abend or in the abend SVRB, taken from the abend 0775 memory dump, to determine which of the CASE1 entries applies to the current failure.

## 0775

**Case 2** A pseudoabend is set up at label NOCORE. The information about message DFS2450I is helpful in determining the cause of the abend 0775 and in obtaining a resolution

See message DFS2450I for additional information. If message DFS2450I was not issued, the amount of storage specified in the PIMAX parameter might have been exhausted. Increase the size of the PIMAX parameter in member DFSPBIMS in a DB/DC environment, or in member DFSPBDBC in a DBCTL environment. For batch data sharing, refer to *CICS Transaction Server for z/OS Messages and Codes (GC34-6827)*.

**Analysis:** If the abend is issued because DFSFXC10 was unable to acquire storage, then message DFS2450I is also issued. This message uses the routing codes obtained from SSCDROUT. The default routing codes 2 and 7 route the message to the master console. The master console is the main system console and any other console that has been specified as a master console during system definition. If the message is not going to the correct console, then the routing codes can be changed by using the MCS parameter of the IMSCTRL macro.

The message identifies the name of the ENQ/DEQ block for the storage that could not be acquired. The last 2 or 3 digits of this name is a block number. The message also contains a reason code that is helpful in determining what initiated the abend. The following reason codes and meanings indicate what action to take.

Code	Meaning	Action
8	There is no storage available for the first ENQ/DEQ block. The message indicates block 01 and a standard abend occurs.	For z/OS, increase the allocation for CSA.
0	A block was requested that is not the first or the last block. This means that the amount of storage specified in the PIMAX parameter has not been exhausted, but a GETMAIN for another block of storage was unsuccessful.	For z/OS, increase the allocation for CSA.

If message DFS2450I was not issued, the amount of storage specified in the PIMAX parameter might have been exhausted. Increase the size of the PIMAX parameter in member DFSPBIMS in a DB/DC environment, or in member DFSPBDBC in a DBCTL environment. See the information about enqueue and dequeue storage requirements in *Specifying enqueue and dequeue requirements (System Definition)*. This topic describes an overview of how database update events are stored and released in internal storage to maintain database integrity. Estimate separately the storage for each MPP and BMP that is running, and then total the estimates. The PIMAX parameter value should be large enough to hold enqueue/dequeue table entries so that abend 0775 can be avoided.

**Attention:** Taking checkpoints more frequently in a BMP decreases the total storage size required. If running under CICS and not using DBCTL, the value specified by the PIMAX and PIINCR parameters are handled differently. A CICS initialization parameter overrides the second parameter of CORE, which specifies maximum storage.

If this abend still persists, run program isolation (PI) trace (see /TRACE command) and the results printed using DFSERA10 with the DFSERA40 exit routine. Analyze the results should be analyzed to determine the number of ENQs per PST that are still outstanding at the time this abend occurred. Then use by PST in the formula mentioned to recompute the total size of storage required.

If message DFS2450I cannot be located, obtain a system memory dump. This memory dump helps you determine which of the CASE 2 failures occurred.

### CASE 1:

Key	Label	Description
Reg0=minus value Reg7=REQQEL	ALTWC2	This request to enqueue is shareable and did not have to wait. If the wait count is decreased, it becomes minus and the abend is issued.
(Reg5+X'00') Reg8=0's REQCTLW BIT4=0	ALTWCER4	In the QEL for this request there was no matrix row assigned for deadlock protection. The request did not indicate ABORT, so the abend is issued.

Key	Label	Description
Reg1=0's PRMENCTN BIT1=1 (Reg10+X'06')	GTCINT	An internal caller requested a QCB. If no more core can be GETMAINed and no internal QCB is available, the abend is issued.
XCGMID2=X'FB' (Reg11+X'62')	XC10WTOA	Core could not be obtained for the first ENQ/DEQ block, so the abend is issued.
Reg6=BAL(ALTWC) Reg14=BAL	ENQ31	There are two BALs to XC10ABND after this label. This new request will not share control with an existing request. If the wait count is minus or zero on return from the routine to adjust the wait counts, the abend is issued.
Reg6=BAL(ALTWC) Reg14=BAL	ENQ31C	There are three BALs to XC10ABND after this label. On return from the subroutine to adjust the wait counts, the count did not become minus or the adjustment was aborted, so the abend was issued.
Reg2 & Reg3=QCBID Reg4=SAVQCR Reg9=REQQCB Reg14=BAL	QCDEQ3	On return from the subroutine to find the QCB for a DEQ request, the QCB was not found, and the abend is issued.
Reg9=REQQCB Reg14=BAL	PURGE	For an attempt to purge a resource, the QCB chain is searched to find the QCB for this resource. If the QCB cannot be found, the abend is issued.
Reg5=REQBLK base Reg6=BAL(ALTWC)	ADJUP2	A call was made to adjust the wait count for this request and any request currently waiting for control of this resource. If on return from the subroutine to adjust wait counts, wait count becomes minus, the abend is issued.
Reg5=REQPREQ Reg14=BAL	ADJUP3A	If the end of the chain is found without finding the REQ while trying to update the REQ, the abend is issued.
Reg6=BAL(ALTWC) Reg14=BAL	ADJUP3B	On return from the subroutine to adjust the wait count, an impossible condition was indicated. Adjustment was terminated, and the abend is issued.
Reg6=BAL(ALTWC) Reg14=BAL	ADJDN	There are two BALs to XC10ABND after this label. On return from the subroutine to adjust the wait count, either a plus value or an impossible condition was indicated. Adjustment was terminated. For these error conditions, the abend is issued.
Reg5=REQPREQ Reg14=BAL	ADJDN2	When multiple requests are indicated in a QCB, a search is made to find the next request to process. If the next request cannot be found, the abend is issued.
Reg6=BAL(ALTWC) Reg14=BAL	ADJDN3	There are two BALs to XC10ABND after this label. On return from the subroutine to adjust the wait count, either a plus value or an impossible condition was indicated. Adjustment is terminated. For these error conditions, the abend is issued.
Reg5=CURRREQ Reg7=REQQEL Reg8=0 Reg14=BAL	ALTWCER1	No column was available for deadlock detection and an invalid request was made, so the abend is issued.
Reg5=CURRREQ Reg7=REQQEL Reg8=0 Reg14=BAL	ALTWCER3	No row was available for deadlock detection and an invalid request was made, so the abend is issued.
Reg14=BAL	ALTWCY1	When adjusting the wait count, it became negative, so the abend is issued.
Reg4=0 Reg14=BAL	RMVREQ1	While trying to remove an REQ from a QEL and a QCB, the REQ was not found, so the abend is issued.

Key	Label	Description
Reg4=0 Reg14=BAL	RMVREQ3	If this routine finds the request to be removed from the QCB, but is unable to find the previous request, the abend is issued.

*Case 2:*

Key	Label	Description
Reg6=BAL PRMFNCTN=0 Reg7=0 Reg10=PST	NOTFP2	Preinitialization is done. The requested control level has been set up and an attempt is made to assign a QEL. GETQCB was unable to allocate a QCB, so the abend is issued.
Reg6=BAL PRMFNCTN=01 Reg14=BAL	TSTFENQ	This is not an enqueue request, and a matching QCB was not found. GETQCB was unable to get space for a QCB, so the abend is issued.
Reg8=PRIORQCB	NOCORE1 from NOTFASTP	No space was available for an REQ after a QCB was obtained. The QCB is released and the abend is issued.
Reg8=BAL PRMFNCTN=0	ENQ30A	This is the first request for resources. The routine (ADDID) that adds a new resource ID to a list of resources held or requested by a QEL found no space to add the identification, so the abend is issued.
Reg0=0 PRMFNCTN=X'03' Reg8=BAL	ENQ31	A new request for resources will not share control with the existing request. An attempt was made to add the request identification to the list of resources (ADDID). No space was available, so the abend is issued.
REQPCTLW=REQCTLW Reg6=BAL(ALTWC)	ADJUPE0	A call was made to ALTWC to adjust the wait count for this request and any request currently waiting for control of this resource. If the adjustment was terminated, the abend is issued.

**Possible cause:** If you are using the maximum storage available for IMS ENQ/DEQ, and you still get pseudoabend 0775, get a normal ABEND dump at the time the error was detected. To get the memory dump, change the unconditional branch to ENQXTA to an unconditional branch to XC10ABND (a few instructions after label NOCORE).

**System action:** For Case 1, the IMS control region terminates abnormally. For Case 2, the application program terminates abnormally.

**Operator response:** For Case 1, none. For Case 2, decrease the system load by stopping the regions; then start the indicated SMBs and PSBs. If a BMP is not taking checkpoints and has used all available ENQ space, cancel the BMP.


**Programmer response:** For Case 1, contact IBM Software Support for assistance. For Case 2, correct the cause by increasing the maximum storage available for the ENQ or by decreasing the requirements for ENQs.

**Problem determination:** For Case 1: 1, 2, 4, 6, 13

**Source:** IMS abends


**Module:** DFSFXC10

**Related reference:**

 CICS Messages and Codes

**Related information:**

 DFS2450I (Messages and Codes)

 IMS problem determination (Messages and Codes)



---

**0776**

**Explanation:** A failure occurred in dynamic backout while processing a ROLB call. This failure can happen if the current online OLDS does not contain the checkpoint that is required by a ROLB call. The OLDS might have wrapped if its allocation is too small compared to the huge volume of processing.

**Analysis:** This is a pseudoabend set by modules DFSRBOI0 and DFSFXC50, and issued by DFSECP10, DFSECP20, or DFSFDMP0.

*For DFSECP10 or DFSECP20:*

Message DFS981I accompanies this abend and should be used to determine the reason for the failure. The Batch Backout utility must be run for the database named in the DFS981I message. No dump is taken for the abend.

*For DFSFDMP0:*

Message DFS0832I accompanies this abend and should be used to determine the reason for the failure. The Batch Backout utility must be run for the database named in the DFS0832I message. A dump is taken for this abend.

**System action:** The IMS dependent region that encounters the ROLB call failure terminates abnormally.

**Programmer response:** *For DFSECP10 or DFSECP20:*

See message DFS981I, which accompanies this abend, to determine the reason for the failure. Run the Batch Backout utility on the database named in the message.

*For DFSFDMP0:*

See message DFS0832I, which accompanies this abend, to determine the reason for the failure. Run the Batch Backout utility on the database named in the message.

**Problem determination:** *For DFSECP10 or DFSECP20:*

See message DFS981I to determine the reason for the failure.

*For DFSFDMP0:*

See message DFS0832I to determine the reason for the failure.

**Source:** IMS abends

**Module:** DFSECP10, DFSECP20, DFSFDMP0

**Related information:**

 [DFS981I \(Messages and Codes\)](#)

 [DFS0832I \(Messages and Codes\)](#)

---

**0777**

**Explanation:** The application program terminated abnormally because a potential resource was in a deadlock condition, or in the case of an external subsystem resource, the resource was unavailable or in a deadlock condition.

This abend is used to terminate a unit of work and possibly cause the associated transaction message to be rescheduled. In cases where the message is rescheduled, no abend message is produced. IMS produces 67FF deadlock records, which are written to the IMS log, and the 07 record completion code will show U0777 in all cases. CICS DBCTL thread messages are associated with CICS and cannot be rescheduled. The thread will terminate abnormally.

**Analysis:** This is a pseudoabend issued by module DBFIRC10, DBFLRH00, DFSESPR0, DFSLRH00, DFSFXC10, DFSCMD30, or DFSCMD60. If the abend is issued by DFSESPR0, the external subsystem passed IMS a NORMAL CALL return code 04 or a CREATE THREAD return code 24, which indicates a deadlock situation or a resource unavailable condition. The external subsystem can issue a message that indicates the cause of the problem. The external subsystem should be consulted to determine why resources were not available. IMS takes the following actions:

- For a message-driven process, places the input message back at the head of the input queue (MPP or IFP).
- Discards all uncommitted message output and database updates.

When issued from DFSCMD30 or DFSCMD60, the abend breaks a potential deadlock among a DBR command, a MODIFY COMMIT command, and an MPP region with an application program issuing an ICMD or CMD call.

## 0778

| A U0777 abend might also occur if an application program accesses an IMS HALDB partition by using PROCOPT=GOx and the partition is unavailable because a database command is in progress against it, and PPUR=Y is specified in the DFSVSMxx PROCLIB member. In this case, the associated message, if any, is rescheduled, but no 67FF records are produced.

| A U0777 abend might also occur when a non-message-driven BMP that accesses an external subsystem receives a resource unavailable condition, and the PSB includes a GSAM PCB. In this case, the action to be taken depends on the specification of the GSROLBOK parameter in PSBGEN for the associated PSB. If GSROLBOK=YES is specified, the application does not terminate with abend U0777 but receives a failure indication on the ESS call, which is followed by an internal IMS ROLB. For DB2, this is SQL CODE -911. Otherwise, the BMP ends abnormally with abend U0777.

| **System action:** The application program terminates abnormally. If it was running in an MPP or IFP region, the message it was processing will automatically be rescheduled.

| **Operator response:** No action is required.

| **System programmer response:** If a BMP failed, rerun the job.

| **Source:** IMS abends

| **Module:** DBFIRC10, DBFLRH00, DFSESPR0, DFSLRH00, DFSFXC10, DFSCMD30, DFSCMD60

---

## 0778

**Explanation:** A ROLL call was issued by a user application program, or a failure during the Database Backout utility in another region resulted in all regions being terminated as a result of an internally issued ROLL call.

**Analysis:** 0778 is a pseudoabend detected by the DL/I call analyzer program, DFSDLA00. A SNAP of the control blocks is required to diagnosis this problem. Print the type X'67' log records using the field select and formatting print program, DFSERA10. The OPTION PRINT statement must specify EXITR=DFSERA30.

Two situations will initiate this abend, a user-issued ROLL call or failure of the Database Backout utility. In the first situation, the user program issues a ROLL call when it determines that some invalidity exists in the processing it has done. All database activity since the last sync point is backed out and the dependent region in which the user is executing is terminated *without* a dump. In the second situation, the failure of the Database Backout utility will result in all dependent regions being terminated. The SNAP for this situation will contain the control blocks of the other dependent regions that were active at the time of backout failure. As before, no dump of the dependent region is output.

After backout is completed, the original transaction is discarded (if it is discardable) and is not executed again. To notify the remote transaction programs, the system issues the z/OS APPC verb, ATBCMDC TYPE(ABEND), specifying the TPI. This causes all active conversations (including any spawned by the application program) to be DEALLOCATED TYPE(ABEND\_SVC).

A message processing program (MPP) is automatically rescheduled.

Key	Label	Description
Situation one (1)	FUNCLOW	The user application program has issued a ROLL call.
Situation two (2)	DCPCBOK, RET000	A database backout operation failed in another region resulting in the other active dependent region's termination.

**Possible cause:** User-issued ROLL call, review application program database backout terminated abnormally. Abend 0778 is normal for this situation.

**System action:** The application program terminates abnormally without a storage dump.

**Operator response:** A message processing program (MPP) is automatically rescheduled.

**Source:** IMS abends

**Module:** DFSDLA00

---

**0779**

**Explanation:** The application program attempted to issue more \*Q command codes between sync points than specified in the MAXQ parameter of the PSBGEN statement in the PSB.

**Analysis:** This is a pseudoabend initiated in the DL/I call analyzer module, DFSDLA00. Register 12 is the base register.

The applicable registers as they were at entry-to-abend may be found at the thirteenth-level save area in the PST. They start at offset 'C' in the save area that starts at label PSTSAV13 (using the dump of control blocks that was written to the system log).

DFSDLA00, when it determines that the \*Q command codes was specified or the call is being processed, will compare PSBCQCNT (the current \*Q count) with PSBMQCNT (the maximum \*Q count). PSBCQCNT must be less than PSBMQCNT or the abend is issued. Both of these fields are contained in the PSB prefix. The value stored at PSBMQCNT is the result of what was coded in MAXQ field of the PSBGEN statement for the PSB. This represents the maximum number of \*Q commands, which will be allowed for the application program between sync points. It is used as a check to disallow an application from enqueueing more resources than it needs and perhaps going into a loop.

**System action:** The control region continues normally; however, the message region terminates abnormally.

**Programmer response:** If the application program was in error (for example, looping), correct it. If the PSB contains too low a value for MAXQ, correct the PSB and regenerate the ACB.

**Problem determination:** 4, 19

The number of \*Q commands issued since the last sync point is accumulated in the PSB.

**Source:** IMS abends

**Module:** DFSDLA00

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

**0780**

**Explanation:** The IMS latching routine detected a critical system error.

**Analysis:** This is a standard abend, issued by modules DFSCLM00, DFSCLM10, DFSCLM20, DFSDBDR0, DFSDTTA0, and DFSASK00. If the abend is issued from DFSCLM00, DFSCLM10, or DFSCLM20, you can find the abend subcode in the text of message DFS6551, register 15, or in the PSTLATRC field. Register 14 contains the address where the error was detected in the CLM modules. Register 10 contains the address of the input parameter area (use DFSCLM dsect). If the abend is issued from DFSDBDR0, register 15 contains the return code from module DFSCLM00.

To determine the latch type requested at the time of the abend, check CLMPTYPE in the parameter area.

To determine which latches were owned by the unit of work (UOW) at the time of the abend, check the Common Latch List Element (CLLE) block that is attached to SAP (Save Area Prefix) using SAPACLE. The CLLE definitions are listed below.

The following codes are for DFSCLM00 abends. DFSCLM00 gets the latch for the requester:

**Codes****Explanation****X'001'**

Reserved.

**X'002'**

Requester currently has a latch allocated at this level.

**X'003'**

The requested latch level is less than the highest latch held.

**X'004'**

The requester for the exclusive latch is the current owner of the requested latch.

## 0780

### X'005'

The post code of an ECB on the latch manager. Wait queue is not a latch manager post code.

### X'006'

The requester for the share latch is the current owner of the requested latch.

The following codes are for CLM01000 (a subroutine of DFSCLM00) abends. CLM01000 allocates DLLE to the latch requesters:

#### Codes

##### Explanation

### X'011'

Unable to allocate an AWE block for an ITASK requester. Attempting to allocate AWE to call DFSBCB00 for CLLE.

### X'012'

Unable to allocate the CLLE block for an ITASK requester.

### X'013'

Unable to deallocate the AWE block for an ITASK requester. AWE is used to allocate CLLE.

### X'014'

Unable to allocate an AWE block for a non-ITASK requester. Attempting to allocate AWE to call DFSBCB00 for CLLE.

### X'015'

Unable to allocate the CLLE block for a non-ITASK requester.

### X'016'

Unable to deallocate the AWE block for an ITASK requester. AWE is used to allocate CLLE.

The following codes are for CLM01000 (a subroutine of DFSCLM00) abends. CLM01000 allocates DLLE to the latch requesters:

#### Codes

##### Explanation

### X'011'

Unable to allocate an AWE block for an ITASK requester. Attempting to allocate AWE to call DFSBCB00 for CLLE.

### X'012'

Unable to allocate the CLLE block for an ITASK requester.

### X'013'

Unable to deallocate the AWE block for an ITASK requester. AWE is used to allocate CLLE.

### X'014'

Unable to allocate an AWE block for a non-ITASK requester. Attempting to allocate AWE to call DFSBCB00 for CLLE.

### X'015'

Unable to allocate the CLLE block for a non-ITASK requester.

### X'016'

Unable to deallocate the AWE block for an ITASK requester. AWE is used to allocate CLLE.

The following codes are for DFSCLM10 abends. DFSCLM10 releases the latch for the requester:

#### Codes

##### Explanation

### X'102'

Requester does not have a latch allocated at this level.

### X'103'

The requester is not the current owner of the requested latch. The latch request mode is exclusive.

**X'104'**  
Latch request is shared; resource not in shared mode.

**X'105'**  
Latch request is shared; resource count is positive.

The following codes are for CLM11000 (subroutine of DFSCLM10) abends. CLM11000 deallocates CLLE for the latch requester.

**Codes**

**Explanation**

**X'111'**  
Unable to allocate an AWE block for an ITASK requester. AWE blocks are used to deallocate CLLE for a requester using a DFSBCB call.

**X'112'**  
Unable to deallocate a CLLE block for an ITASK requester.

**X'113'**  
Unable to deallocate an AWE block for an ITASK requester. AWE blocks are used to deallocate CLLE for a requester using a DFSBCB call.

**X'114'**  
Unable to allocate an AWE block for a non-ITASK requester. AWE blocks are used to deallocate CLLE for a requester using a DFSBCB call.

**X'115'**  
Unable to deallocate a CLLE block for a non-ITASK requester.

**X'116'**  
Unable to deallocate an AWE block for a non-ITASK requester. AWE blocks are used to deallocate CLLE for a requester using a DFSBCB call.

The following codes are for DFSCLM20 abends. DFSCLM20 recovers (releases) latches by failed units of work (that is, SAPs).

**Codes**

**Explanation**

**X'201'**  
DFSCLM20 was posted with an invalid post code. DFSCLM20 is posted by the initialization routine. DFSCLM20 is an AWE processor.

**X'202'**  
Latch type to recover is invalid.

**X'203'**  
Latch mode (share, exclusive, any) passed is invalid.

**X'204'**  
Exclusive latch being recovered is not owned by SAP.

**X'205'**  
Exclusive latch — No recovery exit routine is specified.

**X'206'**  
Exclusive latch — Recovery exit routine is specified but none is provided.

**X'207'**  
Exclusive latch — Recovery exit routine is specified and driven but failed and returned a nonzero return code.

**X'208'**  
Shared latch is not owned.

| **X'209'**  
| Shared latch -- A recovery exit routine was specified for the latch type being recovered, but the routine address  
| was zero.

## 0783 • 0790

| X'20A'

| Shared latch -- A recovery exit routine was specified and was called. The recovery exit returned a nonzero return  
| code, indicating that the latch was not recoverable.

**System action:** IMS terminates abnormally.

**Programmer response:** Determine the cause of the problem and restart IMS.

**Problem determination:** 4, 5, 6, 35

**Source:** IMS abends

**Module:** DFSCLM00, DFSCLM10, DFSCLM20, DFSDBDR0, DFSDTTA0, DFSASK00

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

### 0783

**Explanation:** The DMB and PSB Pool Space Manager (DFSDPDM0) attempted to delete and release the space for a PSB that is not on its master PDIR's chain.

**Analysis:** This is a standard abend issued by module DFSDPDM0.

Register 12 in the abend SVRB registers is the base register. Register 8 is used as a BAL register. Register 7 points to the PDIR being deleted. Register 10 points to the PST.

**System action:** The system abnormally terminates.

**Programmer response:** No action is required.

**Problem determination:** 4 and 5, or 25

**Source:** IMS abends

**Module:** DFSDPDM0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

### 0790

**Explanation:** The IMS USE Manager routine detected a critical system error.

**Analysis:** This is a standard abend that can be issued by modules DFSUSE00, DFSUSE10, or DFSUSE20.

Register 15 contains the abend subcode. Register 14 contains the address where the error was detected in the USE models. Register 10 contains the address of input parameter area (use DFSUSE DSECT) for DFSUSE00 and DFSUSE10 (use register 9 in DFSUSE20).

DFSUSE00 does in use, lock, connect, merge or inquire functions for the requester.

#### Codes

##### Explanation

X'001'

Reserved.

X'002'

Invalid block type has been specified or the use anchor pointer is zero.

X'003'

The requester did not issue in-use function before issuing a lock function.

X'004'

The INUSE/LOCK/CONNECT wait queue has been posted by someone other than the Use Manager.

X'005'

Use request = LOCK; the requester's use list (CULE) indicates an element has already been allocated to the resource, but the element is not on the resource element chain.

- X'006'**  
User request = LOCK. The requester is the current lock owner of the resource.
- X'007'**  
User request = LOCK/CONNECT. Another INUSE/LOCK/CONNECT is active for this unit of work (UOW). A UOW can have only one active LOCK/CONNECT. Additional requests create the potential for a deadlock.
- X'008'**  
The user request was INUSE. The requester is a current sharer of the resource.
- X'009'**  
The user request was INUSE/LOCK/CONNECT. The resource has been logically deleted.
- X'010'**  
The user request was MERGE. The requestor does not have a lock on the *from* resource.
- USE01000, a subroutine in DFSUSE00, allocates CULE to latch requesters.
- Codes**  
**Explanation**
- X'011'**  
Unable to allocate an AWE block for the ITASK requester. Attempting to allocate AWE to call DFSBCB00 for CULE.
- X'012'**  
Unable to allocate CULE block for ITASK requester.
- X'013'**  
Unable to deallocate AWE block for ITASK requester. AWE used to allocate CULE.
- X'014'**  
Unable to allocate an AWE block for the non-ITASK requester. Attempting to allocate AWE to call DFSBCB00 for CULE.
- X'015'**  
Unable to allocate CULE block for non-ITASK requester.
- X'016'**  
Unable to deallocate an AWE block for ITASK requester. AWE used to allocate CULE. USE01000, a subroutine in DFSUSE00, deallocates a CULE.
- X'021'**  
reserved
- X'022'**  
Unable to deallocate a CULE block for an ITASK requester.
- X'023'**  
reserved
- X'024'**  
Unable to allocate an AWE block for the non-ITASK requester. Attempting to allocate an AWE to call DFSBCB00 to release the CULE.
- X'025'**  
Unable to deallocate a CULE block for a non-ITASK requester.
- X'026'**  
Unable to deallocate an AWE block for a non-ITASK requester. The AWE is used to deallocate the CULE.
- X'061'**  
User request = INUSE. During retry, the element could not be found on the resource element chain.
- X'062'**  
The INUSE/LOCK/CONNECT wait queue was posted by someone other than the Use Manager.
- X'063'**  
User request = INUSE/LOCK/CONNECT. The resource was logically deleted.

## 0790

USE03000, a subroutine in DFSUSE00, allocates DCSL latch.  
USE07000, a subroutine in DFSUSE00, allocates CONV latch.

### Codes

#### Explanation

**X'041'**

Either CULEF2LS or CULEF2LE is not set. This problem is an internal USE Manager error.

**X'042'**

reserved

**X'043'**

reserved

**X'044'**

The latch is already owned.

**X'045'**

A latch hierarchy violation occurred.

**X'046'**

An exclusive latch could not be obtained.

**X'047'**

A shared latch could not be obtained.

**X'048'**

A latch manager internal error occurred.

USE04000, a subroutine in DFSUSE00, release DCSL latch.  
USE08000, a subroutine in DFSUSE00, release CONV latch.

### Codes

#### Explanation

**X'051'**

The requestor does not own the latch.

**X'052'**

The requestor does not have the Common Latch List Element (CLLE) .

DFSUSE10 does no-use, unlock, or disconnect functions.

### Codes

#### Explanation

**X'101'**

A common use element block (CULE) is not allocated to the unit of work (UOW).

**X'102'**

An invalid block type was specified or the use anchor pointer is zero.

**X'103'**

If request = NOUSE, the requester does not have an active INUSE on this resource. If the request = UNLOCK/DISCONNECT, a LOCK/CONNECT has not been done on the resource; or the resource header is zero.

**X'104'**

The workID/callID in the token use list does not match the workID/callID in the parmameter list.

**X'105'**

Invalid post code detected in the lock use queue logic.

**X'106'**

Reserved.

**X'107'**

The requester does not have use ownership on the requested resource.



USE11000, a subroutine of DFSUSE10, deallocates CULE for the latch requester.

**Codes**

**Explanation**

**X'111'**

Unable to allocate an AWE block for the ITASK requester. AWE blocks are used to deallocate CULE for requester using DFSBCB call.

**X'112'**

Unable to deallocate CULE block for the ITASK requester.

**X'113'**

Unable to deallocate an AWE block for the ITASK requester. AWE blocks are used to deallocate CULE for a requester using DFSBCB call.

**X'114'**

Unable to allocate an AWE block for non-ITASK requester. AWE blocks are used to deallocate CULE for requester using DFSBCB call.

**X'115'**

Unable to deallocate CULE block for a non-ITASK requester.

**X'116'**

Unable to deallocate an AWE block for a non-ITASK requester. AWE blocks are used to deallocate CULE for requester using DFSBCB call.

**X'120'**

An invalid block type was specified or the use anchor point is zero.

DFSUSE20 recovers (releases) in use ownerships from the AWE queue.

**Codes**

**Explanation**

**X'201'**

Initialization - invalid post code from idle state.

**X'202'**

Invalid recovery type has been specified.

USE21000, a subroutine of DFSUSE20, deallocates use for a unit of work.

**Codes**

**Explanation**

**X'210'**

Invalid block type was specified.

**X'211'**

Invalid recovery token specified.

**X'212'**

Unit of work's CULE table indicates that resource is locked. However, resource header ownership is not allocated to that unit of work.

**X'213'**

Unit of work's CULE table indicates that resource is in use. However, resource header ownership is not allocated to any unit of work.

**X'214'**

Unit of work's CULE table indicates that resource is in use. However, resource header ownership is not allocated to that unit of work.

**For DFSSLC00, DFSSLC10:**

The IMS latching routine detected a critical system error.

## 0791

Abend 0790 is a standard abend that can be issued by modules DFSSLC00 or DFSSLC10. Register 15 contains the abend subcode. Register 14 contains the address where the error was detected in the SLC modules. Register 10 contains the address of the input parameter area (use DFSSLC DSECT) for DFSSLC00 and DFSSLC10 (use register 9 in DFSSLC20). SLCPTYPE in the parameter area determines the latch type requested at the time of the abend.

The subcodes for DFSSLC00 GETs latch for requester are as follows:

### Codes

#### Explanation

X'01'

Passed ECB mismatched with ECB attached to SAP.

X'02'

Unable to allocate parameter area for DFSBCB GET for AWE.

X'03'

Latch allocation failure. Latch held.

X'04'

Latch deallocation failure.

X'05'

Unable to deallocate parameter area.

The subcodes for DFSSLC10 RELs latch for requester are as follows:

### Codes

#### Explanation

X'01'

Passed ECB mismatched with ECB attached to SAP.

X'02'

Invalid search type has been specified.

**System action:** IMS terminates abnormally.

**Programmer response:** Determine the cause of the USE manager by referring to the memory dump analysis.

**Source:** IMS abends

**Module:** DFSUSE00, DFSUSE10, DFSUSE20

---

## 0791

**Explanation:** The Open Database Access (ODBA) sync point task control block (TCB) initialization or processing detected a critical system error.

**Analysis:** 0791 is a standard abend that can be issued by modules DFSAERS0, DFSAERN0, DFSAERC0, DFSAERG0, DFSAERM0, and DFSAIBEQ.

**For DFSAERS0:**

### Codes

#### Explanation

X'04'

BCB QSAV get failed.

X'08'

DFSCDSP call failed.

X'0C'

IMODULE GETMAIN failed.

X'10'

DFSCWU call failed.

**X'14'**  
DFSCIR call failed.

**X'18'**  
IMODULE LOAD failed.

**X'1C'**  
IPOST failed.

**X'20'**  
IMODULE GETSTOR failed.

**X'24'**  
Name/Token Service call failed.

**X'28'**  
A call to abnormally terminate an ODBA thread failed.

#### For DFSAERN0:

##### Codes

##### Explanation

**X'04'**  
DFSBCB GET call failed.

**X'20'**  
ODS Function invalid.

**X'28'**  
A call to terminate an ODBA thread abnormally failed.

#### For DFSAERC0:

##### Codes

##### Explanation

**X'04'**  
DFSBCB GET call failed.

#### DFSAERG0

##### Codes

##### Explanation

**X'2C'**  
Attach of DFSAERM0 failed. R14 contains the return code from ATTACH.

**X'400'**  
ODBM compatibility was in use but the ODBA address space was not registered with RRMS or ODBM. This is probably an internal error.

**X'404'**  
ODBA failed to associate the Recovery Resource Management Services (RRMS) context after completing the ODBM compatibility call. This is probably an internal error.

**X'408'**  
ODBA failed to disassociate the current RRMS context before making the ODBM compatibility call. This is probably an internal error.

**X'40C'**  
An allocate PSB (APSB) call from ODBA to ODBM failed. Verify that the PSB name is correct and defined to IMS.

**X'410'**  
ODBA failed to locate the local PCB. Verify the application is using a valid PCB name.

#### DFSAERM0

## 0793 • 0794

### Codes

#### Explanation

#### X'30'

A DFSAERM0 initialization error occurred. AIBRSA1 is missing or supplied with an invalid subfunction code. An INIT parameter error was found.

#### X'34'

DFS AERM0 was unable to obtain adequate storage to complete initialization. A failure to obtain AERMWORK area was encountered.

#### X'38'

A DFSAERM0 initialization error occurred. A failed to establish ESTAE occurred.

**System action:** IMS terminates abnormally.

**Operator response:** Emergency restart IMS to resume processing.

**Problem determination:** 1, 3, 4, 5, 11

**Source:** IMS abends

**Module:** DFSAERS0, DFSAERN0, DFSAERC0, DFSAERG0, DFSAERM0, DFSAIBEQ.

#### Related information:

Chapter 2, "IMS problem determination," on page 13

---

### 0793

**Explanation:** Module DFSISTS0 was unable to open either the LOGIN or LOGOUT DD statements.

**Analysis:** This abend is a standard abend issued by module DFSISTS0.

**Possible cause:** A z/OS error occurred.

**System action:** Module DFSISTS0 terminates abnormally.

**Programmer response:** No action is required.

**Problem determination:** 1, 2, 3, 8

Check the JCL for each ddname. Correct the JCL, and resubmit the job.

**Source:** IMS abends

**Module:** DFSISTS0

#### Related information:

Chapter 2, "IMS problem determination," on page 13

---

### 0794

**Explanation:** Module DFSISTS0 attempted to link to the SORT program and encountered a nonzero return code. A zero return code is required.

**Analysis:** This is a standard abend issued by DFSISTS0, the statistics formatter program. DFSISTS0 invokes the SORT program to put each complete message together with either its ENQ, DEQ, or CANCEL records, or all three.

The program status word (PSW) at entry-to-abend will point to the instruction within label LINKRC from which the abend (SVC 13) is issued. The return code from the LINK(SVC 6) is saved at label SAVER15.

Key	Label	Description
Reg1=X'8000031A'	SORTINIT	The LINK for the SORT program failed with a nonzero return code in register 15.

**System action:** Module DFSISTS0 terminates abnormally.

**Problem determination:** 1, 2, 3, 4, 8, 25, 35

**Source:** IMS abends

**Module:** DFSISTS0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

0795

**Explanation:** An invalid record type was encountered in the output phase of the SORT program. Register 7 points to the invalid record. This is probably a SORT program problem.

**Analysis:** This is a standard abend issued by DFSISTS0. The log data set records have been read into an in-line SORT using an input exit routine (E15ON). The records now are passed to DFSISTS0 using the output exit routine (E35ON) for further processing. DFSISTS0 validates the records as they are passed back, and will abend if an invalid record type is encountered. The valid record types are:

**Codes**

**Explanation**

X'36'

DEQ record

X'35'

ENQ record

X'34'

CANCEL record

X'33'

Free queue record

X'31'

GU record

X'03'

Output record

X'01'

Input record

The program status word (PSW) at entry-to-abend points to the instruction within label BADSORT from which the abend (SVC 13) is issued. Register 7 in the abend SVRB points to the invalid record, and register 12 is the base register.

Key	Label	Description
Reg1=X'8000031B' Reg7=A(invalid record)	CHECKED	The SORT output exit routine (E35ON) passed DFSISTS0 an invalid record type.

**Possible cause:** SORT program failure

**System action:** Module DFSISTS0 terminates abnormally.

**Problem determination:** 1, 2, 3, 4, 8, 25, 35

**For APAR processing:** Console sheet, memory dump, copy of IMS log

**Source:** IMS abends

**Module:** DFSISTS0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

## 0796

**Explanation:** While processing a REPL call, the replace module DFSDLDR0, within DFSDL00, found that the root segment was not locked by Retrieve.

**Analysis:** This is a pseudoabend issued from module DFSDLDR0 and is traced in the DL/I Trace Table in a series of entries, each of which is identified by a character D (X'C4') in its first byte. This series of entries immediately precedes the current entry in the trace table.

The important field in each trace entry is the second word (offset 4), called ENTRY1. This word identifies the abend and the routine within the Delete/Replace module that encountered the problem.

The ENTRY1 word in the first entry of the Delete/Replace abend trace is the key used in the following table:

Key	Description
41410118	DFSDLDR0 checks if the root segment has been locked by Retrieve before obtaining an update lock for replace processing. This abend is issued if DSGTOKEN=0 (does not contain the root lock token).

**System action:** The application program terminates abnormally.

**Problem determination:** 17a through 17d, 17f, 17g, 35

**Source:** IMS abends

**Module:** DFSDL00

**Related information:**

Chapter 2, "IMS problem determination," on page 13

## 0797

**Explanation:** Delete or Replace received an unexpected segment from the buffer handler.

**Analysis:** This is issued from module DFSDL00 and can be a standard or a pseudoabend. Abend 0797 is traced in the Buffer Handler Trace Table in a series of entries, each of which is identified by a character D (X'C4') in its first byte. This series of entries immediately precedes the current entry in the trace table.

The important field in each trace entry is the second word (offset 4), called ENTRY1. This word identifies the abend and the routine within the Delete/Replace module that encountered the problem.

The ENTRY1 word in the first entry of the Delete/Replace abend trace is the key used in the following table:

Key	Description
4343011C	In attempting to replace a segment which is neither variable-length nor compressed, the Replace module received an unexpected segment type from the buffer handler.
4343012C	In attempting to replace a variable-length or a compressed segment, IMS finds an invalid segment code navigating down to the segment that is being replaced.
4343021C	In attempting to replace a segment which is neither variable-length nor compressed, after interfacing with the logger to log the old copy, the segment in the buffer was not the same one as before logging.
4343031C	In attempting to replace a variable-length or a compressed segment, the Replace module received an unexpected segment type from the buffer handler.
4343041C	In attempting to replace a variable-length segment, the Replace module found that the old copy of the segment was separated. However, the database was not HD-organized, or the segment code in the data portion of the segment did not match the segment code in the prefix.
4343051C	In attempting to replace a variable-length segment, the Replace module found that the old copy of the segment was separated. However, the data portion pointed to by the prefix did not indicate that it was separated.

**System action:** The application program terminates abnormally.

**Problem determination:** 17a through 17d, 17f, 17g, 35

**Source:** IMS abends

**Module:** DFSDL00

**Related information:**

Chapter 2, "IMS problem determination," on page 13

## 0799

**Explanation:** The DELETE/REPLACE module called the user-written segment edit/compression routine to expand or compress a segment. Upon return, Delete/Replace found the new LL for the segment exceeded the maximum allowable LL.

When using Fast Path, upon return from the exit routine, one of the following problems was encountered:

- The key was altered (subcode 1).
- The length was incorrect (subcode 2).
- The compressed segment length has changed (subcode 3).

**Analysis:** Abend 0799 is issued from module DFSDL00 and can be a standard or a pseudoabend.

Abend 0799 is traced in the Buffer Handler Trace Table in a series of entries, each of which is identified by a character D (X'C4') in its first byte. This series of entries immediately precedes the current entry in the trace table.

The important field in each trace entry is the second word (offset 4), called ENTRY1. This word identifies the abend and the routine within the Delete/Replace module that encountered the problem.

The ENTRY1 word in the first entry of the Delete/Replace abend trace in the key shown in the following table:

Key	Description
04040134	After decompressing a segment, the resultant length is less than 2.
04040234	After decompressing a segment, the resultant length is greater than DMBSGMX
04040334	After compressing a segment, the resultant length is less than 2.
04040434	After compressing a segment, the resultant length is greater than DMBSGMX.
04040534	After compressing a fixed-length segment, the resultant length is greater than DMBDL plus 10.

**Possible cause:** The causes include:

- Compressed-data length for a variable length segment was greater than DMBSGMX.
- Compressed-data length for a fixed-length segment was greater than DMBDL plus 10.
- Error in Edit/Compression routine, or the Edit/Compression routine was passed meaningless data.

### For DBFCMP10:

After returning from the your segment compression exit routine, the key was altered or the length (LL) was incorrect.

This abend is a pseudoabend issued by module DBFCMP10 when the key is altered. Register 1 contains the abend code and register 2 contains subcode 1.

**Reg3** A (segment after invoking user exit routine)

**Reg4** A (segment before invoking user exit routine)

**Reg7** Key length

**Reg10** A (EPST)

DBFCMP10 also issues this abend under the following conditions (register 1 contains the abend code and register 2 contains subcode 2).

1. The LL is less than the specified minimum length in database description (DBD).
2. The LL is greater than the specified maximum length (up to 10 bytes) and less than the specified CI size (less than 120 bytes overhead) in DBD.

**Reg4** A (MLTE)

## 0800

**Reg5** returned LL  
**Reg6** ABEND Code  
**Reg7** CI size (120 bytes)  
**Reg8** A (DEDB area control (DMAC))  
**Reg10** A (EPST)

You can find the maximum (SDBLMAX) and the minimum (SDBLMIN) by locating A (SDBS) from MLTESDBS. The CI size is DMACBLKL.

**System action:** The application program terminates abnormally.

**Programmer response:** Correct the segment edit/compression routine, and rerun the job.

**Problem determination:** 17a through 17g, 35

**Source:** IMS abends

**Module:** DFSDL00, DBFCMP10

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

## 0800

**Explanation:** The Segment Edit/Compression exit routine that you are using has expanded a variable-length compressed segment. The segment returned had a length greater than the maximum allowable or was less than or equal to 2 bytes.

During processing of the VLEXP routine in module DFSDLR00, a PCB with PROCOPT=GO, GON, or GOT (GOx) detected an invalid length in the LL field of a compressed segment. The length value in the LL field exceeded the maximum segment length by more than 10 bytes or was equal-to-or-less than 2 bytes. An update transaction might have changed the segment data before it expanded for the PROCOPT=GOx PCB.

- | When the compression exit for Fast Path DEDB is being used and the expanded segment length is greater than the segment work area because PROCOPT=GOx lost its position, this abend will be issued to prevent overlay.

**Analysis:**

- | 0800 is a pseudoabend issued from several modules.

**For DFSDLR00:** When the abend is detected, Retrieve saves the registers (register 14 through register 12) starting at offset 'X'C' in the last save area in the PST. This save area starts at label PSTSAVL.

Key	Label	Description
Reg0=WKA size Reg15=concatenated key length Reg0=000320 Reg7=PST	VLEN2	Register 15 points to the segment length. It is compared with DMBSGMX, and if the segment length is greater, the abend is issued.
	COMPG020	Register 3 contains the segment length. If Register 3 is less than or equal to 2 bytes, the abend is issued.

- | **For DBFCMP10 and DBFCMP11:** IMS detected an overlaid problem due to PROCOPT GOx processing.

**System action:** The batch or message processing region is abnormally terminated. The control region continues normally.

**Programmer response:** Correct the Segment Edit/Compression exit routine.

If a PCB detected an invalid length in the LL field, rerun the PROCOPT=GOx transaction. If the abend is reissued, check the LL field of the segment in the database.

**Problem determination:** 4, 17c

**Source:** IMS abends

- | **Module:** DFSDLR00, DBFCMP10, DBFCMP11

**Related information:**



Chapter 2, "IMS problem determination," on page 13



## Chapter 18. IMS abend codes 0801 - 0850

This topic identifies and explains the following abend codes issued by IMS. Each code is 4 decimal digits.

### 0801

**Explanation:** There are three reasons to get this abend code:

1. The "PSBIOAWK" work area is too small to hold the concatenated keys or segments to be returned. This problem might be caused if you defined an area too small in the PSBGEN (PSBGEN statement, IOASIZE= parameter) or if a path call was specified in such a way that the combined length of the concatenated segments to be returned to the application would be greater than 65535 bytes. This problem could be an IMS error if the IOASIZE parameter is omitted in the PSBGEN, in which case the size of the "PSBIOAWK" would be determined by the block builder.
2. A nonzero return code from storage manager in response to a GET BUFF call from DFSDLR00 to the "DBWP" pool. A nonzero return code from storage manager typically indicates that there is not enough room in the "DBWP" pool to satisfy the request.
3. The length field in a variable length segment is either less than two or greater than the defined maximum length.

#### Analysis:

0801 is a pseudoabend issued from module DFSDLR00. If the Retrieve trace was active, then it can be used to locate the subroutine that initiated the abend. The contents of registers 14 through 12 have been saved starting at offset X'C' in the last save area in the PST. This save area starts at label PSTSAVL. Register 14 is indicated by an X'AA' in the first byte. If the Retrieve trace was not active, then the base address contained in register 12 can be used to determine the subroutine that initiated the abend. Normal register usage is as follows:

Register 3=JCB  
 Register 4=level table  
 Register 5=SDB  
 Register 7=PST  
 Register 8=DSG

If the level table is not in register 4, it is stored at SAVELEVR in the JCB.

Key	Label	Description
Reg0=WKA size Reg15=concatenated key length	RDLR200 (RDLRABND)	The retrieve is for the concatenated key length for an 01 type logical child with a physical key. A comparison is made between the concatenated key length in register 15 and the size of the work area in register 0. The size of the work area is the difference between the values of PSBIOAWK and PSBSSAWK. If the value in register 15 is greater, the abend is issued.
Reg0=WKA size Reg15=00000004	RDP010 (RDP0RABND)	A comparison is made between the data length of the input SDB in register 1 and the size of the work area. The size of the work area is the difference between the values of PSBIOAWK and PSBSSAWK. If the value in register 1 is greater, the abend is issued.
Reg14=BAL(GETBUF)	CCT0MEB (CCT0MEER)	An IGETBUF is issued for space equal to the key length. On return from DFSIGBUF, if register 15 is nonzero, the abend is issued.
JCBSGMX=WKA size	MVSGA (MVSEXAB)	A comparison is made between the data length, which is the sum of JCBSEGLL and JCBSGMV, and the size of the work area. The work area is calculated as the difference between PSBIOAWK and PSBSSAWK, and is stored at JCBSGMX. If the data length exceeds the work area size, the abend is issued.

## 0802

Key	Label	Description
Reg0=length of data	MVSQ (MVSEXAB)	The data length (JCBSEGLL) must be within a valid range. If JCBSEGLL is less than 2 or greater than the maximum length (DMBSGMX), then the abend is issued.

*Possible cause:* Incorrect PSBGE

*For APAR processing:* Listings of DBDGEN and PSBGEN

**System action:** The application program is terminated abnormally.

**Programmer response:** Ensure that the value defined for the "IOASIZE=" parameter complies with the specifications for that parameter. If the call at the time of the ABEND was a path call, ensure that the combined length of all concatenated segments to be returned does not exceed 65535 bytes. If this size has been exceeded, the user might want to issue a series of path calls, rather than a single call, so that the 65535 byte limitation is not exceeded during any one call. If the abend is the result of an incorrect length field in a variable length segment, correct the invalid length.

**Problem determination:** 17a, 17b

**Source:** IMS abends

**Module:** DFSDLR00

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

## 0802

**Explanation:** Replace tried to obtain space from space management to replace a variable-length segment. The new LL or minimum LL' is greater than the old LL. Therefore, Replace requested additional space from space management, but received a return code indicating that no more space is available in the data set.

**Analysis:** 0802 is issued from module DFSDL00 and can be a standard or a pseudoabend.

Abend 0802 is traced in the Buffer Handler Trace Table in a series of entries, each of which is identified by a character D ('X'C4') in its first byte. This series of entries immediately precedes the current entry in the trace table.

The important field in each trace entry is the second word (offset 4), called ENTRY1. This word identifies the abend and the routine within the Delete/Replace module that encountered the problem.

The ENTRY1 word in the first entry of the Delete/Replace abend trace in the key shown in the following table:

Key	Description
43430028	The Replace module issued a space request to lengthen a segment. The return code from space management indicated no space available.

*Possible cause:* A rearrangement of the data set is needed to make free, or potentially free space available for access. This can be accomplished by an UNLOAD, followed by a reload of the data set.

**System action:** The application program terminates abnormally.

**Programmer response:** Reorganize the database.

**Problem determination:** 17a through 17g, 35

**Source:** IMS abends

**Module:** DFSDL00

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

## 0803

**Explanation:** Delete or replace was unable to find a logically related segment (usually a logical parent or a physically paired logical child); or a logical parent's counter is negative if decremented for deletion of a logical child; or a logical child could not be found on a logical twin chain.

If you were unable to find a physically paired logical child, one of two conditions exists:

- The paired segment does not exist.
- The paired segment intersection data is not equal to the original intersection data.

If it is this last condition, it is caused by the user's load program failing to load both copies with equal intersection data.

**Analysis:** This is issued from module DFSDL00 and can be a standard or a pseudoabend.

Abend 0803 is traced in the Buffer Handler Trace Table in a series of entries, each of which is identified by a character D ('X'C4') in its first byte. This series of entries immediately precedes the current entry in the trace table.

The important field in each trace entry is the second word (offset 4), called ENTRY1. This word identifies the abend and the routine within the Delete/Replace module that encountered the problem.

The ENTRY1 word in the first entry of the Delete/Replace abend trace in the key shown in the following table:

Key	Description
3131012C	A logical child segment is to be deleted. However, the counter in the logical parent prefix is 0, indicating that the logical parent has no logical children.
3535012C	In deleting a logical child segment which the Delete module determined to be the first on the logical chain, it was found that the LCF pointer in the logical parent did not point to this segment.
3535022C	The LCF pointer does not point to the logical child to be deleted. In attempting to find the segment on the chain, it was found that no LTF pointer exists.
3535032C	The logical child to be deleted was not found on the logical twin chain.
6363012C	The Delete/Replace module needs to traverse a relationship to go from the LC database to the LP database. The logical child has a logical-parent pointer; however, its value is 0.
6363022C	The Delete/Replace module needs to traverse a relationship to go from the LP database to the LC database. A logical-child-first pointer exists in the logical parent segment; however, its value is 0.
6565012C	The PSTBYTNM of the segment just retrieved is not as expected. (HISAM)
6565022C	After traversing a logical relationship and obtaining a segment, the Delete/Replace module could not identify the segment. (HISAM)
6565032C	After traversing a logical relationship and obtaining an overflow record, it was found that the overflow record was empty. (HISAM)
6666002C	Cannot find root segment by key.
6666012C	Cannot find segment position by key.
6666022C	
6666032C	Cannot find segment position by pointer.

**Possible cause:** The causes include:

- After database reorganization, Prefix Update was not run prior to any other calls to the database.
- During a delete of a physically-paired logical child, equal intersection data was not loaded for the logical children in a physical pair.
- After the Prefix Resolution utility was run, the return code, if not 0, was not checked for implications before Prefix Update was run.
- When using symbolic keys, the logical parent did not exist.
- If a logical structure exists, pointers can have gotten lost when using a keys that is not unique.

**System action:** The application program terminates abnormally.

## 0804

**Programmer response:** Ensure that the reorganization and load program was performed correctly. See Partial Database Reorganization utility (DFSPRCT1) (Database Administration).

**Problem determination:** 17a through 17g, 35

At the time of the abend, if register 11 points to CSECT DFSDLDS0, delete or replace was unable to find the logical parent or the paired logical child. However, if register 11 points to CSECT DFSDLDD00, register 15 will contain a value to identify the problem. If register 15 is negative, the logical parent counter was zero prior to the delete call. If register 15 is equal to zero, the logical parent does not point to the logical child to be deleted.

**Source:** IMS abends

**Module:** DFSDLDD00

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

## 0804

**Explanation:** A DL/I action module made a request for working storage. Because of insufficient space available in the database working pool, the request could not be satisfied. The database working pool should be at least as large as the number of active dependent regions times 2K.

**Analysis:** This is issued from module DFSDLDD00 and can be a standard or a pseudoabend.

Abend 0804 is traced in the Buffer Handler Trace Table in a series of entries, each of which is identified by a character D (X'C4') in its first byte. This series of entries immediately precedes the current entry in the trace table.

The important field in each trace entry is the second word (offset 4), called ENTRY1. This word identifies the abend and the routine within the Delete/Replace module that encountered the problem.

The ENTRY1 word in the first entry of the Delete/Replace abend trace in the key shown in the following table:

Key	Description
22220130	The delete-work-area (DLTWA) chains were invalid. The Delete module has just found the last work area; however, the PST indicates that another exists.
67670030	A nonzero return code was obtained from GETBUF after a request for storage to build a work area.

**Possible cause:** The causes include:

- VS SORT did not release 80 bytes of allocated storage when SORT was invoked from COBOL V3.1.
- The application was run in a batch region in overlay mode, but the application macro was not coded to reflect this.
- For a DB/DC environment, the DBWP is too small.
- For a batch environment, the region size is too small.

**System action:** The application program is stopped and the current message is re-enqueued on the message queue.

**Operator response:** The application program can be restarted and the transaction will be scheduled again.

**Programmer response:** If the 0804 abend occurs in an online system, the database working pool (DBWP) should be increased in size. This size can be specified as a parameter on the EXEC statement. If the abend occurs in a batch region, the region size should be increased.

**Problem determination:** 4, 5, 6

**Source:** IMS abends

**Module:** DFSDLDD00

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

---

**0805**

**Explanation:** During Reload processing for a partition of a High Availability Large Database (HALDB) it was determined that either the prefix was not valid for a HALDB partition or the prefix length was not valid for a HALDB partition.

**Analysis:** This is a pseudoabend set up by DFSDLA00 and issued from module DFSPR000.

In the dump generated by a SYSABEND or SYSUDUMP DD card, locate the PST that is pointed to by the address in register 11 at time of abend. Locate the correct offset of field PSTFUNCH by assembling the IPST dsect as referenced by the DFSADSCT module.

PSTFUNCH will contain one of the following error codes:

- **RER1** - Invalid partition prefix
- **RER2** - Invalid partition prefix length

**Possible cause:** A database, other than the partition of a HALDB that was unloaded, was specified at reload time.

**System action:** The application program terminates abnormally.

**User response:** Specify the correct partition database that was unloaded and rerun the reload job.

**Problem determination:** 1, 2, 3, 17a, 35, 36

**Source:** IMS abends

**Module:** DFSDLA00

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

**0806**

**Explanation:** Delete/replace received an invalid return from the buffer handler on a byte locate call, or an error occurred attempting to mark buffer altered. The request to the buffer handler might have been either a Mark Buffer Altered, a Byte-Locate call, an Erase-Logical-Record call, an SETL, or a Block-Locate call. This problem is an IMS system error.

**Analysis:** This is issued from module DFSDL00 and can be a standard or a pseudoabend.

The buffer handler function code at PSTFNCTN will indicate the type of call that was made:

Call	PSTFNCTN
X'E1'	Block Locate
X'E2'	Byte Locate
X'E6'	Mark Buffer Altered
X'F1'	Erase Logical Record
X'F2'	SETL

Abend 0806 is traced in the buffer handler Trace Table in a series of entries, each of which is identified by a character D (X'C4') in its first byte. This series of entries immediately precedes the current entry in the trace table.

The ENTRY1 word in the first entry of the Delete/Replace abend trace should be 0202xx34, where xx is the PSTFNCTN code as shown above. The fourth byte, X'34', is the Delete/Replace internal code for abend 0806. The first 2 bytes are routine identification numbers: the first is the originator of this entry and the second is the originator of the abend. The identification number X'02' is for the CALLBH routine within the Delete/Replace module. (See the module prolog for ID numbers.)

Additional items of information that can be obtained from this trace entry are:

Offset	Item
22	Return offset from CSECT (caller).
24	DSG address.
28	PSTBYTNM.

## 0807

If the first trace entry is insufficient to diagnose the problem, the routine that called CALLBH can be identified from the second trace entry, according to the following table:

### Routine IDs

#### Source Module and Routines

#### 01 to 1F

DFSDLDC0: Control/Common subroutines.

#### 20 to 3F

DFSDLDD0: Delete routines.

#### 40 to 5F

DFSDLDR0: Replace routines.

#### 60 to 7F

DFSDLDW0: DLTWA Build routines.

For example, if the ENTRY1 word in the second entry is 4202E234, the PSTFNCTN code, E2, identifies a Byte-Locate call; the Routine ID, 42, identifies the caller's module as DFSDLDR0. A reference to the prolog of DFSDLDR0 identifies routine 42 as CPRDATA. Further problem diagnosis must be done from the code and comments.

For those routines which go to CALLBH from several locations, the exact exit to CALLBH can be determined from offset X'16' in the prior trace entry. This halfword contains the offset from the beginning of the caller's CSECT to the instruction following the 'BALR R10,R11' instruction that branched to CALLBH. To continue problem diagnosis, read the code and comments.

**System action:** The application program terminates abnormally.

**Problem determination:** 17a through 17g and 35

**Source:** IMS abends

**Module:** DFSDLD00

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

## 0807

**Explanation:** Delete/replace cannot obtain a position in the physical structure (physical path to root required). The failure may have occurred while searching the logical blocks or while searching the physical path to the root. This problem is probably a system error.

**Analysis:** This is issued from module DFSDLD00 and can be a standard or a pseudoabend.

Abend 0807 is traced in the Buffer Handler Trace Table in a series of entries, each of which is identified by a character D (X'C4') in its first byte. This series of entries immediately precedes the current entry in the trace table.

The important field in each trace entry is the second word (offset 4), called ENTRY1. This word identifies the abend and the routine within the Delete/Replace module that encountered the problem.

The ENTRY1 word in the first entry of the Delete/Replace abend trace in the key shown in the following table:

Key	Description
05050038	After obtaining the overflow record from the buffer handler, the Delete/Replace module found that it did not contain a valid segment code where expected.
25250138	In processing a DELETE in which a logical relationship exists in VSAM KSDS database, when returning the delete scan to the database of the logical child a retrieve-by-key for the root segment could not find the segment.
2320138	While attempting to update the PTF pointer or the PCF pointer, just before free-spacing the segment, the Delete module could not find the segment on the twin chain.
64640138	The level table in the Delete work area contains an entry which has an RBN of 0.
6660038	The entry for the root segment in the level table (in the Delete work area) has an RBN of 0 with no key available.

**System action:** The application program terminates abnormally.



**Programmer response:** No action is required.

**Problem determination:** 17a through 17g, and 35

If register 11 points to CSECT DFSDLDS0, delete/replace was unable to follow the logical blocks (SDB and level tables) from the segment to be deleted or replaced up to the root segment. This is usually caused by retrieve not posting position where delete expected it. If register 11 points to CSECT DFSDLDD00, the segment to be removed is not pointed to by the physical chain (PCF, PCL, PTF, PTB, or RAP).

**Source:** IMS abends

**Module:** DFSDLDD00

**Related information:**

Chapter 2, "IMS problem determination," on page 13

## 0808

**Explanation:** Delete/replace cannot resolve database blocks description of logical relationships or could not calculate the address of the PSDB for the segment code just retrieved.

**Analysis:** This is issued from module DFSDLDD00 and can be a standard or a pseudoabend.

Abend 0808 is traced in the Buffer Handler Trace Table in a series of entries, each of which is identified by a character D (X'C4') in its first byte. This series of entries immediately precedes the current entry in the trace table.

The important field in each trace entry is the second word (offset 4), called ENTRY1. This word identifies the abend and the routine within the Delete/Replace module that encountered the problem.

The ENTRY1 word in the first entry of the Delete/Replace abend trace is the key used below.

Key	Description
2020013C 2828013C 3838013C 2A2A013C 2A2A023C 2A2A033C 2A2A043C 3232013C 3232043C 3A3A013C 3C3C013C 3131023C	The address of the PSDB for the segment code just retrieved could not be calculated.
2121013C 2626013C	The address of the PSDB for a logical child could not be calculated using the logical parent's SECLST entry for that logical child.
3131013C	The Delete module determined that the logical parent's counter must be decremented before deleting the logical child. The logical parent has no counter.
3232023C 6565003C	In attempting to find a segment on a twin chain with no backward pointing, the Delete/Replace module must start with the parent segment. However, the parent has no pointer to the child segment.
3232033C 3838023C	In traveling along a forward chain, a segment has been encountered which has no forward pointer.
3434013C 3434023C 6868043C	The address of the PSDB for the parent segment (of the segment whose PSDB is known) could not be calculated.
3B3B013C 4242023C 4444003C 6868023C	The logical child SECLST entries contain no entry for the logical parent.

## 0810

Key	Description
3B3B023C 2626013C 4242033C	The address of the PSDB for a logical parent could not be calculated using the logical child's SECLST entry for that logical parent.
3B3B033C 4242053C	The logical parent SECLST entries contain no entry for a logical child.
4242013C 6868013C	The PSDB for a logical child does not contain an address for the SECLST entries.
4242043C	The PSDB for a logical parent does not contain an address for the SECLST entries.
4242063C 7272033C 7272043C	The logical child SECLST entries contain no entry for the paired logical child.
4242073C 7272013C 7767013C	The address of the PSDB for the paired logical child could not be calculated using the original logical child's SECLST entry for that pair.
6363013C	No path exists from the logical child to its logical parent. The logical child has no logical-parent pointer, nor does it carry the key of the logical parent.
6363023C	In a bidirectional relationship, no path exists from the logical parent to the logical child. The logical parent has no logical-child-first pointer.
6666003C	In HIDAM, the SECLST entries for the root segment contain no entry for the primary index.
6868033C	The sequence field of the logical child is contained in a virtual concatenated key of the logical parent.
6969033C	In attempting to obtain position and keys from the logical blocks, the PSDB address of the current segment as obtained from its SDB does not match the PSDB address in the Delete work-area level-table for that segment.
7272023C	The address of the PSDB for either a logical parent or a logical child (not paired) could not be calculated from a SECLST entry.

**System action:** The application program terminates abnormally.

**Problem determination:** 17a through 17g and 35

**Source:** IMS abends

**Module:** DFSDDL00

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

## 0810

**Explanation:** An application program attempted to load a database and was not sensitive to all data set groups within the database.

**Analysis:** 0810 is a pseudoabend detected by DFSDLOC0, the DL/I data base OPEN/CLOSE module. The SNAP of the control blocks is required to diagnosis this problem. Print the type X'67' log records using the field select and formatting print program, DFSERA10. The OPTION PRINT statement must specify EXITR=DFSERA30.

To find the registers applicable for this pseudoabend it is necessary to locate the save area set used by DFSDLOC0. Get the entry point address from the SCD at label SCDDLICL and search down the save sets for a matching address in register 15. This save set contains the registers when the open/close module was called. Register 1 in this save set points to the PST.

An OPEN has been requested for a PCB (PSTFNCTN=X'0A'). A test is made, if the PCB processing option is load (PROCOPT=L), that the AMPs associated with each DSG in the JCB are either all open or all closed.

Key	Label	Description
PSTABTRM=X'1000032A' PSTFNCTN=X'80'	LDOCHKEG	A work byte (OCWORKSW) is tested to ensure that all AMPs are either opened or closed. A failure results in branch to label NOTOPNLD to issue the abend.

**Possible cause:** The PSBGEN is not sensitive to at least one segment in each data set group.

**For APAR processing:** PSBGEN, DBDGEN parameter, copy of the system log, and dependent region dump

**System action:** The application program terminates abnormally.

**Programmer response:** Perform PSB generation again, and make certain that the program is sensitive to at least one segment in each data set group.

**Problem determination:** 2, 3, 9, 18, 19

**Source:** IMS abends

**Module:** DFSDL00

**Related information:**

Chapter 2, "IMS problem determination," on page 13

## 0811

**Explanation:** Delete/Replace was attempting to change a pointer in a segment's prefix. The system found that the old value of the pointer was not as expected. Previous pointer maintenance was erroneous.

**Analysis:** Abend 0811 is issued from module DFSDL00 and can be standard or a pseudoabend. Abend 0811 is traced in the Buffer Handler Trace Table in a series of entries, each of which is identified by a character D (X'C4') in its first byte. This series of entries immediately precedes the current entry in the trace table.

The important field in each trace entry is the second word (offset 4), called ENTRY1. This word identifies the abend and the routine within the Delete/Replace module that encountered the problem.

The ENTRY1 word in the first entry of the Delete/Replace abend trace is the key used below:

Key	Description
37370140	The deleted segment was either first or last on a physical-child chain. However, when attempting to change the pointer in the parent segment, the Delete module found that the pointer did not contain the RBA of the deleted segment.
38380140	The deleted segment was on a physical or logical chain
39390140	In attempting to chain a pointer in the prior or next segment, the Delete module found that the pointer did not contain the RBA of the deleted segment.

**System action:** The application program terminates abnormally.

**Problem determination:** 17a through 17g and 35

**Source:** IMS abends

**Module:** DFSDL00

**Related information:**

Chapter 2, "IMS problem determination," on page 13

## 0812

**Explanation:** A return code greater than 4 was returned by the randomizer routine.

**Analysis:** This is a pseudoabend issued by modules DFSDLDC0 and DFSDLR00.

**Source:** IMS abends

**Module:** DFSDLDC0, DFSDLR00

## 0814

**Explanation:** An unrecoverable error was detected during data capture processing when changed data logging or changed data exits were defined for the database.

**Analysis:** The type of error depends upon which module generated the abend:

**DFSCPY40**

The abend was issued because an error was detected while preparing to call z/OS data compression services, or the return code from z/OS data compression services indicated an error condition. If register 5 is negative, an error occurred while calculating the output length to pass to z/OS data compression services. If register 5 is not negative, an error was returned from z/OS data compression services in register 15. See *z/OS MVS Programming: Assembler Services Reference Vol 1 (SA22-7606)* for an explanation of CSRCE SRV return codes.

**DFSECP10/DPSECP20**

The pseudoabend occurred because DFSDCAP0 detected an error during data capture for a full-function database. Field PSTCDCRC in the PST contains diagnostic information. (The PST is formatted in the X'67FF' log record created for this abend.) The contents of field PSTCDCRC are the same as the contents of register 15 when DFSDCAP0 issues this abend (see below).

**DFSPCC30**

A DSPSERV or ALESERV request received an invalid return code in register 15 at the time of the abend. For return code 8, the creation of the data space was disallowed because the IEFUSI user exit routine prevented the creation of key 8 data spaces.

**DFSDCAP0**

An error occurred during data capture for a full-function database in a Batch region (DLI/DBB). Register 15 contains the following information:

Byte	Subcodes (hex in parentheses) and description
First byte - module detecting the error	<b>04 (04)</b> DFSDCAP0
Second byte - routine detecting the error	<b>04 (04)</b> DO BYTE LOCATE <b>08 (08)</b> BYTE LOCATE <b>12 (C)</b> BUILD LOGICAL PARENT CONCATENATED KEY <b>16 (10)</b> FIND POINTER <b>20 (14)</b> GET WORK <b>24 (18)</b> GET STORAGE <b>28 (1C)</b> DELETE PATH DATA
Third byte - reason code for the error	<b>04 (04)</b> Subcode is DMB segment code. <b>08 (08)</b> Subcode is PST return code. <b>16 (10)</b> Subcode is capture function that failed.
Fourth byte - subcode for the error	<i>nn</i> PST return code if REASON code 04. <i>nn</i> DMB segment code if REASON code 08. <b>04 (04)</b> UNABLE TO GET WORK STORAGE IF REASON CODE 16. <b>08 (08)</b> NO DATA SPACE STORAGE AVAILABLE IF REASON CODE 16.


**System action:** The application program terminates abnormally.

**Problem determination:** Check the JCL for the batch region.

**Source:** IMS abends

**Module:** DBFDCAP0, DFSECP10, DFSECP20, DFSPCC30, DFSCPY40

**Related reference:**

 z/OS: CSRCE SRV return and reason codes

## 0816

**Explanation:** Index synchronization error occurred. The contents of a secondary index did not correspond to the contents of the target database. This problem could be the result of the target and secondary DBDs being rebuilt without rebuilding the PSBs referencing them.

**Analysis:** This is a pseudoabend issued by module DFSDLR00.

**System action:** The program terminates abnormally.

**Programmer response:** Ensure that the index is recovered to the same level as the target database; inspect the index segment to ascertain whether its contents are valid. It can be found at the address contained in PSBXIOWK. The PSB is found by following the pointers at PSTPSB to the PDIR and PDIRADDR to the PSB. Reload the database and all secondary indexes. An effort should be made to trace the source of the synchronization error to a specific database call.

**Problem determination:** 4, 5, 11, 17b, 17c, 17g, 35

**Source:** IMS abends

**Module:** DFSDLR00

**Related information:**

Chapter 2, "IMS problem determination," on page 13

## 0819

**Explanation:** An internal IMS system error was encountered. One of the following occurs:

- A CLOSE failure occurs during phase 1 of sync point, and sync point processing is aborted.
- Any participant votes NO in phase 1 of sync point.
- A module called the synchronization point processor (DFSFXC30) and requested an undefined function.
- In a Database Control (DBCTL) environment, a Coordinator Control (CCTL) subsystem thread requested a sync point COMMIT before doing a sync point PREPARE.
- A failure occurs during processing of the FLD call due to errors other than deadlocks or verification errors.

**Analysis: For DFSFXC30:**

The sync point processor (DFSFXC30) was called to process an undefined function.

If SPOOL/API processing was occurring, the system terminated abnormally for the following reasons:

- A CLOSE failure occurred during phase 1 of syncpoint and the decision was made to abort syncpoint processing.
- A participant voted N0in phase 1 of syncpoint.

This is a standard abend issued by DFSFXC30. The program status word (PSW) at entry-to-abend points to the abend SVC. Register 1 is loaded with the completion code at label ABEND819.

The acceptable encoded function codes are located in the PST at label PSTFUNCT, and are as follows:

### Codes

#### Explanation

X'01'

MSG get unique code

X'42'

Normal/abnormal termination

X'79'

Synchronization call

X'85'

Checkpoint call

X'89'

ROLB

X'8E'

Internal ROLB

Register 13 in the abend SVRB points to the current save area set. In this save area set, register 14 is the caller's return address, and register 1 is the PST's address.

## 0820

Key	Label	Description
Reg1=X'80000333'	(prior to label) WASHRTNE	The PSTFUNCT field is checked for the existence of a valid function. If an invalid code was passed, the abend is issued.

### For DFSDSC00:

A COMMIT request was made without a previous PREPARE request.

**System action:** IMS abnormally terminates.

**Operator response:** (system operator) Inform the master terminal operator of the IMS control region abend.

(master terminal) Execute an emergency restart.

**Problem determination:** 4, 5, 11, 35

**Source:** IMS abends

**Module:** DFSFXC30, DFSDSC00

### Related information:

Chapter 2, "IMS problem determination," on page 13

---

## 0820

**Explanation:** An error was detected in either module DFSDLA30 or DFSFXC40. The following are possible errors:

- DFSDLA30:
  - An invalid return code was received from the queue manager. This is an IMS system error.
  - A failure occurred while trying to locate the call destination.
- DFSFXC40
  - An invalid function code was passed into DFSFXC40. This is an IMS system error.
  - A nonzero return code was received from the message router DFSICLRO. This is an IMS system error.
  - The I/O PCB destination output represents an SMB and the dependent region is message driven. This is an IMS system error.

**Analysis:** This is a standard abend issued by either DFSDLA30 or DFSFXC40. Use the program status word (PSW) at entry-to-abend to isolate the failing module.

### For DFSDLA30:

0820 is a standard abend issued by DFSDLA30, the DL/I communications call handler.

The program status word (PSW) at entry-to-abend points to the instruction within label xxxAB820 from which the abend (SVC 13) is issued. Use register 14 to isolate to a specific label. Register 12 is the DL/I call function routine base register. Register 9 contains the PST address.

Key	Label	Description
Reg9=PST address	ROLB1000	ROLB call processing. PSTSMB address is zero.
Reg15=RC from DFSCON10	ROLB2400	ROLB call processing. Invalid return code received from DFSCONC10.
	GETGU20	GU call processing. Pseudo WFI subroutine detected that the call flag of the SAP waiting for GU call is set. DFSDLA30 is out of sync with the SMB DEQ routine.
Reg15=RC from DFSQMGR		GU, GN, CHKP or ROLB call processing invalid return code received from DFSQMGR call.
Reg15=RC from DFSSLC		GU, GN, CHKP or ROLB call processing invalid return code received from DFSSLC CNTS search call.
Reg15=RC from DFSSLC	GMOVABND	GU, GN, CHKP or ROLB call processing invalid return code received from DFSSLC LNBS search call.

Key	Label	Description
Reg15=RC from DFSFNDST	GMOV1900	GU, GN, CHKP or ROLB processing made a findest call (using DFSSLC) and received an invalid return code.
Reg15=RC from DFSSLC		GU, GN, CHKP or ROLB call processing invalid return code received from DFSSLC CNTS search call.
Reg15=RC from DFSFNDST		GU, GN, CHKP or ROLB processing made a findest call (using DFSSLC) and received an invalid return code.
Reg15=RC from DFSQMGR	CANMSG	GU, GN, CHKP or ROLB call processing invalid return code received from DFSQMGR call.
Reg15=RC from DFSQMGR	CMSG1000	GU or CHKP call processing. Invalid return code received from DFSQMGR Reject (QMRJDLI) call.
Reg15=RC from DFSCMD30	GCMDEXIT	GCMD call processing. Invalid return code received from DFSCMD30.
Reg15=RC from DFSFXC40	PURG4000	PURG call processing. Invalid return code received from DFSFXC40.
Reg15=RC from DFSQMGR	ISRT085	ISRT call processing. Invalid return code received from DFSQMGR Insert Prefix (QMIPDLI) call while attempting to update MODNAME.
Reg15=RC from DFSQMGR	ISRT098	ISRT call processing. Invalid return code received from DFSQMGR Insert Prefix (QMIPDLI) call while attempting to update MODNAME.

#### For DFSFXC40:

This is a standard abend issued by DFSFXC40, the communications syncpoint processor. DFSFXC40 is called by DFSFXC30 to handle the communications portion of a syncpoint. A function vector is passed to DFSFXC40. Should this vector value be invalid or a message router error occur, the abend will be issued.

The program status word (PSW) at entry-to-abend will point to the instruction within label ABEND820 from which the abend (SVC 13) is issued. Register 10 in the abend SVRB will contain the function vector value in hexadecimal or a BAL address for routine ROUTERR. The caller's registers are saved (register 14 through register 12) at register 13 plus X'C'.

Key	Label	Description
R10=function code	ABEND820	An invalid function code was passed to DFSFXC40. Register 14 in DFSFXC40's save area points to the calling module.
	DLAWQ6	An IOPCB was invalidly pointing to an SMB. This is valid only when a non-message driven BMP with the OUT= parameter specifies another transaction as the destination.
Reg15=RC from DFSICLR0	ABEND820	Message Router (DFSICLR0) returned a nonzero code. Register 14 points to the instruction after the call to the message router.

**Possible cause:** Coding change, a user modification, in DFSFXC30 if vector Router-Qmgr enqueue return code is greater than zero.

**System action:** IMS terminates abnormally.

**Problem determination:** 4, 5, 11, 34

**Source:** IMS abends

**Module:** DFSDLA30, DFSFXC40

**Related information:**

Chapter 2, "IMS problem determination," on page 13

## 0821

**Explanation:** An error has occurred during initialization of the program and database control blocks while executing a DBB type IMS region.

**Analysis:** This is a standard abend issued by DFSBIND0, the DL/I Batch ACBLIB block loader interface module. Register 10 in the abend SVRB should be used to isolate to the specific label; register 11 points to the SCD and register 12 is the base register. In addition to the abend dump, an informational message is included in the message class for the job and written to the system output device (SYSPRINT).

The batch job selects either the OLCSTAT or the MODSTAT data set to read the online change status in the following sequence:

1. From the OLCSTAT data set allocated via JCL if one exists.
2. From the OLCSTAT data set defined in the MDA member if one exists.
3. From the MODSTAT data set allocated via JCL if there is no OLCSTAT DD statement coded in JCL and no OLCSTAT MDA member is defined.
4. The default is to use the ACBLIB from the IMSACB DD statement when there is no OLCSTAT and no MODSTAT DD statement coded in the JCL.

After the OLCSTAT or the MODSTAT or the IMSACB data set have been selected, an error can occur either because the data set name is incorrect or it cannot be accessed.

Key	Label	Description
Reg5=A(DCB) Reg10=BAL	DFSBIND0	The ACBLIB data set must be opened. A not open condition results in message DFS823I and the abend is issued.
Reg5=A(BLDL area) Reg10=BAL	DFSBIND0	A BLDL SVC (SVC 18) for a PSB resulted in return code equal to 8 and message DFS824I.
Reg 5=A(SELACBDD)	SELACBDD	The OLCSTAT or MODSTAT data set could not be opened or contains invalid data or is coded as DD DUMMY. See previous error messages prior to the ABEND0821 for an explanation.
Reg9=A(BLDL area) Reg10=BAL	DFSBIND0	The BLDL SVC completed with an error return code less than 8, but the TTR field was zero. This indicates a not found condition. See message DFS830I.
Reg9=A(BLDL area) Reg10=BAL	DFSBIND0	This code tests for an alias name; an alias is <i>not</i> allowed and treated as an error condition. See messages DFS830I and DFS826I for reference to ALIAS.
Reg9=A(BLDL area) Reg10=BAL	DFSBIND0	The BLDL was for a PSB but the bits in BLDLFLAG field indicate a DMB. See messages DFS830I and DFS826I for reference to NOTPS.
Reg0=zero Reg8=A(PDIR) Reg10=BAL	DFSBIND0	The PSB size is determined to be zero, this is a critical error. See messages DFS830I and DFS826I for reference to NOBUF.
Reg10=BAL	IDMBD7	A BLDL SVC (SVC 18) for a DMB resulted in return code equal to 8. See message DFS824I.
Reg10=BAL	IDMB20	Message DFS826I shows DBD names and reason for failure.
Reg10=BAL	EODAD	An I/O error occurred during a read operation of a shared secondary index. See message DFS838I.
Reg10=BAL	BINMODRD	The MODSTAT data set could not be opened or contains invalid data. See message DFS3467I.

**System action:** The IMS region terminates abnormally. One or more messages will have been written to the system console. If the batch initialization module (DFSBIND0) detects the error then one or more of the following messages appears: DFS2855I, DFS823I, DFS824I, DFS826I, DFS830I, DFS838I, DFS3467I.

If the error is detected by the block mover module DFSDBLM0, or by any module called by block mover, any of the



messages put out by those modules in a batch DBB region might appear. These include DFS561I, DFS564I, and DFS592I.

**Programmer response:** See the message listed above to determine the cause of failure and problem determination steps.

**Source:** IMS abends

**Module:** DFSBIND0

## 0822

**Explanation:** A single message queue segment did not fit into the PSB index work area. This problem is an IMS system error.

**Analysis:** This is a standard abend issued by DFSDLA30, the communications DL/I call handler. A message segment can require multiple queue segments; each call to the QMGR returns one queue segment. If the message spans multiple queue segments, a QMGR call is required for each block. A QMGR *GU* function retrieves the first block of the message and subsequent *GN* calls retrieve all additional blocks of the message. The PSB index work area always holds at least one block.

The program status word (PSW) at entry-to-abend points to the instruction within label GETAB822 from which the abend (SVC 13) is issued. Register 14 is the address from which the abend was issued. Register 12 is the base register.

Key	Label	Description
Reg1=X'80000336'	GETEND1	This is the first call (register 8=zero). The complete segment will <i>not</i> fit into the XWA.

**System action:** IMS terminates abnormally.

**Problem determination:** 4 or 11 and 35

**Source:** IMS abends

**Module:** DFSDLA30

**Related information:**

Chapter 2, "IMS problem determination," on page 13

## 0824

**Explanation:** IMS received an invalid return code from the user's program routing exit. Register 15 contains the invalid return code.

**Analysis:** 0824 is a standard abend issued from DFSDLA30, the communications DL/I call handler. The multiple system (MS) program routing exit module (DFSCMPRO) was called to route a response to a remote terminal component other than the one that entered the message. DFSDLA30 tests the return code in register 15 to determine if the "CHNG" call should be processed. The valid return codes are:

### Codes

#### Explanation

**X'00'**

Use destination name as supplied by the CHNG call to find the destination.

**X'04'**

Use originating system ID to route the message back to originating system.

**X'08'**

Return 'A1' status code to the application program.

The program status word (PSW) at entry-to-abend points to the instruction within label ABEND824 from which the abend (SVC 13) is issued. Register 2 in the abend SVRB is the pointer to the destination name supplied by the "CHNG" call, register 15 will contain the invalid return code, register 12 is the base register, and register 14 is the address from which the abend was issued.

## 0825

Key	Label	Description
Reg14=BAL Reg15<zero or>X'08'	CHGNONC	An invalid return code was passed back to DFSDLA30 by DFSCMPRO.

**Source:** IMS abends

**Module:** DFSDLA30

## 0825

**Explanation:** An unexpected return code from the buffer handler (DFSDDBH0) was found, or a catastrophic error occurred while executing a move character long (MVCL) instruction.

**Analysis:** 0825 is a pseudoabend detected by DFSDXMT0, the DL/I index maintenance module. The SNAP of the control blocks are required to diagnosis this problem. Print the type X'67' log records using the utility DFSERA10 with EXITR=DFSERA30.

To find the applicable registers on a pseudoabend (using the SNAP of the control blocks) it is necessary to locate the save set used by DFSDXMT0. Get the entry point address of DFSDXMT0 from the SCD at label SCDDXMT0 and search down the save area sets for that address in register 15. This save set will contain the registers as they were when the index maintenance module was called. Register 1 in this save set will contain the pointer to the PST. The next lower save set will contain the registers when either subroutine DPRSYS was called or when module DFSDDBH0 was called. If the registers are those when subroutine DPRSYS was called the next, or second lower, save set will contain the entry registers to the buffer handler (DFSDDBH0).

The parameter of the call to the buffer handler and the actual return codes can be seen in the buffer handler trace table. This trace table can be found by its label (BFSP) in a formatted SNAP or by following the pointer at SCDDBFSP. The current trace table entry indicates the DSG in which the call was performed and the contents of PSTBYTNM passed to DFSDBVH0.

Next the index work area (XWORKARA) must be located. Scan the SNAP of the control blocks for the *PSB*. The first word of the *PSB* is a pointer to the *PST*; The second word is the pointer to the index work area (XWORKARA). Field ASUPRO within the XWORKARA is used as a save area for register 14 prior to exiting with an error. This register 14 BAL should be used to isolate to a specific label. In addition to the index work area, fields within the *PST* that are pertinent are:

- PSTFNCTN requested function
- PSTRTCDE buffer handler return code
- PSTBYTNM relative byte address (RBA)

Key	Label	Description
Reg14=BAL (ASUPRO)	DRETSEG	Buffer handler returned an unexpected error code.
Reg14=BAL (ASUPRO)	DSPNOTSX	DFSDDBH0 passed back an unexpected return code.
Reg14=BAL (ASUPRO)	SPRETCL	DFSDDBH0 passed back an unexpected return code.
Reg14=BAL (ASUPRO)	MVTRBL	The MVCL instruction failed to process without error.

**System action:** The batch or message processing region terminates abnormally. The control region continues.

**Programmer response:** Review the *PST* fields. Pertinent *PST* fields and contents are:

Name	Description
PSTFNCTN	Requested function
PSTRTCDE	Return code

If the return code is X'04', the following fields contain relevant information:

Name	Description
PSTBYTNM	Incorrect relative byte address

**PSTDBPCB**

Address of the PCB

**DBPCBJCB**

Address of the JCB

If an I/O error is indicated, take the action required to recover the database. Any other error is probably a DL/I error.

**Problem determination:** 4, 5, 11, 17b, 17c, 17g, 35

Re-create the problem in a batch environment and snap control blocks and buffer pool on two calls prior to this abend.

**Source:** IMS abends

**Module:** DFSDDBH0, DFSDXMT0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

**0826**

**Explanation:** A situation was encountered by index maintenance that prohibits further processing of the database. Message DFS0840I identifies the index and gives a status code. Probable causes are:

- Read error indicated by buffer handler. The status code is 'NO'.
- Unable to open the index database. The status code is 'NI'.
- I/O error indicated when inserting to an index DB. The status code is 'NO'.
- Open error on work file data set during DB load with secondary indexes. The status code is 'NI'.
- Index Maintenance is unable to handle the return code it received from the buffer handler. The status code is blanks.

**Analysis:** This is a pseudoabend detected by DFSDXMT0, the DL/I index maintenance module. The SNAP of the control blocks should be available for problem diagnosis purposes. Print the type X'67' log records using the File Select and Formatting Print utility (DFSERA10). The OPTION PRINT statement must specify EXITR=DFSERA30.

Two items are required to isolate to a specific label:

1. PST, which can be found on the SNAP of the controls blocks
2. Message DFS0840I, which is issued to the IMS master console. The message identifies the failing index and gives a status code.

The PSTENCTN field within the PST contains the requested DL/I function.

To find the applicable registers for a pseudoabend locate the save area set used by DFSDXMT0. Get the entry point address from the SCD at label SCDDXMT0 and search down the save sets for a matching address in register 15. This save set contains the registers when the index maintenance module was called. The next lower save set contains the registers for the buffer handler (DFSDDBH0 or for VSAM DFSDBVH0) or OPEN/CLOSE (DFSLOC0).

Key	Label	Description
Status code=blank	DREPLS	The replace processor subroutine (DREPLACE) utilizes the buffer handler to replace an index segment. The buffer handler passed back a nonzero return code.
Status code='NO' PSTFNCTN=X'E2'	DRETSEG	A read error was detected on a buffer handler retrieve request.
Status code='NI' PSTFNCTN=X'48'	DINITDSG	For a LOAD request the database <i>open</i> failed.
Status code='NO' PSTFNCTN=X'F4'	DISRTVSM	A VSAM insert of an index segment failed.
Status code='NI'	DTAPERA	The root load subroutine (DLOAD) failed because workfile DFSURWF1 could not be opened.

**System action:** The application program terminates abnormally.

## 0827

**Programmer response:** If an open error is indicated, check JCL parameters or missing DD statements. For an I/O error, investigate the need to recover the affected data set.

**Problem determination:** 35

**Source:** IMS abends

**Module:** DFSDXMT0

**Related information:**

 DFS0840I

Chapter 2, "IMS problem determination," on page 13

---

## 0827

**Explanation:** An error occurred during LATCH, UNLATCH, or IMODULE LOAD processing related to use of the index exit routine. The probable causes are:

- Exit routine not found in Job or Step libraries under the name indicated in the DBDGEN.
- Read error encountered on reading exit routine from library.
- Latch ECB posted incorrectly while serializing the Index exit routine.

**Analysis:** 0827 is a pseudoabend presented by DFSDXMT0, the DL/I index maintenance module. The SNAP of the control blocks should be available for problem diagnosis purposes. Print the type X'67' log records using the file select and formatting print program, DFSERA10. The OPTION PRINT statement must specify EXTITR=DFSERA30.

To find the registers applicable for a pseudoabend it is necessary to locate the save area set used by DFSDXMT0. Get the entry point address from the SCD at label SCDDXMT0 and search down the save sets for a matching address in register 15. This save set contain sthe registers when the index maintenance module was called.

This abend only occurs if the DBDGEN has been coded to include an XDFLD statement with the keyword EXTRTN. This optional operand (EXTRTN) is used to specify the name of a user supplied index maintenance exit routine that is used to suppress the creation of selected index pointer segments.

There are two reasons for this abend. Either the IMODULE LOAD failed attempting to load the user exit routine, or the serialization of the DMB latch (DMBLATCH) failed. The first reason can only occur in a DB/DC environment. In the case for an IMODULE failure register 14 through register 12 are stored in the second lower save set prior to abend. Register 3 in this save set points to the Index Maintenance parameter list (DMXMPRM). At label DMBMXNM is the name of the user supplied exit routine, and register 15 contains the IMODULE LOAD return code.

In the second case, either the latch or unlatch operation failed; the third lower save set should contain the registers at entry to LATCH/UNLATCH. If the error occurs here, the environment is not saved prior to exit; therefore, labels are provided as a guide through the logic and the KEY information is absent.

Key	Label	Description
Reg15 nonzero (2nd lower save set)	DSPRETAA	The IMODULE LOAD for the exit routine failed.
	DSUPISLD	For a LATCH operation, either the owner of the latch is <i>not</i> the caller, or the post code (PSTDECB) is invalid after an ISERWAIT. For an UNLATCH operation, the owner of the latch is not the caller.

**Possible cause:** JOBLIB or STEPLIB JCL does not point to the library where the user exit routine resides.

**For APAR processing:** Assembly of DBDs, dependent region dump, and copy of the system log

**System action:** IMS terminates abnormally.

**Programmer response:** Analyze the possible library errors that might have caused the problem. Latch problems can only exist in the online environment. If not a hardware or a user error, proceed with the problem determination.

**Problem determination:** 3, 8, 11, 17b, 18, 35

**Source:** IMS abends

**Module:** DFSDXMT0, DFSURGU0

**Related information:**

Chapter 2, “IMS problem determination,” on page 13

**0828**

**Explanation:** Index Maintenance attempted to insert a new index entry, with index specified as unique, and encountered a duplicate index entry.

**Analysis:** This is a pseudoabend presented by DFSDXMT0, the DL/I index maintenance module. The SNAP of the control blocks should be available for problem diagnosis. Print the type X'67' log record using the file select and formatting print program, DFSERA10. The OPTION PRINT statement must specify EXITR=DFSERA30.

To find the registers applicable for a pseudoabend it is necessary to locate the save area set used by DFSDXMT0. Get the entry point address from the SCD at label SCDDXMT0 and search down the save sets for a matching address in register 15. This save set contains the registers when the index maintenance module was called.

The printout of the PST is important. Fields PSTRTCDE and PSTFNCTN are the two important fields. PSTWRK4 contains the SSA of the index entry segment. Message DFS0840I precedes the abend and is issued to the IMS master console. The status code is NI.

If duplicate secondary index entries occur, the index should be specified non-unique, and an overflow entry-sequence (EDS) data set should be provided. If running batch, backout should be executed to resynchronize data base indexes.

Key	Label	Description
PSTFNCTN=X'F4' PSTRTCDE=X'28'	DISRTVSM	A duplicate index record was encountered while attempting to insert a new index entry.

**System action:** At the time the error was encountered, the segment being inserted was already added to the database and the other indexes might already have been updated. If you are executing in a batch system with disk logging and if the execution parameter BKO=Y (dynamic backout) was not specified, it is necessary to run the Database Backout utility. Otherwise the system runs dynamic backout.

**Programmer response:** If duplicate secondary index entries occur, the index should be specified as non-unique, and an overflow entry-sequenced data set should be provided.

**Problem determination:** 1, 11, 35

*For APAR processing:* A copy of the system log

**Source:** IMS abends

**Module:** DFSDXMT0

**Related information:**

 DFS0840I

Chapter 2, “IMS problem determination,” on page 13

**0829**

**Explanation:** When a VSAM ERASE call was issued, an invalid return code was returned from the VSAM buffer handler.

**Analysis:** 0829 is a pseudoabend presented by DFSDXMT0, the DL/I index maintenance module. The SNAP of the control blocks should be available for problem diagnosis. Print the type X'67' log records using the file select and formatting print program. DFSERA10. The OPTION PRINT statement must specify EXITR=DFSERA30.

To find the registers applicable for a pseudoabend it is necessary to locate the save area set used by DFSDXMT0. Get the entry point address from the SCD at label SCDDXMT0 and search down the save sets for a matching address in register 15. This save set contains the registers when the index maintenance module was called.

An index deletion was requested. The correct index segment is found, a type X'50' log record is written and the buffer handler is called to execute an erase of the index segment entry. The PSTRTCDE field of the PST should be referenced to determine the buffer handler return code. Also, the buffer handler trace (BFSP) should be located and the order of the calls preceding the erase recorded.

## 0830

Key	Label	Description
PSTABTRM=X'1000033D' PSTFNCTN=X'F1'	DDLTVSAM	An erase operation returned an invalid buffer handler return code.

*For APAR processing:* Copy of the system log, dependent region dump

**System action:** IMS terminates abnormally.

**Programmer response:** No action is required.

**Problem determination:** 11, 17, 35

**Source:** IMS abends

**Module:** DFSDXMT0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

## 0830

**Explanation:** An application program issued a GU call, and the DC call handler (DFSDLA30) detected the message has invalid fields such as MSC SYSIDS, IMSID, or SOURCE name.

This error can occur:

- While the message was in flight in a remote IMS in an MSC network (and on the queue in a shared-queue environment), the resources from which these fields were set have changed.
- When one or more systems in a shared-queues environment are defined with MSC links, while others are defined without MSC links. In an IMSplex, if one IMS system is MSC capable, then all IMS systems in the IMSplex must be MSC capable. To enable an IMS system for MSC, define an MSC link for the IMS system during system definition. This MSC link definition does not have to be functional.
- Due to an IMS internal error.

**Analysis:** 0830 is a standard pseudoabend that is issued by DFSDLA30. The following events occur:

- Message DFS554A is issued to the master terminal.
- A pseudoabend 0830 is issued to back out the message and return it to the message queue. The message is put on the local queue (non-SQs) or ready queue (SQs) and is unlocked.
- The transaction and program are stopped.
- A type 6701-LA3E record is written to the IMS log. A type 67D0 record might also be written. Type 67FF pseudoabend records are not written.

Print the type X'6701' ID=LA3E and X'67D0' log records using the Log File Select utility DFSEAR10.

The 6701-LA3E records contain the following control blocks:

- PCB (the I/O PCB in the PST).
- SMB (transaction code and characteristics).
- INP\_QBUF (first or only QBUF of input message).
- PST (Partition Specification Table for the dependent region).
- Additional information logged in the second 6701-LA3E record includes:

**AB830RSN**

ABENDU0830 reason code - the ABEND 0830 R15 reason code from the DFSDLA30 registers saved at label GETAB830 (see table below)

**PSTJOBNM**

Job name and step name from PST

Additionally, the second 6701-LA3E record might contain the following fields for MSC-related problems:

**SIDPFXHD**

SID table prefix and header (if MSC is included in IMS as part of the system definition process - see macro DFSMSSIT).

**SIDENT**

SID table entry (if MSC is included in IMS as part of the system definition process - see macro DFSMSSIT).

**IMIDENT**

IMS ID table entry (if MSC is included in IMS as part of the system definition process - see macro DFSMSIMT).

Assemble the following CSECT to get the DSECTs for these control blocks:

```
CSECT
IDLI TPCBASE=0
IAPS SMBBASE=0
ILOGREC TYPE=DSECT,RECID=01
IPST ISIT=YES
END
```

**Remember:** The PST contains information fields that are used to process the GU call. The PST also contains the save areas PSTSAV1 through PSTSAVxx (where xx=20). The save area for DFSDLA30 is PSTSAV4 or PSTSAV5. When the pseudoabend is detected, the current registers are stored there, including the abend 0830 reason code in Reg15 and return code (if applicable) in Reg1.

The following information fields are logged:

- The input message in error is logged in the INP\_QBUF. The format of this message is in the QLOGMSGP macro.
- The pseudoabend code 1000033E (X'33E' = decimal 0830) is saved in field PSTABTRM. Assemble the IPST DSECT using the IPST macro.

The registers from DFSDLA30 when the error was detected are in save area PSTSAV4. This save area and all other save areas have the following format:

```
PSTSAV4 DC F'0'          0 or SAP pointer for first set
PSTBCK4 DC A(PSTSAV3)   Backward pointer to PSTSAV3
PSTFRD4 DC A(PSTSAV5)   Forward pointer to PSTSAV5
         DC 15F'0'      REG14 through REG12 saved here
```

- PSTSAV4 contains the registers in DFSDLA30 when the error was detected.
- Reg1 is the return code from the call that detected the error (for example, a FINDEST error), if the error is related to such a call.

The PSTSAV5 save area contains some additional diagnostic information as follows:

- Reg14, Reg15, Reg0 contain the 12 byte UTC time from SCDCKTIM. This time is when this IMS was restarted. Use this time to compare the message origin UTC time, MSGUTC field in the system extension prefix (type 8A) in the INP\_QBUF. Determine if this message was sent before this IMS was restarted.
- Reg1 contains the PSEUDOABEND code X'1000033E'. This code is also in PSTABTRM in the PST.
- Reg2 and Reg3 contain the IMSID of this IMS. Compare this IMSID with the message origin IMSID (field MSGORGID in the basic prefix in the INP\_QBUF) to determine whether this message originated from this IMS or another IMS in the IMSplex or MSC network.

The 67D0 record, if written by DFSQLD00, logs the following information:

- SCD, the IMS System Contents Directory
- ECB (the same address as PST)
- 32 byte UOW
- 10 byte service request detail
- 8 byte destination name
- 24 byte service request that failed
- 4 byte return code
- 4 byte reason code
- Dump entry 1
- Dump entry N

DFSQCLD0 is OCO. You might need to contact IBM Software Support to analyze this information.

The abend 0830 Reg15 reason code descriptions are provided in the following table:

Key	Label	Description
Reg15=00000001		<p>A local source name at PSTSYMBO was found, but it is not a valid LTERM (that is, the block returned by FINDEST was either an SMB or an LNB). It must be either a local CNT or a remote RCNT.</p> <p>The most likely cause is either:</p> <ol style="list-style-type: none"> <li>1. A system definition error was encountered.</li> <li>2. If you are using shared queues (SQ), the IMSplex was restarted before all the IMS systems have joined the shared queues group (SQG).</li> </ol> <p>To correct the problem, do:</p> <ol style="list-style-type: none"> <li>1. If case 1, correct the system generation.</li> <li>2. If case 2, either restart all the IMS systems in the SQG before starting the regions, or dequeue this message from the SQ.</li> </ol>
Reg15=00000002		A non-zero return code was returned in REG15 from the call to QMGR. The return code is moved to R1 and saved in REG1 in PSTSAV4 or PSTSAV5. The return code was determined to be invalid. This problem is an IMS internal error.
Reg15=00000003		Message has an APPC/OTMA source name at MSGIDSTN/PSTIDSTN but the message did not have an APPC/OTMA prefix type 'X'87' segment. This problem is an IMS internal error.
Reg15=00000004		A CQS error occurred.
Reg15=00000006		The source SYSID at PSTSIDS, taken from the message at MSGMSIID (one-byte SYSID) or MSGRSID (two-byte SYSID) is remote, but the source name at PSTIDSTN/MSGIDSTN is not an MSNAME. Determine if this SYSID has changed since the message was created. If it has changed, a procedure error has occurred. Otherwise, this error is probably an IMS internal error.
Reg15=00000007		The source SYSID at PSTSIDS, taken from the message at MSGMSIID (one-byte SYSID) or MSGRSID (two-byte SYSID) is remote. The source is a local APPC/OTMA name. However, the message is processed on a BE IMS, which is invalid. Determine if the source SYSID has changed since this message was created. If yes, a procedure error has occurred. That is, a remote IMS with remote SYSIDs was opened alone without starting the front-end IMS in the SQG. Start the front-end IMS and release the message to process again. Otherwise, this error is probably an IMS internal error.
Reg15=00000008		The source SYSID at PSTSIDS, taken from the message at MSGMSIID (one-byte SYSID) or MSGRSID (two-byte SYSID) is invalid. Verify whether this SYSID has changed since the message was created. If it has changed, a procedure error has occurred. If the value of PSTSIDS is zero, the message probably came from an IMS that is not MSC capable or was not capable when the message was put on the shared queue. Verify that all the IMS systems in the shared-queue group are MSC capable or non-MSC capable. In an IMSplex, if one IMS system is MSC capable, then all IMS systems in the IMSplex must be MSC capable. To enable an IMS system for MSC, define an MSC link for the IMS system during system definition. This MSC link definition does not have to be functional. Otherwise this abend is most likely an IMS internal error.
Reg15=00000009		The source SYSID at PSTSIDS, taken from the message at MSGMSIID (one-byte SYSID) or MSGRSID (two-byte SYSID) is invalid. The message is from a non-APPC or non-OTMA LTERM that was local to this IMS. Determine if this SYSID has changed since the message was created. If it has changed, a procedure error has occurred. Otherwise, this error is probably an IMS internal error.



Key	Label	Description
Reg15=0000000A		The message source SYSID at MSGMSIID (one-byte SYSID) or MSGRSID (two-byte SYSID) was remote or invalid. The message was determined to be a valid local message. An attempt was made to correct the source SYSID using the MTO terminal CNTSIDL, but this SYSID was not local, so the SYSID could not be converted. This error is probably an IMS internal error.
Reg15=0000000B		The source name is an MSNAME type, but the source name at PSTIDSTN/MSGIDSTN is an APPC/OTMA token name. This problem is invalid and is probably an IMS internal error.
Reg15=0000000C		The source SYSID is remote and the system source name at PSTIDSTN/MSGIDSTN matches the terminal source name at PSTSYMBO/MSGMSINM. These names only match for local messages. This problem is probably an IMS internal error.
Reg15=0000000D		A FINDDEST call to locate the source LTERM name for a non-APPC or non-OTMA local LTERM at PSTIDSTN/MSGIDSTN failed. Force create was used, so the LTERM could not be located or created. Verify that ETO is active. If it is not active, activate ETO and see if the problem is resolved. If ETO is active, this problem is probably an IMS internal error.
Reg15=0000000E		A FINDDEST call to locate the source MSNAME at PSTIDSTN/MSGIDSTN received a non-zero return code in Reg15. Verify that the MSC MSNAME is valid for this IMS. If not, this MSNAME might have been removed while the message was on the SQ to be processed. Otherwise, this problem is probably an IMS internal error.
Reg15=0000000F		A request to create a dynamic MSNAME for the MSNAME at PSTIDSTN/MSGIDSTN failed. The MSNAME does not exist and cannot be created. Verify that this name is a valid MSNAME and does not exist as an LTERM or TRANCODE. If this is not the problem, this problem is probably an IMS internal error.
Reg15=00000010		A FINDDEST call to locate the source MSNAME at PSTIDSTN/MSGIDSTN was not successful. This IMS doesn't support shared queues. Verify that the MSNAME was not removed from the system definition after the message was created. Also verify that IMS was started with warm start and the message was left on the queue. Changing MSNAMES requires at least a CTLBLKS system generation and a cold start to remove messages from the queue. Otherwise, this problem is probably an IMS internal error.
Reg15=00000011		A request to create a dynamic MSNAME for the MSNAME at PSTIDSTN/MSGIDSTN failed. The MSNAME does not exist and cannot be created. Verify that this name is a valid MSNAME and does not exist as an LTERM or TRANCODE. If not, this problem is probably an IMS internal error.
Reg15=00000012		A FINDDEST call to locate the source LTERM name for a non-APPC or non-OTMA local LTERM at PSTSYMBO/MSGMSINM received a non-zero return code in Reg15. This indicates that the MSC message originated in this IMS and was sent to a remote IMS to process. The remote IMS issued a program-to-program switch back to the source IMS. Verify that the name is still a valid LTERM and not a TRANCODE or MSNAME. If not, this problem is probably an IMS internal error.
Reg15=00000013		A FINDDEST call to locate the source LTERM name for a non-APPC or non-OTMA local LTERM at PSTSYMBO/MSGMSINM failed. Force create was used, so the LTERM could not be located or created. Verify that ETO is active to create the LTERM. This indicates that the MSC message originated in this IMS and was sent to a remote IMS to process. The remote IMS issued a program-to-program switch back to the source IMS. Verify that the name is still a valid LTERM and not a TRANCODE or MSNAME. If not, this problem is probably an IMS internal error.

Key	Label	Description
Reg15=00000014		A request to create a dynamic MSNAME for the MSNAME at PSTIDSTN/MSGIDSTN failed. The MSNAME does not exist and cannot be created. Verify that this name is a valid MSNAME and does not exist as an LTERM or TRANCODE. If not, this problem is probably an IMS internal error.
Reg15=00000015		A TMR prefix MSGMSC, type 8C, could not be located by the DLA3MGPL routine in DFSDLA30. The message is in the INP_QBUF area. This problem is an IMS internal error.
Reg15=00000016		An MSC extension prefix MSGMSCE, type 8B, could not be located by the DLA3MGPL routine in DFSDLA30. The message is in the INP_QBUF area. This problem is an IMS internal error.
Reg15=00000017		The TMR prefix MSGMSC, type 8C, could not be located by the DLA3MGPL routine in DFSDLA30. The message is in the INP_QBUF area and is an APPC/OTMA message processing in the front-end SQ IMS. This problem is an IMS internal error.
Reg15=00000018		The TMR prefix MSGMSC, type 8C, could not be located by the DLA3MGPL routine in DFSDLA30. The message is in the INP_QBUF area and is an APPC/OTMA message processing in the front-end SQ IMS. This problem is an IMS internal error.
Reg15=00000019		A request to create a dynamic MSNAME for the MSNAME at PSTIDSTN/MSGIDSTN failed. The MSNAME does not exist and cannot be created. Verify that this name is a valid MSNAME and does not exist as an LTERM or TRANCODE. If not, this problem is probably an IMS internal error.

**System action:** The following actions occur:

- A pseudoabend 0830 is issued and message DFS554A is sent to the master terminal.
- The transaction and program are stopped.
- The message is placed back on the ready queue (unlocked).
- A 6701-LA3E record is written to the IMS log.

**Operator response:** If you are using a shared queues IMS, first verify that the message was not processed successfully by another IMS in the shared queues group. If the message is still on the queue, it might be removed by issuing a /DEQUEUE TRANSACTION *trannname* PURGE1. *Trannname* is the TRANCODE name in message DFS554A. You can also issue a /START command to restart the transaction and program. If the error is corrected, let the application program schedule again and GU the message.

If you are using a shared queues IMSplex that is multiple systems coupling (MSC) capable, and the message was entered on an IMS that was taken down, and the error is due to an invalid MSC SYSID, the error might be corrected by starting all the IMS systems in the shared queues group. This action causes the SYSIDs to be exchanged between the IMS systems in the SQG and make all the SYSIDs valid.

**Programmer response:** This problem might be a procedural or IMS internal error. To determine which and a possible resolution, the type 6701-LA3E records (and X'67D0' records if produced) need to be printed with the File Select and Formatting Print utility (DFSERA10), and analyzed.

**Problem determination:** 5, 6, 17h

The following information applies to 17h:

- Print the log records type x'6701'ID=LA3E.
- Use the Log File Select utility DFSERA10.
- Use the following control cards:
 

```
OPTION PRINT 0=5,V=6701,L=2,C=M,T=X,E=DFSERA30
OPTION PRINT 0=9,V=LA3E,L=4,C=E,T=C,E=DFSERA30
OPTION PRINT 0=5,V=67D0,L=2,C=E,T=X,E=DFSERA30
```

**Source:** IMS abends

**Module:** DFSDLA30

**Related information:**

 DFS554A (Messages and Codes)

---

**0831**

**Explanation:** IMS detected a logic error during processing block serialization for full function databases. The abend was issued because DFSBLSER determined that a task has requested to release the block serialization lock on a block that it does not own.

**Analysis:** Abend 0831 is a standard abend issued when a PST attempts to release the block serialization lock on a block that it does not own. Starting from the first element at SCDHDPTR, space management has scanned the entire chain of elements searching for the element for this PST. When the end of the chain is reached and the element is not found then a branch is taken to label WERELOST.

At the time of the abend the normal register usage is as follows:

- R3** NEXT contains zeros. This indicates that the entire block serialization chain has been scanned and the element for this task was not found on the chain.
- R4** OURS contains the pointer to element that was not found on the chain. PSTDHDEL also contains the element.
- R10** PST address for the task that should be waiting. R11 - SCD address.

**System action:** IMS terminates abnormally with an IMS abend 0831.

**Programmer response:** If the abend occurs during IMS dependent region activity then the IMS can be emergency restarted.

If the abend occurs during emergency restart of IMS during dynamic backout then IMS must be restarted with either option COLDBASE or COLDSYS on the /ERESTART command and backout must be performed using batch backout.

If the abend occurs during batch backout then the databases must be recovered using forward recovery.

**Problem determination:** 1, 2, 3, 4, 5, 6, 8, 14, 15, 25, 40

**Source:** IMS abends

**Module:** DFSMMUD0, DFSRDBC0

---

**0832**

**Explanation:** While requesting or freeing space in an HDAM, HIDAM, PHDAM, or PHIDAM database, an incorrect free space element (FSE) was found or created in the block whose prefix address is in PSTBUFFA. The current call might not have been completed. Several conditions are checked, any of which will cause the abend to be issued:

- The FSE length exceeds the DCB buffer length.
- The FSE pointers or length fields are not positive signed numbers.
- The FSE position overlaps the next FSE or the end of the block.

**Analysis:**

0832 is a pseudoabend detected in one of two modules: DFSRCHB0, the search block routine, or DFSFRSP0, the free space routine. At label AB832 in both of these routines, a direct branch is made to CSECT DFSAB832, which is in the source for DFSRCHB0. DFSAB832 saves the current registers in the last-level save area, PSTSAVL, starting at offset X'C'. An X'FF' is in the first byte at X'C', which indicates register 14. Register 12, which is the base register for both DFSRCHB0 and DFSFRSP0, should be used to determine which module detected the error condition.

DFSAB832 also traces the current HD space management request in the DL/I trace table. This trace entry contains the register 14 return address of the caller of space management. The caller is backout, delete or replace, or load or insert.

Field PSTDATA points to the buffer containing the block or CI currently being used.

**For DFSRCHB0:**

DFSRCHB0 is entered to search through the FSE chain and locate a block of free space which makes the best fit for a segment of data. Prior to searching the chain, three place holders are initialized to contain;

1. A perfect fit
2. Enough space for the segment plus a minimum length segment
3. Space for two segments

## 0832

For DFSRCHB0, register 2 points to the start of the block.

Key	Label	Description
Reg2=A(BLOCK) Reg3>Reg11	CTNUSRCH	If free space pointer shows a position that exceeds buffer length, the abend is issued.
Reg2=A(BLOCK) Reg7>Reg11	MAXOK	If FSE length exceeds buffer length, then the abend is issued.
Reg2=A(BLOCK) Reg8>Reg3	MAXOK	If the prior FSE overlaps into our current position, the abend is issued.
Reg2=A(BLOCK) Reg8=PSTWRK2	SKIPZRO	Check if new FSE exceeds buffer length. A compare is done between the sum of the contents of PSTWRK3 and PSTWRK2, and what is in register 11. If register 11 is lower, the abend is issued.
Reg2=A(BLOCK) (Reg9+Reg3)>Reg11	TWOHOLES	If free space pointer exceeds buffer length, the abend is issued.
Reg2=A(BLOCK) PSTWRK2>Reg11	TWOHOLES	If previous FSE overlays current FSE, the abend is issued.
Reg2=A(BLOCK) (PSTWRK2+PSTWRK3) >Reg11	TWOHOLES	If FSE exceeds buffer length, the abend is issued.

### For DFSFRSP0:

DFSFRSP0 is called due to a request to free a segment. The segment is located and the determination made as to whether a new FSE must be built or whether this free space could be appended to an existing FSE. In either case, the task is performed and the bit map updated as required. For DFSFRSP0, register 6 will point to the start of the block.

Key	Label	Description
Reg6=A(BLOCK) Reg3>Reg8 Reg6 $\neq$ RBASE	NOTODD	The buffer length (DCB plus X'18') is compared to the number of bytes to be freed. If number of bytes to be freed is greater, the abend is issued. RBASE will later be defined as PSTDATA minus PSTOFFST.
Reg6=A(BLOCK)	NTVSM	The creation of an FSE would overlay a bit map.
Reg6=A(BLOCK) Reg9<0	SCAN	Check that FSE length is not negative. If it is negative, the abend is issued.
Reg6=A(BLOCK) Reg9>Reg8	SCAN	Check if FSE length exceeds buffer length. If it does, the abend is issued.
Reg6=A(BLOCK) Reg9>Reg3	SCAN	Check that prior FSE does not overlap into our current position. If it does, the abend is issued.
Reg4=Reg3 Reg6=A(BLOCK)	SCAN	Check that FSE is not already free or that the area is not zeros. If it is, the abend is issued.
Reg6=A(BLOCK) Reg11>(Reg3 $\neq$ 0)	ONWARD	Check that the next FSE is not overlaid. If it is, the abend is issued.
Reg6=A(BLOCK) (Reg11+PSTWRK3)>Reg8 PSTFNCTN=E6	XITDOCH	Check that the FSE does not exceed buffer length; if it does, the abend is issued.
Reg6=A(BLOCK) (PSTWRK4+2)+ (PSTWRK3+2)>Reg8	XITDOCH	After building a second FSE, check that its length does not exceed buffer length; if it does, the abend is issued.

**System action:** The application program terminates abnormally.

**Programmer response:** To identify which database is in error, obtain the data management block number from

PSTDMBNM and the DDIR from the SCD; use the data management block number to find the DDIR number that applies. Reorganize the database to correct FSEs, or do the following:

1. Set the FSE anchor point in the block to zero. This will prevent any more use of the FSEs in that block.
2. Create a memory dump of the database using the Database Image Copy Utility. This is necessary because backout of activity in this block, which preceded setting the FSE to zero, should not be attempted. Backout will not replace segments correctly if the space expected is not there, and results will be unpredictable.

**Problem determination:** 17a, 17b, 17g, 35, and save the log tapes for reference to activity on the incorrect data block

**Source:** IMS abends

**Module:** DFSRCHB0, DFSFRSP0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

### 0833

**Explanation:** An invalid SYSID was encountered. Register 15 contains the invalid SYSID. This error is probably either a logic error or a destroyed data area.

**Analysis:** 0833 is a standard abend issued by DFSDLA30, the communications DL/I call handler. This module processes DL/I calls that reference the message queues.. The program status word (PSW) at entry-to-abend should be used to isolate the failing module.

The subroutine at label GETLNB is invoked, using a BAL on register 10, to obtain the logical link name block (LNB) associated with the originating SID. The originating SID value is obtained from the PST (label PSTSIDS) and inserted into register 15. If the value is zero (register 15=zero) or greater than the maximum defined (SCDSIDN), the abend is issued.

The program status word (PSW) at entry-to-abend points to the instruction within label xxxABEND from which the abend (SVC 13) is issued. You can use register 14 to isolate to the specific label. Register 12 is the base register. Register 9 is the pointer to the PST and register 11 points to the SCD.

Key	Label	Description
	CHG1400	A conversational program's input originated in a remote system, but the destination LNB associated with the input system was not found defined in this system.
	CHG8200	A nonconversational program's input originated from a remote system, but the destination LNB associated with the input system was not found in this system.
Reg15=RC from get dynamic LNB (DLA3DLNB) routine		GU, GN, CHKPT, or ROLB processing made a request for a dynamic LNB for the MSNAME at PSTIDSTN and SYSID at PSTSIDS, and received a return code of 8 or greater.
Reg15=RC from get dynamic LNB (DLA3DLNB) routine		GU, GN, CHKPT, or ROLB processing made a request for a dynamic LNB for the MSNAME at PSTIDSTN and SYSID at PSTSIDS, and received a return code of 8 or greater. REG3 contains an error code that indicates which process in the DLA3DLNB routine failed.
Reg15=RC from get dynamic LNB (DLA3DLNB) routine		GU, GN, CHKPT, or ROLB processing made a request for a dynamic LNB for the MSNAME at PSTIDSTN and SYSID at PSTSIDS, and received a return code of 8 or greater. REG3 contains an error code that indicates which process in the DLA3DLNB routine failed.
Reg15=RC from get dynamic LNB (DLA3DLNB) routine	GMOV2830	GU, GN, CHKPT, or ROLB processing made a request for a dynamic LNB for the MSNAME at PSTIDSTN and SYSID at PSTSIDS, and received a return code of 8 or greater. REG3 contains an error code that indicates which process in the DLA3DLNB routine failed.
	ISRT0106	The message originated in the remote system, but the destination LNB could not be found defined in this system.
	ISRT0464	The destination system ID was found not to be the same as the local system ID.

## 0834 • 0840

*Possible cause:* Internal logic error exists or data areas have been destroyed.

**System action:** IMS terminates abnormally.

**Problem determination:** 35

**Source:** IMS abends

**Module:** DFSDLA30

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

### 0834

**Explanation:** While searching for space, Space Management received three consecutive read error returns from the buffer handler. An I/O error message was issued by the buffer handler prior to this abend.

**Analysis:** This is a pseudoabend issued by DFSGGSP0 at label ABEND834 whenever the buffer handler, during calls from space management, has encountered three I/O errors reading the same data set.

**System action:** The application program terminates abnormally.

**Programmer response:** Refer to the I/O error message issued prior to the termination of the transaction.

**Source:** IMS abends

**Module:** DFSGGSP0

---

### 0835

**Explanation:** An error was detected in a field descriptor block (FDB) for a field within an index source segment. The field represented by the FDB was specified in the SUBSEQ or DDATA operand of an XDFLD statement and the FDB indicates that it is a system-related field, but the field name does not begin with /CK or /SX. The index maintenance module, DFSDXMT0, detected the error.

**Analysis:** 0835 is a pseudoabend issued by DFSDXMT0, the DL/I index maintenance module. The SNAP of the control blocks should be available for problem diagnosis.

To find the registers applicable for a pseudoabend, it is necessary to locate the save area set used by DFSDXMT0. Get the entry point address from the SCD at label SCDDXMT0, and search down the save sets for a matching address in register 15. This save set will contain the registers when DFSDXMT0 was called.

The abend occurs when processing an index source segment. An FDB for a field that was specified in the SUBSEQ or DDATA operand of an XDFLD statement indicates that the field is a system-related field, but the field name does not begin with /CK or /SX. The address of the PST is in register 1 in the save set for DFSDXMT0, and the PST has the address of the SDB or the PSDB for the index source segment at label PSTWRK1.

**System action:** IMS terminates abnormally.

**Programmer response:** This problem is probably an IMS internal error.

**Problem determination:** 4, 35

**Source:** IMS abends

**Module:** DFSDXMT0, DFSURGU0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

### 0840

**Explanation:** The Segment Edit/Compression routine detected a processing error while attempting compression or expansion services.

**Analysis:** Use the following information to help determine the cause of the abend:

#### SYSTEM LOG

The pseudoabend SNAP of control blocks written to the system log will be needed to diagnose the error.

Print the type X'67' log records by using the file select and formatting print program, DFSERA10. The OPTION PRINT statement must specify EXITR=DFSERA30.

#### DATA COMPRESSION

For hardware data compression (DFSZLDX0) problems, see the return codes for abend 3325. For software data compression problems, see the return codes for abend 2990 (DFSCMPX0) and abend 2991 (DFSKMPX0).

#### PSTLOGWA

PSTLOGWA contains diagnostic information that is related to the processing error.

Offset	Item
X'00'	The label name or error reason that is returned by the Segment Edit/Compression routine.
04	The user abend code that would have been issued by the routine.
08	Parameter registers passed to the routine: <b>word 1</b> Does not contain valid information. <b>word 2</b> Address of source segment. This is an input field. <b>word 3</b> Address of destination segment. This is an output field. <b>word 4</b> PSDB address, or zero if Fast Path. <b>word 5</b> Address of DMBCPAC or DBFCMPC if Fast Path. <b>word 6</b> Entry Code.
X'20'	A copy of the DMBCPAC or DBFCMPC if Fast Path.

**For DFSDLR00:** This is a pseudoabend that is issued from module DFSDLR00 at label COMPCALL. For analysis, see SYSTEM LOG and PSTLOGWA.

**For DFSDXMT0:** This is a pseudoabend that is issued from module DFSDXMT0 at label NOKEY. For analysis, see SYSTEM LOG and PSTLOGWA.

**For DFSDDLE0:** This is a pseudoabend that is issued from module DFSDDLE0 at label NDXREP3, NOASRT2, or CKKEYLP. For analysis, see SYSTEM LOG and PSTLOGWA.

**For DFSDLDC0:** This is a pseudoabend that is issued from module DFSDLDC0 in the EXPAND routine. For analysis, see SYSTEM LOG and PSTLOGWA.

| **For DFSDCAP0:** This is a pseudoabend that is issued from module DFSDCAP0 at routine PSEUDO\_ABEND. For  
| analysis, see SYSTEM LOG and PSTLOGWA.

**For DFSPRIMS:** This is a pseudoabend that is issued from module DFSPRIMS at routine IMS0B. Register 14 contains the address of PSTLOGWA.

| **For DBFCMP10:** This is a pseudoabend that is issued from module DBFCMP10 at routine CALLEXIT. At abend,  
| register 2 contains the error reason that is returned by the Segment Edit/Compression routine.

| **For DBFCMP11:** This is a pseudoabend that is issued from module DBFCMP11 at routine CALLEXIT. At abend,  
| register 2 contains the error reason that is returned by the Segment Edit/Compression routine.

| **For DBFDAC50:** This is a pseudoabend that is issued from module DBFDAC50 at routine DECOMPRESS. At abend,  
| register 15 contains the label name or error reason that is returned by the Segment Edit/Compression routine.  
| Register 14 contains the abend code that is returned from the routine.

**System action:** The application abnormally ends.

**Programmer response:** Ensure the edit/compression routine has been built and linked properly.

**Problem determination:** 4, 17a, 17b, 18, 19, and the source and link listings of the edit/compression routine

**Source:** IMS abends

| **Module:** DFSDLR00, DFSDXMT0, DFSDDLE0, DFSDLDC0, DFSDCAP0, DFSPRIMS, DBFCMP10, DFSURGU0,  
| DBFDAC50

**Related information:**

Abend 2990

Abend 2991

Abend 3325

---

**0841**

**Explanation:** The address of the Segment Edit/Compression routine was invalid when IMS attempted to start the routine.

**Analysis:** Register 2 and Register 3 contain the name of the Segment Edit/Compression routine.

**System action:** The batch or message processing region is abnormally ended. The control region continues normally.

**Programmer response:** Check message DFS0733I for additional information. Ensure that the Segment Edit/Compression routine is accessible and that IMS has enough storage to load the routine.

**Source:** IMS abends

**Module:** DBFCMP10, DBFCMP11

**Related information:**

 DFS0733I (Messages and Codes)

---

**0842**

**Explanation:** An error occurred while attempting to do a pseudo data set extend to an OSAM database in a data sharing environment.

**Analysis:** For **DFSDBH40:** This is a pseudoabend while extending the data set in a data sharing environment.

Message DFS0842I is issued to the master terminal operator prior to the abend indicating the DDNAME, DBDNAME, and DSNNAME of the data set with the error. Refer to message DFS0842I for the definitions of the reason codes.

For **DFSDVBH0:** This is a standard abend issued for a non-Fast Path failure when the user codes DUMP=YES on the OPTIONS statement. Module DFSDVBH0 intercepts the pseudoabend 0842 and changes it to a standard abend.

**System action:** Message DFS0842I was issued prior to the abend indicating the DBDNAME, DDNAME, and DSNNAME of the data set that encountered the extend error. The application program terminates abnormally with pseudoabend 0842, and IMS continues processing. However, if DUMP=YES was specified on the DL/I buffer options statement, the application terminates with standard abend 0842, and the control region terminates abnormally.

**Programmer response:** Ensure that all database data sets can be shared from all subsystems that share the data set.

**Problem determination:** 1, 2, 4, 17

**Source:** IMS abends

**Module:** DFSDBH40, DFSDVBH0

**Related information:**

 DFS0842I

Chapter 2, "IMS problem determination," on page 13

---

**0843**

**Explanation:** The block number passed to DL/I from your randomizing routine, when multiplied by the block size, yields a value that exceeded the maximum addressable data address. The data address must be addressable by 32 bits in VSAM or 31 bits in OSAM.

**Analysis:** 0843 is a pseudoabend issued by module DFSDLR00.

**System action:** The processing task is terminated abnormally.

**Programmer response:** Examine the value returned from the randomizing routine in DMBDACP. If the block number portion multiplied by the CI size is greater than the corresponding maximum value allowable, change the randomizing routine or reduce the size of the root addressable area.

**Problem determination:** 11, 25, 35

**Source:** IMS abends



**Module:** DFSDLR00

**Related information:**

Chapter 2, "IMS problem determination," on page 13

**0844**

**Explanation:** No space is available in the database, or the data set is defined as DSNAME=NULLFILE or DD DUMMY. This problem can also occur if an error occurs during CLOSE TYPE=T processing for a VSAM data set that has an extent.

**Analysis:** The following defines the modules that issues this abend:

**DBFIRC10**

No space is available in the DEDB area.

**DFSDVSM0, DFSGGSP0**

No space is available in the database, or the data set is defined as DSNAME=NULLFILE or DD DUMMY. This problem can also occur if an error occurs during CLOSE TYPE=T processing for a VSAM data set that has an extent.

**DFSDVBH0**

A non-Fast Path failure that has an OPTIONS statement with the DUMP=YES parameter.

**For DBFIRC10:** This pseudoabend is issued when the DEDB area is full. Message DFS2765W or DFS2767I indicates that the DDNAME of the data set that is out of space was sent to the master terminal operator before abnormal termination. A return code of 16 is set by DBFMCLX0, which is called to process a DEDB DL/I call. Sync return code X'36' is set in the EPSTSPRC field.

**For DFSDVSM0, DFSGGSP0:** This standard abend is issued for a non-Fast Path failure when DUMP=YES is specified on the OPTIONS statement. Module DFSDVBH0 intercepts the pseudoabend 0844 and changes it to a standard abend. For CLOSE TYPE=T failures, a X'4C' log record is written to indicate that the database is stopped. The PSTRTCDE field is set to new return code X'50'.

**For DFSDVBH0:** This standard abend is issued for a non-Fast Path failure when DUMP=YES is specified on the OPTIONS statement. Module DFSDVBH0 intercepts the pseudoabend 0844 and changes it to a standard abend.

**System action:** For Fast Path data sets, message DFS2765W or DFS2767I is issued before the abend, indicating that the DDNAME data set is out of space.

For non-Fast Path data sets, message DFS0844I is issued before the abend, indicating the DBDNAME and DDNAME of the data set that encountered the out-of-space condition. The application program terminates abnormally with pseudoabend 0844, and IMS continues processing. However, if DUMP=YES was specified on the DL/I buffer OPTIONS statement, the application terminates with standard abend 0844, and the control region terminates abnormally.

**Programmer response:** For Fast Path data sets, the size of the data set must be increased. Space might be made available by running the DEDB Direct Reorganization utility.

For non-Fast Path data sets, the database might be invalid. Run the Database Backout utility or re-create the database, allocating more space. If a LIST VTOC job is run, it might provide misleading information. While offline (using the /DBR command), you must copy the database to a larger space allocation. Then make the database available to the online region by using the /START command.

**Problem determination:** 4, 12

**Source:** IMS abends

**Module:** DBFIRC10, DFSDVBH0, DFSDVSM0, DFSGGSP0

**Related information:**

 DFS0844I

 DFS2765W

 DFS2767I

Chapter 2, "IMS problem determination," on page 13

## 0845

**Explanation:** An unexpected condition was encountered in DFSTOCL0, DFSDVBH0, DFSNOTB0, DFSDVSM0, DFSNOT0, or DFSFXC50. This problem is an IMS system error. Using either the reason code from the DFS1058E message or the value in register 6 from the register at Abend, locate the matching value in the key fields in one of the tables. The label referenced for that key is the label in the module where the error was detected.

**Explanation: For DFSDVBH0:**

0845 is a standard abend issued from several labels in module DFSDVBH0. The PSW points to the section of DFSNOTB0 that detected the error.

If the PSW points to the label NOTBGMER, register 10 points back to the address in DFSDVBH0 that branched to NOTBGMER.

Key	Label	Description
Reg1=X'034D' Reg6=X'E4084510' Rsn Code=010	AB845R10	BQEL is not a VSAM BQEL.
Reg1=X'034D' Reg6=X'E4084515' Rsn Code=015	AB845R15	BUFC write error indicator not set.

**For DFSTOCL0:**

0845 is a standard abend issued from several labels in module DFSTOCL0. The PSW points to the section of DFSTOCL0 that detected the error.

If the PSW points to the label NOTBGMER, register 10 points back to the address in DFSNOTB0 that branched to NOTBGMER.

Key	Label	Description
Reg1=X'034D' Reg6=X'E4084520' Rsn Code=020	AB845020	BQEL is not a VSAM BQEL.
Reg1=X'034D' Reg6=X'E4084525' Rsn Code=025	AB845025	BUFC write error indicator is not set.

**For DFSDVSM0:**

This problem is a standard abend issued from several labels in module DFSDVSM0. The program status word (PSW) at entry-to-abend and the registers in 'REGS AT ENTRY TO ABEND' can be used in problem isolation.

Find the PSW address in the dump. Then scan downward from that address to the DC statements where the labels and error descriptions are located.

Key	Label	Description
Reg1=X'034D' Reg6=X'E4084530' Rsn Code=030	AB845R30	An RPL was already active in VSAM on entry to module DFSDVSM0.
Reg1=X'034D' Reg6=X'E4084531' Rsn Code=031	AB845R31	A nonzero return code was received from VSI update during insert ESDS processing.
Reg1=X'034D' Reg6=X'E4084532' Rsn Code=032	AB845R32	A nonzero return code was received from VSI update call using VSAM SVC 109 during insert ESDS processing.

Key	Label	Description
Reg1=X'034D' Reg6=X'E4084533' Reg8=RPL Block Rsn Code=033	AB845R33	The RPL passed to the JRNAD exit routine is not an IMS RPL.
Reg1=X'034D' Reg6=X'E4084534' Rsn Code=034	AB845R34	A write error was detected in every buffer in the subpool.
Reg1=X'034D' Reg6=X'E4084536' Reg2=buffer address Reg4=RBA of CI Rsn Code=036	AB845R36	Either an invalid RDF, CIDE, or both, was found in the buffer in the VSAM buffer pool during shared-read processing.
Reg1=X'034D' Reg6=X'E4084537' Reg8=RPL Block Rsn Code=037	AB845R37	The RPL passed to the UPAD exit routine is not an IMS RPL.
Reg1=X'034D' Reg6=X'E4084538' Reg2=buffer address Reg3=CI size Reg4=RBA of CI Rsn Code=038	AB845R38	Either an invalid RDF, CIDE, or both, was found in the buffer that was to be written. Message DFS0442A is issued before this abend.
Reg1=X'034D' Reg6=X'E4084539' Reg2=buffer address Reg3=CI size Reg4=RBA of CI Rsn Code=034	AB845R39	Either an invalid RDF, CIDE, or both, was found in the buffer that was to be written. Message DFS0442A is issued before this abend.
Reg1=X'034D' Reg6=E408453A Reg2=buffer address Reg3=CI size Reg4=RBA of CI Rsn Code=03A	AB845R3A	Either an invalid RDF, CIDE, or both, was found in the buffer that was to be written. Message DFS0442A is issued before this abend.

**For DFSFXC50:**

At sync point time, after the buffer pools have been purged, a buffer was found that was not written back to the database.

0845 is a standard abend issued at label ABEND845. At the time of the abend register 2 contains the address of the buffer prefix, and register 10 contains the address of the buffer queue element (BQEL). Determining when the buffer was modified and what happened since the buffer was modified can help resolve the problem.

**Possible cause:** The buffer was not written or was written before the BQEL was created.

Key	Label	Description
Reg1=X'034D' Reg6=X'E4084540' Rsn Code=040	AB845R40	BQEL is not a VSAM BQEL.
Reg1=X'034D' Reg6=X'E4084545' Rsn Code=045	AB845R45	BUFC write error indicator not set.

## 0845

For DFSNOTB0:

0845 is a standard abend issued from several labels in module DFSNOTB0. The PSW points to the section of DFSNOTB0 that detected the error.

If the PSW points to the label NOTBGMER, register 10 points back to the address in DFSNOTB0 that branched to NOTBGMER.

Key	Label	Description
Reg1=X'034D' Reg6=X'E4084560' Rsn Code=060	AB845060	Buffer invalidate error. Invalidated buffer altered.
Reg1=X'034D' Reg6=X'E4084561' Rsn Code=061	AB845061	Buffer invalidate error. Invalidated buffer altered.
Reg1=X'034D' Reg6=X'E4084562' Rsn Code=062	AB845062	Buffer invalidate error. VSAM VSI address is not equal to IMS VSI address.
Reg1=X'034D' Reg6=X'E4084563' Rsn Code=063	AB845063	Getmain for save area storage failed.
Reg1=X'034D' Reg6=X'E4084564' Rsn Code=064	AB845064	VSAM KSDS data VSI does not match IMS KSDS data VSI.
Reg1=X'034D' Reg6=X'E4084565' Rsn Code=065	AB845065	VSAM ESDS data VSI does not match IMS ESDS data VSI.
Reg1=X'034D' Reg6=X'E4084566' Rsn Code=066	AB845066	VSI is not a key 0.
Reg1=X'034D' Reg6=X'E4084567' Rsn Code=067	AB845067	Invalid VSI block ID.
Reg1=X'034D' Reg6=X'E4084568' Rsn Code=068	AB845068	VSI key does not equal 0.
Reg1=X'034D' Reg6=X'E4084569' Rsn Code=069	AB845069	VSAM/IMS data set names do not match.
Reg1=X'034D' Reg6=X'E408456A' Rsn Code=06A	AB84506A	Buffer invalidate error. Invalidated altered buffer.
Reg1=X'034D' Reg6=X'E408456B' Rsn Code=06B	AB84506B	Getmain for save area failed.
Reg1=X'034D' Reg6=X'E408456C' Rsn Code=06C	AB84506C	Getmain for save area failed.
Reg1=X'034D' Reg6=X'E408456D' Rsn Code=06D	AB84506D	Getmain for save area failed.
Reg1=X'034D' Reg6=X'E408456E' Rsn Code=06E	AB84506E	Getmain for AWE failed.

Key	Label	Description
Reg1=X'034D' Reg6=X'E4084570 Rsn Code=070	AB845070	Getmain for AWE failed.

**System action:** IMS terminates abnormally.

**Problem determination:** 17a, 35

For a pseudoabend, register 14 is saved in the PST in the field named PSTDATA. The content of register 14 is the address in the buffer handler where the abnormal termination was issued.

**Source:** IMS abends

**Module:** DFSDVBH0, DFSTOCL0, DFSDVSM0, DFSFXC50, DFSNOTB0, DFSNOT0

**Related information:**

 DFS1058E (Messages and Codes)

Chapter 2, "IMS problem determination," on page 13

#### 0846

**Explanation:** Module DFSDBH00 was posted with an unexpected post code. This problem is probably an IMS system error.

**Analysis:** The content of register 14 is the address in DFSDBH00 where the abnormal condition was detected.

**Possible cause:** Module DFSDBH00 was waiting for some buffer handler function to complete. When posted, the post code was erroneous.

**System action:** IMS terminates abnormally.

**Problem determination:** 4, 35

**Source:** IMS abends

**Module:** DFSDBH00

**Related information:**

Chapter 2, "IMS problem determination," on page 13

#### 0847

**Explanation:** An unexpected condition occurred in the OSAM buffer handler. This problem is an IMS system error. Register 14 contains the address in DFSDBH00 where the abnormal condition was detected.

**Analysis:** A partition specification table (PST) should be waiting for a buffer, but the stack is empty. The program status word (PSW) at entry-to-abend points to the instruction in the routine at label ABENDWE from which the abend (SVC 13) is issued. This routine is branched to from various locations in DFSDBH00 when an error is detected. Register 14 is the BAL register to the abend routine, and will contain the address of the location from which control was passed.

Abend 0847 could also occur if an invalid local vector index was detected while testing for buffer validity using coupling facility services.

**System action:** IMS terminates abnormally.

**Problem determination:** 4

**Source:** IMS abends

**Module:** DFSDBH00

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

**0848**

**Explanation:** All the buffer prefixes in the OSAM buffer subpool were locked because of write errors.

**Analysis:** 0848 is a pseudoabend issued by DFSDBH20. Message DFS0762I is printed for each buffer marked in error.

**System action:** The application program terminates abnormally.

**Problem determination:** 27, 35

**Source:** IMS abends

**Module:** DFSDBH20

**Related information:**

 DFS0762I

Chapter 2, "IMS problem determination," on page 13

---

**0849**

**Explanation:** Retrieve attempted to issue a pseudoabend for a DBPCB that had PROCOPT=GO. The abend code was changed to abend 0849.

**Analysis:**

Module DFSDLR00 detected a failure and attempted to issue a pseudoabend for a DB PCB that had PROCOPT=GOx. The original abend was changed to user abend 0849. The register contents for the original abend code were saved in the last level save area in the PST. This save area starts at label PSTSAVL. The original abend code can be found in the two lower bytes of REG0, which is saved at PSTSAVL+X'14'.

**System action:** The application program terminates abnormally.

**Problem determination:** 17a, 17b

**Source:** IMS abends

**Module:** DFSDLR00

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

**0850**

**Explanation:** Retrieves request to the buffer handler for a key type call or a byte locate call failed. DFSDLR00 received an invalid return code from a call to the buffer handler.

**Analysis:** This is a pseudoabend issued from module DFSDLR00. The buffer handler return code is in either field PSTRTCDE or JCBRC. The requested function is in field PSTFNCTN.

The contents of registers 14 through 12 at the time of abend have been saved starting at offset X'C' in the last save area in the PST. This save area starts at label PSTSAVL. Normal register usage is as follows:

- R3=JCB
- R4=level table
- R5=SDB
- R7=PST
- R8=DSG

If the level table address is not in register 4, it is stored in field SAVELEVR in the JCB.

Key	Label	Description
Reg0=X'00000352' Reg14=BAL	IBYTERR	Register 15 is nonzero on return from the buffer handler. PSTFNCTN=X'E2' if entered from IBYTE routine. PSTFNCTN=X'F0', X'F2', X'F4', or X'F8' if entered from the BYKEY routine.

*Possible cause:* This abend might be a result of the DBD used for creating the database being different from the DBD used for updating the database.

**System action:** The application program terminates abnormally.

**Programmer response:** Analyze the DBDs you use. Ensure that the DBD you used to create the database and the DBD you used to update the database were the same.

**Problem determination:** 17a through 17e

**Source:** IMS abends

**Module:** DFSDLR00

**Related information:**

Chapter 2, "IMS problem determination," on page 13





---

## Chapter 19. IMS abend codes 0851 - 0900

This topic identifies and explains the following abend codes issued by IMS. Each code is 4 decimal digits.

---

### 0851

**Explanation:** The retrieve did not have a root segment returned as the first segment of an LRECL. The buffer handler had been passed a key to find the LRECL for retrieve.

For a HISAM or INDEX database, the first segment code in the logical record was not segment code 1. This abend can occur if the database is being processed using a DBD from a DBDLIB or ACBLIB that is different from the DBD under which the database was most recently loaded.

**Analysis:**

0851 is a pseudoabend issued from module DFSDLR00. The PSTFNCTN is equal to X'F0', X'F2', or X'F8'. The contents of registers 14 through 12 at the time of abend have been saved starting at offset X'C' in the last save area in the PST. This save area starts at label PSTSAVL. Normal register usage is as follows:

- Register 3=JCB
- Register 4=level table
- Register 5=SDB
- Register 7=PST
- Register 8=DSG

If the level table address is not in register 4, it is stored in SAVELEVR in JCB. Register 1 and JCBBUFSC point to the byte which was expected to be a segment code of X'01'.

Key	Label	Description
Reg0=X'00000353'	ABENDX	The address of the start of the LRECL is in PSTDATA. DMBPFOFF has a constant which is the offset from the start of the LRECL to the first segment code in the LRECL. If the first segment code in the LRECL is not a X'01', and if it is not a HISAM secondary data group, and not simple HISAM, then the abend is issued.

**System action:** The application program terminates abnormally.

**Problem determination:** 17a through 17e

**Source:** IMS abends

**Module:** DFSDLR00

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

### 0852

**Explanation:** A byte locate call to the buffer handler for an HDAM, HIDAM, HISAM, PHDAM, or PHIDAM database returned an invalid segment code to Retrieve.

In response to a PSTBYLCT call to the buffer handler, the segment code returned was not a valid segment code for the database. This abend can occur if the database is being processed using a DBD from a DBDLIB or ACBLIB that is different from the DBD under which the database was most recently loaded. .

**Analysis:**

This is a pseudoabend issued from module DFSDLR00. Register 1 points to the invalid segment code. The contents of registers 14 through 12 at the time of abend have been saved starting at offset X'C' in the last save area in the PST. This save area starts at label PSTSAVL. Normal register usage is as follows:

## 0853

- R3=JCB
- R4=LEVEL TABLE
- R5=SDB
- R7=PST
- R8=DSC

If the level table address is not in register 4, it is stored in field SAVELEVR in the JCB.

Key	Label	Description
Reg0=X'00000354' Reg6=last segment accessed Reg12=GETPSDB Reg15=DMB	GTPSDBER	This routine is branched to from GTPSDB10. The segment code used is from JCBACSC. It is the segment code from the last segment accessed. Because the segment code is not within the range of valid segment codes for the database, it is invalid.
Reg0=X'00000354' Reg5=SDB Reg8=DSG Reg9=PSDB SDBHH=0 Reg15=SAVELEVR	ABEND	This routine is branched to from SETR1. Register 5 is the base register for the SDB. Because SDBHH=0 and the segment code returned on the byte locate call do not equal SDBPHYCD, the segment type must have a twin type pointer. (LEVCOMMT=X'80'.)
Reg0=X'00000354' Reg6=address of start of PSDB Reg9=0	ABEND	This routine is branched to from SETR1. Register 5=SDB. If the SDB is not on the hierarchic chain (SDBHH=0), the segment code returned on the byte locate call must equal SDBPHYCD, and a zero segment code is invalid.
Reg0=X'00000354' Reg6=address of start of PSDB Reg9=PSDB offset	ABEND	This routine is branched to from SETR1. A test is made to see if the segment code returned is one of the valid segment codes within the DMB. If register 9 is not less than register 0, the abend is issued.

**System action:** The application program terminates abnormally.

**Programmer response:** Analyze the database. If there is an invalid pointer, recover the database.

**Problem determination:** 17a through 17e, 35

**Source:** IMS abends

**Module:** DFSDLR00

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

## 0853

- | **Explanation:** For a HIDAM, HDAM, PHDAM, or PHIDAM database, in attempting to locate a root segment, the  
| segment returned either did not have a segment code of 1, or (for HIDAM or PHIDAM) had a different key value  
| than the key value in the index pointer segment. This abend can occur if the database is being processed using a  
| DBD from a DBDLIB or ACBLIB that is different from the DBD under which the database was most recently loaded.  
| This abend may also be returned when doing a GHU on a database from which all segments have been deleted.

**Analysis:**

This is a pseudoabend issued from module DFSDLR00. The contents of registers 14 through 12 at the time of abend have been saved starting at offset X'C' in the last save area in the PST. This save area starts at label PSTSAVL.

Normal register usage is as follows:

- R3=JCB
- R4=level table
- R5=SDB
- R7=PST
- R8=DSG

If the level table address is not in register 4, it is stored in field SAVELEVR in the JCB. Register 14 is the return register from the POSTW routine if the segment code was not X'01'.

Key	Label	Description
Reg0=X'30000355' Reg1=A(segment), segment code != X'01' Reg14=return address of POSTW caller Reg15=RC8	ABEND33	Register 1 should point to the beginning of a segment with a X'01' segment code. Because it is not, this abend is called by branching to ABEND33 from POSTW.
Reg0=X'30000355' Reg1=A(segment) Reg4=Level table address Reg5=root (SDB) Reg8=root (DSG) Reg15=RC8	ABEND33	Register 1 points to the segment code of the segment returned. The key value of the segment returned did not match the key value of the index pointer segment. This abend is called by branching to ABEND33 from HIDBLCT.

To find the keys that resulted in the abend described in the second table entry, obtain the following pointers.

- The DSGLRKEY from the root DSG.
- The index SDB from SDBTARG in the root SDB.
- The DSG from the index SDB.
- The DSGLRKEY from the index DSG.

**System action:** The application program terminates abnormally.

**Problem determination:** 17a through 17e

**Source:** IMS abends

**Module:** DFSDLR00

**Related information:**

Chapter 2, "IMS problem determination," on page 13

## 0854

**Explanation:** For a HSAM database, a segment code was encountered which was not a valid segment code for the data management block. The retrieve service routine was looking for the next root in the current block and encountered an invalid segment code.

**Analysis:**

0854 is a pseudoabend issued from module DFSDLR00. The contents of registers 14 through 12 at the time of abend have been saved starting at offset X'C' in the last save area in the PST. This save area starts at label PSTSAVL.

Normal register usage is as follows:

- Register 8=DSG
- Register 9=PST
- Register 3=JCB
- Register 4=Level Table
- Register 5=SDB

If the level table address is not in register 4, it is stored in SAVELEVR in the JCB.

Key	Label	Description
Reg0=X'00000356' Reg14=block address PSTOFFST=offset in block	HSABEND	This abend is called by branching to the SETL subroutine from NXTSGGHS where the segment code is validity checked. The segment code in error is pointed to by register 1. The segment code in the block does not equal the segment code in the DMB.

## 0855

**System action:** The application program terminates abnormally.

**Problem determination:** 17a through 17e

**Source:** IMS abends

**Module:** DFSDLR00

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

## 0855

**Explanation:** The lock request handler detected an error while processing a lock request. PSTLRPRM contains the request parameters set by the DFSLR macro.

Pseudoabend DFSDLA00 is issued when a required user exit (DFSDBUX1) cannot be loaded or DATXEXIT=YES is specified in the DBDGEN but the user exit set SRCHFLAG to X'FF'.

**Analysis: For DFSLRH00, DBFLRH00:**

This is a pseudoabend issued by module DFSLRH00 or DBFLRH00 the lock request handler. The abend is issued from the common abend routine at label AB855COM. Normal register usage is as follows:

DFSLRH00-register 2=PST, register 6=AMPB, register 7=DSG, register 9=SCD. These registers, available only if DFSLRH00 calls DFSPIEX0, are saved in the save area containing the entry point of DFSPIEX0. If DFSDLR00 calls DFSLRH00, then, upon return, DFSDLR00 saves its own registers in the last level save area in the PST starting at offset 'C' beginning at label PSTSAVL.

DBFLRH00 issues this abend after label AB855. It saves the registers in SAV13 as follows:

### Register

#### Contents

**Reg0** Holds the return code from the lock manager (PI/IRLM), if one is available  
**Reg1** DBFLRH00 caller's return address  
**Reg6** ESCD address  
**Reg7** DMAC address  
**Reg8** LRHPARM address  
**Reg9** EPST address  
**Reg10** PST address  
**Reg11** SCD address  
**Reg14** Points to the subroutine that detected the error  
**Reg15** Points to an ABENDU0855 debug area eye-catcher

PSTLRPRM (4 bytes) contains the request parameters set by the DFSLR macro. PSTLRSUB has a subcode indicating the reason for the abend.

Key	Label	Description
PSTLRSUB=X'01' PSTLRPRM=X'30', X'60' X'80', or X'81' PSTLRPRM+2=X'02' DSGTOKEN=0 DSGFLGB=X'20' or X'02'	RTLPROC	While processing a root lock request, the LRH determined from DSGFLGB that an update occurred in the database record whose root lock token is in DSGTOKEN. However, the request is for a share state lock. Share state is invalid for updates.
PSTLRSUB=X'02' PSTLRPRM=X'60' or X'81' DSGTOKEN=0	RTLNOTOK	While processing a request to release a root lock, the LRH determined that the DSG did not contain the root lock token.
PSTLRSUB=X'03' SCDCMDTK=0 PSTLRPRM=X'26' and PSTLRPRM+2=X'03' or PSTLRPRM=X'56'	FBLCIDTK	While processing a request to release the command lock or to obtain an update state lock on the command lock held in read state, the LRH determined that the SCD did not contain the read state command lock token.

Key	Label	Description
PSTLRSUB=X'04' DMBPFTOK=0 PSTLRPRM=X'55'	FBLDIDB	While processing a request to release a data set reference lock, the LRH determined that the AMPB did not contain the lock token.
PSTLRSUB=X'05' PSTLRPRM=X'13', X'22', X'31', X'33', X'43', X'52', X'62', or X'63'	CRBN0040 CRBNOSHD FBIDRNA FBIDV060	While processing a request for a buffer lock, the LRH determined that the RBA/RRN passed in PSTBLKNM is invalid. The branch to AB85505 is taken if the RRN is zeros for a HISAM OSAM data set, or if the RBA is less than the block size for an ESDS data set, or an HD OSAM data set.
PSTLRSUB=X'07' (DBFLRH00)		Function code of request to Fast Path lock request was invalid. PSTDECB has the function code.
PSTLRSUB=X'08' (DBFLRH00)		STATE parm in Fast Path lock request was invalid. PST has STATE parm.
PSTLRSUB=X'10' (DBFLRH00)		LOCKID of the resource name for Fast Path lock request was invalid in nonblock level sharing. LOCKID is located at EPSTLKID.
PSTLRSUB=X'11' (DBFLRH00)		LOCKID of the resource name for Fast Path block request was invalid in block level sharing. LOCKID is located at EPSTLKID.
PSTLRSUB=X'12' (DBFLRH00)		Token address at the Fast Path unlock or change request was not specified. Token save area address pointed by EPSTTKNA is zero or EPSTTKNA is zero.
PSTLRSUB=X'13' (DBFLRH00)		Lock scope flag in DMAC was reset at the Fast Path change request. DMACLKSF flag has neither local lock flag nor global lock flag.
PSTLRSUB=X'DD'	RELBUF	A nonzero return code was set by the buffer handler when called by the LRH to release buffer ownership.
PSTLRSUB=X'FE' Register4=BAL	LMGLGTAB LMGLRTAB	PSTLRIPM contains an invalid function code for a global lock request. Register 4 has the return address of the calling subroutine.
PSTLRSUB=X'FB'		An unexpected return code was returned from VSAM while testing for buffer validity using coupling facility services.
PSTLRSUB=X'FD'		An invalid local vector index was detected while testing for buffer validity using coupling facility services.
PSTLRSUB=X'FF'	LR00020 BRTABLE	An invalid function code was passed to the LRH. PSTDECB has the function code.

**For DFSDLA00:**

Before restarting IMS and rerunning the transaction, ensure that the exit is bound into an APF authorized library. If DATXEXIT=YES was specified in the DBDGEN but the user exit set SRCHFLAG to X'FF', remove DATXEXIT=YES from the DBDGEN or change the user exit so that it does not set SRCHFLAG to X'FF'.

**System action:** The application program terminates abnormally.

**Problem determination:** 17a, 17b, 17e, 17f, 35

**Source:** IMS abends

**Module:** DFSLRH00, DBFLRH00, DFSDLA00

**Related information:**

Chapter 2, "IMS problem determination," on page 13

## 0857

**Explanation:** Verification of position for insertion of an HDAM, HIDAM, PHDAM, or PHIDAM segment failed to locate a valid chain.

The insert position was determined. Then, in attempting to verify the position, the segment to which the inserted segment would point could not be found starting from its parent. The database would appear to have invalid pointers. This abend can occur if the database is being processed using a DBD from a DBDLIB or ACBLIB that is different from the DBD under which the database was most recently loaded.

**Analysis:** 0857 is a pseudoabend issued from module DFSDLR00. For an HDAM, HIDAM, PHDAM, or PHIDAM insert, routine ISRTVER is called to verify that the segment in SDBPOSP points to the segment in SDBPOSN so that no hierarchic chains are broken.

The contents of registers 14 through 12 at the time of abend have been saved starting at offset X'C' in the last save area in the PST. This save area starts at label PSTSAVL. Normal register usage is as follows:

- Register 3=JCB
- Register 4=Level Table
- Register 5=SDB
- Register 7=PST
- Register 8=DSG

If the Level Table address is not in register 4, it is stored in SAVELEVR in the JCB.

Key	Label	Description
JCBR4 BIT3=1	IVRERR	A branch is taken to this routine from the IVRB routine. The chain has been followed to locate the SDBPOSN segment from its parent. The search is unsuccessful, so the abend is issued.
SDBPCF=0 SDBTFLG BIT3=1	IVRERR	This routine is branched to from IVRB which had determined that alternate chaining is being used. However, the physical child forward pointer to the segment cannot be found at SDBPCF, so the abend is issued.

**System action:** The application program terminates abnormally.

**Problem determination:** 17a, 17b, 17c, 17g

**Source:** IMS abends

**Module:** DFSDLR00

**Related information:**

Chapter 2, "IMS problem determination," on page 13

## 0858

**Explanation:** The pair of a physically paired segment could not be located.

**Analysis:**

This is a pseudoabend issued from module DFSDLR00. The contents of registers 14 through 12 at the time of abend have been saved starting at offset X'C' in the last save area in the PST. This save area starts at label PSTSAVL.

Normal register usage is as follows:

- Register 3=JCB
- Register 4=Level Table
- Register 5=SDB
- Register 7=PST
- Register 8=DSG

If the level table is not in register 4, it is stored in SAVELEVR in the JCB.

Key	Label	Description
Reg0=X'0000035A'	DPRERR	This routine is branched to from the RDPRGC routine because a search for a segment's physical pair, when passed the SDB, returned a X'04' return code in register 15 indicating that the pair could not be found.

**System action:** The application program is terminated abnormally.

**Problem determination:** 17a, 17b, 17c, 17g

**Source:** IMS abends

**Module:** DFSDLR00

**Related information:**

Chapter 2, "IMS problem determination," on page 13

### 0859

**Explanation:** IMS found a logical child on a GET call and the logical parent could not be found.

**Analysis:**

0859 is a pseudoabend issued by module DFSDLR00. The contents of registers 14 through 12 at the time of abend have been saved starting at offset X'C' in the last save area in the PST. This save area starts at label PSTSAVL.

Normal register usage is as follows:

- Register 3=JCB
- Register 4=Level Table
- Register 5=SDB
- Register 7=PST
- Register 8=DSG

If the level-table address is not in register 4, then it is stored in field SAVELEVR in the JCB.

**System action:** The application program terminates abnormally.

**Problem determination:** 17a through 17e

**Source:** IMS abends

**Module:** DFSDLR00

**Related information:**

Chapter 2, "IMS problem determination," on page 13

### 0860

**Explanation:** A return error from a call to the database handler occurred. This abend is issued from DFSDDLE0 for any of three reasons.

1. Bad insert position in SDBs from retrieve (DFSDLR00).
2. Database had bad pointers; error in DFSDDLE0.
3. There was insufficient space in the data set to initially load the data set.

**Analysis:** To diagnose an abend 0860, you will need the pseudoabend SNAP dump of control blocks. Use the SCD to get the entry points for DFSDDLE0 (SCDDLIO7), space management (SCDDHDS0), the buffer handler (SCDDBH0), and DFSDDL00 (SCDDLIDR). Search down the save area sets until you find a save area with register 12 containing the entry point of DLI/LOAD (DFSDDLE0). Register 1 in this save area will point to the PST; register 8 has the return point within DFSDDLE0; register 4 points to the SDB; register 6 points to the JCB. Check PSTRTCDE. If the PSTRTCDE is *not zero*, then there was a bad return code from space management or the buffer handler. If PSTRTCDE is *zero*, the error was detected in the variable-length HISAM Replace Routine.

When PSTRTCDE is not zero, then the save area with register 12 having entry point of DFSDDLE0 is the one to use. In this save area, register 15 will contain the entry point of space management or the buffer handler. Use register 14 to determine where in DFSDDLE0 the error condition was detected for bad return codes. For a bad return code problem, check the following problem related areas for use in the label definitions which follow this Analysis section.

## 0860

<b>Name</b>	<b>Description</b>
<b>PSTFNCTN</b>	requested function
<b>PSTBYTNM</b>	requested RRN or RBN
<b>PSTRTCDE</b>	return code
<b>PSTDSGA</b>	data set group

If the problem is in the buffer handler and you have the buffer handler trace table, use it to trace the calls through the buffer handler. If you don't have the buffer handler Trace Table, you can recreate the problem, then rerun the job with tracing turned on.

All calls to the buffer handler are (BAL or BALR, GOTOFUNC) or (BAL or BALR, GOTOBUFF). When they go to 'GOTOFUNC' or 'GOTOBUFF', the address of the buffer handler is loaded into register 15 followed by a branch to register 15 unconditionally.

When PSTRTCDE is equal to zero, the save area preceding the save area with register 15 having the entry point of DFSDDLE0 will have register 15 pointing to entry point of DFSDDL00. In this case, the abend was issued by the HISAM variable-length replace using a branch to DLETERR from one of the four labels shown. This is a special call to LOAD/INSERT from DELETE/REPLACE to shift segments for the replace of a variable-length segment in a HISAM database.

Key	Label	Description
PSTRTCDE=0 TM SDBORGN for X'0C'=0 (Reg4+X'09')	DLETSHFT	The routine which shifts the segment for HISAM database, did a test to verify that it is a HISAM database. Because it was not a HISAM database, the abend was issued.
PSTRTCDE=0 TM DMBFLAG for X'48'=0 (Reg5+X'20')	CHKPP	Previous test showed that the database should have a physical pair. A test was done to verify physical pair. It was not a physical pair, so the abend was issued.
PSTRTCDE=0 SDBPSDB=PSTDLIW7	FREPP	When a physical pair is found, if it is not the segment to be replaced, the abend is issued.
PSTRTCDE=0 TM DMBVLDLFG for X'0C'=0 X'0C'=0 (Reg5+X'18')	SDBREP	A check is made for variable or compact segment. This segment is neither, so the abend is issued.
Reg14=BAL PSTFNCTN=X'E2'	NDXREPL	This routine, doing index maintenance after a replace in the database, went to the buffer handler with function code of E2 (Requesting a segment). Register 7 will have a return address within DFSDDLE0 where the BAL to index maintenance for Replace was executed. A bad return code was set by the buffer handler, so the abend is issued.
Reg14=BAL PSTFNCTN=X'E2'	NDXREP1	This routine, doing index maintenance after a replace, went to the buffer handler with function code 02 (Get RBA of data). PSTDATA will have the address of data and register 7 will have the return address within DFSDDLE0 that BAL to this routine. A bad return code from the buffer handler resulted in this abend.
Reg14=BAL PSTFNCTN=X'E2'	LPSPLIT	While working with logical parent and with split data, a call went to the buffer handler to get data portion. PSTDLIW1 has the prefix RBN, PSTBYTNM has the data RBN. A bad return code from the buffer handler caused this abend.
Reg14=BAL PSTFNCTN=X'E2'	VRLPREP	This routine was working with a variable-length segment that was not split. PSTBYTNM has the prefix RBA. A bad return code from the buffer handler caused this abend.



Key	Label	Description
Reg14=BAL PSTFNCTN=X'E4'	ISIS0001	A call was made to the buffer handler to get a buffer for a new OSAM record. PSTBYTNM has the OSAM RRN. A bad return code from the buffer handler caused this abend.
Reg14=BAL PSTFNCTN=X'EB'	ISIS0010	A call was made to the buffer handler to get the prior OSAM record. PSTBYTNM has the RRN of the prior record. A bad return code from the buffer handler caused this abend.
Reg14=BAL PSTFNCTN=X'F4'	SHISAM01	When setting up to insert a new root, a new KSDS LRECL is created, then a retrieve by key is issued to find out if it is a duplicate record. A bad return code from the buffer handler caused this abend.
Reg14=BAL PSTFNCTN=X'E2'	NOTC2N	A call was made to the buffer handler to retrieve a buffer. PSTBYTNM has the RRN of record. A bad return code from the buffer handler caused this abend.
Reg14=BAL PSTFNCTN=X'EB'	ANCLB	A call was made to the buffer handler to retrieve the original record that data was shifted. PSTBUFFA is the buffer prefix address. PSTBYTNM is the RRN of original record. A bad return code from the buffer handler caused this abend.
Reg14=BAL PSTFNCTN=X'EB'	ANCLD	An attempt was made to move RRN of the new OSAM record to the original OSAM record. But since the intermediate space was used, it had to retrieve the original OSAM record again. PSTBYTNM has RRN of original OSAM record. Bad return code from the buffer handler caused this abend.
Reg14=BAL PSTFNCTN=X'EB'	RHX5	After shifting data and expanding the segment, the routine must get the old record. PSTBYTNM is RRN of the old record. A bad return code from the buffer handler caused this abend.
Reg14=BAL PSTFNCTN=X'EB'	SPACEOK	When an enqueue command for an anchor point was issued, a WAIT occurred, necessitating that the record again be obtained. PSTBYTNM=RBA. A bad return code from the buffer handler caused this abend.
Reg14=BAL PSTFNCTN=X'EB'	UPHDA	An attempt was made to UPDATE an anchor point for HD type organization. PSTBYTNM has RBA of anchor point. Bad return code from the buffer handler caused this abend.
Reg14=BAL PSTFNCTN=X'EB'	LOGRELNO	This is the call to the buffer handler to get the parent, so the prefix could be updated. PSTBYTNM is RBN of parent. An error code returned by the buffer handler caused this abend.
Reg14=BAL PSTFNCTN=X'EB'	UPPREFIX	An attempt was made to update the prefix of a segment. PSTBYTNM is RBN of the segment requesting update. An error code returned from the buffer handler caused this abend.
Reg14=BAL PSTFNCTN=X'EB'	HDREPL	This routine replaces a segment which was deleted in the physical path. PSTBYTNM has RBN of segment to be updated. The buffer handler returned with an error code, causing this abend.
Reg14=BAL	CALLFUCN	This is the call to the buffer handler subroutine, and is entered from many places within DFSDDLE0. The routine is entered using a BAL with REG 'A' being the return register. A bad return code from the buffer handler was received. Go to where REG 'A' points to find out why the call to the buffer handler was made.
Reg14=BAL PSTFNCTN=X'01'	TOSPACE	This subroutine is entered from many places in DFSDDLE0. The routine is entered using a BAL with register 8 being the return register. Space management gave an error return code, causing this abend.

Key	Label	Description
Reg14=BAL	DHDCALL	This subroutine is entered from many places in DFSDDLE0. The routine is entered using a BAL with register 8 being the return register. Space management gave an error return code, causing this abend.
Reg14=BAL PSTFNCTN=X'81'	REPVLS50	While trying to replace variable-length segments previously separated, an error returned from space management caused this abend.
Reg14=BAL PSTFNCTN=X'EB'	SAMEDSG	The buffer handler was called to read an OSAM record for a HISAM database. PSTBYTNM will have the RRN that was requested. A bad return code from the buffer handler caused this abend.
Reg14=BAL PSTFNCTN=X'E4'	NOTVSAM2	JCBSTOR4 has the code byte telling why the HISAM load write routine was entered. PSTBYTNM has RRN of the OSAM record. Register 7 has the return address to return to within DFSDDLE0. A bad return code from the buffer handler caused this abend.
Reg14=BAL PSTFNCTN=X'F4'	ISRTROR2	While doing a root insert in the HISAM routine, PSTBYTMN pointed to the key to be inserted. A bad return code from the buffer handler caused this abend.
Reg14=BAL PSTFNCTN='E2'	NOSWAPIE	PSTBYTNM has RRN of the OSAM record. A bad return code from the buffer handler caused this abend.
Reg14=BAL PSTFNCTN=X'E6'	LOGAFTNX	A call was made to the buffer handler to mark a buffer altered. A bad return code from the buffer handler caused this abend.
Reg14=BAL PSTFNCTN=X'EB'	GETLP	A call was made to the buffer handler to get the logical parent. PSTBYTNM has RRN of the logical parent. This routine is entered using a BALR with R8 being the return register. A bad return code from the buffer handler caused this abend.
Reg14=BAL PSTFNCTN=X'F4'	NORMLOAD	A call was made to the buffer handler to retrieve by KEY-RECORD. A bad return code from the buffer handler caused this abend.
Reg14=BAL PSTFNCTN=X'EB'	YESVSAM1	A call was made to the buffer handler to get the new record back. A bad return code from the buffer handler caused this abend.
Reg14=BAL PSTFNCTN=X'E6'	REINVLSB	A call was made to the buffer handler to mark buffer altered. A bad return code from the buffer handler caused this abend.
Reg14=BAL PSTFNCTN=X'E6'	CKSEGSZ	A call was made to the buffer handler to mark buffer altered. A bad return code from the buffer handler caused this abend.
Reg14=BAL PSTFNCTN=X'E4'	SEGTOOLD	A call was made to the buffer handler to get the next OSAM/ESDS LRECL. A bad return code from the buffer handler caused this abend.
Reg14=BAL PSTFNCTN=X'E4'	SEGTONNEW	A call was made to the buffer handler to get the next OSAM/ESDS LRECL. A bad return code from the buffer handler caused this abend.
Reg14=BAL PSTFNCTN=X'E4'	ANCLG	A call was made to the buffer handler to get next LRECL in OVERFLOW. A bad return code from the buffer handler caused this abend.
Reg14=BAL PSTFNCTN=X'EB'	INSADJUS	A call was made to the buffer handler to retrieve the LRECL of the segment. A bad return code from the buffer handler caused this abend.
Reg14=BAL PSTFNCTN=X'E6'	MOVEITIN	A call was made to the buffer handler to mark buffer altered. A bad return code from the buffer handler caused this abend.

Key	Label	Description
Reg14=BAL PSTFNCTN=X'E6'	SHFTRITE	A call was made to the buffer handler to mark buffer altered. A bad return code from the buffer handler caused this abend.
Reg14=BAL PSTFNCTN=X'E4'	NEWOLDTL	A call was made to the buffer handler to get the next OSAM/ESDS LRECL. A bad return code from the buffer handler caused this abend.
Reg14=BAL PSTFNCTN=X'E4'	RHX8	A call was made to the buffer handler for the next LRECL. A bad return code from the buffer handler caused this abend.
Reg14=BAL PSTFNCTN=X'E6'	REPRELD	For a UCF load restart, the Index entry needs to have its pointer corrected to point to the newly inserted high key (X'FFFF..FF') root. The address of the record is in PSTDATA. A bad return code from the buffer handler caused the abend.
Reg14=BAL PSTFNCTN=X'EB'	UPCTR5	For a HISAM database, physical paired segments have been inserted. The logical parent counters must be updated. PSTBYTNM contains the RRN of the logical parent. A bad return code from the buffer handler resulted in an abend.
Reg14=BAL PSTFNCTN=X'ED'	CHECKPT	The utility control facility (UCF) requested a check point. All the database buffers must be flushed. The buffer handler returned with an error indicating the cause of the abend.
Reg14=BAL PSTFNCTN=X'EB'	REHID	For a Reload call on a HIDAM database with twin backward pointers, the routine unchains higher roots and deletes index entries. PSTBYTNM contains the RRN of the segment. An error code returned from the buffer handler caused this abend.
Reg14=BAL PSTFNCTN=X'EB'	UNCAP	For a RELOAD call on a HIDAM database with twin forward pointers, the anchor points needs resetting. PSTBYTNM points to the block containing the root. A bad return code from the buffer handler caused this abend.
Reg14=BAL PSTFNCTN=X'F2'	DELROOT	This routine is entered to delete a KSDS record. PSTBYTNM contains the address of the Key. A bad return from the buffer handler resulted in abend.

**System action:** The application program terminates abnormally.

**Programmer response:** Probable causes are:

- Bad insert position in SDBs from retrieve (DFSDDL00).
- Database had bad pointers. Error in DFSDDL00.
- There was insufficient space in the data set to initially load the data set.

**Problem determination:** 4, 17a, 17b, 17g, 18, 19, 35

**Source:** IMS abends

**Module:** DFSDDL00

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

## 0861

**Explanation:** When doing a dependent segment insert in HISAM, or replacing a variable-length segment in HISAM whose length has changed, an invalid segment code was encountered by DFSDDL00 (LOAD/INSERT) in determining length of segments to the right of the insert point.

**Analysis:** 0861 is issued from DFSDDL00, at label ABEND861. A SNAP of the control blocks issued to the log is needed to diagnose the cause of 0861.

Search down through the save areas until you find the last save area. Register 12 equals the entry address of DFSDDL00 (SCDDL107). Use the registers in this register save area. Register 14 indicates the point in DFSDDL00 from which this subroutine was entered. Register 10 points to the insert point or the first segment to be shifted.

## 0862

Register 9 points to the invalid segment code. Register 1=PST, register 4=SDB, and register 6=JCB. PSTABTRM equals the pseudoabend code in hexadecimal.

Key	Label	Description
Reg14=BAL	NOX2	This BAL was to SCANREC to compute the length of shift data. Label ABEND861 was branched to a few instructions past label 'COMP SHFT'. It was determined that the address calculated, using the segment code of the segment pointed to by register 9, was beyond the end of the PSDB range.
Reg14=BAL	SHFTRITE	This BAL was to SCANREC to compute the length of shift data. Label ABEND861 was branched to a few instructions past label 'COMP SHFT'. It was determined that the address (calculated using the segment code of the segment pointed to by register 9) was beyond the end of the PSDB range.
Reg14=BAL	HISNLP	It was determined that the insert point offset into the record was zero bytes. A test is made to see if it is a VSAM DATA SET. If it is, it causes an error condition to have a Multi-Data Set Group.

**System action:** The application program terminates abnormally.

**Programmer response:** Probable causes are:

- DBD redone, changing length of segments without reloading database.
- Bad insert position in SDBs from retrieve (DFSDLR00) or replace (DFSDDL00)

**Problem determination:** 4, 17a, 17b, 17g, 18, 19, 35

**Source:** IMS abends

**Module:** DFSDDLE0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

## 0862

**Explanation:** On insert of logical child/logical parent concatenated segment with logical parent insert rule virtual, format of the user I/O area was correct, but, when attempting to replace logical parent, the key in the logical parent did not match the corresponding portion of the concatenated key of the logical parent in the logical child. Module DFSDLR00 positioned for the logical parent using the concatenated key of the logical parent in the logical child. This position was placed in the SDBPOSC field of the SDB for the logical parent. When subsequent replace of the logical parent was attempted, this position was not consistent with the concatenated key of the logical parent.

**Analysis:** 0862 is issued from DFSDDLE0, at label ABEND862. To diagnose this abend, you need the pseudoabend SNAP of control blocks that was put out to the log. Registers at the time of abend are stored in the last save area in the PST. Use the registers in the register save area. Register contents are as follows:

### Register

	Contents
1	PST
2	PCB
4	SDB of logical parent
6	JCB
9	Address of the key of the logical parent within the concatenated key of the logical child in the user I/O area
10	Address of the key of the logical parent within the logical parent segment retrieved from the data set

Key	Label	Description
Reg11=X'035E'	ABEND862	When register 9 (key of logical parent within concatenated key of logical child) was compared with register 10 (key of the logical parent within logical parent segment), there was an unequal condition.

**System action:** The application program terminates abnormally.

**Programmer response:** Check the problem-related data.

**Problem determination:** 4, 17a, 17b, 17g, 18, 19, 35

**Source:** IMS abends

**Module:** DFSDDLE0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

0863

**Explanation:** An error in the edit/compression routine resulted in one of the following:

- The segment length was greater than the defined maximum; or if the segment is fixed length, the length is greater than the defined length +10.
- For fixed-length segments, the length was greater than the defined length.
- The key was changed without key compression being specified.

**Analysis:** Abend 0863 is detected by DFSDDLE0, at label VLDERR. To diagnose abend 0863 the pseudoabend SNAP of control blocks put out to the system log is needed. Use the save area set with register 12 equal to the entry point address of DFSDDLE0. The register contents are listed below:

**Register**

	Contents
1	PST
2	Address of the segment in the user's I/O area
5	The address of the PSDB for this segment
10	Address of the compressed segment,
4	The SDB address

Key	Label	Description
Reg14=BAL DMBVLDFG=X'04' (Reg5+X'25')	NOASRT2	Just before label VLDVSFXL, there is a CH instruction. Register 7 has the data length of this segment, and is compared against DMBSGMX (register 5+X'1E'), the maximum length for variable-length segment. Since this is a variable-length segment, if the length is greater than the maximum length, the abend is issued.
Reg14=BAL DMBVLDFG=X'02' (Reg5+X'25')	CKEYCMP	A check is made to verify that this segment is not a level 01 segment. An 01 level is a root segment and the user cannot compress a HISAM root key. If the segment level (register 4+X'08') is equal to X'01', and SDBORGN has SDBORH1 on for HISAM, this abend is issued.
Reg14=BAL DMBVLDFG=X'04' (Reg5+X'25')	VLDVSFXL	This is a fixed-length segment and a check is made to verify that the segment data length (register 7) is not greater than the fixed-length (DMBDL, register 5+X'0A'). If the data length is greater than the fixed length, the abend is issued.
Reg14=BAL Data at Reg8 Data at Reg A		All the compressions were okay, but a check is made to assure that all data from the front of the segment through the key was not changed. Register 8 and register 10 point to the first bytes that do not compare, as the result of the CLCL instruction. If it has changed, the abend is issued.

**System action:** The application program terminates abnormally.

**Programmer response:** Ensure that the edit/compression routine is executing properly. Obtain the information listed under Problem Determination for analysis.

**Problem determination:** 4, 17a, 17b, 18, 19, and a source listing of the edit/compression routine

**Source:** IMS abends

**Module:** DFSDDLE0

## 0864 • 0867

### Related information:

Chapter 2, "IMS problem determination," on page 13

---

### 0864

**Explanation:** An unexpected return was received from the program isolation ENQ/DEQ routine, or from an invalid post to IWAIT. This is an IMS system error.

**Analysis:** 0864 is a pseudoabend which is not issued but the code that sets it is in DFSDDL0 at label ABEND864. The code remains in DFSDDL0 for possible diagnostic efforts.

**System action:** The application program terminates abnormally.

**Problem determination:** 17a, 17b, 35

**Source:** IMS abends

**Module:** DFSDDL0

### Related information:

Chapter 2, "IMS problem determination," on page 13

---

### 0865

**Explanation:** When inserting a replacing segment in a HISAM database, a work area was required to hold one LRECL. Space was not available using the ICREATE macro.

**Analysis:** 0865 is a pseudoabend. To diagnose an abend 0865, you need the pseudoabend SNAP of control blocks that was written to the log data set. You will need the entry point of DFSDDL0. (It can be found in the SCD at label SCDDLI07.)

Search down the save area set until you find the save area with register 15 equal to entry point of DFSDDL0. In the *next lower* save area, register 12 should contain the entry point of DFSDDL0 and register 14 should contain the address within the module from which the BAL to label GETSPC was made.

Using these registers, register 6 plus X'92' points to the length of the area requested.

In the next lower save area, you can find the registers that were current when the call was made to DFSICRET. These are the registers you should use to diagnose 0865.

Key	Label	Description
Reg14=BAL	GETPOOL	On return from DFSICRET (ICREATE) when ICREATE was unable to get space, 0865 is issued. Register 2 contains the 4-byte name of the POOL. Register 3 contains the POOL size requested in bytes.

**System action:** The application program terminates abnormally.

**Programmer response:** Increase the storage available to the program.

**Problem determination:** 17a, 17b

**Source:** IMS abends

**Module:** DFSDDL0

### Related information:

Chapter 2, "IMS problem determination," on page 13

---

### 0867

**Explanation:** When inserting a logical child, an attempt is made to locate and update a logical parent through the logical child's secondary list. No secondary list entry was found for the just inserted logical child, indicating an incomplete ACBGEN.

**Analysis:** 0867 is a pseudoabend issued by module DFSDDL0. At entry to this abend, register 11 points to the logical child's SDB and register 5 points to the DMBPSDB for which no valid secondary list entry was found. Registers at the time of abend are stored in the last save area in the PST.

**System action:** The application program terminates abnormally.

**Programmer response:** At entry to this abend, register 11 or register 5 points to the logical child's SDB for which no valid secondary list entry was found. Correct DBD in error or rerun ACBGEN, or both; and then re-execute the program.

**Problem determination:** 4, 9, 17d

**Source:** IMS abends

**Module:** DFSDDLE0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

## 0868

**Explanation:** When inserting a HIDAM or PHIDAM root with a twin backward pointer in update mode, the twin forward pointer was found to be zero. With a twin backward pointer specified at the root level, there should never be a twin forward pointer of zero. This indicates an invalid database or the loss of the all FFs key in the index record.

**Analysis:** 0868 is issued from DFSDDLE0 at label ABEND0868. To diagnose this abend occurring in an online region, you need the pseudoabend SNAP of the IMS control blocks that was put on the log. Use the SCD to get the entry point for DFSDDLE0 (SCDDLI07), and search the save areas until you find a save set with a register 15 equal to the entry address of DFSDDLE0. Then below this save set, find the save set with an 'AA' in the high-order byte of register 14 and the entry address of DFSDDLE0 in register 12. This is the save set at abend time. Register 4 in this save set will contain the address of the SDB for the segment we are trying to insert. The offsets in the SDB contain the information as follows:

### Codes

#### Explanation

#### X'34'

Starts the three position words that reflect the position for the segment just inserted.

#### X'34'

Is the RBA of the prior twin, if any.

#### X'38'

Is the RBA where the inserted segment is located.

#### X'3C'

Is the RBA of the next twin. Abend 0868 denotes that X'3C' was zeros.

**System action:** The application program terminates abnormally.

**Programmer response:** Analyze the database to see if the twin forward and twin backward chains are valid and if the all FFs key is missing in the record. In either case, the database must be recovered.

**Problem determination:** 17a through 17g, 35

**Source:** IMS abends

**Module:** DFSDDLE0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

## 0869

**Explanation:** The DL/I buffer handler router was asked to perform an undefined function. This problem is an IMS system error.

**Analysis:** 0869 is a standard abend issued by module DFSDVH0. The pertinent PST field is PSTFNCTN.

**System action:** IMS terminates abnormally.

**Operator response:** Submit all output to your IMS system programmer. Follow the IMS restart procedures.

**Problem determination:** 4, 11, 12, 35

## 0870 • 0880

**Source:** IMS abends

**Module:** DFSDVBH0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

### 0870

**Explanation:** Retrieve detected a bad twin chain when positioning for the insert of a HIDAM or PHIDAM root that has PTR=TB. The prior HIDAM or PHIDAM root that the DL/I Retrieve module obtained using the VSAM index has a key value higher than the key of the root being inserted.

**Analysis:** 0870 is a pseudoabend issued from module DFSDLR00. The contents of registers 14 through 12 at the time of abend have been saved starting at offset X'C' in the last save area in the PST. This save area starts at label PSTSAVL. A X'AA' in the first byte at X'C' indicates R14. Normal register usage is as follows:

- Register 3=JCB
- Register 5=SDB
- Register 7=PST
- Register 8=DSG

Key	Label	Description
R0=X'10000366' R6=pointer to the ABEND870 label	ABEND870	The RBA in JCBWC matches the RBA in PRIORRBA. The RBA in PRIORRBA is from the index entry returned by VSAM on a retrieve by a key equal to or greater than the call (X'F2'). When the prior root was read, it had a key greater than the key being inserted. The call to VSAM was redone. Abend 0870 is issued when VSAM returns the same next higher key a second time.

**Possible cause:** The causes include:

- Broken index (that is, missing index entries)
- HIDAM roots previously inserted out of sequence

**System action:** The application program terminates abnormally.

**Problem determination:** 17a through 17d, 35

**Source:** IMS abends

**Module:** DFSDLR00

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

### 0880

**Explanation:** An unexpected condition occurred during processing of a HALDB. The error might have been detected during processing for one of these events:

- Creation of an extended pointer set
- Creation of an indirect list entry
- Correction of an extended pointer set
- Selection of a HALDB partition
- Update of an indirect list entry
- Validation of an extended pointer set

**Analysis:** Abend 0880 occurs when an unexpected condition was detected during DL/I processing of a HALDB for one of the following functions:

- Partition selection
- Creation of an extended pointer set
- Creation of an indirect list entry
- Update of an indirect list entry
- Validation of an extended pointer set
- Correction of an extended pointer set



The contents of registers 14 through 12 at the time of the abend have been saved starting at offset X'C' in the last save area in the PST. This save area starts at label PSTSAVL. The error reason code can be found at offset X'1C' in this save area. The possible reason codes are listed below:

**Reason code**

**Explanation**

- X'C001'  
Non-partitioned DB access.
- X'C002'  
Partitions not active.
- X'C003'  
Partition stopped.
- X'C004'  
Partition structure terminating.
- X'C005'  
Partition structure not initialized.
- X'C006'  
Partition structure rebuild failed.
- X'C009'  
ILDS DSG not provided.
- X'C010'  
Unusual structure status.
- X'C030'  
Invalid authorization reason code.
- X'C031'  
An OPEN error occurred. Check for a DFS0730I message to determine the cause of the problem.
- X'C040'  
Invalid partition DDIR.
- X'C041'  
Invalid partition set operation.
- X'C042'  
Unable to allocate PSETE storage.
- X'C043'  
Invalid partition set selection action.
- X'C050'  
User selection exit not started.
- X'C060'  
Invalid function code.
- X'C070'  
Buffer handler byte locate request failed.
- X'C071'  
Buffer handler locate by key request failed.
- X'C073'  
Buffer handler mark altered request failed.
- X'C074'  
Buffer handler byte locate and mark altered request failed.
- X'C075'  
A request to obtain a root lock failed.
- X'C076'  
A DFS BCB request to get storage failed.

## 0881 • 0882

X'0077'

A HALDB OLR cursor refresh failed.

**System action:** The application program terminates abnormally.

**Problem determination:** 17a through 17d, 35

**Source:** IMS abends

**Module:** DFSDLR00, DFSDDLE0, DFSDXMT0, DFSDDLDC0

**Related information:**

 DFS0730I (Messages and Codes)

Chapter 2, "IMS problem determination," on page 13

---

### 0881

**Explanation:** An unexpected condition occurred in module DFSFXC50 during processing of a HALDB PINT (partition initialization) call or UNLD call.

**System action:** The application program terminates abnormally.

**Problem determination:** 17a through 17d, 35

**Source:** IMS abends

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

### 0882

**Explanation:** A failure occurred while running the Index/ILDS Rebuild utility (DFSPREC0). The reason code identifies the type of error and action to be taken:

#### Codes

##### Explanation

**04** The call to the Partition Selection and Initialization routine failed due to a severe error.

**08** The call to open the prime data set failed.

**0C** The call to open the index data set failed.

**10** The call from the buffer handler to locate the first block of the prime data set failed.

**14** The call to close the prime data set failed.

**18** The call to close the index data set failed.

**20** The call to insert an indirect list entry failed.

**24** An invalid status code was returned on a GN DL/I call.

**28** The call from the buffer handler to locate the next block in the database failed.

**2C** While scanning a database block, the end of the PSDBs was reached before the PSDB was found for the level of the segment being processed.

**System action:** The job terminates abnormally.

**Programmer response:** Save the dump and registers provided and call IBM Software Support for assistance.

**Problem determination:** 1, 3, 6, 8, 17a through 17d, 35, 40

**Source:** IMS abends

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

---

**0885****Explanation:**

A required user exit (DFSDBUX1) incurred an error for the following reasons:

1. The exit could not be loaded.
2. DATXEXIT=YES was specified in the DBDGEN but the user exit set SRCHFLAG to X'FF'.

**Analysis:** You must ensure that the exit is bound into an APF authorized library before restarting IMS and rerunning the transaction. If the second option applies, either remove the DATXEXIT=YES from the DBDGEN or change the user exit so that it does not set SRCHFLAG to x'FF'.

**System action:** IMS stops the database and the transaction is terminated with a pseudoabend.

**Programmer response:** Correct the problem and rerunning the transaction.

**Source:** IMS abends

**Module:** DFSDLA00

---

**0888**

**Explanation:** The active format library did not have any members and the IMS system has MFS supported terminals. As a minimum, the active format library should contain the default format blocks generated by the IMS system definition.

**Analysis:** This is a standard abend issued from the MFS buffer pool initialization module DFSIING0 (XA) or DFSIINF0 (non-XA). The abend occurs when a value of zero is detected for the count of format blocks. At a minimum, the active format library should contain the default format blocks generated by the IMS system definition.

**System action:** IMS terminates abnormally with an abend code of 0888.

**Programmer response:** Correct the format library specification on the FORMATA DD statement to a valid data set.

**Source:** IMS abends

**Module:** DFSIINF0, DFSIING0

---

**0889**

**Explanation:** A DEVTYPE macro was issued with an invalid area address.

**Analysis:** 0889 is a standard abend issued by the module to initialize the message format block pool, DFSIINF0. The program status word (PSW) at entry-to-abend points to the instruction within label ABEND9D from which the abend (SVC 13) is issued. This abend is the result of an unconditional branch to label ABENDA, which issues a WTO (SVC 35) to write out IMS error message DFS891A to the IMS master console prior to terminating abnormally.

Register 11 in the abend SVRB registers is the base register for this module. Register 15 contains the return code (in this case, an X'08') from the DD statement validity check routine, DFSIDDP0.

Key	Label	Description
Reg3=ABEND code X'00000379'	DDRCCHKS	A branch has been taken to the DD validity check routine (DFSIDDP0), which loads register 15 with a return code if an error is detected. In this case, a DEVTYPE parameter error occurred (area address specified was invalid), and an X'08' is loaded into register 15. The X'08' return code causes a branch to label ABEND.
Reg14=BALR		
Reg15=return code X'08'		

**System action:** IMS terminates abnormally.

**Operator response:** Notify your IMS system programmer.

**Problem determination:** 2, 3, 4, 35

**Source:** IMS abends

**Module:** DFSIINF0, DFSIING0

**Related information:**

 DFS891A

Chapter 2, “IMS problem determination,” on page 13

**0890**

**Explanation:** A DEVTYPE macro was issued for the ddname of the active format data set, FORMATA or FORMATB, and a not-found condition occurred.

**Analysis:** The active format library ddname is identified in the message text of DFS3410I.

0890 is a standard abend issued by the module to initialize the message format block pool, DFSIINF0. The program status word (PSW) at entry-to-abend points to the instruction within label ABEND9D from which the abend (SVC 13) is issued. This abend is the result of an unconditional branch to label ABENDA, which issues a WTC(SVC 35) to write out IMS error message DFS891A to the IMS master console prior to abending.

Register 11 in the abend SVRB registers is the base register for this module. Register 15 contains X'04', the return code from DFSIDDP0, the DD statement validation routine. DFSIDDP0 is the module that detected the error condition. X'04' indicates that the ddname was not found.

Key	Label	Description
Reg3=ABEND code, X'0000037A' Reg14=BALR Reg15=return code X'04'	DDRC0416	A branch has been taken to the DD validity check routine (DFSIDDP0), which loads register 15 with a return code if an error is detected. In this case, a DD statement was missing for the DEVTYPE macro, and an X'04' is loaded into register 15. The X'04' return code will cause a branch to label ABENDA to issue abend 0890.

**Possible cause:** A DD statement defining the active format data set with ddname FORMATA or FORMATB was omitted from the job stream.

**System action:** IMS terminates abnormally.

**Operator response:** Rerun the job, making sure that there is a valid DD statement for the active format data set with ddname FORMATA or ddname FORMATB. The active format library ddname is identified in the message text of DFS3410I.

**Problem determination:** 2, 4

**Source:** IMS abends

**Module:** DFSIINF0, DFSIING0

**Related information:**

Chapter 2, “IMS problem determination,” on page 13

 DFS891A

 DFS3410I

**0891**

**Explanation:** During IMS initialization, the active format data set could not be opened.

**Analysis:** 0891 is a standard abend that can be issued from one of two modules: DFSIINF0 or DFSTMEI0. The program status word (PSW) at entry-to-abend and the registers in the abend SVRB isolates the failure to a particular module.

The active format data set is defined to the system with a DDNAME statement of FORMATA or FORMATB. DCBOFLGS are in the DCB at X'30'.

**For DFSTMEI0:**

This can be issued by DFSTMEI0 because of a failure to open any of four format pool DCBs. Register 2 contains the format pool address. Register 12 is the entry point address and the base register for the module.

Key	Label	Description
Reg1='8000037B'	PREOPEN	During PREFETCH ECB initialization, in any system in which 3270s are defined, the format data set is opened. A failure to open any one of the four DCBs associated with the active format data set ddname FORMATA or FORMATB results in this abend.

**For DFSIINF0:**

This can be issued by module DFSIINF0 as a result of a failure to open any of those DCBs defined by DDNAME for the active format library (FORMATA or FORMATB).

Register 12 points to the DCB for which open failed, and message DFS891A is issued to the master console prior to abend. Register 11 contains the entry point address of the module.

Key	Label	Description
Reg1='8000037B' Reg12=DCB (failing) DCBOFLGS=X'10'	OPENM	This module initializes the format block pool. The OPEN macro instruction is issued to those DCBs defined by DDNAME FORMATA or FORMATB; should any DCBs fail to open, a branch is taken to OPENERR, then to label ABENDA where message DFS891A and this abend are issued.

**System action:** IMS terminates abnormally.

**Operator response:** Contact your IMS system programmer for assistance.

**Source:** IMS abends

**Module:** DFSIINF0, DFSTMEI0

**Related information:**

 DFS891A

**0892**

**Explanation:** Sufficient storage was not available in the message format block pool (MFBP) to complete the initialization of the pool.

**Analysis:** 0892 is a standard abend issued by DFSIINF0, the module that initializes the message format block pool (MFBP). The program status word (PSW) at entry-to-abend points to the instruction within label ABEND9D from which the abend (SVC 13) is issued. This abend is the result of an unconditional branch by the routine at label SPACERR to the routine at label ABEND0, which issues message DFS892 to the IMS master console before abending.

Register 11 in the abend SVRB registers is the base register for this module.

Key	Label	Description
Reg2=address of DFS892A message Reg3=ABEND completion code, X'8000037C'	FREINIT	Register 3 at this point contains the calculated size of the MFBP as computed from the FBP parameter of the EXEC statement for the IMS procedure. Register 15 contains the computed MFBP size (total FRE the size, plus fixed length of pool, plus size of 1 DECB, plus the size of 12 directory entries, plus 40 bytes of slack space). The two registers are compared; if register 3 at this point is lower, a branch is taken to label SPACERR to handle the abend. Note that at the time of abend, register 3 contents have been overlaid by the processing abend.
Reg3=ABEND completion code, X'8000037C' Reg2=address of DFS892A message	MOVELAST	After the directory entry length in storage is subtracted from the free space in the MFBP, register 3 (at this point, register 3 contains the amount of free space left in MFBP) and directory entry length are compared. If register 3 is lower, indicating no free space is left, a branch is taken to SPACERR to handle the abend. Note that at the time of the abend, register 3 contents have been overlaid by the processing abend.

## 0893

Key	Label	Description
Reg8=size of \$\$IMSDIR Reg9=amount of free space left in pool	HITIMS	A compare is made to determine if there is sufficient space for the in-storage index. If the value in Reg8 is higher than or equal to the value at register 9+X'04', a branch is taken to label SPACERR to handle the abend.
Reg2=address of DFS892A message Reg3=ABEND completion code, X'8000037C'	CLSE	During this routine, the contents of register 2 (at this point, register 2 contains the minimum size for dynamic space in MFBP) and register 3 (at this point, register 3 contains the amount of dynamic space in MFBP) are compared to determine if there is minimum dynamic space in the MFBP. If register 3 at this point is lower, there is not enough minimum space, and a branch is taken to label SPACERR to handle the abend. At the time of abend, the contents of register 2 and register 3 have been overlaid by the processing abend.

**Possible cause:** Check the FBP parameter on the EXEC statement in the IMS procedure. It might be necessary to define a larger pool (the pool size for FBP is defined in 1 KB blocks). Check the FBPR parameter on the EXEC statement in the IMS procedure.

**System action:** IMS terminates abnormally.

**Operator response:** Rerun the job after defining a larger format pool with the FBP parameter on the EXEC statement in the IMS procedure.

**Programmer response:** See Specifying buffers (Database Administration) for the factors to be considered when defining pool size.

**Problem determination:** 4

**Source:** IMS abends

**Module:** DFSIINF0, DFSIING0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

## 0893

**Explanation:** An I/O error occurred and issued a POINT or FIND macro for the active format library (IMS.FORMATA or IMS.FORMATB) during initialization of the message format block pool (MFBP).

**Analysis:** 0893 is a standard abend issued by DFSIINF0, the module that initializes the message format block pool (MFBP). The program status word (PSW) at entry-to-abend points to the instruction within label ABEND9D from which the abend (SVC 13) is issued. This abend is the result of an unconditional branch by the routine at label ABEND4 to the routine at label ABEND0, which issues message DFS983I to the IMS master console prior to abending.

Register 11 in the abend SVRB registers is the base register for this module. Register 3 contains the abend completion code, X'8000037D'.

Key	Label	Description
Reg8=BAL to READ routine R1ERRBIT=X'20'	Reg14=0 POINTD1	In the routine to read in all of the directory blocks, register 14 should contain the resident directory address. Register 14 is tested and found to be zero, indicating an I/O error occurred, and a branch is taken to label ABEND3. The field R1ERRBIT (register 1 on entry to SYNAD exit routine) is tested for a X'20'; if the bit is set, a branch is taken to label ABEND4 to handle the abend.

Key	Label	Description
Reg8=BAL for READ routine R1ERRBIT=X'20'	Reg14=0 HIT	While reading the \$\$IMSDIR directory block, register 14 should contain the address of the resident directory. Register 14 is tested and found to be zero, indicating an I/O error occurred during the reading of the \$\$IMSDIR directory block. A branch is taken to label ABEND3, and the field R1ERRBIT (register 1 on entry to SYNAD exit routine) is tested for a X'20'. If the bit is set, a branch is taken to label ABEND4 to handle the abend.
Reg14=end of directory block Reg8=address of next directory entry name R1ERRBIT=X'20'	IMSCOMP	Register 8 and register 14 are compared. If the address in register 8 is higher than or equal to the address in register 14, a branch is taken to label ABEND3. The field R1ERRBIT (register 1 on entry to SYNAD exit routine) is tested for a X'20'. If the bit is set, a branch is taken to label ABEND4 to handle the abend.
Reg8=BAL for READ routine R1ERRBIT=X'20'	Reg14=0 HITIMS	While reading the resident directory into storage, register 14 should contain the address of the resident directory. Register 14 is tested and found to be zero, and a branch is taken to label ABEND3. The field R1ERRBIT (register 1 on entry to SYNAD exit routine) is tested for a X'20'; if the bit is set, a branch is taken to label ABEND4 to handle the abend.

**Possible cause:** There is probably an error in the PDS directory for the active format data set. The ddname FORMATA or FORMATB in the message text of DFS3410I identifies the active format data set. Ensure that the MFS utility executed properly.

**System action:** IMS terminates abnormally.

**Programmer response:** There is probably an error in the PDS directory for the format data set. Ensure that the MFS Utility executed properly, and rerun the job.

**Problem determination:** 2, 3, 4, 11, 16

**Source:** IMS abends

**Module:** DFSIINF0, DFSIING0

**Related information:**

 DFS983I

 DFS3410I

Chapter 2, "IMS problem determination," on page 13

---

## 0894

**Explanation:** An I/O error occurred issuing a READ macro for the active format data set during initialization of the message format block pool, or an invalid directory block was read.

**Analysis:** 0894 is a standard abend issued by DFSIINF0, the module that initializes the message format block pool (MFBP). The program status word (PSW) at entry-to-abend points to the instruction within label ABEND9D from which the abend (SVC 13) is issued. This abend is the result of an unconditional branch by the routine at label ABEND3 to the routine at label ABEND0, which issues message DFS894I to the IMS master console prior to abending.

Register 11 in the abend SVRB registers is the base register for this module. Register 3 contains the abend completion code, X'8000037E'.

Key	Label	Description
Reg8=BAL to READ routine Reg14=0 R1ERRBIT= X'20'	POINTD1	In the routine to read in all of the directory blocks, register 14 should contain the resident directory address. Register 14 is tested and found to be zero, indicating an I/O error occurred during the READ, and a branch is taken to label ABEND3. The field R1ERRBIT (register 1 on entry to SYNAD exit routine) is tested for an X'20'; if the bit is not set, a branch is taken to label ABEND0 to abend.
Reg8=BAL for READ routine Reg14=0 R1ERRBIT= X'20'	HIT	While reading the \$IMSDIR directory block, register 14 should contain the address of the resident directory. Register 14 is tested and found to be zero, indicating an I/O error occurred reading the \$IMSDIR directory block. A branch is taken to label ABEND3, and the field R1ERRBIT (register 1 on entry to SYNAD exit routine) is tested for an X'20'; if the bit is not set, a branch is taken to label ABEND0 to abend.
Reg14=end of directory block Reg8=address of next directory entry name R1ERRBIT= X'20'	IMSCOMP	Register 8 and register 14 are compared. If the address in register 8 is higher than or equal to the address in register 14, a branch is taken to label ABEND3. The field R1ERRBIT is tested for an X'20'; if the bit is not set, a branch is taken to label ABEND0 to abend.
Reg8=BAL for READ routine Reg14=0 R1ERRBIT= X'20'	HITIMS	While reading the resident directory into core, register 14 should contain the address of the resident directory. Register 14 is tested and found to be zero, and a branch is taken to label ABEND3. The field R1ERRBIT is tested for an X'20'; if the bit is not set, a branch is taken to label ABEND0 to abend.

**Possible cause:** There is probably an error in the PDS directory for the active format data set. The active format data set must not be updated while IMS is active. Ensure that the MFS utility was executed properly.

**System action:** IMS terminates abnormally.

**Programmer response:** There is probably an error in the PDS directory for the format data set. The active format data set must not be updated while IMS is active. Ensure that the MFS utility executed properly, and rerun the job.

**Problem determination:** 2, 3, 4, 11, 16

**Source:** IMS abends

**Module:** DFSIINF0, DFSIING0

**Related information:**

 DFS894I

Chapter 2, "IMS problem determination," on page 13

## 0895

**Explanation:** Insufficient storage was available in the IMS extended private area to build the PDS directory indexes used to read MFS control blocks from the active MFS format library.

**Operator response:** Rerun the job when more space is available in the IMS extended private area.

**Source:** IMS abends

**Module:** DFSIING0

## 0896

**Explanation:** The DD DUMMY parameter is not supported for the active format data set. The ddname is FORMATA or FORMATB. Refer to message DFS3410I that identifies which of the ddnames is the active format data set.

**Analysis:** This is a standard abend issued by DFSIINF0, the module that initializes the message format block pool



(MFBP). The program status word (PSW) at entry-to-abend points to the instruction within label ABENDND from which the abend (SVC 13) is issued.

This abend is the result of an unconditional branch by the routine at label ABEND6 to label ABENDND. In its initial processing and set up, DFSIINF0 does a BALR to module DFSIDDP0, the DD statement validation routine. DFSIDDP0 has found an error on a DD statement, specifically a data set name error, and passes the error return code (X'10') back to DFSIINF0 in register 15. Based on the return code, a branch is taken in DFSIINF0 from the routine at DDRC0416 to the routine to handle the abend.

The abend 0896 is not accompanied by a dump. No provisions are made in the code at label ABENDND to set the dump option indicator (high-order bit of register 1). This abend is the direct result of coding DD DUMMY on the DD statement for the FORMAT data set. This data set must have a valid data set name (DUMMY not allowed).

If a dump is necessary, rerun the failing job after altering the branch instruction at label ABEND6 to result in an unconditional branch to label ABEND9D (47F0 Bxxx).

**System action:** IMS terminates abnormally.

**Operator response:** Rerun the job with the DD statement corrected to specify a valid data set name.

**Problem determination:** 2

**Source:** IMS abends

**Module:** DFSIINF0, DFSIING0

**Related information:**

 DFS3410I

Chapter 2, "IMS problem determination," on page 13

#### 0897

**Explanation:** Insufficient storage was available in the IMS private area to allocate the staging buffers used to read MFS control blocks from the active MFS format library.

**Operator response:** Rerun the job when more space is available in the IMS private area.

**Source:** IMS abends

**Module:** DFSIING0

#### 0898

**Explanation:** Insufficient storage was available in the IMS extended private area to build the MFS Dynamic Directory used to read MFS control blocks from the active MFS format library.

**Operator response:** Rerun the job when more space is available in the IMS extended private area.

**Source:** IMS abends

**Module:** DFSIING0

#### 0899

**Explanation:** Insufficient storage was available in the IMS private area to allocate the MFS buffer pool control area.

**Operator response:** Rerun the job when more space is available in the IMS private area.

**Source:** IMS abends

**Module:** DFSIING0

#### 0900

**Explanation:** Insufficient storage was available in the IMS private area to allocate storage for a dummy ECB required in order to read the MFS format library in 31-bit mode.

**System action:** Rerun the job when more space is available in the IMS private area.

**Source:** IMS abends

---

## Chapter 20. IMS abend codes 0901 - 0950

This topic identifies and explains the following abend codes issued by IMS. Each code is 4 decimal digits.

---

### 0904

**Explanation:** During the processing of PDS directory entries by an MFS utility program, an invalid entry with a format block size greater than 32K was found. The MFS format build modules limit the size of the format block to 32K. The directory entry has been corrupted.

**Analysis:** 0904 is a standard abend issued by:

- DFSIINF0, the format block pool initiator
- DFSFDIR0, the MFS XA directory build module
- DFSICV50, the /MODIFY COMMIT processor

The MFS Format Build modules limit the size of the format block to 32K. The PDS directory entry has an invalid size. Register 2 contains the address of the directory entry, and register 8 contains the size of the format block.

**System action:** IMS terminates abnormally.

**Programmer response:** At entry-to-abend, register 2 points to the directory entry. Register 8 contains the format block size. Delete the invalid entry; then re-execute the program.

**Source:** IMS abends

**Module:** DFSICV50, DFSIINF0, DFSFDIR0

---

### 0905

**Explanation:** The block builder was unable to obtain sufficient storage to build the required control blocks.

**Analysis:** 0905 is a standard abend issued from multiple CSECT module DFSDLBL0. Message DFS0905I accompanies this abend.

The registers in the abend SVRB and those saved at label ERRORSVE within DFSDLBL0 can be used in problem diagnosis. The register 14 BAL from the ERRORSVE save area should be used to isolate to a particular label.

A GETMAIN macro is issued to obtain the storage to build the control blocks. One of the following return codes is passed to the routine in register 15:

#### Codes

##### Explanation

**X'00'**

The virtual storage requested was allocated.

**X'04'**

No virtual storage was allocated.

The abend is issued from a common routine at label SETPSEU. The flow of control through the module is as follows: The subroutine BALs to GETMAINR which stores register 15 and register 0 at SAVE0 and issues the GETMAIN macro. (In the cases where DFSDLBA0 is the base, the subroutine BALs to label GETMAIN to resolve the addressability problems and then GETMAIN BALs to GETMAINR.) On return, ADCON ADDR (address of GETMAIN area) is checked for a X'04' and, if equal, branches to ERROR905 which saves register 0 through register 15 at label ERRORSVE and loads register 6 with the PSB name address. A branch is then taken to RETURN where message DFS0905I is issued and PSTSTAT is set to X'16'. RETURN branches to SETPSEU which issues the abend (SVC 13).

The only exception to this the instance is where the subroutine SUMSPDX BALs to DFSDLBA0 (entry point INTPROP) and the GETMAIN macro is issued at label BLDLIMT2. Register 15 is tested for a zero return, and, if not

zero, branches to INTRET4. INTRET4 sets the return code and branches to INTRET which does some housekeeping and returns to DFSDLBL0. At this point, a branch is taken to ERROR905 and control continues as previously stated.

Key	Label	Description
Reg2=CURLIM addr (Current Limit Table) Reg11=DFSDLBL0 base Reg12=DFSDLBA0 base Reg14=BAL Reg15=X'04'	SUMSPDX BLDLIMIT2	This routine builds an enqueue list based on intent by segment type. It BALs to the intent propagation routine (DFSDLBA0 entry point INTPROP) where a table of PSDB addresses is to be constructed starting with the passed PSDB. Subroutine BLDLIMT2 issues the GETMAIN macro. If storage is not available to construct the table, DFSDLBA0 passes a return code of X'04' in register 15 to DFSDLBL0 which then issues the abend.
Reg2=DDIR space required Reg3=PSB address Reg4=DBPCB list address Reg14=BAL Reg15=X'04'	ISDBPCB GETMAINR	This routine acquires space needed to build a list of temporary DDIRs for all DBDs referenced by the PSB. A nonzero return code from the GETMAIN request indicates that no storage was available to satisfy the request so the abend is issued.
Reg2=SDB space required Reg14=BAL Reg15=X'04'	SGTABGSD GETMAINR	The number of SDBs needed by the PSB is calculated by scanning the PCBs and counting the number of SENSEG statements. When the last PCB is processed, this routine attempts to acquire the storage needed to build the SDBs. A nonzero return code indicates that no space is available to satisfy the request, so the abend is issued.
Reg1=new PSB space required Reg2=SDB expansion size Reg14=BAL Reg15=X'04'	LOOPN GETMAINR	This routine acquires storage for an SDB expansion block. A nonzero return code from the GETMAIN request indicates that no space is available, so the abend is issued.
Reg2=DMB space required Reg14=BAL Reg15=X'04'	NOINDXTD GETMAINR	The size of storage needed to build the DMB has been calculated, and this routine attempts to obtain the storage. A nonzero return code from the GETMAIN request indicates that no space is available, and the abend is issued.
Reg2=SDB space required Reg3=PSB Reg6=SDB Reg13=DFSDLBL0 base Reg14=BAL Reg15=X'04'	GETSDBA GETMAINR	This subroutine was branched to from BLDSDBEB to obtain an SDB for a logical database. GETSDBA attempts to allocate space for building eight SDBs. A nonzero return code from the GETMAIN request indicates that no storage is available, so the abend is issued.
Reg2=PRLSTSZ SIZE Reg14=BAL Reg15=X'04'	BLDLOGLV GETMAINR	This routine is attempting to get storage to build a list describing the generated SDBs for physical pairing. A nonzero return code from the GETMAIN request results in the abend.
Reg0=SECLIST Table entry length Reg2=size of SECLIST Table Reg4=DDIR Reg14=BAL Reg15=X'04'	BLDDBNDA GETMAINR	This routine is attempting to get storage to process the SECLISTS. It does a GETMAIN for the SECLIST Table based on one entry for each SECTAB + 1. A nonzero return code from GETMAIN indicating no core available results in the abend.

Key	Label	Description
Reg2=size of ENQUEUE List Reg14=BAL Reg15=X'04'	ENQACUMA GETMAINR	The size of the enqueue list (intent list) for a physical DBD has been calculated and an attempt is made to obtain storage for it. A nonzero return code from a GETMAIN request indicates that no space is available, so the abend is issued.

**System action:** IMS terminates abnormally.

**Programmer response:** At entry to this abend, register 6 points to the PSB being processed. Increase the region or partition size, and re-execute the program. See message DFS0905I.

**Problem determination:** 4, 9, 17d, 35

**Source:** IMS abends

**Module:** DFSDLBL0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

## 0906

**Explanation:** A SENSEG statement had an invalid processing option specified.

**Analysis:** 0906 is a standard abend issued from multiple CSECT module DFSDLBL0. Message DFS0906I accompanies this abend.

The registers saved at label ERRORSVE within DFSDLBL0 should be used for problem isolation. Register 6 and register 7 in the abend SVRB point to the SENSEG name and PSB name, respectively.

The abend is issued from a common routine at label SETPSEU. When an error is detected, a branch is taken to ERROR906. Registers 0 through 15 are saved at label ERRORSVE. Register 7 is loaded with the PSB name address. Register 14 points to the instruction following the branch to ERROR906. A branch is then taken to RETURN, where message DFS0906I is issued and PSTSTAT is set to X'16'. RETURN branches to SETPSEU, which issues the abend (SVC 13).

Key	Label	Description
Reg6=SDB Reg7=option table address Reg15=A(PROCOPT)	BLDSDB LOOPA	This routine is building an SDB from the SENSEG statement. A comparison of the table entries containing processing options is made for the PROCOPT in register 15. When no matching option is found, it is determined that an invalid PROCOPT exists, and the abend is issued.

**System action:** IMS terminates abnormally.

**Programmer response:** At entry to this abend, register 6 points to the SENSEG name, and register 7 points to the PSB name. Correct the PSB, and re-execute the program. See message DFS0906I.

**Problem determination:** 4, 9, 17d, 35

**Source:** IMS abends

**Module:** DFSDLBL0

**Related information:**

 DFS0906I

Chapter 2, "IMS problem determination," on page 13

---

**0907**

**Explanation:** A PSB had a PCB that referenced a logical DBD and had a PROCOPT of L or LS.

**Analysis:** 0907 is a standard abend issued from multiple CSECT module DFSDLBL0. Message DFS0907I accompanies this abend.

The registers saved at label ERRORSVE within DFSDLBL0 should be used for problem isolation. Register 5 and register 6 in the abend SVRB point to the DBD name and the PSB name, respectively.

The abend is issued from a common routine at label SETPSEU. When an error is detected, the subroutine branches to label ERR907, which restores DFSDLBL0 as the base and branches to ERROR907. Here, register 0 through register 15 are saved at label ERRORSVE. Register 14 points to the instruction following the branch to ERROR907. A branch is taken to RETURN, where message DFS0907I is issued and PSTSTAT is set to X'16'. RETURN branches to SETPSEU, which issues the abend (SVC 13).

Key	Label	Description
Reg1=DDIR base DDIRADDR=X'22'	CLRDDIR9	The routine is working with a logical DDIR. The high-order byte of the DMB storage address (DDIRADDR) is tested for an X'22', which specifies a logical DBD and the load option. The load PROCOPT is invalid for this situation, so the abend is issued.

**System action:** IMS terminates abnormally.

**Programmer response:** At entry to this abend, register 6 points to the PSB name, and register 5 points to the PCB name. Correct the PSB, and re-execute the program.

**Problem determination:** 4, 9, 17d, 35

**Source:** IMS abends

**Module:** DFSDLBL0

**Related information:**

 DFS0907I

Chapter 2, "IMS problem determination," on page 13

---

**0908**

**Explanation:** The PSB is not a valid PSB.

**Note:** If no accompanying DFS0908I message is received, then this abend is not issued by IMS, but by a different product.

**System action:** IMS terminates abnormally.

**Programmer response:** At entry to this abend, register 6 points to the PSB name. Create the PSB, using IMS PSBGEN, and re-execute the program.

**Problem determination:** 4, 17d, 35

**Source:** IMS abends

**Related information:**

 DFS0908I (Messages and Codes)

Chapter 2, "IMS problem determination," on page 13

---

**0909**

**Explanation:** The DBD name is not a valid DBD. Message DFS0909I is issued prior to this abend.

**Analysis:** 0909 is a standard abend issued from multiple CSECT modules DFSDLBL0 and DFSDPSB0.

**For DFSDLBL0:**

The registers saved at label ERRORSVE within DFSDLBL0 should be used for problem isolation. Register 6 in the abend SVRB point to the DBD.

The abend is issued from a common routine at label SETPSEU. When an error is detected, the subroutine branches to label ERROR909 which saves register 0 through register 15 at label ERRORSVE and loads register 6 with the DBD name. Register 14 points to the instruction following the branch to ERROR909. A branch is then taken to RETURN, where message DFS0909I is issued and PSTSTAT is set to X'16'. RETURN branches to SETPSEU, which issues the abend (SVC 13).

Key	Label	Description
Reg2=DBD addr Reg4=DDIR addr AMODLEV=X'30'	LOADDBD	A DBD has been loaded (TYPESW=X'20'). Field AMODLEV is tested for a Version 1 DBD (Bits 1 and 2=X'30'). If the DBD is not valid, the abend is issued.

#### For DFSDPSB0:

This module is called by DFSDLBL0 if the region parameter is UDR or ULU. A parameter list address is passed in register 1.

Entry registers (register 14 through register 12) are saved at register 13 plus X'C'. The registers in the abend SVRB should be used for problem isolation.

The error condition for abend 0909 is detected within this module; however, the abend (SVC 13) is actually issued from module DFSDLBL0. DFSDPSB0 passes the abend code in register 15 to DFSDLBL0 which issues the abend at label SETPSEU.

Key	Label	Description
Reg10=DBD base AMODLEV=X'30'	NOLOAD (DFSDLBL0) GETDBD (DFSDPSB0)	This routine validity checks the DBD for the current level. If it is not a Version 1 DBD (bits 1 and 2=X'30' in field AMODLEV), the abend is issued.

**System action:** IMS terminates abnormally.

**Programmer response:** At entry to this abend, register 6 points to the DBD. Create the DBD, using IMS DBDGEN, and re-execute the program.

**Problem determination:** 4, 17d, 35

**Source:** IMS abends

**Module:** DFSDLBL0, DFSDPSB0

**Related information:**

 DFS0909I

Chapter 2, "IMS problem determination," on page 13

#### 0910

**Explanation:** An internal programming error has occurred while processing the PSB. Message DFS0910I is issued before this abend.

**Analysis:** 0910 is a standard abend issued from multiple CSECT module DFSDLBL0. Message DFS0910I accompanies this abend.

The registers saved at label ERRORSVE within DFSDLBL0 should be used for problem isolation. Register 6 in the abend SVRB points to the PSB name.

The abend is issued from a common routine at label SETPSEU. When an error is detected, the subroutine branches to label ERROR910, which saves register 0 through register 15 at label ERRORSVE and loads register 6 with the PSB name address. Register 14 points to the instruction following the branch to ERROR910. A branch is then taken to RETURN, where message DFS910I is issued and PSTSTAT is set to X'16'. RETURN branches to SETPSEU, which issued the abend (SVC 13).

## 0911

Key	Label	Description
Reg5=alternate index database name Reg8=PCB address	LOOPO1	This routine is trying to process a DDIR. It loops through the DDIRs (comparing the DMB name in Reg5 with the DDIR in Reg15) looking for the one needed for the alternate processing sequence for this PCB until it reaches the last. No match is found and the abend is issued.
Reg5=start of index tables Reg5 + X'8'=X'40' or X'20'	LOOPU	This routine is processing a secondary index database name that is referenced by the PSB but not by any PCB or SENSEG within it. It tests the index table entry bits for alternate processing sequence and the INDICES operand to see if all the entries have been processed. If not, the abend is issued because these bits should have been cleaned up by now.

**System action:** IMS terminates abnormally.

**Programmer response:** At entry to this abend, register 6 points to the PSB name which caused the error. Remove the offending PSB from the processing list, and re-execute the program.

**Problem determination:** 4, 17d, 35

**Source:** IMS abends

**Module:** DFSDLBL0

**Related information:**

 DFS0910I

Chapter 2, "IMS problem determination," on page 13

## 0911

**Explanation:** The processing option intent list length was calculated incorrectly for the named PSB. Message DFS0911I is issued prior to this abend.

**Analysis:** 0911 is a standard abend issued from multiple CSECT module DFSDLBL0. Message DFS0911I accompanies this abend.

DFSDLBL0 (SUMSPDH) BALs to the intent propagation routine (DFSDLBA0 Entry Point INTPROP) to build an enqueue list based on intent by segment type. Various subroutines within INTPROP BAL to routine SETNT, which locates the passed PSDB and sets the intent entry associated with it. A comparison is made to ensure that the intent list is large enough to contain the entry. If it is not, an error has occurred.

Routines INTPROP and SETNT save their caller's registers (register 14 through register 12 and register 14 through register 7, respectively) at register 13 plus X'C'. Beginning with register 13 in the abend SVRB and tracing through these save area sets, the flow of control which led up to the error condition can be traced. The register 14 BAL, saved at label ERRORSVE, should be used to isolate to a particular label.

The abend is issued from a common routine at label SETPSEU. When an error is detected at label SETNT, branches are taken to INTRET8 and INTRET, where housekeeping is done and a return code of X'08' is set in register 15. INTRET returns to CSECT DFSDLBL0 with the return code, and a branch is taken to ERROR911, which saves register 0 through register 15 at label ERRORSVE and loads register 6 with the PSB name address. Register 14 points to the instruction following the branch to ERROR911. A branch is then taken to RETURN, where message DFS911I is issued and PSTSTAT is set to X'16'. RETURN branches to SETPSEU, which issues the abend (SVC 13).

Key	Label	Description
Reg4=BAL Reg8=PSDB address Reg9=DDIR address Reg14=BAL	SETINTNT	There are six Reg4 BALs to this label which should be used to determine the type of intent entry to be set. This routine BALs on Reg14 to SETNT.
Reg3=current limit table entry address Reg14=BAL	DLETUP	Update intent will be set for the current PSDB and related PSDBs for 'D' PROCOPT. This routine BALs on Reg14 to SETNT.



Key	Label	Description
Reg3=current limit table entry address Reg14=BAL	UPISRTB	Update intent will be set for the current PSDB and related PSDBs for T PROCOPT. This routine BALs on Reg14 to SETNT.
Reg3=current limit table entry address Reg14=BAL	UPISRTE	Update intent will be set for the current PSDB and related PSDBs for T PROCOPT. This routine BALs on Reg14 to SETNT.
Reg8=current limit table entry address Reg14=BAL	UPISRPTR	Update intent for the current PSDB will be propagated to the logical parent. This routine BALs on Reg14 to SETNT.

**System action:** IMS terminates abnormally.

**Programmer response:** At entry to this abend, register 6 points to the PSB name. This is an internal program logic error. Remove the offending PSB, and re-execute the program.

**Problem determination:** 4, 17d, 35

**Source:** IMS abends

**Module:** DFSDLBL0

**Related information:**

 DFS0911I

Chapter 2, "IMS problem determination," on page 13

## 0912

**Explanation:** A program specification block (PSB) referenced a SEGM in the DBD. The SEGM did not exist in the DBD. The PSB, SECM, and DBD are identified in message DFS0912I, that is issued before this abend.

**Analysis:** 0912 is a standard abend issued from multiple CSECT module DFSDLBL0. Message DFS0912I accompanies this abend.

The registers saved at label ERRORSVE within DFSDLBL0 should be used for problem isolation. Register 5, register 6, and register 7 in the abend SVRB point to the PSB, segment name, and DBD name, respectively.

The abend is issued from a common routine at label SETPSEU. When an error is detected, a branch is taken to ERROR912. Registers 0 through 15 are saved at label ERRORSVE. Register 14 points to the instruction following the branch to ERROR912. A branch is then taken to RETURN, where message DFS912I is issued and PSTSTAT is set to X'16'. RETURN branches to SETPSEU, which issues the abend (SVC 13).

Key	Label	Description
Reg6=SDB address	BLDSDBDA	This routine is trying to find the matching SEGTab entry in the DBD for the referenced skeleton SDB. If no match is found in the DBD, the abend is issued.

**System action:** IMS terminates abnormally.

**Programmer response:** At entry to this abend, register 5 points to the PSB name, register 6 points to the segment name, and register 7 points to the DBD name. Correct the PSB or DBD, and re-execute the program.

**Problem determination:** 4, 17d, 35

**Source:** IMS abends

**Module:** DFSDLBL0

**Related information:**

 DFS0912I

Chapter 2, "IMS problem determination," on page 13

---

**0913**

**Explanation:** The named DBD contained an invalid or unknown access method. Message DFS0913I is issued before this abend.

**Analysis:** 0913 is a standard abend issued from multiple CSECT module DFSDLBL0.

The registers saved at label ERRORSVE within DFSDLBL0 should be used for problem isolation. Register 6 in the abend SVRB points to the DBD name being initially loaded, reloaded, or scanned.

The abend is issued from a common routine at label SETPSEU. When an error is detected, a branch is taken to ERROR913. Registers 0 through 15 are saved at label ERRORSVE. Register 14 points to the instruction following the branch to ERROR913. A branch is then taken to RETURN, where message DFS0913I is issued and PSTSTAT is set to X'16'. RETURN branches to SETPSEU, which issued the abend (SVC 13).

Key	Label	Description
Reg9=PREFIX base PREACCES~='X'0F'	NOPASS	While loading the DBD referenced by the DDIR, a comparison is made to see if the access method is defined as X'0F' or less. If higher, then the access method is invalid and the abend is issued.
Reg1=0	SIGMAIA	When checking for a valid access method the characters from field PREACCES are inserted in register 1. If register 1 equals zero, the abend is issued because no access method is specified.

**System action:** IMS terminates abnormally.

**Programmer response:** At entry to this abend, register 6 points to the DBD name being initially loaded, reloaded, or scanned. Correct the DBD, and re-execute the program.

**Problem determination:** 4, 17d, 35

**Source:** IMS abends

**Module:** DFSDLBL0

**Related information:**

 DFS0913I

Chapter 2, "IMS problem determination," on page 13

---

**0914**

**Explanation:** The PSB contained a SENSEG statement for which there was no corresponding SEGM in the DBD.

**System action:** The application program terminates abnormally and the program is stopped.

**System programmer response:** Replace the DBD with a new DBD copy using one of the following, and rerun the ACBGEN utility program.

- BUILD PSB=ALL
- BUILD DBD=dbdname
- BUILD PSB=psbname, DBD=dbdname

**Problem determination:** 11, 17a, 17b, 17d, 35

**Source:** IMS abends

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

**0915**

**Explanation:** The named INDEX DBD had a sequence field length specified that is not equal to the indexed field length. The correct length is indicated. Message DFS0915I is issued prior to this abend.

**System action:** IMS terminates abnormally.

**Programmer response:** At entry to this abend, register 6 points to the DBD name, and register 5 points to the required length. Correct the DBD in error, and re-execute the program.

**Problem determination:** 4, 17d, 35

**Source:** IMS abends

**Related information:**

 DFS0915I (Messages and Codes)

Chapter 2, "IMS problem determination," on page 13

## 0916

**Explanation:** The named DBD requires that a sequence field be specified for the root segment. A sequence field was not specified, or the field was specified as non-unique. Message DFS0916I is issued before this abend.

**Analysis:** 0916 is a standard abend issued from multiple CSECT module DFSDLBL0.

The registers saved at label ERRORSVE within DFSDLBL0 should be used for problem isolation. Register 6 in the abend SVRB points to the DBD name.

The abend is issued from a common routine at label SETPSEU. When an error is detected, the subroutine branches to label ERROR916, which restores DFSDLBL0 as the base and branches to ERROR916. Here, register 0 through register 15 are saved at label ERRORSVE. Register 14 points to the instruction following the branch to ERROR916. A branch is taken to RETURN, where message DFS0916I is issued and PSTSTAT is set to X'16'. RETURN branches to SETPSEU, which issues the abend (SVC 13).

Key	Label	Description
Reg14=0	ISHISAME	This subroutine is getting the FDB address for a HISAM or index database root segment. If no field has been defined for the root segment, then no sequence field exists and the abend is issued.
Reg14=FDB address	ISHISAME	The FDB has been located for a HISAM or index database root segment. The abend is issued because bits 1 and 2 in field FDBDCENF are either both on or both off, indicating a nonunique root key or no sequence field, respectively.
Reg14=0	ISHDORGC	This subroutine is getting the FDB address for an OSAM HDAM or HIDAM database root segment. If no field has been defined for the root segment, then no sequence field exists and the abend is issued.
Reg14=FDB address	ISHDORGC	The FDB has been located for the root segment. Bits 1 and 2 of field FDBDCENF indicate that no sequence field has been specified or that duplicate sequence fields are allowed. In either case, the abend is issued.
Reg14=0	ISVHIBA	This subroutine is getting the FDB address for a shared index (VSAM HISAM) database root segment. If no field has been defined for the root segment, then no sequence field exists and the abend is issued.
Reg14=FDB address FDBDCENF=X'40'	ISVHIBA	The FDB has been located. Field FDBDCENF indicates that no sequence field has been specified, so the abend is issued.
Reg14=0	ISVHIBC	This subroutine is getting the FDB address for a VSAM HISAM or index database root segment. If no field has been defined for the root segment, then no sequence field exists and the abend is issued.
Reg14=FDB address FDBDCENF=X'40' and X'20' DMBORG=X'0F'	ISVHIBCA	The FDG has been located for a VSAM HISAM or index database root segment. If field FDBDCENF indicates no sequence field was specified, the abend is issued. If field FDBDCENF indicates a sequence field has been specified and that duplicate sequence fields are allowed, while field DMBORG does not indicate VSAM index organization, the abend is issued because only VSAM indexes are allowed duplicate sequence fields.

## 0917

Key	Label	Description
Reg14=0	ISVHDC	This subroutine is getting the FDB address for a VSAM HIDAM database root segment. If no field has been defined for the root segment, then no sequence field exists and the abend is issued.
Reg14=FDB address	ISVHDC	The FDB has been located. Bits 1 and 2 of field FDBDCENF indicate that no sequence field has been specified or that duplicate sequence fields are allowed, so the abend is issued.

**System action:** IMS terminates abnormally.

**Programmer response:** At entry to this abend, register 6 points to the DBD name. Correct the DBD, and resubmit the program.

**Problem determination:** 4, 17d, 35

**Source:** IMS abends

**Module:** DFSDLBL0

**Related information:**

 DFS0916I

Chapter 2, "IMS problem determination," on page 13

---

## 0917

**Explanation:** The first DBD referenced a SEGM in the second DBD. The SEGM does not exist in the second DBD. The DBDs and the SEGM are identified in message DFS0917I, which is issued before this abend.

**Analysis:** 0917 is a standard abend issued from multiple CSECT module DFSDLBL0.

The registers saved at label ERRORSVE within DFSDLBL0 should be used for problem isolation. Register 5, register 6, and register 7 in the abend SVRB point to the referenced DBD, the referenced segment, and the referenced DBD, respectively.

The abend is issued from a common routine at label SETPSEU. When an error is detected, a branch is taken to ERROR917. Registers 0 through 15 are saved at label ERRORSVE. Register 14 points to the instruction following the branch to ERROR917. Branches to various subroutines are taken for housekeeping purposes before a branch to RETURN is made, where message DFS0917I is issued and PSTSTAT is set to X'16'. RETURN branches to SETPSEU, which issues the abend (SVC 13).

Key	Label	Description
Reg5=secondary list DMBSECDB=DDIR DMBSECNM=Segment name	EPSILONV	The DDIR minus 4 points to a table containing segment names. This subroutine searches through the table comparing a logical child or logical parent segment name (DMBSECNM) with the table entry plus zero. If no match is found, the abend is issued.

**System action:** IMS terminates abnormally.

**Programmer response:** At entry to this abend, register 5 points to the referencing DBD, register 6 points to the referenced segment, and register 7 points to the referenced DBD. Correct one or both of the DBDs, and re-execute the program.

**Problem determination:** 4, 17d, 35

**Source:** IMS abends

**Module:** DFSDLBL0

**Related information:**

 DFS0917I

Chapter 2, "IMS problem determination," on page 13

**0918**

**Explanation:** The named PSB contained a SENSEG statement for the named segment. The SENSEG statement referenced a logical child segment as the first part of a segment concatenation. The second part of the concatenation, the logical or physical parent of the logical child, did not have a valid physical parent. Message DFS0918I is issued before this abend.

**System action:** IMS terminates abnormally.

**Programmer response:** At entry to this abend, register 5 points to the PSB name, and register 6 points to the SENSEG name. Correct the PSB in error or the DBDs in error, or both; and then re-execute the program.

**Problem determination:** 4, 17d, 35

**Source:** IMS abends

**Related information:**

 DFS0918I

Chapter 2, "IMS problem determination," on page 13

**0919**

**Explanation:** A PCB in a program specification block (PSB) contains a KEYLEN parameter which is too small to hold the longest fully concatenated key. The correct length is indicated. The PSB and PCB are identified in message DFS0919, which is issued prior to this abend.

**Analysis:** 0919 is a standard abend issued from multiple CSECT module DFSDLBL0.

The registers saved at label ERRORSVE within DFSDLBL0 should be used for problem isolation. Register 5, register 6, and register 7 in the abend SVRB point to the PCB name, the PSB name and the required length, respectively.

The abend is issued from a common routine at label SETPSEU. When an error is detected, a branch is taken to ERROR919. Registers 0 through 15 are saved at label ERRORSVE. Register 14 points to the instruction following the branch to ERROR919. A branch is then taken to RETURN, where message DFS0919I is issued and PSTSTAT is set to X'16'. RETURN branches to SETPSEU, which issues the abend (SVC 13).

Key	Label	Description
TYPESW=X'08'	SUMSPNAB	The SDB key length and feedback position for all segments have been determined. Field TYPESW is tested for an X'08' indicating that the PCB key feedback length is too short. If so, the abend is issued.

**System action:** IMS terminates abnormally.

**Programmer response:** At entry to this abend, register 5 points to the PCB name, register 6 points to the PSB name, and register 7 points to the required length. Correct the PSB, and re-execute the program.

**Problem determination:** 4, 17d, 35

**Source:** IMS abends

**Module:** DFSDLBL0

**Related information:**

 DFS0919I

Chapter 2, "IMS problem determination," on page 13

**0920**

**Explanation:** The named PSB contained at least one reference to the named DBD with a PROCOPT of L and at least one additional reference to the same DBD with a PROCOPT of something other than L. The reference might be direct, such as in a PCB statement, or the reference might be indirect, such as in a DBD that references another DBD using logical or index relationships. Message DFS0920I is issued prior to this abend.

**Analysis:** 0920 is a standard abend issued from multiple CSECT module DFSDLBL0.

## 0921

The registers saved at label ERRORSVE within DFSDLBL0 should be used for problem isolation. Register 5 and register 6 in the abend SVRB point to the DBD name and PSB name, respectively.

The abend is issued from a common routine at label SETPSEU. When an error is detected, a branch is taken to ERROR920. Registers 0 through 15 are saved at label ERRORSVE. Register 14 points to the instruction following the branch to ERROR920. A branch is then taken to RETURN, where message DFS0920I is issued and PSTSTAT is set to X'16'. RETURN branches to SETPSEU, which issues the abend (SVC 13).

Key	Label	Description
DDIRADDR=X'20' and X'80'	ISHISAM	This routine builds DCBs for HISAM and INDEX type DBDs. The high order byte of DDIRADDR is tested for a X'A0', which indicates load and scan processing options. If conflicting PROCOPTs are specified, the abend is issued.

**System action:** IMS terminates abnormally.

**Programmer response:** At entry to this abend, register 5 points to the DBD name and register 6 points to the PSB name. Correct the PSB or one or more of the DBDs, or both, and then re-execute the program.

**Problem determination:** 4, 17d, 35

**Source:** IMS abends

**Module:** DFSDLBL0

**Related information:**

 DFS0920I

Chapter 2, "IMS problem determination," on page 13

---

## 0921

**Explanation:** A program specification block (PSB) was loaded. Upon examination it was discovered the PSB was not a valid PSB. Message DFS0921I is issued before this abend.

**Analysis:** 0921 is a standard abend issued from multiple CSECT module DFSDLBL0.

The registers saved at label ERRORSVE within DFSDLBL0 should be used for problem isolation. Register 6 in the abend SVRB points to the DBD name.

The abend is issued from a common routine at label SETPSEU. When an error is detected, the subroutine branches to label ERROR921, which saves register 0 through register 15 at label ERRORSVE. A branch is then taken to RETURN, where message DFS0921I is issued and PSTSTAT is set to X'16'. RETURN branches to SETPSEU, which issues the abend (SVC 13).

Key	Label	Description
Reg3=PSB Reg15=0	RETLOAD	This routine is trying to verify that the loaded PSB is a valid one. If the index I/O area address in the PSB is not zero, then this is in fact a DBD and the abend is issued.

**System action:** IMS terminates abnormally.

**Programmer response:** At entry to this abend, register 6 points to the PSB name. Check the JCL or the control statements, or both, correct the errors, and then re-execute the program.

**Problem determination:** 4, 17d, 35

**Source:** IMS abends

**Module:** DFSDLBL0

**Related information:**

 DFS0921I

Chapter 2, "IMS problem determination," on page 13

**0922**

**Explanation:** The named DBD was loaded. Upon examination it was discovered the DBD was not a valid DBD. Message DFS0922I is issued before this abend.

**Analysis:** 0922 is a standard abend issued from multiple CSECT module DFSDLBL0.

The registers saved at label ERRORSVE within DFSDLBL0 should be used for problem isolation. Register 6 in the abend SVRB points to the DBD name.

The abend is issued from a common routine at label SETPSEU. When an error is detected, a branch is taken to ERROR922. Registers 0 through 15 are saved at label ERRORSVE. Register 14 points to the instruction following the branch to ERROR922. A branch is then taken to RETURN, where message DFS0922I is issued and PSTSTAT is set to X'16'. RETURN branches to SETPSEU, which actually issues the abend (SVC 13).

Key	Label	Description
Reg2=DBD address Reg15=0	BLDSDBC	This routine attempted to load a DBD. The load was successful, but the item loaded was a PSB, not a DBD. Therefore, the abend was issued.
Reg2=PSB name address Reg4=PDIR address	RETLOOP	On a HD unload, HD reload, or database recovery region, no DBD name was supplied. Therefore, the abend was issued.

**System action:** IMS terminates abnormally.

**Programmer response:** At entry to this abend, register 6 points to the DBD name. Check the JCL or the control statements, or both, correct the errors, and then re-execute the program.

**Problem determination:** 4, 17d, 35

**Source:** IMS abends

**Module:** DFSDLBL0

**Related information:**

 DFS0922I

Chapter 2, "IMS problem determination," on page 13

**0923**

**Explanation:** The indicated logical child segment in the indicated DBD had a BYTES specification shorter than its logical parents fully concatenated key. The minimum acceptable length is indicated. Message DFS0923I is issued before this abend.

**Analysis:** 0923 is a standard abend issued from multiple CSECT module DFSDLBL0.

The registers saved at label ERRORSVE within DFSDLBL0 should be used for problem isolation. Register 6, register 7, and register 5 in the abend SVRB point to the segment name, the DBD name, and the minimum length, respectively.

The abend is issued from a common routine at label SETPSEU. When an error is detected, a branch is taken to ERROR923. Registers 0 through 15 are saved at label ERRORSVE. Register 14 points to the instruction following the branch to ERROR923. Branches to various subroutines are taken for housekeeping purposes before a branch to RETURN is made, where message DFS0923I is issued and PSTSTAT is set to X'16'. RETURN branches to SETPSEU, which actually issues the abend (SVC 13).

Key	Label	Description
Reg0=negative value code of referenced segment DMBSFD=logical parent key length	EPSILONW	While resolving logical relationships, this routine tests to see if the data length of the logical child segment is less than its logical parent's concatenated key. If so, it is an invalid segment length and the abend is issued.

**System action:** IMS terminates abnormally.

## 0924 • 0925

**Programmer response:** At entry to this abend, register 6 points to the segment name, register 7 points to the DBD name, and register 5 points to the minimum length. Correct the DBD, and re-execute the program.

**Problem determination:** 4, 17d, 35

**Source:** IMS abends

**Module:** DFSDLBL0

**Related information:**

 DFS0923I

Chapter 2, "IMS problem determination," on page 13

---

### 0924

**Explanation:** The root segment in an INDEX DBD had a data length that was too small to hold the required index data. The minimum acceptable length is indicated. Message DFS0924I is issued before this abend and provides the DBD name.

**Analysis:** 0924 is a standard abend issued from multiple CSECT module DFSDLBL0.

The registers saved at label ERRORSVE within DFSDLBL0 should be used for problem isolation. Register 6 and register 5 in the abend SVRB point to the DBD name and the minimum length, respectively.

The abend is issued from a common routine at label SETPSEU. When an error is detected, a branch is taken to ERROR924. Registers 0 through 15 are saved at label ERRORSVE. Register 14 points to the instruction following the branch to ERROR924. A branch is then taken to RETURN, where message DFS0924I is issued and PSTSTAT is set to X'16'. RETURN branches to SETPSEU, which actually issues the abend (SVC 13).

Key	Label	Description
Reg1=Index DDIR addr Reg7=PSDB DMBDL<DDIROPT	EPSILOMO	While working with an index DBD, this routine compares the data length of the target segment concatenated key and the index segment (which, since it is pointed to symbolically, must have the entire concatenated key of its target). If the index segment is not long enough, the abend is issued.

**System action:** IMS terminates abnormally.

**Programmer response:** At entry to this abend, register 6 points to the DBD name, and register 5 points to the minimum length. Correct the DBD in error, and re-execute the program.

**Problem determination:** 4, 17d, 35

**Source:** IMS abends

**Module:** DFSDLBL0

**Related information:**

 DFS0924I

Chapter 2, "IMS problem determination," on page 13

---

### 0925

**Explanation:** A logical child segment had a sequence field defined which fell within the logical parent's concatenated key area in the logical child and the key was specified as VIRTUAL. The key must be specified as PHYSICAL if it is to be used as part of the physical twin sequence field. Message DFS0925I is issued before this abend and provides the name of the segment.

**Analysis:** 0925 is a standard abend issued from multiple CSECT module DFSDLBL0.

The registers saved at label ERRORSVE within DFSDLBL0 should be used for problem isolation. Register 6 and register 7 in the abend SVRB point to the segment name and the DBD name, respectively.

The abend is issued from a common routine at label SETPSEU. When an error is detected, a branch is taken to ERROR925. Registers 0 through 15 are saved at label ERRORSVE. Register 14 points to the instruction following the



branch to ERROR925. Branches to various subroutines are taken for housekeeping purposes before a branch to RETURN is made, where message DFS0925II is issued and PSTSTAT is set to X'16'. RETURN branches to SETPSEU, which actually issues the abend (SVC 13).

Key	Label	Description
Reg0=sequence field start offset Reg0<Reg14 + X'8'	EPSILOAA	While resolving logical relationships, this routine encountered a logical child segment with a sequence field which starts within the virtual logical parent's key area. This is an error, so the abend is issued.

**System action:** IMS terminates abnormally.

**Programmer response:** At entry to this abend, register 6 points to the segment name, and register 7 points to the DBD name. Correct the DBD in error, and re-execute the program.

**Problem determination:** 4, 17d, 35

**Source:** IMS abends

**Module:** DFSDLBL0

**Related information:**

 DFS0925I

Chapter 2, "IMS problem determination," on page 13

## 0926

**Explanation:** The indicated PCB in the indicated PSB had an alternative processing sequence specified. The specified secondary index is not valid for the specified SENSEG. Message DFS0926I is issued before this abend.

**Analysis:** 0926 is a standard abend issued from multiple CSECT module DFSDLBL0.

The registers saved at label ERRORSVE within DFSDLBL0 should be used for problem isolation. Register 7 and register 8 in the abend SVRB point to the PCB name, and the PSB name, respectively.

The abend is issued from a common routine at label SETPSEU. When an error is detected, a branch is taken to ERROR926. Registers 0 through 15 are saved at label ERRORSVE. Register 14 points to the instruction following the branch to ERROR926. A branch is then taken to RETURN, where message DFS0926I is issued and PSTSTAT is set to X'16'. RETURN branches to SETPSEU, which actually issues the abend (SVC 13).

Key	Label	Description
Reg9=root segment's last SEC list DMBSCDE=X'80' Reg15=BAL	LOOPJ	This routine is BALed to from NOTPL12 where, while processing the PCB, it was determined that alternate processing sequence has been specified, so an SDB for the secondary index must be built. This routine obtains the secondary list (Reg9) to get the index entry. If the end of the list is reached without locating the entry, an error has occurred and the abend is issued.
Reg8=root segment's last FDB FDBXDFLG=X'80' Reg15=BAL	LOOPR	If the secondary list entry is located, the FDB are scanned (Reg8) to get the proper sequence field. If the end of the FDB is reached without locating the proper field, an error has occurred and the abend is issued.

**System action:** IMS terminates abnormally.

**Programmer response:** At entry to this abend, register 6 points to the SENSEG name, register 7 points to the PCB name, and register 8 points to the PSB name. Correct the PSB or the DBDs, or both, and then re-execute the job.

**Problem determination:** 4, 17d, 35

**Source:** IMS abends

**Module:** DFSDLBL0

**Related information:**

 DFS0926I

Chapter 2, “IMS problem determination,” on page 13

**0927**

**Explanation:** A INDEX DBD specified an indexed field in the INDEXED DBD which did not exist. See message DFS0927I, issued before this abend, for the name of the DBD.

**Analysis:** 0927 is a standard abend issued from multiple CSECT module DFSDLBL0.

The registers saved at label ERRORSVE within DFSDLBL0 should be used for problem isolation. Register 6 and register 5 in the abend SVRB point to the INDEX DBD name and the INDEXED DBD name, respectively.

The abend is issued from a common routine at label SETPSEU. When an error is detected, a branch is taken to ERROR927. Registers 0 through 15 are saved at label ERRORSVE. Register 14 points to the instruction following the branch to ERROR927. A branch is then taken to RETURN, where message DFS0927I is issued and PSTSTAT is set to X'16'. RETURN branches to SETPSEU, which actually issues the abend (SVC 13).

Key	Label	Description
Reg15=LASTPSDB	INDXLOC	The PSDBs for an indexed database are being scanned for the indexed field. Reg15 is compared with LASTPSDB to see if all the PSDBs have been processed. If so, then the index reference is invalid and the abend is issued.
Reg1=0 Reg15=target PSDB	EPSILOMM	The secondary list entries in the target PSDB are being scanned for a X'60'/X'64' type list entry. If the X'60'/X'64' type list entry is not chained to the index DDIR, the index reference is invalid and the abend is issued.

**System action:** IMS terminates abnormally.

**Programmer response:** At entry to this abend, register 6 points to the INDEX DBD name, and register 7 points to the INDEXED DBD name. Correct the DBD in error, and re-execute the program.

**Problem determination:** 4, 17d, 35

**Source:** IMS abends

**Module:** DFSDLBL0

**Related information:**

 DFS0927I

Chapter 2, “IMS problem determination,” on page 13

**0928**

**Explanation:** A INDEX DBD indexes a valid field, but the field is not a sequence field. The DBD is identified in message DFS0928I, which is issued before this abend.

**Analysis:** 0928 is a standard abend issued from multiple CSECT module DFSDLBL0.

The registers saved at label ERRORSVE within DFSDLBL0 should be used for problem isolation. Register 6 in the abend SVRB points to the INDEX DBD name. Register 1 in the abend SVRB points to the index DDIR.

The abend is issued from a common routine at label SETPSEU. When an error is detected, a branch is taken to Harrower. Registers 0 through 15 are saved at label ERRORSVE. Register 14 points to the instruction following the branch to Harrower. A branch is then taken to RETURN, where message DFS0928I is issued and PSTSTAT is set to X'16'. RETURN branches to SETPSEU, which actually issues the abend (SVC 13).

Key	Label	Description
Reg1=FDB address FDBDCENF=X'40' or X'10'	EPSICHKF	This routine has gotten the index secondary list entry and the FDB. It tests field FDBDCENF for a X'10' and a X'40' to see if a sequence field exists. If it does not, an error has occurred and the abend is issued.

**System action:** IMS terminates abnormally.

**Programmer response:** At entry to this abend, register 6 points to the INDEX DBD name. Correct the DBD in error, and re-execute the program.

**Problem determination:** 4, 17d, 35

**Source:** IMS abends

**Module:** DFSDLBL0

**Related information:**

 DFS0928I

Chapter 2, "IMS problem determination," on page 13

## 0929

**Explanation:** A BLDL macro call was issued to construct a directory entry list for the named member. The member was not found in the DBD or PSB library. Message DFS0929I is issued before this abend.

**Analysis:** 0929 is a standard abend issued from multiple CSECT module DFSDLBL0.

The registers saved at label ERRORSVE within DFSDLBL0 should be used for problem isolation. Register 6 in the abend SVRB points to the member name.

This abend is also issued if the IMS catalog is enabled in the DFSDFxxx member of the IMS.PROCLIB data set, or if the IMS Catalog Definition exit routine (DFS3CDX0) is included with function code 1. In either case, ensure that all of the catalog members (DFSCP000, DFSCD000, and DFSCX000) are included in the correct library.

The abend is issued from a common routine at label SETPSEU. When an error is detected, a branch is taken to ERROR929. Registers 0 through 15 are saved at label ERRORSVE. Register 14 points to the instruction following the branch to ERROR929. A branch is then taken to RETURN, where message DFS0929I is issued and PSTSTAT is set to X'16'. RETURN branches to SETPSEU, which actually issues the abend (SVC 13).

Key	Label	Description
Reg2=PSB name Reg15=0	MODLOAD	Because this is not a special region request, the passed PSB is to be loaded. This routine moves the PSB name into the BLDL list and issues the BLDL to locate the member. If the member PSB name is not in the library, the abend is issued.
Reg2=PSB name PDSAV<X'B' Reg15=0	MODLODA	While attempting to load the PSB, this routine determined that the binder has not created the PSB, so the abend is issued.
Reg4=DDIR Reg4 + X'8'=DBD name Reg15=0	MODLOAD	The DBD referenced by the DDIR is to be loaded. This routine moves the DBD name into the BLDL list and issues the BLDL to locate the member. If the member DBD name is not in the library, the abend is issued.
Reg4 + X'8'=DBD name PDSAV<X'B' Reg15=0	MODLODA	While attempting to load the DBD, this routine determined that the binder has not created the DBD, so the abend is issued.

**System action:** IMS terminates abnormally.

**Programmer response:** At entry to this abend, register 6 points to the member name. Correct the error in the appropriate library, and re-execute the program.

If the abend was caused by a missing IMS catalog member, rerun the appropriate PSB generation or DBD generation utility for DFSCP000, DFSCD000, and DFSCX000.

**Problem determination:** 4, 17d, 35

**Source:** IMS abends

**Module:** DFSDLBL0

**Related information:**

 DFS0929I (Messages and Codes)

Chapter 2, "IMS problem determination," on page 13

### 0930

**Explanation:** A DBD contained an LCHILD statement which referenced the named SEGM in a PAIR= operand. The named SEGM could not be found or the named SEGM was a virtual segment and the source segment contained a PTR=PAIRED operand. The DBD is identified in message DFS0930I, issued before this abend.

**Analysis:** 0930 is a standard abend issued from multiple CSECT module DFSDLBL0.

The registers saved at label ERRORSVE within DFSDLBL0 should be used for problem isolation. Register 6 and register 5 in the abend SVRB point to the DBD name and the segment name, respectively.

The abend is issued from a common routine at label SETPSEU. When an error is detected, a branch is taken to ERROR930. Registers 0 through 15 are saved at label ERRORSVE. Register 14 points to the instruction following the branch to ERROR930. A branch is then taken to RETURN, where message DFS0930I is issued and PSTSTAT is set to X'16'. RETURN branches to SETPSEU, which actually issues the abend (SVC 13).

Key	Label	Description
Reg2=DBD Reg5=SEC LIST address DMBSCDE=X'01' and X'10' Reg8=Offset of first LCHILD	EPSILOXE	A logical child segment contains a displacement to its paired segment. The secondary list has been built describing the logical parent segment to provide a path to the physical pair of a paired logical child. Register 15 did not contain the offset to the paired segment, so the LCHILD dsect is searched (Reg14 and Reg8 comparison) until it reaches the end. The abend is issued if the paired segment could not be found.
Reg6=SDB address SDBTFLG=X'10'	BLDSDBSR	This routine determined that the named segment has a source entry and that this is not a logical DBD. SDBTFLG=X'10' indicates that the SDB is for a physical pair. A physical pair cannot be a virtual segment, so the abend is issued.

**System action:** IMS terminates abnormally.

**Programmer response:** At entry to this abend, register 6 points to the DBD name, and register 5 points to the segment name. Correct the DBD, and re-execute the program.

**Problem determination:** 4, 17d, 35

**Source:** IMS abends

**Module:** DFSDLBL0

**Related information:**

 DFS0930I

Chapter 2, "IMS problem determination," on page 13

### 0931

**Explanation:** An INDEXED DBD had an index relationship with the named INDEX DBD. Either the INDEX DBD did not have a similar relationship to the INDEXED DBD, or another DBD referenced in the named PSB also had an index relationship with the INDEX DBD. See message DFS0931I, that is issued before this abend, for the name of the DBD.

**Analysis:** 0931 is a standard abend issued from multiple CSECT module DFSDLBL0.

The registers saved at label ERRORSVE within DFSDLBL0 should be used for problem isolation. Register 6, register 7, and register 5 in the abend SVRB point to the PSB name, the INDEXED DBD name, and the INDEX DBD name, respectively.

The abend is issued from a common routine at label SETPSEU. When an error is detected, a branch is taken to ERROR931. Registers 0 through 15 are saved at label ERRORSVE. Register 14 points to the instruction following the branch to ERROR931. A branch is then taken to RETURN, where message DFS0931I is issued and PSTSTAT is set to X'16'. RETURN branches to SETPSEU, which actually issues the abend (SVC 13).

Key	Label	Description
Reg1=FDB of indexed field Reg5=type X'44' secondary list Reg10=DBD Reg15=index prefix size DMBXPSDB=related index PSDB	EPSILOMN	While building a type X'44' secondary list, the indexed segment indicated that symbolic pointing was not checking the index pointer. The index prefix size in the DBD is less than or equal to 2. This indicated it was symbolic pointing, which did not match the information in the FDB. The abend is issued.
Reg5=type X'60' secondary list SAVE0=index PSDB	EPSILOME	The type X'60' secondary list points to a target PSDB. That target PSDB is different from the target PSDBs pointed to by the secondary list entries chained to the index PSDB. The index relationship is invalid and the abend is issued.
Reg1=index value Reg13+12=index table entry	INDXLOOP	The index table entry indicates what type of secondary list is to be processed. The entry is invalid and the abend is issued. Valid entry values are 0, 4, 8, and 12.

**System action:** IMS terminates abnormally.

**Programmer response:** At entry to this abend, register 6 points to the PSB name, register 7 points to the INDEX DBD name, and register 5 points to the INDEXED DBD name. Correct the appropriate PSB or DBDs, or both, and then re-execute the program.

**Problem determination:** 4, 17d, 35

**Source:** IMS abends

**Module:** DFSDLBL0

**Related information:**

 DFS0931I

Chapter 2, "IMS problem determination," on page 13

## 0932

**Explanation:** An INDEX DBD does not have a sequence field defined for the index segment. Message DFS0932I is issued before this abend and identifies the DBD.

**Analysis:** 0932 is a standard abend issued from multiple CSECT module DFSDLBL0.

The registers saved at label ERRORSVE within DFSDLBL0 should be used for problem isolation. Register 6 in the abend SVRB points to the DBD name in error.

The abend is issued from a common routine at label SETPSEU. When an error is detected, a branch is taken to ERROR932. Registers 0 through 15 are saved at label ERRORSVE. Register 14 points to the instruction following the branch to ERROR932. A branch is then taken to RETURN, where message DFS0932I is issued and PSTSTAT is set to X'16'. RETURN branches to SETPSEU, which actually issues the abend (SVC 13).

Key	Label	Description
FDBDCENF=X'10' FDBDCENF=X'80' Reg1=index PSDB Reg5=index secondary list	FDBNXT	This routine has located the index secondary list entry and the FDB. It tests field FDBDCENF for a X'10' to see if this is in the sequence field. If it is not, FDBDCENF is tested for a X'80' to see if this is the last FDB. If it is, no sequence field has been specified and the abend is issued.
Reg5=type X'60' secondary list entry Reg15=FDB of index key	SETNDXB	While processing a type X'60' secondary list entry, no key was found in the index field and the abend was issued.

**System action:** IMS terminates abnormally.

## 0933 • 0934

**Programmer response:** At entry to this abend, register 6 points to the DBD name. Correct the DBD, and re-execute the program.

**Problem determination:** 4, 17d, 35

**Source:** IMS abends

**Module:** DFSDLBL0

**Related information:**

 DFS0932I

Chapter 2, "IMS problem determination," on page 13

---

### 0933

**Explanation:** A PSB contained an invalid INDICES operand. The indicated value of the INDICES operand was not a valid index name for the associated SENSEG statement. Message DFS0933I, which is issued before this abend, identifies the PSB.

**Analysis:** 0933 is a standard abend issued from multiple CSECT module DFSDLBL0.

The registers saved at label ERRORSVE within DFSDLBL0 should be used for problem isolation. Register 6 and register 7 in the abend SVRB point to the PSB name and the DBD name of the index database, respectively.

The abend is issued from a common routine at label SETPSEU. When an error is detected, a branch is taken to ERROR933. Registers 0 through 15 are saved at label ERRORSVE. Register 14 points to the instruction following the branch to ERROR933. A branch is then taken to RETURN, where message DFS932I is issued and PSTSTAT is set to X'16'. RETURN branches to SETPSEU, which actually issues the abend (SVC 13).

Key	Label	Description
Reg6=SDB Reg7=indexed PSDB	RELOCJA	An index table entry was found for an index referenced with INDICES=. The PSDB address for the referencing SENSEG was obtained from the corresponding SEGTAB entry, but did not match the indexed PSDB.
Reg4=X'04' FDBDCENF=X'80' Reg6=FDB Reg6=SDB Reg8=FDB	RELOCG	The FDB from the indexed PSDB is loaded into register 8. The FDB's are scanned for the alternate sequence field (FDBXDFLG=X'10'). A test is made to see if it is the last FDB. FDBDCENF=X'80' (last FDB) indicates that an invalid INDICES operand exists on the PSB and the abend is issued.

**System action:** IMS terminates abnormally.

**Programmer response:** At entry to this abend, register 6 points to the PSB name, and register 7 points to the INDICES=value. Correct the PSB or DBDs in error, and re-execute the job.

**Problem determination:** 4, 17d, 35

**Source:** IMS abends

**Module:** DFSDLBL0

**Related information:**

 DFS0933I

Chapter 2, "IMS problem determination," on page 13

---

### 0934

**Explanation:** A PSB referenced the named logical child SEGM in the named DBD. The logical child requires the logical parent's concatenated key to be stored physically but VIRTUAL was specified, or the LP segment could not be located in the specified DBD. Message DFS0934I is issued before this abend and identifies the PSB.

**Analysis:** 0934 is a standard abend issued from multiple CSECT module DFSDLBL0.

The registers saved at label ERRORSVE within DFSDLBL0 should be used for problem isolation. Register 5 and register 7 in the abend SVRB point to the PSB name and the DBD name, respectively.

The abend is issued from a common routine at label SETPSEU. When an error is detected, a branch is taken to

ERROR934. Registers 0 through 15 are saved at label ERRORSVE. Register 14 points to the instruction following the branch to ERROR934. A branch is then taken to RETURN, where message DFS0934I is issued and PSTSTAT is set to X'16'. RETURN branches to SETPSEU, which actually issues the abend (SVC 13).

Key	Label	Description
Reg6=SDB SDBTFLG=X'20' SDBORGN=X'08' Reg15=logical parent SDB	SUMSPNA	This routine checks a segment's logical parent. The logical parent is in a HISAM database. The logical parent's concatenated key is not stored physically. Because the organization is HISAM, this is an error and the abend is issued.

**System action:** IMS terminates abnormally.

**Programmer response:** At entry to this abend, register 5 points to the PSB name, register 6 points to the segment name, and register 7 points to the DBD name. Correct the appropriate DBD, and re-execute the program.

**Problem determination:** 4, 17d, 35

**Source:** IMS abends

**Module:** DFSDLBL0

**Related information:**

 DFS0934I

Chapter 2, "IMS problem determination," on page 13

## 0935

**Explanation:** The named PSB referenced the named DBD through a SENSEG statement. However, a logical structure or relationship within this segment definition is invalid. Message DFS0935I is issued before this abend.

**Analysis:** 0935 is a standard abend issued from multiple CSECT module DFSDLBL0.

The registers saved at label ERRORSVE within DFSDLBL0 should be used for problem isolation. Register 7, register 6, and register 5 in the abend SVRB point to the PSB name, the segment name, and the DBD name, respectively.

The abend is issued from a common routine at label SETPSEU. When an error is detected, a branch is taken to ERROR935. Registers 0 through 15 are saved at label ERRORSVE. Register 14 points to the instruction following the branch to ERROR935. A branch is then taken to RETURN, where message DFS0935I is issued and PSTSTAT is set to X'16'. RETURN branches to SETPSEU, which actually issues the abend (SVC 13).

Key	Label	Description
Reg6=SDB SDBPOSN+1=X'01' SDBLEVEL=X'01'	CHKVIRRT	This routine determines SDB parentage for the current SDB and validates the logical structure. The current SDB is a logical child and is a root segment. The structure is invalid and the abend is issued.
Reg1=DMB SECLIST Reg6=SDB DMBSECDB=SDBDDIR	SETPRNTB	This routine checks secondary list entries for the logical parent. It could not find a X'02' type secondary list, so the abend is issued.
Reg6=SDB SDBLEVEL=X'01' SDBPHYCD=X'01' SDBF4=X'40'	PSRNTSTQ	SDB parentage for the SDB in register 6 is being determined. The logical level of the segment is a root, however it is not a physical root nor is there an alternate processing sequence. The abend is issued because the PCB does not begin with a root segment definition.
Reg1=parent SDB Reg6=SDB	PARNTSTC	This routine has located the parent SDB (Reg1) and its target SDB. It determines that the current SDB and the parent's target SDB are both in the same database and gets the target's PSDB (Reg14). A comparison of segment codes determines that the SDB and the parent's target SDB are describing the same segment. This is an invalid structure and the abend is issued.

Key	Label	Description
Reg1=parent SDB SDBTFLG=X'80' Reg6=SDB	PARNTSTI	This routine is entered because either the parent SDB does not have a logical parent, it has an inverted structure, or its target is not in the same database as the current SDB (Reg6). The parent SDB's PSDB is acquired. The current SDB (Reg6) is the physical parent of the parent SDB (REG1). There is no indication that the structure is inverted. It is an invalid structure and the abend is issued.
DMBSCDE=X'80'	SUMPSPCD	The current segment is a logical child. Its secondary lists are scanned for the logical parent. A test is made to see if this is the last SEC LIST entry. If it is, an invalid structure exists and the abend is issued.

**System action:** IMS terminates abnormally.

**Programmer response:** At entry to this abend, register 7 points to the PSB name, register 6 points to the segment name, and register 5 points to the DBD name. Correct the named DBD or the logical DBD referenced by the PSB, and re-execute the program.

**Problem determination:** 4, 17d, 35

**Source:** IMS abends

**Module:** DFSDLBL0

**Related information:**

 DFS0935I

Chapter 2, "IMS problem determination," on page 13

## 0936

**Explanation:** A SEGM was referenced in the named PSB with a PROCOPT of L or LS. The SEGM is a virtual segment and as such cannot be loaded. See message DFS0936I, issued before this abend, for the identification of the SEGM.

**Analysis:** 0936 is a standard abend issued from multiple CSECT module DFSDLBL0.

The registers saved at label ERRORSVE within DFSDLBL0 should be used for problem isolation. Register 5 and Register 6 in the abend SVRB point to the name of the invalid segment and PSB name, respectively.

The abend is issued from a common routine at label SETPSEU. When an error is detected, a branch is taken to ERROR936. Registers 0 through 15 are saved at label ERRORSVE. Register 14 points to the instruction following the branch to ERROR936. A branch is then taken to RETURN, where message DFS936I is issued and PSTSTAT is set to X'16'. RETURN branches to SETPSEU, which actually issues the abend (SVC 13).

Key	Label	Description
Reg6=SDB SDBTFLG=X'08' SDBF3=X'01' Reg10=SEGTAB DBDSSOFF=X'FF'	NOTER930	This routine has determined that the segment has a source entry and is in a physical DBD (thus being a virtual segment), and that the call sensitivity (SDBF3) is for a LOAD. A virtual segment can not be loaded, so the abend is issued.

**System action:** IMS terminates abnormally.

**Programmer response:** At entry to this abend, register 5 points to the name of the invalid segment, and register 6 points to the PSB name. Correct the PSB, and re-execute the program.

**Problem determination:** 4, 17d, 35

**Source:** IMS abends

**Module:** DFSDLBL0

**Related information:**

Chapter 2, "IMS problem determination," on page 13



**0937**

**Explanation:** The named DBD specified the named segment as a virtual segment. The SOURCE parameter in the SEGM statement contained more than one operand. The probable cause is a missing pair of parentheses on the SOURCE operand. Message DFS0937I is issued before this abend.

**Analysis:** 0937 is a standard abend issued from multiple CSECT module DFSDLBL0.

The registers saved at label ERRORSVE within DFSDLBL0 should be used for problem isolation. Register 5 in the abend SVRB points to the DBD name.

The abend is issued from a common routine at label SETPSEU. When an error is detected, a branch is taken to ERROR937. Registers 0 through 15 are saved at label ERRORSVE. Register 14 points to the instruction following the branch to ERROR937. A branch is then taken to RETURN, where message DFS0937I is issued and PSTSTAT is set to X'16'. RETURN branches to SETPSEU, which issues the abend (SVC 13).

Key	Label	Description
Reg0=0 DDIRADRC= X'02'	BLDSDBEB	This routine has determined that the virtual segment has source in a physical database with more than one entry (register 0=source segment count). A virtual segment can have only one source, so the abend is issued.

**Possible cause:** Missing pair of parentheses on the SOURCE operand

**System action:** IMS terminates abnormally.

**Programmer response:** At entry to this abend, register 6 points to the segment name, and register 5 points to the DBD name. Correct DBD source segments, and re-execute DBDGEN.

**Problem determination:** 4, 17d, 35

**Source:** IMS abends

**Module:** DFSDLBL0

**Related information:**

 DFS0937I

Chapter 2, "IMS problem determination," on page 13

**0938**

**Explanation:** A database PCB within the named PSB had no SENSEG statements defined at PSBGEN time. The PSBGEN was invalid. Message DFS0945I, issued before this abend, identifies the PCB.

**Analysis:** 0938 is a standard abend issued from multiple CSECT module DFSDLBL0.

The registers saved at label ERRORSVE within DFSDLBL0 should be used for problem isolation. Register 5 and register 6 in the abend SVRB point to the PCB name and the PSB name, respectively.

The abend is issued from a common routine at label SETPSEU. When an error is detected, a branch is taken to ERROR938. Registers 0 through 15 are saved at label ERRORSVE. Register 14 points to the instruction following the branch to ERROR938. A branch is then taken to RETURN, where message DFS945I is issued and PSTSTAT is set to X'16'. RETURN branches to SETPSEU, which issues the abend (SVC 13).

Key	Label	Description
Reg1=0 Reg8=PCB address	SGNOTPLI	PCBs are being scanned to count the number of SENSEGs in order to calculate the number of SDBs needed by this PSB. This routine encountered a PCB with no SENSEGs and the abend is issued.

**System action:** IMS terminates abnormally.

**Programmer response:** At entry to this abend, register 5 points to the PSB name, and register 6 points to the PCB name. Correct PSB source statements, and re-execute PSBGEN.

**Problem determination:** 4, 17d, 35

**Source:** IMS abends

**Module:** DFSDLBL0

**Related information:**

 DFS0945I

Chapter 2, "IMS problem determination," on page 13

### 0939

**Explanation:** An INDEX DBD contained an INDEX= operand which specified a field name with a /CK as the first 3 characters. Message DFS0946I, issued before this abend, identifies the DBD.

**Analysis:** 0939 is a standard abend issued from multiple CSECT module DFSDLBL0. Messages DFS0946I and DFS0939I accompany this abend.

The registers saved at label ERRORSVE within DFSDLBL0 should be used for problem isolation. Register 6 in the abend SVRB points to the DBD name in error.

The abend is issued from a common routine at label SETPSEU. When an error is detected, a branch is taken to ERROR939. Registers 0 through 15 are saved at label ERRORSVE. Register 14 points to the instruction following the branch to ERROR939. A branch is then taken to RETURN, where message DFS0946I is issued and PSTSTAT is set to X'16'. RETURN branches to SETPSEU, which issues the abend (SVC 13).

Key	Label	Description
Reg1=FDB address FDBSYSNM=C/' FDBDCENF=X'10' Reg14=Indexed SEC LIST address	EPSILOMK	The index secondary list entry and FDB have been acquired. The FDB is a special one (that is, system related or indexed). FDBSYSNM is tested for a C/' indicating a system-related field. If it is, the abend is issued because of an invalid index.

**System action:** IMS terminates abnormally.

**Programmer response:** At entry to this abend, register 6 points to the DBD name in error. Correct the DBD, and re-execute the job.

**Problem determination:** 4, 17d, 35

**Source:** IMS abends

**Module:** DFSDLBL0

**Related information:**

 DFS0939I

 DFS0946I

Chapter 2, "IMS problem determination," on page 13

### 0941

**Explanation:** A segment in a DBD was a physically paired logical child and of variable length. The opposite pair was fixed length. Physically paired segments must have the same length attributes.

**Analysis:** 0941 is a standard abend issued from multiple CSECT module DFSDLBL0. Message DFS0948I accompanies this abend and identifies the segment and DBD.

The registers in the abend SVRB and those saved at label ERRORSVE within DFSDLBL0 can be used in problem diagnosis. Register 6 and register 7 in the abend SVRB point to the segment name and the DBD name, respectively.

The abend is issued from a common routine at label SETPSEU. When an error is detected, a branch is taken to ERROR941. Registers 0 through 15 are saved at label ERRORSVE. The abend code is loaded in register 1. Register 14 points to the instruction following the branch to ERROR941. A branch is then taken to RETURN, where message DFS948I is issued and PSTSTAT is set to return code of X'16'. RETURN branches to SETPSEU, which issues the abend (SVC 13).

The only exception to the above is the first instance (label UPISRTK), where DFSDLBL0 (SUMSPDX) BALs to the intent propagation routine (DFSDLBA0 Entry Point INTPROP) to build an enqueue list based on intent by segment type. Update intent for the segment has been set and is to be propagated to its logical parent and its physical pair. After processing the logical parent's PSDB, a test is made to ensure that the physical pair has the same length

attribute. If it does not, an error has occurred and a branch is taken to INTRET12 and INTRET, where housekeeping is done and a return code of X'0C' is set in register 15. INTRET returns to DFSDLBL0 with the return code and there, a branch is taken to ERROR941 and control continues as previously stated.

Routine INTPROP saves its caller's registers, register 14 through register 12, at register 13 plus X'C'.

Key	Label	Description
Reg8=segment's PSDB address UPISRTK DMBVLDG=X'C' Reg15=physical pair's PSDB address DMBVLDG=X'C'	UPISRTK	A segment's logical parent PSDB has been processed. It has been determined that a physical pair exists for the segment. The physical pair's secondary list is located. A test is made to see if the segment has a compression routine or if the segment has variable length (DMBVLDG=X'C'). It does, so the physical pair's PSDB is obtained and a test is made to see if it has a compression routine or variable length. (DMBVLDG=X'C'). The physical pair does not have either of these attributes, so the abend is issued.
Reg3=PSDB address Reg15=physical pair's PSDB address DMBVLDG=X'04'	CLRDIR6	The cleaning up of DDIRs and checking for valid logical relationships are being done. This routine encountered a segment which is a variable-length logical child. However, it has a physical pair which does not have variable length, so the abend is issued.
Reg3=PSDB address Reg15=physical pair's PSDB address DMBVLDG=X'04'	CLRDIRA	This routine encountered a segment which is not a variable-length logical child. However, it has a physical pair which does not have a variable length, so the abend is issued.

**System action:** IMS terminates abnormally.

**Programmer response:** At entry to abend, register 6 points to the segment name, and register 7 points to the DBD name. Correct the DBD in error, and re-execute the job.

**Problem determination:** 4, 17d, 35

**Source:** IMS abends

**Module:** DFSDLBL0

**Related information:**

 DFS0948I

Chapter 2, "IMS problem determination," on page 13

## 0942

**Explanation:** An INDEX DBD contained an LCHILD statement that contained the same database name for the indexed database as the name for this index database.

**Analysis:** 0942 is a standard abend issued from multiple CSECT module DFSDLBL0. Message DFS0949II accompanies this abend and identifies the DBD..

The registers saved at label ERRORSVE within DFSDLBL0 should be used for problem isolation. Register 6 in the abend SVRB points to the DBD name in error.

The abend is issued from a common routine at label SETPSEU. When an error is detected, a branch is taken to ERROR942. Registers 0 through 15 are saved at label ERRORSVE. Register 14 points to the instruction following the branch to ERROR942. A branch is then taken to RETURN, where message DFS0949I is issued and PSTSTAT is set to return code of X'16'. RETURN branches to SETPSEU, which issues the abend (SVC 13).

Key	Label	Description
Reg1=DDIR for DBD reference DDIRADDR=X'04' Reg1+X'8'=Reg4+X'8' Reg4=index DBD DDIR address NTNDXB Reg15=database ref table address	NTNDXB	The loaded index DBD has been scanned for DBDs referenced and the new DDIR is to be chained in for processing. Prior to processing the referenced DBD's DDIR, the name of the loaded index DBD is compared with the name of the referenced DBD to ensure that the index DBD does not reference itself. If it does, the abend is issued.

## 0943

**System action:** IMS terminates abnormally.

**Programmer response:** At entry to this abend, register 6 points to the DBD name. Correct the DBD, and re-execute the job.

**Problem determination:** 4, 17d, 35

**Source:** IMS abends

**Module:** DFSDLBL0

**Related information:**

 DFS0949I

Chapter 2, "IMS problem determination," on page 13

---

## 0943

**Explanation:** An invalid logical relationship was found. One of the following conditions was found:

- A logical child references the named segment in the indicated database, and the named logical parent does not have an LCHILD statement.
- A logical parent references a logical child segment in the named database, and the logical child does not have a reference for the logical parent.
- A logical DBD does not contain a reference to the named segment in the SOURCE= operand that references its corresponding logical child or logical parent.

**Analysis:** 0943 is a standard abend issued from multiple CSECT module DFSDLBL0. Message DFS0950I accompanies this abend.

The registers saved at label ERRORSVE within DFSDLBL0 should be used for problem isolation. Register 7 and register 8 in the abend SVRB point to the DBD name and the referenced segment name, respectively.

The abend is issued from a common routine at label SETPSEU. When an error is detected, a branch is taken to ERROR943. Registers 0 through 15 are saved at label ERRORSVE. Register 14 points to the instruction following the branch to ERROR943. Branches to various subroutines are taken for housekeeping purposes before a branch to RETURN is made, where message DFS0950I is issued and PSTSTAT is set to return code of X'16'. RETURN branches to SETPSEU, which issues the abend (SVC 13).

Key	Label	Description
Reg1=DDIR addr of reg database OR DMBSCs=segment code of ref segment Reg8=SEC LIST entry address Reg15=DMB	CLRDDIR3	The cleaning up of DDIRs and checking for valid logical relationships are being done. The segment is either a logical child or a logical parent and its associated secondary list has been acquired. The DDIRs of both the segment and its associated SEC LIST entry are compared. If they are equal, then segment codes are compared. If they are not equal and the end of the list is reached, then an error has occurred. If the segment codes are not equal and the end of the SEC LIST is reached, an error has occurred. For either situation, the abend is issued.

**System action:** IMS terminates abnormally.

**Programmer response:** At entry to this abend, register 7 points to the DBD name, and register 8 points to the referenced segment name. Correct the DBD, and rerun the job.

**Problem determination:** 4, 17d, 35

**Source:** IMS abends

**Module:** DFSDLBL0

**Related information:**

 DFS0950I

Chapter 2, "IMS problem determination," on page 13

---

**0944**

**Explanation:** Message ID not found in the message text module (DFSUMGT0).

**Analysis:** 0944 is a standard abend issued from either DFSUMSG0 or DFSDMSG0. Given a message number, the module selects a message from a message list and sends it out, either to the SYSPRINT data set, for DFSUMSG0, or to the JCL data set using a WTO, for DFSDMSG0.

The registers in the abend SVRB should be used for problem isolation. The program status word (PSW) at entry-to-abend will point to label TBLEND within the module that issued the abend (SVC 13).

Register 1 in the save area pointed to by register 13 contains a pointer to a parameter list set up by the UERR macro. The parameter list format is:

**halfword**

Message ID in binary

**halfword**

H'0' indicating no inserts to message

H'-1' indicating inserts specified Addresses of insertions, if any.

Register 4 contains the message ID from the ID table entry pointed to by register 9; it contains X'FFFF' if register 9 points to the end of the table.

Key	Label	Description
Reg2=message ID from parameter list Reg3=second halfword from parameter list Reg4=X'FFFF'	TESTID	In searching the ID table for the ID in the parameter list, register 9 is pointed to successive entries in the table and register 4 is loaded with the ID from the table (from register 9). Register 4 is compared with X'FFFF' to test for the last table entry. If this is the end of the table (register 4=X'FFFF'), the abend is issued because the message ID passed in the UERR parameter list was not found in the table.

**System action:** IMS terminates abnormally.

**Programmer response:** Ensure invalid modifications were not made to the message text module (DFSUMGT0).

**Problem determination:** 1, 2, 3, 4, 8, 9, 12, 13, 35

**Source:** IMS abends

**Module:** DFSDMSG0, DFSUMSG0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

**0945**

**Explanation:** The message formatter expected inserts to be passed; however, none or an incorrect number of parameters was passed to the message formatter, module DFSUMSG0, through the UERR macro.

**Analysis:** 0945 is a standard abend issued from either DFSUMSG0 or DFSDMSG0. Given a message number, the module selects a message from a message list and sends it out, either to the SYSPRINT data set, for DFSUMSG0, or to the JCL data set using a WTO, for DFSDMSG0.

Use the registers in the abend SVRB for problem isolation. The program status word (PSW) at entry-to-abend will point to label TBLEND within the module from which the abend (SVC 13) is issued.

Register 1 in the save area pointed to by register 13 contains a pointer to a parameter list set up by the UERR macro. The parameter list format is:

**halfword**

Message ID in binary

**halfword**

H'0' indicating no inserts to message

H'-1' indicating inserts specified Addresses of insertions, if any

Key	Label	Description
Reg2=message ID from UERR parameter list Reg3=insert indicator from UERR parameter list (=F'O') Reg7=address of text-table entry in message buffer Reg7 + 0 = X'FF'	IDEQU	The requested message ID (in register 2) from the UERR parameter list has been found in the message ID table. The corresponding message-text-table entry has been moved to the message buffer and is pointed to by register 7. That register 7 + 0 is X'FF' indicates an insert-type message, and register 3 is tested to see if insert variables were specified in the UERR call. If not (register 3≠X'FFFF'), the abend is issued to indicate that at least one insert was expected but was not coded in the UERR macro instruction.
Reg1= message ID from ERR parameter list Reg3= format bytes for text insert in text-table entry (X'FF') Reg7=address of text-table entry in message buffer Reg1 + 0 = X'FFFF'	GETMSG	At least one insert variable has been moved into the message text pointed to by register 7. Register 3 points to the offset for the next insert variable and not the end of the format list, because register 3 + 0 is not equal to X'FF'. Register 1 + 0 is compared to X'FFFF' to see if another insert variable was specified in the UERR macro instruction. If not (register 1 + 0 = X'FFFF'), the abend is issued to indicate that too few insert variables were specified in the UERR call.

**System action:** IMS terminates abnormally.

**Programmer response:** Ensure that invalid program modifications were not made to the message text module (DFSUMGT0).

**Problem determination:** 1, 2, 3, 4, 9, 12, 13, 35

**Source:** IMS abends

**Module:** DFSDMSG0, DFSUMSG0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

## 0947

**Explanation:** An invalid secondary list code was found in a data management block. This problem is an IMS system error.

**Analysis:** 0947 is a standard abend issued from module DFSUAMB0. Messages DFS0570I and DFS0951I accompany this abend.

Use the registers in the abend SVRB for problem isolation. The abend is issued from the abend routine at label INVSEC. The program status word (PSW) at entry-to-abend will point to the label from which the abend (SVC 13) is issued.

Key	Label	Description
Reg5=DDIR entry being processed Reg6=DMB Reg7=PSDB Reg8=invalid code Reg10=PST DMBSCDE≠CODETAB	NXSECL	This section of code relocates DMB addresses to offsets from the start of the DMB. The secondary list is obtained. A TRANSLATE AND TEST instruction is done to the DMB code byte (DMBSCDE) to change the code to a branch table index. The abend is issued if an invalid code exists.

**System action:** IMS terminates abnormally.

**Programmer response:** At entry to this abend, register 8 points to the invalid code, register 5 points to the DDIR entry being processed, and register 7 points to the PSB being processed.

**Problem determination:** 1, 2, 3, 4, 9, 35

**Source:** IMS abends

**Module:** DFSUAMB0

**Related information:**

DFS0951I

DFS0570I

Chapter 2, "IMS problem determination," on page 13

### 0948

**Explanation:** The SYSPRINT data set had a permanent I/O error.

**Analysis:** 0948 is a standard abend issued from module DFSUACB0 and is invoked by IOS through the SYNAD exit routine IOERRORQ because an I/O error occurred while doing a PUT to the SYSPRINT data set.

The registers in the abend SVRB should be used for problem isolation. The register 14 BAL (branch and link) can be used to determine from which label the PUT was issued.

The program status word (PSW) at entry-to-abend will point to the abend routine (ABEND2) within IOERRORQ.

Key	Label	Description
Reg2=SYSPRINT DCB Reg14=BAL	PM0	The print message routine (PRTMSG) was BAlEd to by routine GETT. A PUT has been issued to the SYSPRINT data set to print a page header.
Reg2=SYSPRINT DCB Reg14=BAL	PM11	The print message routine (PRTMSG) was BAlEd to by routine GETT. A PUT has been issued to the SYSPRINT data set to print a requested line.

**Attention:** DFSUMSG0, the ACBGEN Error Message Formatter also calls the PRTMSG routine to print error messages.

**Possible cause:** Hardware error. Check LOGREC to determine the unit causing the error and the reason for the error.

**System action:** IMS terminates abnormally.

**Programmer response:** Check LOGREC to determine the cause and resubmit the job.

**Problem determination:** 1, 2, 3, 4, 9, 23, 35

**Source:** IMS abends

**Module:** DFSUACB0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

### 0949

**Explanation:** The SYSPRINT data set could not be opened.

**Analysis:** 0949 is a standard abend issued from module DFSUACB0.

Use the registers in the abend SVRB for problem isolation. The program status word (PSW) at entry-to-abend points to the common abend routine at label ABEND.

In the first instance where the error is detected, label CMPR3, the SYSPRINT DCB is inline coded.

Key	Label	Description
DCBOFLGS=X'10'	CMPR3	An inplace compression on ACBLIB is being done by linking to IEBCOPY. The SYSPRINT DCB is reopened. If the DCB is not opened, the abend is issued.
Reg4=DCB DCBOFLGS=X'10'	OPENERR	This routine writes an error message for each unopened DCB. When a DCB is not open, a comparison is made to see if it is for SYSPRINT. If it is, the error message cannot be written, so the abend is issued.

**Possible cause:** Invalid SYSPRINT DD statement in the JCL

**System action:** IMS terminates abnormally.

**Programmer response:** Ensure that the JCL contains a valid DD statement for SYSPRINT, and resubmit the job.

## 0950

**Problem determination:** 1, 2, 3, 4, 9, 23, 35

**Source:** IMS abends

**Module:** DFSUACB0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

## 0950

**Explanation:** The work data set generator (DFSDSEH0) was attempting to read a control data set. The data set was either not opened successfully or, if opened, contained either no data or erroneous data or all the blocks of data were not read.

**Analysis:** This abend is a standard abend issued from module DFSDSEH0. The program status word (PSW) at entry-to-abend and the register 14 BAL in the abend SVRB should be used to determine the label at which the error was detected.

The DCB for the control data set is in-line coded (CSDSCB). CSDSCB plus X'30' contains the DCB open error flags (DCBOFLGS).

Key	Label	Description
DCBOFLGS=X'10'	TEST #@LB18	This routine issues an OPEN for the CDS DCB in preparation to read in the control data set. DCBOFLGS is tested for an open error. If there is an error, the abend is issued.
Reg3=read area address Reg14=BAL DCBOFLGS=X'10'	TEST #@LB20	A new block of data has been read in and the old one freed. The CDS DCB has been closed and an OPEN has been issued against it for input. DCBOFLGS is tested for an open error. If there is an error, the abend is issued.
Reg2=blockcount (number of blocks to be read in) Reg3=next block to be read Reg14=BAL (GET routine to read in block)	CDSEOF (#@LB22) #@LB30	Register 2 is tested to see if all the blocks have been read. If register 2 is not zero, the abend is issued.
Reg2=0 Reg3=CDSHD (CDS read area pointer) Reg14=BAL	CDSEOF	Register 3 is loaded with the pointer to the first block in the CDS. If all the blocks were read (register 2=0), but register 3=X'00', then the CDS is empty, so the abend is issued.
Reg3=CDSHD (CDS read area pointer) Reg3=X'04C "control data set" Reg14=BAL	CDSEOF (#@LB22) #@LB34	The first block of data has been read. A check is made for the characters CONTROL DATA SET. If they are not present, the CDS is invalid and the abend is issued.

**System action:** The program terminates abnormally.

**Programmer response:** The control data set must be specified by a DD statement named DFSURCDS. The control data set must have been generated by the Database Preorganization utility (DFSURPR0).

**Problem determination:** 1, 2, 3, 4, 17d, 18 and 19, and a printout of the DFSURCDS data set

**Source:** IMS abends

**Module:** DFSDSEH0

**Related information:**

Chapter 2, "IMS problem determination," on page 13



---

## Chapter 21. IMS abend codes 0951 - 1000

This topic identifies and explains the following abend codes issued by IMS. Each code is 4 decimal digits.

---

### 0951

**Explanation:** The work data set generator utility (DFSDSEH0) was performing a dynamic limit-check, and the limit-check failed. Before the database reorganizes, a diagnostic message is issued during its execution that indicates segments for which limit-check failure occurred.

**Analysis:** 0951 is a standard abend issued from module DFSDSEH0. The program status word (PSW) at entry-to-abend and the registers in the abend SVRB can be used for problem diagnosis.

Register 11 contains the address of the DCB. DCB plus X'52' is the logical record length (LRECL). Register 15 contains the maximum sort record length (X'C8').

Key	Label	Description
Reg4=length of CK for record	GS #@LB185	The concatenated key for the record being checked (register 4) is compared to the maximum record length in register 15. If the value in register 4 exceeds the value in register 15, a limit-check failure has occurred and the abend is issued.

**System action:** The utility terminates abnormally.

**Programmer response:** The Database Preorganization utility (DFSURPR0) provided diagnostic messages during its execution that indicate the segments for which the limit-check failure occurred. Use these messages for the appropriate action to take.

**Source:** IMS abends

**Module:** DFSDSEH0

---

### 0952

**Explanation:** The work data set generator program (DFSDSEH0) tried to open a work data set, but was not successful. The work data set must be specified by a DD statement named DFSURWF1.

**Analysis:** 0952 is a standard abend issued from module DFSDSEH0. The program status word (PSW) at entry-to-abend and the registers in the abend SVRB should be used for problem diagnosis.

Register 11 contains the address of the DFSURWF1 work file DCB. Open error flags (DCBOFLGS) are at register 11 plus X'30'. Other registers at the time of abend are: Register 10=PST, register 9=PCB, register 7=SDB, register 6=PSDB.

Key	Label	Description
DCBOFLGS=X'10'	RNDX	The work file data set is not open, so an OPEN is issued. DCBOFLGS is tested for open errors. If the DCB for DFSURWF1 cannot be opened, an abend is issued.

**System action:** The program terminates abnormally.

**Programmer response:** This DD statement must be present during initial database load, database reload, and database scan whenever the affected database contains segments involved in logical relationships.

**Problem determination:** 4, 8

**Source:** IMS abends

**Module:** DFSDSEH0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

**0953**

**Explanation:** The work data set generator (DFSDSEH0) was attempting to locate DL/I control blocks for segments involved in a logical relationship. The DL/I control blocks could not be found.

**Analysis:** 0953 is a standard abend issued from module DFSDSEH0. The program status word (PSW) at entry-to-abend and the registers in the abend SVRB should be used for problem diagnosis.

Register 7 contains the address of the SDB. Register 3 contains the address of DMBSEC, the DMB secondary list.

Key	Label	Description
Reg3=DMBSEC DMBSND=X'01'	#@LB80	DFSDSEH0 searches the DMB secondary lists for one describing a logical child segment. It loops through until the last list has been checked (bit 0 of byte 1 in DMBSND). Because it has reached the end, the routine determines that the DL/I control blocks could not be found, and the abend is issued.

**Possible cause:** DBDGEN error. Verify that valid DBDs are available for the database being initially loaded, reloaded, or scanned.

**System action:** The program terminates abnormally.

**Programmer response:** Verify that valid DBDs are available for the databases being initially loaded, reloaded, or scanned.

**Problem determination:** 1, 2, 3, 4, 17d, and 20, a printout of the DFSURWF1 data set, and a printout of the DFSURCDS

**Source:** IMS abends

**Module:** DFSDSEH0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

**0955**

**Explanation:** This abend code is issued by modules DFSURGP0, DFSURGS0, and DFSURG10 when an abend control statement is supplied in the SYSIN input stream.

**Analysis:** This abend is a standard abend issued by module DFSURGP0, DFSURGS0, DFSURG10, or DFSURPR0.

**System action:** The abend is issued at the end of program execution.

**Operator response:** Determine why the abend statement was supplied to the input, and take appropriate action.

**Problem determination:** 1, 2, 3, 4, 17d, and 20, a printout of DFSURWF1, or a printout of DFSURWF3, or both

**Source:** IMS abends

**Module:** DFSURGP0, DFSURGS0, DFSURG10, DFSURPR0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

**0956**

**Explanation:** The work data set generator is called by a HDAM initial load, reload, or scan to create a work tape for building logical relationships. While attempting to resolve a logical relationship, it encountered a logical parent segment with a concatenated key length of zero bytes.

**Analysis:** 0956 is a standard abend issued from module DFSDSEH0. The program status word (PSW) at entry-to-abend and the registers in the abend SVRB should be used for problem diagnosis.

Register 9 contains the address of the PCB. DBPCB is used to calculate the logical parent concatenated key length.

Key	Label	Description
Reg2=DBPCBLKY	LPLC	This routine determines that the concatenated key of the logical parent should be used to resolve a relationship. Register 2 is tested for the length of the concatenated key of the logical parent. If register 2=0, the abend is issued.
Reg4=CENT	#@LB62	
Reg8=PSDB		
Reg9=PCB		
Reg10=PST		

**Possible cause:** Error in DBDGEN. Either a logical child segment was defined as a root segment or the sequence field was not defined on a root segment by specifying NAME=xx instead of NAME=(xx, seq, u).

**System action:** IMS terminates abnormally.

**Programmer response:** Correct the field definition, and resubmit the job.

**Problem determination:** 1, 2, 3, 4, 17d, 35

**Source:** IMS abends

**Module:** DFSDSEH0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

## 0957

**Explanation:** A segment is physically paired. Either the two intersection data lengths were not equal (for fixed-length segments); or the maximum intersection data lengths for the two paired segments were not equal. Message DFS0951I identifies the segment.

**Analysis:** 0957 is a standard abend issued from multiple CSECT module DFSDLBL0 Message DFS951I accompanies this abend.

The registers saved at label ERRORSVE within DFSDLBL0 should be used for problem isolation. In the abend SVRB registers, register 6 points to the referenced segment name and register 2 and register 4 contain the two calculated data lengths.

The abend is issued from a common routine at label SETPSEU. When an error is detected, the subroutine branches to label ERR957, which restores DFSDLBL0 as the base and branches to ERROR957. Here, register 0 through register 15 are saved at label ERRORSVE. A branch is taken to RETURN, where message DFS0951I is issued and PSTSTAT is set to X'16'. RETURN branches to SETPSEU, which issues the abend (SVC 13).

Key	Label	Description
Reg2=physical pair's data length	CLRDDIR6	The cleaning up of DDIRs and checking for valid logical relationships are being done. The data length of the referenced segment is loaded into register 4. The segment has a physical pair and its data length is loaded into register 2. If a comparison of the two registers is not equal, then the abend is issued because data lengths for paired segments must be equal.
Reg3=physical pair's PSDB		
Reg4=segment's data length		
Reg15=segment's DMB		

**System action:** IMS terminates abnormally.

**Programmer response:** At entry to this abend, register 6 points to the referenced segment name. Register 2 and register 4 contain the two calculated data lengths. Correct the DBD, and re-execute the job.

**Problem determination:** 4, 17d, 35

**Source:** IMS abends

**Module:** DFSDLBL0

**Related information:**

 DFS0951I

Chapter 2, "IMS problem determination," on page 13

## 0958

**Explanation:** A PSB cannot reference a primary index DBD. Message DFS0952I, which accompanies this abend, identified the DBD.

**Analysis:** 0958 is a standard abend issued from multiple CSECT module DFSDLBL0.

The registers saved at label ERRORSVE within DFSDLBL0 should be used for problem isolation. Register 5 in the abend SVRB point to the DBD name in error.

The abend is issued from a common routine at label SETPSEU. When an error is detected, the subroutine branches to label ERROR958, which saves register 0 through register 15 at label ERRORSVE. A branch is then taken to RETURN, where message DFS952I is issued and PSTSTAT is set to X'16'. RETURN branches to SETPSEU, which issues the abend (SVC 13).

Key	Label	Description
Reg1=SEC LIST address DMBXDFLGSVRB should be used for problem diagnosisX'10' Reg6=SDB address Reg7=PSDB	SUMSPAC	The PCB is being processed. The database is an index one. Field SDBF3 is tested for any PROCOPT other than "G". Updating is specified, so a test is made to see if the database is a secondary index. If it is not a secondary index, an error occurs. The abend is issued because primary HIDAM indexes only allow the read processing option.

**System action:** IMS terminates abnormally.

**Programmer response:** Change the PSB so that the primary index will not be processed, and resubmit the job.

**Problem determination:** 3, 4, 17d, 35

**Source:** IMS abends

**Module:** DFSDLBL0

**Related information:**

 DFS0952I

Chapter 2, "IMS problem determination," on page 13

## 0959

**Explanation:** An error exists in the logical child SEGM statement in the database referenced by message DFS0953I. It is not permissible to specify a direct pointer to a database with HISAM organization. PTR=LP is an incorrect specification; only PTR= should be specified.

**Analysis:** 0959 is a standard abend issued from multiple CSECT module DFSDLBL0.

The registers saved at label ERRORSVE within DFSDLBL0 should be used for problem isolation. Register 5 in the abend SVRB points to the DBD name in error.

The abend is issued from a common routine at label SETPSEU. When an error is detected, the subroutine branches to label ERROR959, which saves register 0 through register 15 at label ERRORSVE. A branch is then taken to RETURN, where message DFS0953I is issued and PSTSTAT is set to X'16'. RETURN branches to SETPSEU, which issues the abend (SVC 13).

Key	Label	Description
Reg3=PSDB address DMBPTR=X'02' DMBLOG=X'04' Reg8=SEC LIST entry Reg10=SEC LIST entry	CLRDDIR4	Validity checking of logical relationships is being done. A logical child segment's secondary list has been obtained. The segment has a logical parent pointer. The logical parent is in HISAM. Direct pointers are not allowed for HISAM, so the abend is issued.

**System action:** If running under DL/I, IMS terminates abnormally.

**Programmer response:** Correct the logical child SEGM statement to remove the LP specification. Specify instead only PTR=, and reassemble the DBD. Rerun the job.

**Source:** IMS abends

**Module:** DFSDLBL0

**Related information:**

 DFS0953I

## 0960

**Explanation:** An error exists in the LCHILD statement in the database referenced in message DFS0954I. You have specified either PTR=SNGL or PTR=DBLE; both specifications are incorrect. It is not permissible to specify a direct pointer to a database with HISAM organization.

**Analysis:** 0960 is a standard abend issued from multiple CSECT module DFSDLBL0. Message DFS954I accompanies this abend.

The registers saved at label ERRORSVE within DFSDLBL0 should be used for problem isolation. Register 5 in the abend SVRB points to the DBD name in error. The abend is issued from a common routine at label SETPSEU. When an error is detected, the subroutine branches to label ERROR960, which saves register 0 through register 15 at label ERRORSVE. A branch is then taken to RETURN, where message DFS954I is issued and PSTSTAT is set to X'16'. RETURN branches to SETPSEU, which issues the abend (SVC 13).

Key	Label	Description
Reg3=PSDB address DMBSLCFLSVRB should be used for problem diagnosis0 DMBLORG=X'04' Reg15=DMB address	CLRDDIRC	Validity checking of logical relationships is being done. A logical child statement has logical child forward pointer specified. The logical child segment is in a HISAM database. Direct pointers are not allowed for HISAM, so the abend is issued.

**System action:** If running under DL/I, IMS terminates abnormally.

**Programmer response:** Correct the LCHILD statement, reassemble the DBD, and rerun the job.

**Source:** IMS abends

**Module:** DFSDLBL0

**Related information:**

 DFS0954I

## 0961

**Explanation:** An error existed in the XDFLD statement in a DBD that points to the Shared Index (see message DFS0955I). You have specified more than one XDFLD constant of the same value. Each XDFLD statement for a Shared Index must have a unique CONST= specification.

**Analysis:** 0961 is a standard abend issued from multiple CSECT module DFSDLBL0. Message DFS0955I accompanies this abend.

The registers saved at label ERRORSVE within DFSDLBL0 should be used for problem isolation. Register 5 in the abend SVRB points to the DBD name in error.

The abend is issued from a common routine at label SETPSEU. When an error is detected, the subroutine branches to label ERROR960, which saves register 0 through register 15 at label ERRORSVE. A branch is then taken to RETURN, where message DFS0955I is issued and PSTSTAT is set to X'16'. RETURN branches to SETPSEU, which issues the abend (SVC 13).

Key	Label	Description
Reg2=DMBXDPAD Reg5=PSDB address Reg6=SEC LIST address Reg8=FDB address	CKCONSTB	This routine is working with a shared index. It is looking at the PSDBs (other than the first PSDB) to check the XDFLD which specifies a constant value against a table of constant values pointed to by register 2. If a PSDB is encountered with the same constant value as that in the register 2 table, then the abend is issued because the constant value must be unique.

**System action:** If running under DL/I, IMS terminates abnormally.

**Programmer response:** Correct the XDFLD statements in error, reassemble the DBDs, and rerun the job.

## 0962 • 0963

**Source:** IMS abends

**Module:** DFSDLBL0

**Related information:**

 DFS0955I

---

### 0962

**Explanation:** The SENSEG statements within the named PSB were not specified in hierarchic sequence (top to bottom, left to right). Refer to message DFS0956I, which accompanies this abend, for more information.

**Analysis:** 0962 is a standard abend issued from multiple CSECT module DFSDLBL0.

The registers saved at label ERRORSVE within DFSDLBL0 should be used for problem isolation. Register 5 and register 6 in the abend SVRB point to the PSB name and the segment name, respectively.

The abend is issued from a common routine at label SETPSEU. When an error is detected, the subroutine branches to label ERROR962, which saves register 0 through register 15 at label ERRORSVE. A branch is then taken to RETURN, where message DFS0956I is issued and PSTSTAT is set to X'16'. RETURN branches to SETPSEU, which issues the abend (SVC 13).

Key	Label	Description
Reg6=SDB Reg7=parent SDB SDBFCSDB=0 Reg10=displacement from parent SDB to current SDB	SKIPAND	The inline SDBs are being built. It has been determined that the current SDB is the first child of its parent. Because a segment's first child immediately follows it hierarchic sequence, these two SDBs must be contiguous. If they are not, a hierarchic sequence error exists and the abend is issued.

**System action:** IMS terminates abnormally.

**Programmer response:** Correct the order of the SENSEG statements, and re-execute the job.

**Source:** IMS abends

**Module:** DFSDLBL0

**Related information:**

 DFS0956I

---

### 0963

**Explanation:** Backout was unable to open the checkpoint control data set (SYSIN), or the DEVTYPE macro failed.

**Analysis:** 0963 is a standard abend issued from module DFSBACK0.

Use the registers in the abend SVRB for problem isolation. Register 12 is the base register. The program status word (PSW) at entry-to-abend points to the common abend routine at label ABSYSIN.

DFSBACK0 is responsible for processing a log used as input to the database backout module DFSRDBC0. The DEVTYPE macro is issued to check for SYSIN input of the checkpoint ID as a stopping point. Register 15 contains the return code from DEVTYPE:

#### Codes

##### Explanation

**X'04'**

The ddname was not found.

**X'08'**

Invalid area address. The address of the output area either violates protection, or is out of the range of virtual storage.

Normal register usage for the module: Register 7=PDIR, register 9=SCD, register 10=PSB, register 11=PST.

Key	Label	Description
Reg15>X'04'	DLITCBL	The DEVTYPE (SVC 24) macro is issued to verify the existence of a checkpoint data set. A nonzero return code greater than X'04' results in an error and the abend is issued.
CHKPDCB=DCB DCBOFLGSSVRB should be used for problem diagnosisX'10'	DLITDBL	The OPEN (SVC19) macro is issued for the checkpoint SYSIN data set. The DCB is internal to the module and is located at label CHPKDCB. If the OPEN was unsuccessful, DCBOFLGSSVRB should be used for problem diagnosisX'10', the abend is issued.

**System action:** The program terminates abnormally.

**Programmer response:** Correct the SYSIN DD statement or the data set, or both, and rerun backout.

**Problem determination:** 2, 4

**Source:** IMS abends

**Module:** DFSBACK0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

## 0965

**Explanation:** There was an inconsistency in the DL/I control blocks involving a segment in a database being reorganized or initially loaded. Control blocks for a logically related database were incomplete or in error. This problem might be an IMS system error, or an improperly defined logical relationship.

**Analysis:** This is a standard abend issued by module DFSURPR0.

**System action:** The IMS Utility Preorganization module (DFSURPR0) terminates abnormally.

**Programmer response:** Verify that the DBDs for all logically related databases are accurate and available.

**Problem determination:** 2, 8, 11, 18, 35

**Source:** IMS abends

**Module:** DFSURPR0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

## 0966

**Explanation:** The PCB contained a PROCOPT of H but the DBD is not a DEDB. The PROCOPT H is only valid for a DEDB. Message DFS964I is issued prior to this abend and contains the name of the PCB.

**Analysis:** 0966 is a standard abend issued from multiple CSECT module DFSDLBL0.

The registers saved at label ERRORSVE within DFSDLBL0 should be used for problem isolation. Register 6 and register 7 in the abend SVRB points to the PCB name and PSB name, respectively.

The abend is issued from a common routine at label SETPSEU. When an error is detected, a branch is taken to ERROR966. Registers 0 through 15 are saved at label ERRORSVE. Register 14 points to the instruction following the branch to ERROR967. A branch is then taken to RETURN, where message DFS964I is issued and PSTSTAT is set to X'16'. RETURN branches to SETPSEU, which actually issues the abend (SVC 13):

Key	Label	Description
Reg1=address 'H' or Reg3=address 'H'	HMCHK1	These routines check the PROCOPT of non-DEDB DBDs. The characters in the PBPCBPRO are checked for 'H'. If the character 'H' is found the abend is issued.
	HCHK1	

**System action:** IMS terminates abnormally.

**Programmer response:** At entry to this abend, register 6 points to the PCB name and register 7 points to the PSB name. Correct the PSB, and run the program again.

## 0967 • 0969

**Problem determination:** 4, 17d, 35

**Source:** IMS abends

**Module:** DFSDLBL0

**Related information:**

 DFS964I

Chapter 2, "IMS problem determination," on page 13

---

### 0967

**Explanation:** The PSB contained more than one explicit reference to a PHIDAM DBD with a PROCOPT of L or LS. Message DFS1008I is issued before this abend and contains the name of the PSB.

**Analysis:** This abend is a standard abend issued from multiple CSECT module DFSDLBL0.

The registers saved at label ERRORSVE within DFSDLBL0 should be used for problem isolation. Register 5 and register 6 in the abend SVRB point to the DBD name and PSB name, respectively.

The abend is issued from a common routine at label SETPSEU. When an error is detected, a branch is taken to ERROR967. Registers 0 through 15 are saved at label ERRORSVE. Register 14 points to the instruction following the branch to ERROR967. A branch is then taken to RETURN, where message DFS1008I is issued and PSTSTAT is set to X'16'. RETURN branches to SETPSEU, which actually issues the abend (SVC 13):

Key	Label	Description
DDIRADDR=X'40'	ISVHDORG ISHDORG	These routines build DCBs for HIDAM type DBDs. The high order byte of DDIRADDR is tested for a X'40', which indicates multiple load. If multiple load PROCOPTs are specified, the abend is issued.

**System action:** IMS terminates abnormally.

**Programmer response:** At entry to this abend, register 6 points to the PSB name and register 5 points to the DBD name. Correct the PSB and run the program again.

**Programmer response:** 4, 17d, 35

**Source:** IMS abends

**Module:** DFSDLBL0

**Related information:**

 DFS1008I

Chapter 2, "IMS problem determination," on page 13

---

### 0969

**Explanation:** Reading a change accumulation data set, a detail record was encountered before a header record, or a record with an invalid record type code was encountered.

**Analysis:** 0969 abend is issued when the input record from the Change Accumulation data set is not a valid record type. The abend SVRB contains the registers from DFSURIO0. The program status word (PSW) at entry-to-abend points to the instruction that issued the abend (SVC 13). The routine that detects the error branches to the routine issues the abend (SVC 13).

Key	Label	Description
Reg1=address of input record Reg14=BAL to DFSURIO0	GETFILE	DLOGCODE is not equal to return codes of X'00', X'24', X'25', X'50', X'5B', or X'5947'.

**Possible cause:** Invalid data sets were used as input to change accumulation data set or Recovery utilities.

**System action:** IMS terminates abnormally.

**Programmer response:** Correct the problem, and rerun the job.



**Problem determination:** 1, 2, 3, 4

**Source:** IMS abends

**Module:** DFSURIOO

**Related information:**

Chapter 2, "IMS problem determination," on page 13

## 0970

**Explanation:** One of the several possible error conditions occurred while restart processing was attempting to determine the checkpoint to be used in the restart. Register 10 at entry to abend contains a return code that pinpoints the error.

**Analysis:** 0970 is a standard abend issued from module DFSRBCP0. This abend only occurs during restart.

The program status word (PSW) at entry-to-abend can be used to determine which instruction issued the abend (SVC 13). Use the registers in the abend SVRB for problem isolation. Register 10 contains the subcode.

Key	Label	Description
Reg2=AWE Reg10=1		An attempt to initialize the checkpoint table from the restart data set (RDS) failed. The feedback field in the AWE (AWRFDBK) contains a return code from DFSRDS00, indicating the error detected. The feedback field, AWRFDBK, is a 4-byte area. The low-order byte contains a return code from a lower service routine, and is not documented here. The 3 high-order bytes contain one of the following eye-catchers indicating the type of error encountered. <b>OPNx</b> A restart data set open or initialization error occurred. For information about the restart data set open failure, see message DFS3128A, which precedes this abend. <b>RLVx</b> The IMS release level of the active and backup are different. Both the active and backup system must be at the same IMS version and release level. <b>GIBx</b> An IOSB, used in the I/O for the restart data set, could not be obtained. This is a system error. <b>BCPx</b> An entry in the control block table could not be found for the checkpoint table, BCPT. This is a system error. <b>BCBx</b> A quick save area could not be obtained. This is a system error. <b>LOGx</b> The checkpoint table could not be restored from the RDS data set. An attempt to obtain the latest copy of the table from the log (X'42' log record) failed. This error implies an OLDS open/read error. <b>CBTx</b> An entry in the control block table could not be found for one of the following blocks: RRE, LCRE, SIDX. This is a system error. <b>ALTx</b> An entry in the control block table could not be found for one of the following blocks: RRE, LCRE, SIDX. This is a system error. <b>QCSx</b> A storage allocation request failed for one of the following: BCPT, RRE, SIDX, LCRE. Or, if the return code is 04, the MAXPST startup parameter is less than the MAXPST value was at the previous IMS restart. <b>IDMX</b> A mismatch between the IMSID (non-xrf) or the RSENAME (xrf) has occurred. This change is not allowed during a warm start of IMS.
Reg2=AWE Reg10=2 Reg15=BCB return code		IMS was unable to obtain a subsystem index block (SIDX). An internal IMS error.
Reg10=3		An attempt was made to restart the system before a valid checkpoint was written. This usually happens when an /ERE is attempted before a good cold start is accomplished.
Reg10=4		IMODULE failure to obtain storage for buffer used in reading log for restart.
Reg10=5 Reg15=logger return code		Error in opening the log for reading.
Reg10=6 R15=logger return code		Error in reading the first checkpoint record.
Reg10=7 Reg3=A(BLOCK)		Cannot find '4001', begin checkpoint, record in block returned by the logger in response to the first read request.
Reg10=8 Reg3=A(BLOCK)		Block returned by the logger in response to the first read request contained the wrong checkpoint ID.

## 0971

Key	Label	Description
Reg10=9 Reg3=A(BLOCK)		The checkpoint found in the block was the correct one. However, the checkpoint is not compatible with the type of restart requested. Either a /ERE CHKPT0 was specified but the checkpoint returned was not taken during a cold start; or, a BUILDQ was specified but the checkpoint does not contain the queue records.
Reg10=0A Reg15=return code from FIND		An emergency restart was requested. The checkpoint table was restored from the RDS data set. DFSRBCP0 was unable to find the SIDX for the local subsystem.
Reg10=0B		DFSRBCP0 determined that the value in the TOD clock was earlier than the value in the current checkpoint in the checkpoint table.
Reg10=0C		DFSRBCP0 was unable to dequeue or release the RRE.

*For APAR processing:* Storage dump

**System action:** IMS terminates abnormally.

**Programmer response:** Be sure that all proper log data sets are allocated to the control region. Also, be sure that enough storage is allocated to the control region. This abend is probably an IMS system error.

**Problem determination:** 4, 5, 17b, 35

**Source:** IMS abends

**Module:** DFSRBCP0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

## 0971

**Explanation:** One of several possible error conditions occurred while restart was in progress.

**Analysis:** Register 15 at entry to abend contains one of the following codes:

### Codes

#### Explanation

#### X'01'

An attempt to open the system data sets failed. A message indicating which data set failed to open is also provided.

#### X'02'

Restart was unable to obtain storage for its backout table, DFSRPSTB. Ensure that the control region was allocated enough storage and rerun.

#### X'03'

An internal error occurred in attempting to dequeue a LCRE.

#### X'04'

An internal error occurred in attempting to release a LCRE.

#### X'05'

An internal error occurred in attempting to dequeue a RRE.

#### X'06'

An internal error occurred in attempting to release a RRE.

For codes 01-02 the problem can probably be resolved by examining the JCL and issued messages. For codes 03-06 an internal examination of the problem is required.

**System action:** IMS terminates abnormally.

**Programmer response:** For codes 01-02 the problem can probably be resolved by examining the JCL and issued messages. For codes 03-06 an internal examination of the problem will be required.

**Problem determination:** 4, 5, 17b, 35

**Source:** IMS abends

**Module:** DFSRLP00, DFSRST00, DFSRBCP0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

### 0979

**Explanation:** The database change logger, DFSRDBL0, encountered an error condition or a process failure and cannot complete its request processing. Register 15 contains a hexadecimal reason code.

**Analysis:** 0979 is a standard abend issued by module DFSRDBL0 when it detects an error in the parameters passed to it, or when a process required to complete the change log request fails.

Register 15 in the abend SVRB contains a reason code for the abend. (See table below.) Register 14 is used to BAL to the common routine at label ABND979, where the abend is issued. Registers 2, 4, 8, 9, and 11 contain the PST, DDIR, log work area, JCB, and SCD addresses, respectively.

Key	Nearest Label	Description
Reg15=1	ORGDONE	Invalid database organization
Reg15=1	SETFUNC	Invalid PSTWRK1 function.
Reg15=1	BADFUNC	Invalid DLOGPARM function
Reg15=2	RBAKSDS	HD RBA is zero.
Reg15=3	GOODLGWX	LGWX IPAGE GET failed. Extended private storage in CTL (LSO=Y) or DLI (LSO=S) might need to be expanded.
Reg15=4	ALLOCALL	SWAREQ request failed.
Reg15=5	ALLOCALL	IRLM Quit request failed.
Reg15=6	ALLOCALL	DBRC ALLOC call failed.
Reg15=7	ALLOCALL	TIOT DDNAME does not match DMB.
Reg15=8	SERVCMPR	Compression service error
Reg15=9	BFSERCH	Invalid PSTBUFFA
Reg15=A	XRFL0100	A DDIR could not be found by a FUNC=SCAN using the local DMB number.
R15=B	SRG000B	DBRC indicated not registered or not a part.
R15=C	SRG000C	DBRC encountered an I/O error.
R15=D	SRG000D	DBRC detected an undefined call.
R15=E	SRG000E	DBRC detected an invalid REORG # passed.
R15=F	SRG000F	Unknown DBRC return code
R15=10	SRG000G	DBRC open RECON problem
R15=11	SRG000H	DBRC close RECON problem
R15=12	SRG000I	DBRC detected an REORG# was not passed .

**System action:** IMS terminates with a memory dump.

**Programmer response:** This is an IMS internal logic error.

**Problem determination:** 1, 4, 11, 35

**Source:** IMS abends

**Module:** DFSRDBL0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

**0982**

**Explanation:** An emergency restart was unable to read the backout queue for the referenced program and database.

**Analysis:** 0982 is a standard abend issued by DFSRBOI0.

During emergency restart backout processing, an I/O error occurred on the log data set. Message DFS982I is sent to the master console and abend occurs. The I/O error is detected by the ICHECKOS routine (DFS AOS30) and a return code is passed back in register 15.

The program status word (PSW) at entry-to-abend points to the instruction within the routine labeled X100 from which the abend (SVC 13) is issued. Register 10 contains the address of the database log record. Register 5 contains the address of the current PST. Register 11 contains the address of RSRDSECT. Register 12 contains the base register. These registers are stored in the abend SVRB.

Key	Label	Description
Reg1=X'03D6'	RBOI0040	The dynamic log data set has experienced a read I/O error, thus preventing completion of the required backout. Message DFS982I and abend result.

**Possible cause:** Defective dynamic log data set

**System action:** IMS terminates abnormally.

**Programmer response:** If this error occurred because the backout records have been archived and are no longer in the OLDS, use the Database Backout utility with the SLDS as input to perform the backout.

**Problem determination:** 1, 2, 3, 4, 5

**Source:** IMS abends

**Module:** DFSRBOI0

**Related information:**

 DFS982I

Chapter 2, "IMS problem determination," on page 13

**0985**

**Explanation:** An emergency restart was unable to perform a backout. Register 15 at entry to abend contains a return code.

**Analysis:** 0985 is a standard abend issued by DFSRBLB0.

The program status word (PSW) at entry-to-abend will point to abend (SVC 13). Register 15 of the registers at entry-to-abend contain a reason code with the following meanings:

**Reason code****Explanation****X'01'**

Unable to create an ITASK or release a QSAV for the backout task. (internal).

**X'04'**

Invalid log record address for backout found.

**X'05'**

Unable to obtain a QSAV or RRE for a backout task, or unable to enqueue an RRE to the SIDX.

**X'06'**

Unable to determine the DDIR of an altered database.

**X'07'**

An /ERE command was unable to determine the RBN of a database record.

**X'08'**

An XRF alternate system failed to find a DPST while processing a type X'31' log record.

**X'09'**

Unable to obtain subpool 231 storage for an LCRE.

**X'0A'**

An XRF alternate system does not have a DPST for database record processing.

*For APAR processing:* Storage dump and log used for emergency restart

**System action:** IMS terminates abnormally.

**Problem determination:** 4, 5, 17b, 35

**Source:** IMS abends

**Module:** DFSRBLB0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

**0986**

**Explanation:** A segment in a logical database was specified as its source a segment in another logical database. Refer to message DFS2425I immediately preceding this abend.

**Analysis:** 0986 is a standard abend issued by module DFSDLBL0. Message DFS2425I accompanies this abend.

The registers saved at label ERRORSVE in CSECT DFSDLB80 within DFSDLBL0 should be used for problem determination. The registers saved in the abend SVRB contain pointers to the names used in message DFS2425I, as follows:

- Register 5 points to the segment name.
- Register 6 points to the first DBD name.
- Register 7 points to the source DBD name.

The abend is issued from a common routine at label SETPSEU. When an error is detected, the subroutine that detected the error issues a BAL register 14 to label ERROR986 in CSECT DFSDLB80, which saves register 0 through register 15 at label ERRORSVE. A branch is then taken to RETURN, where message DFS2425I is sent and PSTSTAT is set to X'16'. RETURN branches to SETPSEU, which issues the abend (SVC 13).

Key	Label	Description
Reg4=address of the second logical DDIR Reg6=address of the SDB for the segment Reg10=address of the second source segment	BCDSDBEF	The SDBPOSP field contains the name of a source segment. This source segment indicates that it also has a source segment.

**Source:** IMS abends

**Module:** DFSDLBL0

**Related information:**

 DFS2425I

**0987**

**Explanation:** An LCHILD statement in an index DBD specified an incorrect segment name for the indexed segment. Refer to message DFS2426I immediately preceding this abend.

**Analysis:** 0987 is a standard abend issued by module DFSDLBL0.

The registers saved at label ERRORSVE in CSECT DFSDLB80 within DFSDLBL0 should be used for problem determination. The registers saved in the abend SVRB contain pointers to the names used in message DFS2426I, as follows:

- Register 5 points to the index DBD name.
- Register 6 points to the index segment name.
- Register 7 points to the indexed segment name.

Register 8 points to the indexed DBD name.

The abend is issued from a common routine at label SETPSEU. When an error is detected, the subroutine that detected the error issues a BAL register 14 to label ERROR986 in CSECT DFSDLB80, which saves register 0 through register 15 at label ERRORSVE. A branch is then taken to RETURN, where message DFS2426I is sent and PSTSTAT is set to X'16'. RETURN branches to SETPSEU, which actually issues the abend (SVC 13).

Key	Label	Description
Reg5=DDIR extension containing the indexed segment name and DBD name. Reg7=index LCHILD.	NOTER988	The index LCHILD segment name does not match the indexed segment name.

**Source:** IMS abends

**Module:** DFSDLBL0

**Related information:**

 DFS2426I

## 0988

**Explanation:** An LCHILD statement in an indexed DBD referred to a nonexistent segment in the index DBD. Refer to message DFS2427I immediately preceding this abend.

**Analysis:** 0988 is a standard abend issued by module DFSDLBL0.

The registers saved at label ERRORSVE in CSECT DFSDLB80 within DFSDLBL0 should be used for problem determination. The registers saved in the abend SVRB contain pointers to the names used in message DFS2427I, as follows:

- Register 5 points to the index DBD name.
- Register 6 points to the incorrect index segment name.
- Register 7 points to the indexed segment name.
- Register 8 points to the indexed DBD name.

The abend is issued from a common routine at label SETPSEU. When an error is detected, the subroutine that detected the error issues a BAL register 14 to label ERROR988 in CSECT DFSDLB80, which saves register 0 through register 15 at label ERRORSVE. A branch is then taken to RETURN, where message DFS2427I is sent and PSTSTAT is set to X'16'. RETURN branches to SETPSEU, which actually issues the abend (SVC 13).

Key	Label	Description
Reg5=DDIR extension containing the indexed segment name and DBD name. Reg10=index segtab entry	BLDDON	The index segment name does not match the segment name in the DDIR extension.

**Source:** IMS abends

**Module:** DFSDLBL0

**Related information:**

 DFS2427I

## 0989

**Explanation:** In resolving index relationships, module DFSDLBL0 found that an index DBD was referenced by multiple LCHILD statements for indexed segments. See the accompanying message DFS2428I for the name of the index DBD.

**Analysis:** This abend is a standard abend issued by module DFSDLBL0.

The registers saved at label ERRORSVE in CSECT DFSDLB80 within DFSDLBL0 should be used for problem determination. The registers saved in the abend SVRB contain pointers to the following items:

- Register 5 points to the DDIR of the index DBD.
- Register 2 points to the DBD of the indexed database.
- Register 4 points to the DBD LCHILD table entry for one referencing DBD.
- Register 10 points to the current SEGTAB entry in the DBD.
- Register 14 points to the end of the BAL instruction, at BLDXTDOK, in the subroutine that branched to ERROR989.

The abend is issued from a common routine at label SETPSEU. When an error is detected, the subroutine that detected the error issues a BAL register 14 to label ERROR989 in CSECT DFSDLB80, which saves register 0 through register 15 at label ERRORSVE. A branch is then taken to RETURN, where message DFS2428I is sent and PSTSTAT is set to X'16'. RETURN branches to SETPSEU, which actually issues the abend (SVC 13).

Key	Label	Description
Reg5 + X'14'	DDIRBLDL	Address of DDIR Index Extension entry which refers to the segment name and DDIR of the first segment designated as indexed by this index.

**System action:** IMS terminates abnormally.

**Programmer response:** Revise the references in the indexed DBDs so that each indexed segment has its own unique index.

**Source:** IMS abends

**Module:** DFSDLBLO

**Related information:**

 DFS2428I

## 0990

**Explanation:** Module DFSDLBLO detected an error in the order that sibling segments were referenced in a logical DBD or PSB. The sibling segment dependents of a parent with PTR=HIER or in a HISAM DBD cannot be referenced in a different order from that established in the physical DBD. Refer to message DFS2429I immediately preceding this abend.

**Analysis:** 0990 is a standard abend issued from multiple-CSECT module DFSDLBLO.

The registers saved at label ERRORSVE in CSECT DFSDLB80 within DFSDLBLO should be used for problem isolation. Register 6 points to the SDB of one of the two segments out of sequence; register 15 points to the SDB of the other; register 8 points to the database PCB; and register 14 points to the end of the BAL instruction in the subroutine that branched to ERROR990.

The abend is issued from a common subroutine at label SETPSEU. When an error is detected, the subroutine that detected the error issues a BAL register 14 to label ERROR990 in CSECT DFSDLB80, which saves registers 0 through 15 at label ERRORSVE. A branch is then taken to RETURN, where message DFS2429I is written and PSTSTAT is set to X'16'. RETURN branches to SETPSEU, which actually issues the abend (SVC 13).

Key	Label	Description
Reg14	SUMSPSBNB in DFSDLB40	See above.

**System action:** IMS terminates abnormally.

**Programmer response:** Recode the logical DBD or PSB, and re-execute the procedure.

**Source:** IMS abends

**Module:** DFSDLBLO

**Related information:**

 DFS2429I

## 0991

**Explanation:** The logical child destination parent concatenation was invalid. Message DFS2430I accompanies this abend.

**Analysis:** 0991 is a standard abend issued from module DFSDLB70.

The registers saved at label ERRORSVE in CSECT DFSDLB80 within DFSDLBL0 should be used for problem isolation. Registers saved in the abend SVRB contain pointers to the names used in message DFS2430I. Register 5 points to the SDB being processed; register 6 points to the current PCB.

The abend is issued from a common subroutine at label SETPSEU. When an error is detected, the subroutine that detected the error issues a BAL register 14 to label ERROR991 in CSECT DFSDLB80, which saves registers 0 through 15 at label ERRORSVE. A branch is then taken to RETURN, where message DFS2430I is written and PSTSTAT is set to X'16'. RETURN branches to SETPSEU, which actually issues the abend (SVC 13).

Key	Label	Description
SDBPOSN +1 = X'01' Reg7=A(PSDB) DMBFLAG=OMBLPEX Reg14 = BAL	CHKVIRT	SDBPOSN indicates that the SDB is specified by the SOURCE= operand, but if DMBLPEX (=X'20') is OFF in DMBFLAG in the PSDB, it indicates that the segment is not a logical child segment and the error exit routine is taken.
SDBTFLG = SDBSLP Reg7=A(PSDB) DMBFLAG=DABLPEX Reg14 = BAL	PARNTSTM	SDBTFLG indicates that the SDB is for a concatenation of logical child to logical parent, but if DMBFLAG is not DMBLPEX, then the first source segment specified in the SOURCE= operand is not a logical-child segment and the reference is invalid.
Reg1=A(logical parent secondary list). Reg7=A(PSDB) DMBFLAG = DMBLPEX Reg14 = BAL SDBTFLG = SDBSLP	CHKNAMES	To verify that the destination-parent segment specified in the SOURCE= operand is the segment's logical parent, the segment-name table entry is loaded into register 15. When the name in the table entry does not match the name in the segment's logical-parent secondary list (register 1), the error exit routine is taken.
SDBTFLG = SDBSLP Reg7=A(PSDB) DMBFLAG = DMBLPEX Reg14 = BAL Reg15=A (segment-name table-entry) for this segment.	GET02LSA	To verify that the destination parent references the segment as a logical child, this segment's segment-name-table entry is loaded into register 15. A loop through the destination parent's secondary lists is taken, using register 1. When no secondary list is found for a logical-child segment with the same ODIR and segment code as this segment, and with a name matching the name in the table entry, the error exit routine is taken.
SDBTFLG=SDBSLP DMBFLAG=DABLPEX Reg1=A(destination parent's PSDBPSDB). Reg1 + X'20' * X'20' Reg14 = BAL	GETLCNME	To verify that the destination parent specified by the SOURCE= operand is a logical parent segment, register 1 is loaded with the address of the destination parent's PSDB. If DMBLCEX is OFF in DMBFLAG-DMBPSDB (.R1), then the exit routine is taken.

**System action:** IMS terminates with a memory dump.

**Programmer response:** Follow the steps indicated by DFS2430I.

**Source:** IMS abends

**Module:** DFSDLB70

**Related information:**

 DFS2430I

## 0992

**Explanation:** A DBD was referenced in an LCHILD statement as an index DBD invalidly. The referenced DBD was not designated as ACCESS=INDEX. Further information is given in the DFS2431I message which precedes the abend.

**Analysis:** 0992 is a standard abend issued from multiple-CSECT module DFSDLBL0.

The registers saved at label ERRORSVE in CSECT DFSDLB80 within DFSDLBL0 should be used for problem isolation. Register 2 points to the DBD of the indexed database; register 10 points to the segment-table entry in the DBD for the indexed segment; and register 14 points to the end of the BAL instruction in the subroutine that branched to ERROR992.



The abend is issued from a common subroutine at label SETPSEU. When an error is detected, the subroutine that detected the error issues a BAL register 14 to label ERROR992 or ERR992A in CSECT DFSDLB80, which saves registers 0 through 15 at label ERRORSVE. A branch is then taken to RETURN, where message DFS2431I is written and PSTSTAT is set to X'16'. RETURN branches to SETPSEU, which actually issues the abend (SVC 13).

The labels referenced in the following table are in DFSDLB00:

Key	Label	Description
Reg1=A(target DBD) Reg5=A(index DDIR) Reg10=A(target segment table entry) Reg14=BAL	BLDXTDOK	Index flagged as built before target database was completed
Reg4=DDIR of DBD illegally referenced as an index Reg4 + 20 (DDIRBLDL)=address of DDIR extension containing a segment name and DDIR address of DBD that made an invalid reference. Reg14=BAL	NOPROT	Nonindexed database has an index extension built, which means that it was referenced as an indexed DBD in an LCHILD statement.

**System action:** IMS terminates abnormally.

**Programmer response:** Correct the DBDs and rerun the job.

**Source:** IMS abends

**Module:** DFSDLBLO

**Related information:**

 DFS2431I

## 0993

**Explanation:** An invalid field name in the segment definition was referenced in a SENFLD statement. See message DFS2432I for additional information.

**Analysis:** 0993 is a standard abend issued from the composite module DFSDLBLO.

The registers saved at label ERRORSVE in DFSDLB80 within DFSDLBLO should be used for problem isolation. Register 5, register 6, register 7, and register 8 in the abend SVRB point to the PSB name, the segment name, the field name, and the DBD name, respectively.

The abend is issued from a common routine at label SETPSEU in DFSDLB80. When an error is detected, the subroutine in DFSDLB40 that detected the error issues a BAL register 14 to label ERROR993 in DFSDLB80, which saves register 0 through register 15 at label ERRORSVE. A branch is then taken to RETURN, where message DFS2432I is issued and PSTSTAT is set to X'16'. RETURN branches to SETPSEU, which actually issues the abend (SVC 13).

The labels referenced below are in DFSDLB40.

Key	Label	Description
Reg5=PSDB Reg6=SDB Reg7=FSB Reg14=BAL	FLSMRG00	No match has been found between the name in the FSB and the field names in the FDBs for the PSDB in register 5.
Reg5=PSDB Reg6=SDB Reg7=FSB	FLSMRG34 FLSMRG32 FLSLP30	If the SDB in register 6 references a physical logical child, then no match has been found between the name in the FSB and the field names in the FDBs for the PSDB of the logical parent or the field names in the FDBs for the logical child. If the SDB in register 6 references a virtual logical child, then no match has been found between the name in the FSB and the field names in the DMBSEC field entries for this segment, the field names in the FDBs for the PSDB of the logical parent, or the field names in the FDBs for the PSDB of the paired segment.

**System action:** IMS terminates abnormally.

## 0994 • 0995

**Programmer response:** Correct the PSB, and rerun the job.

**Source:** IMS abends

**Module:** DFSDLBL0

**Related information:**

 [DFS2432I \(Messages and Codes\)](#)

---

### 0994

**Explanation:** Field mapping specified in PSBGEN caused destructive overlap. See message DFS2433I for additional information.

**Analysis:** 0994 is a standard abend issued from the composite module DFSDLBL0.

The registers saved at label ERRORSVE in DFSDLB80 within DFSDLBL0 should be used for problem isolation. Register 5, register 6, and register 8 in the abend SVRB point to the PSB name, the segment name, and the DBD name respectively.

The abend is issued from a common routine at label STEPSEU in DFSDLB80. When an error is detected, the subroutine in DFSDLB40 that detected the error issues a BAL register 14 to label ERROR994 in DFSDLB80, which saves register 0 through register 15 at label ERRORSVE. A branch is then taken to RETURN, where message DFS2433I is issued and PSTSTAT is set to X'16'. RETURN branches to SETPSEU, which actually issues the abend (SVC 13).

The labels referenced below are in DFSDLB40:

Key	Label	Description
Reg1=FSB1 Reg3=FSB2 Reg6=SDB Reg8=PCB Reg14=BAL	FLSSRT22	The application program is sensitive to overlapping fields in the database segment, at least one of which is not character-type data.
Reg1=FSB1 Reg3=FSB2 Reg6=SDB Reg8=PCB Reg14=BAL	FLSSRT25	Fields overlap in the database segment but do not exhibit the same degree of overlap in the user's I/O area and the segment has update sensitivity.
Reg1=FSB1 Reg3=FSB2 Reg6=SDB Reg8=PCB Reg14=BAL	FLSSRT30	Two separate fields from the database segment are destined for the same location in the user's I/O area.

**System action:** IMS terminates abnormally.

**Programmer response:** Correct the PSB, and rerun the job.

**Source:** IMS abends

**Module:** DFSDLBL0

**Related information:**

 [DFS2433I \(Messages and Codes\)](#)

---

### 0995

**Explanation:** PCB of segment to be inserted did not specify sensitivity to sequence field. See message DFS2434I for additional information.

**Analysis:** 0995 is a standard abend from the composite module DFSDLBL0.

The registers saved at label ERRORSVE in DFSDLB80 within DFSDLBL0 should be used for problem isolation. Register 5, register 6, and register 8 in the abend SVRB point to the PSB name, the segment name, and the DBD name, respectively.

The abend is issued from a common routine at label SETPSEU in DFSDLB80. When an error is detected, the subroutine in DFSDLB40 that detected the error issues a BAL register 14 to label ERROR995 in DFSDLB80, which saves register 0 through register 15 at label ERRORSVE. A branch is then taken to RETURN, where message DFS2434I is issued and PSTSTAT is set to X'16'. RETURN branches to SETPSEU, which actually issues the abend (SVC 13).

The labels referenced below are in DFSDLB40:

Key	Label	Description
Reg6=SDB Reg8=PCB Reg14=BAL	FLSSRT90	The routine has determined that the segment has a key field to which the user is not sensitive and that the PROCOPT for this segment is either I or L.

**System action:** IMS terminates abnormally.

**Programmer response:** Correct the PSB, and rerun the job.

**Source:** IMS abends

**Module:** DFSDLBL0

**Related information:**

 DFS2434I

## 0996

**Explanation:** Field level sensitivity was specified for a logical child with PROCOPT=I or L. See message DFS2435I for additional information.

**Analysis:** This is a standard abend issued from the composite module DFSDLBL0.

The registers saved at label ERRORSVE in DFSDLB80 within DFSDLBL0 should be used for problem isolation. Register 5, register 6, and register 8 in the abend SVRB point to the PSB name, the segment name, and the DBD name, respectively.

The abend is issued from a common routine at label SETPSEU in DFSDLB80. When an error is detected, the subroutine in DFSDLB40 that detected the error issues a BAL register 14 to label ERROR996 in DFSDLB80, which saves register 0 through register 15 at label ERRORSVE. A branch is then taken to RETURN, where message DFS2435I is issued and PSTSTAT is set to X'16'. RETURN branches to SETPSEU, which actually issues the abend (SVC 13).

The labels referenced below are in DFSLB40:

Key	Label	Description
Reg6=SDB Reg8=PCB Reg14=BAL	BLDFSBS	Control was passed to this routine because the field level sensitivity was specified for this segment, and this routine determined that the processing option is either I or L.

**System action:** IMS terminates abnormally.

**Programmer response:** Correct the PSB, and rerun the job.

**Source:** IMS abends

**Module:** DFSDLBL0

**Related information:**

 DFS2435I

## 0997

**Explanation:** The required PSB control block could not be built because the logical relationships involved caused internal processing limits to be exceeded. Message DFS2436I was issued, indicating the PSB and PCB for which this occurred.

**Analysis:** 0997 is issued from the composite module DFSDLBL0. The registers saved at label ERRORSVE in

## 0998

DFSDLB80 within DFSDLBL0 should be used for problem isolation. Register 5 and register 6 in the abend SVRB point to the PCB name and the PSB name respectively.

The abend is also issued from a common routine at label SETPSEU in DFSDLB80. When an error is detected, the subroutine in DFSDLB70 that detected the error issues a BAL REG14 to label ERROR997 in DFSDLB80, which saves register 0 through register 15 at label ERRORSVE. A branch is then taken to RETURN.

Where message DFS2436I is issued and PSTSTAT is set to X'16', RETURN branches to SETPSEU, which issues the abend (SVC 13).

The labels referenced in the following table are in DFSDLB70 :

Key	Label	Description
Reg5=PCB	SUMNOIX1	Control was passed to the routine to save the highest internal process number and the routine determined that the internal process number had exceeded its limit.
Reg14=BAL	SUMALTSD	
	SUMNOIX3	
	SUMGENP2	
	SUMNDSG	
	SUMLDND	

**System action:** IMS terminates abnormally after issuing message DFS2436I.

**Programmer response:** Modify the PSB or the DBDs, or both, to reduce the number of SENSEG statements in the PCB that reference LOGICAL CHILD/ LOGICAL PARENT concatenations. re-execute PSBGEN, then re-execute the job.

**Source:** IMS abends

**Module:** DFSDLBL0, DFSDLB80

**Related information:**

 DFS2436I

---

## 0998

**Explanation:** A field defined for a virtual logical child did not fit in the virtual logical child. The field must begin and end within the logical child.

**Analysis:** 0998 is issued from the composite module DFSDLBL0, and is accompanied by message DFS2437I. The registers saved at label ERRORSVE in DFSDLB80 within DFSDLBL0 should be used for problem isolation. Register 3, register 5, register 6, and register 8 in the SVRB point to the logical parent DBD name, the virtual logical child field name, the real logical child segment name, and the real logical child DBD name, respectively.

This error is detected in module DFSDLB50. DFSDLB50 issues a BAL REG14 to label ERROR998 in DFSDLB80. DFSDLB80 saves registers 0 through 15 at label ERRORSVE, and calls DFSLBLM0, which issues message DFS2437I. DFSDLB80 then branches to label SETPSEU, which issues the abend (SVC 13).

The labels referenced below are in DFSDLB50.

Key	Label	Description
Reg2=address of X'08' seclist Reg3=address of LC PSDB Reg10=address of X'02' seclist Reg14=BAL Reg15=address of current PSDB	CLRDLP08	The virtual field does not start within the virtual logical child.
Reg2=address of X'08' seclist Reg3=address of LC PSDB Reg5=virtual field length Reg10=address of X'02' seclist Reg14=BAL Reg15=address of current PSDB	NOTER998	The virtual field is too long and extends beyond the end of the virtual logical child.

**System action:** Message DFS2437I is issued, and IMS terminates abnormally.

**Programmer response:** Correct the DBD, and rerun the job. The FIELD statement immediately after the definition of

the logical child is wrong. Either the field did not start within the logical child, or the field was too long to fit within the logical child.

**Source:** IMS abends

**Module:** DFSDLBL0, DFSDLB50

**Related information:**

 [DFS2437I](#)



## Chapter 22. IMS abend codes 1001 - 1050

This topic identifies and explains the following abend codes issued by IMS. Each code is 4 decimal digits.

### 1001

**Explanation:** The batch DL/I or DBB region could not be initialized because one of the databases used by the PSB, named in the third positional operand of the PARM field on the EXEC job control statement is an MSDB or DEDB. A subcode is set in Register 15.

**Analysis:** 1001 is a standard abend issued from DBFUDLB0 or DFSBIND0 upon detection of an invalid IMS dependent region type specified in the first positional operand of the PARM field on the EXEC job control statement.

Key	Label	Description
Reg15=X'04' or X'08'		Fast Path database management control blocks are only built at an application control maintenance utility time. No dynamic control block building is supported in any other environment. If register 15=4, DLS region type was specified. If register 15=8, DBB region type was specified.

**Possible cause:** The region type parameter specified in the first positional operand of the PARM field on the EXEC job control statement of the dependent region startup procedure was specified incorrectly. For the batch backout utility: if the PSB includes Fast Path databases, make sure region type DBB is specified.

**System action:** The batch region is terminated abnormally.

**Programmer response:** Correct the PSB, re-execute the PSBGEN, and rerun the job. If all PCBs in the PSB have access to an MSDB or DEDB, a Fast Path (IFP) region procedure should be used.

If batch backout is being attempted in a batch DL/1 region, rerun the job with a DBB region procedure.

**Problem determination:** 17a, d

**Source:** IMS abends

**Module:** DBFUDLB0, DFSBIND0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

### 1002

**Explanation:** One of the following situations occurred:

While making a DL/I call to update the IMS catalog, one of the IMS catalog utilities received a DL/I status code that the utility is not designed to handle. Message DFS4425E, DFS4432E, or DFS4474E precedes this abend.

Or, there was a GETMAIN failure due to unavailable storage. Message DFS4477E will be issued under this condition.

**Analysis:** The following table shows what the utility was doing when the DL/I status code was received:

Table 8. Key descriptions for register 15

Key in register 15	Utility	Description
X'04'	DFS3CNTL DFS3DBLC DFS3DCNC DFS3LU10 DFS3LU20 DFS3PBLC DFS3PCNC DFS3PDBD DFS3PLOC DFS3PPSB DFS3PSDC DFS3PU00 DFS3REAC DFS3SDBD DFS3SPSB DFS3SREF	GETMAIN failed due to unavailable storage. Check the REGION size, and consider setting it to 0M.

Table 8. Key descriptions for register 15 (continued)

Key in register 15	Utility	Description
X'0C'	DFS3PU00	Inserting a DBD or PSB instance in the IMS catalog. For more information about key X'0C', see the following table.
X'10'	DFS3LU10 DFS3LU20 DFS3PU00	Retrieving or deleting a DBD or PSB instance from the IMS catalog.

If the key in register 15 is X'0C', register 2 contains the address of an error block that contains the following information:

Table 9. Contents of segment instance error block (SIEB) for key X'0C'

Contents	Decimal offset	Length	Data type
'SIEB'	0	4	Character
Segment number (first, second, third, and so forth) among the catalog segments in the record being inserted	4	4	Binary
Catalog segment name	8	8	Character
PCB status code from the ISRT call	16	2	Character
Not used	18	2	Character
Address of the catalog segment being inserted	20	4	Pointer
Displacement of the catalog segment from the beginning of the root segment	24	4	Binary

**System action:** IMS issues an error message and the utility terminates abnormally.

**System programmer response:** Check the value of register 15 and review the description of the value in the preceding list.

Consult the status code information in DFS4425E, DFS4432E, or DFS4474E and, if possible, correct the underlying problem.

If the problem cannot be resolved, contact IBM Software Support.

**Source:** IMS abends

**Module:** DFS3PU00, DFS3PU10

**Related reference:**

[IMS Catalog Populate utility \(DFS3PU00\) \(System Utilities\)](#)

[IMS Catalog Record Purge utility \(DFS3PU10\) \(System Utilities\)](#)

**Related information:**

[DFS4425E \(Messages and Codes\)](#)

[DFS4432E \(Messages and Codes\)](#)

[DFS4474E \(Messages and Codes\)](#)

[DFS4478E \(Messages and Codes\)](#)



---

**1003**

**Explanation:** A Fast Path (IFP) region could not be initialized because the set timer (STIMER) option in the tenth positional operand of the PARM field on the EXEC job control statement was incorrectly specified.

**Analysis:** 1003 is a standard abend issued from DFSRRA00 upon detection of an invalid STIMER operand specification. The valid STIMER operand is 0 or 1.

Key	Label	Description
Reg2=address of RCPARMS	RAIFP	STIMER specification was validated during an IFP region initialization. If an invalid specification is detected, the abend is issued.

**Possible cause:** The STIMER parameter specified in the PARM field on the EXEC job control statement of the IFP startup procedure was specified incorrectly.

**System action:** The dependent region is terminated abnormally.

**Programmer response:** Correct the STIMER operand on the EXEC job control statement and rerun the job.

**Problem determination:** 8, 11

**Source:** IMS abends

**Module:** DFSRRA00

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

**1004**

**Explanation:** The Fast Path (IFP) region could not be initialized because the abnormal termination limit count (TLIM) option in the seventh positional operand of the PARM field on the EXEC job control statement was incorrectly specified.

**Analysis:** This abend is a standard abend issued from DFSRRA00 upon detection of a TLIM operand specification of zero. The valid TLIM operand is from 1 to 99.

Key	Label	Description
Reg2=address of RCPARMS	RAFP1	TLIM specification is validated during an IFP region initialization. If an invalid specification is detected, the abend is issued.

**Possible cause:** The TLIM parameter specified in the PARM field on the EXEC job control statement of the IFP startup procedure was specified incorrectly.

**System action:** The dependent region is terminated abnormally.

**Programmer response:** Correct the TLIM operand on the EXEC job control statement, and rerun the job.

**Source:** IMS abends

**Module:** DFSRRA00

---

**1005**

**Explanation:** A batch message processing (BMP) step or a Fast Path (IFP) region was not scheduled for one of the following reasons:

- The dependent region was a BMP, but the PSB was defined as a Fast Path application.
- The dependent region was an IFP, but the PSB was not defined as a Fast Path application.
- The dependent region was an IFP, but the PSB was defined as a message-driven application and the subsystem type was DBCTL.
- The dependent region was an IFP and the PSB was defined as a Fast Path application, but the application was neither message-driven nor a Fast Path utility.

**Analysis:** This abend is a pseudoabend detected by DFSSBMP0 and issued by DFSPCC20.

Key	Label	Description
	BMPRCFP	The IMS dependent region type and the PSB definition are validated when a BMP or an IFP region is scheduled. If an invalid specification is detected, the pseudoabend is initiated.

**Possible cause:** The causes include:

- If the dependent region type specified in the first positional operand of the PARM field on the EXEC control statement was BMP, the PSB specified in the third positional operand of the PARM field might be defined as a Fast Path application in IMS system definition.
- If the dependent region type specified was IFP, the PSB might not be defined as a Fast Path application.
- If the dependent region type specified was IFP, the PSB might be defined as a message-driven application and the subsystem type might be DBCTL.
- If the dependent region type specified was IFP and the PSB was defined as a Fast Path application, the PSB must be either message-driven or defined as a Fast Path utility. Otherwise, an abend 1005 will be issued.

**System action:** The dependent region is abnormally terminated.

**Programmer response:** Determine the cause of the problem, and make the appropriate correction.

**Problem determination:** 8, 10, 11

**Source:** IMS abends

**Module:** DFSSBMP0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

## 1006

**Explanation:** The number of buffers requested for page fixing exceeded the total number of buffers currently available when using the buffer manager as defined by DBBF, DBFX or BSIZ. Alternately, the amount of 64 bit storage requested exceeds the maximum amount when using the 64 bit buffer manager as defined by FPBP64 and set by the user with the FPBP64M parameter. This abend can be issued for any region that has been defined to access Fast Path databases.

**Analysis:** 1006 is a pseudoabend issued during dependent region scheduling when a Fast Path buffer page fix request cannot be satisfied. Abend 1006 is detected by DFSSMSC0 or DFSSBMP0, and issued by DFSPCC20.

This abend is set by DBFINTE0 with a return code 08, and is passed back and detected by DFSSMSC0, DFSSBMP0, DFSSJBMP, or DFSSJMP0 and issued by DFSISI00 or DFSRRC20.

The number of page-fixed buffers required is determined as follows, for the current buffer manager as defined by DBBF, DBFX, and BSIZ:

Total page-fixed buffer required = value of DBFX operand  
 specified + sum of NBA operands specified in all  
 scheduled regions + maximum OBA operands specified  
 in all scheduled regions + number of DEDB areas opened.

If the computed total is greater than the value of the DBBF operand specified in the IMS control region EXEC PARM field, the pseudoabend is initiated. This check is made for each dependent region that was defined to access Fast Path databases, even if NBA=0 and OBA=0 was specified on the region JCL.

For the FP 64 bit buffer manager as defined by FPBP64, the current amount of 64 bit storage allocated to the buffer pool exceeds the maximum amount of 64 bit storage as you established with the FPBP64M parameter, or the UPDATE POOL TYPE(FPBP64) command.

**Possible cause:** The values specified for the NBA and OBA parameters in the EXEC PARM field of the IFP, BMP, or MPP region JCL can be too large, or the value specified for the DBBF parameter in the EXEC job control statement of the IMS control region can be too small. Enter the /DIS POOL FPDB command to see the number of Fast Path buffers available.

For the FP 64 bit buffer manager as defined by FPBP64, the current allowable amount of 64 bit storage as set by FPBP64M or by an UPDATE POOL TYPE(FP64) is less than the amount of 64 bit storage currently in use.

**System action:** The dependent region is terminated abnormally.

**Programmer response:** After checking the NBA and OBA operands specified on the EXEC job control statement, either

- Decrease the operand values and rerun the job.
- Wait for another region to terminate, and rerun the job.

For the Fast Path 64 bit buffer manager as defined by FPBP64, increase the amount of allowed 64 bit storage with the UPDATE POOL TYPE(FPBP64) command and rerun the job.

**Problem determination:** 8, 10, 11

**Source:** IMS abends

**Module:** DFSISI00, DBFBPN61, DBFCBHL0, DBFINTE0, DBFMBED0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

## 1007

**Explanation:** During a Fast Path (IFP) initialization, program DBFCPY00 detected that the scheduled Fast-Path PSB contained one or more GSAM PCBs.

**Analysis:** 1007 is a standard abend issued by module DBFCPY00.

Note that GSAM is not supported in an IFP region. The APPLCTN statement in the IMS system definition should have FPATH=NO specified if GSAM access is required. In this case, change the IMS system by redefining the PSB as a non-Fast Path application, and rerun the PSB in a BMP region. If GSAM access is not required, remove PCB statements with TYPE=GSAM from the PSBGEN input stream, and rerun the PSBGEN and ACBGEN.

**System action:** The dependent region is terminated abnormally.

**Programmer response:** The PSB in error needs to be corrected. Either redefine the PSB as a non-Fast Path application, or remove the GSAM PCBs from the PSB.

**Problem determination:** 4, 9, 18, 19

**Source:** IMS abends

**Module:** DBFCPY00

**Related information:**

Chapter 2, "IMS problem determination," on page 13

## 1008

**Explanation:** During a MPP, BMP, or IFP dependent region termination, program DBFATRM0 detected an error.

- An IFP message-driven application program returned normally but without a QC status code posted in the I/O PCB; or without releasing the Fast Path buffers.
- An MPP, BMP, or IFP nonmessage-driven application program returned normally without releasing Fast Path database buffers.

**Analysis:** 1008 is a pseudoabend issued from DBFATRM0. The following rules are enforced during application program normal termination cleanup process.

1. A message-driven application must receive a 'QC' status code posted in I/O PCB and release Fast Path database buffers.
2. A MPP/BMP/IFP nonmessage-driven application must release Fast Path database buffers.

A violation of the above rules will result in abnormal termination.

Key	Label	Description
Get DBFEPST0 diagnostic area copied to the dependent region dump and check the EPSTDREG, EPSTXCOC and EPSTNRBH fields in the EPST. (See DBFATRM0 source code for Fast Path control block copy at an abnormal termination.)		The IFP message-driven application program did not receive a 'QC' status code prior to its normal return. EPSTDRQC flag is off, or the application program did not release buffers prior to its normal return. Either EPSTXCOC or EPSTNRBH contains a nonzero value.

## 1009

*Possible cause:* Application program error

**System action:** IMS continues processing.

**Programmer response:** Correct the application program. For a message-driven application program, return control to the IMS program controller only after a 'QC' status code is posted in the I/O PCB; else after issuing a "SYNC" or a "ROLB" call to release the buffers.

**Problem determination:** 17a

**Source:** IMS abends

**Module:** DBFATRM0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

## 1009

**Explanation:** Either IMS Fast Path (IFP) was specified for the region type and Fast Path was not generated for this IMS system or a critical condition that should not occur was detected in modules that process IFP region initialization and termination. A subcode is set in register 15.

**Analysis:** Abend 1009 is a standard abend.

**For DBFHIEL0, DBFHQMI0:**

An FP module received a nonzero return code while attempting to release an EMHB using the DFSPOOL REL=EMHB function.

1009 is a standard abend issued from modules DBFHIEL0 and DBFHQMI0.

Key	Label	Description
Reg15=X'21'		

**For DBFIBUF0:**

| A critical condition that should not occur was detected in DBFIBUF0.

This is a standard abend issued from DBFIBUF0. The content of register 15 indicates the reason for the failure.

Key	Label	Description
Reg15=X'01'		A loop is detected in the DMHR chain.
Reg15=X'05'		An unexpected error occurred in an authorized page-fix or page-free call.

*Possible cause:* Internal program logic error

| *For diagnostic documentation:* z/OS console sheet, IMS control region abend memory dump, and log data set printout.

**For DBFINI20:**

A critical condition that should not occur was detected in DBFINI20 while performing the load of the user hash module.

1009 is a standard abend issued from DBFINI20. The content of register 15 indicates the reason for the failure.

Key	Label	Description
Reg15=X'0D'		Unable to load the user hash module. Nonzero return code was passed from the lower modules of DFSQCSS FUNC=LOAD

*Possible cause:* IMS system definition for the user hash module has an error.

**For DBFINTE0:**

A critical condition that should not occur was detected in DBFINTE0 during an IFP region initialization and termination.

1009 is a standard abend issued from DBFINTE0. The content of subcode in register 15 indicates the reason for the failure.

Key	Label	Description
Reg15=X'01'		Unable to locate a proper PST verify table entry in ESCD.
Reg15=X'02'		Unable to locate the current PST entry from the PST verify table in ESCD.
Reg15=X'03'		Unable to locate a proper load and balancing group entry (BALG) in ESCD.
Reg15=X'04'		The first RCTE was not found on the alphabetic chain.

*Possible cause:* Internal program logic error

*For APAR processing:* z/OS console sheet, IMS control region abend memory dump, and log data set printout.

**For DBFLRLS0:**

A DEDB resource that is to be freed cannot be found on the DMAC chain.

This is a standard abend issued by module DBFLRLS0.

Key	Label	Description
Reg15=X'18'		A DEDB resource that is to be freed cannot be found on the DMAC chain.

**For DBFPUXR0:**

A DEDB resource that is to be freed cannot be found on the DMAC or EPST chain.

This is a standard abend issued by module DBFPUXR0.

Key	Label	Description
Reg15=X'17'		A DEDB resource that is to be freed cannot be found on the DMAC chain after all associated XCRBs for the UXRb have been freed.
Reg15=X'18'		A DEDB resource that is to be freed cannot be found on the EPST chain after all associated XCRBs for the UXRb have been freed.

**For DBFXPIX0:**

A DEDB resource is to be freed, but cannot be found on the DMAC chain.

1009 is a standard abend issued by module DBFXPIX0. Registers contain the following information:

- Register 3 = DMAC-chain anchor-point address
- Register 4 = Address of UXRb
- Register 5 = Resource ID (RBA)
- Register 7 = Address of XCRb to be freed
- Register 8 = Address of DMAC

## 1010

Key	Label	Description
Reg15=X'10'		Module DBFXPIX0 was entered to free DEDB resources, but the resources cannot be located on the DMAC chain.
Reg15=X'11'		Internal error: DBFXPIX0 cannot find a UXRБ in the UXRБ DMAC chain when all associated XCRBs for the UXRБ have been freed.
Reg15=X'12'		Internal error: DBFXPIX0 was called to free a UXRБ with a UOW lock not owned by OTHREАD.

*Possible cause:* Internal program logic error

*For APAR processing:* IMS control region memory dump

**For DFSDBLM0, DFSDBLP0, DFSDLBL0, DFSSABN0, DFSSMSC0, DFSCST00:**

A critical condition that should not occur was detected while an exit routine to the Fast Path routine was attempted.

1009 is a standard abend issued from one of the modules listed above. The content of register 15 indicates the reason for the failure.

Key	Label	Description
Reg15=X'07'		Module DBFINTE0 was not included in the IMS nucleus.
Reg15=X'08'		Module DBFUDLB0 was not included in module DFSDLBL0.
Reg15=X'09'		Module DBFDBAC0 was not included in module DFSCST00.

*Possible cause:* Internal IMS system definition error

*For APAR processing:* IMS control/dependent region abend memory dump and IMS system definition STAGE 1 output.

**For DFSPCC20:**

IFP was specified on the EXEC statement in the PARM field, but Fast Path was not generated in this IMS system. This is a pseudoabend issued by module DFSPCC20.

*Possible cause:* IFP specified when Fast Path was not generated in this IMS system.

*For APAR processing:* IMS control region memory dump

**System action:** The IMS control region or the dependent region terminates abnormally.

**Problem determination:** 1, 4, 5, 35

**Source:** IMS abends

**Module:** DBFHIEL0, DBFHQMI0, DBFIBUF0, DBFINI20, DBFINTE0, DBFLRLS0, DBFPUXR0, DBFXPIX0, DFSASK00, DFSDBLM0, DFSDBLP0, DFSDLBL0, DFSPCC20, DFSSABN0, DFSSMSC0.

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

## 1010

**Explanation:** DBFDBIL0 found the segment pointer in an ECNT entry had already been used for another segment. This is a system error.

**Analysis:** Only one terminal-related MSDB segment or non-related MSDB segment with terminal-related keys is valid per LTERM for each MSDB.

Key	Label	Description
Reg4=address of MSDB segment prefix Reg7=address of ECNT Reg9=address of BHDR	DBIL0270	The MSDB records, read from sequential data sets, reside in a DASD and are placed in a main storage pool area within DBFCONT0. During this process the abend is issued if an MSDB segment area associated with an ECNT is being used by another LTERM.

**Possible cause:** An MSDB database was created with duplicate keys.

**System action:** The IMS control region is terminated abnormally.

**Problem determination:** None.

**Source:** IMS abends

**Module:** DBFDBILO

## 1011

**Explanation:** IMS control region initialization failed because a Fast Path initialization error was encountered during IMS startup or restart.

For an explanation, refer to any error message that was issued before this abend occurred. The most likely cause is insufficient storage in CSA or ECSA. Insufficient contiguous storage space in ECSA (due to fragmentation), especially during restart, might also cause this abend to occur.

| **Analysis:** 1011 is a standard abend that is issued from DBFBPNIO, DBFBPNJ1, DBDBPNS0, DBFBPN50, DBFBPN80, DBFCHKP0, DBFDAC50, DBFICIR0, DBFICI10, DBFINI20, DBFINI21, DBFINI22, DBRINI2A, DBFINI23, DBFINI24, DBFINI25, DBFINI26, DBFINI27, DBFINI29, or DBFINI2R.

### | For DBFDAC50:

| DBFDAC50 attempted to obtain storage, but the storage request could not be satisfied. DEDB Alter is abnormally terminated with abend U1011. Free some storage and rerun DEDB Alter.

| **R2=** Reason (STOR)

| **R3=** RSN code (for service use)

### For DBFICIR0:

DBFICIR0 invokes lower-level modules to create Fast Path OTHREAD ITASKS and Fast Path ASYNCHRONOUS process ITASKS using the DFSCIR macro. The nonzero return code is passed back to DBFICIR0 from one of the lower-level modules. The PSW at entry-to-abend points to the instruction label ABEND, where the abend was issued. Register 14 in the abend SVRB is used as the KEY. Register 12 is the base register, and register 15 contains the return code.

Key	Label	Description
Reg1=completion code Reg14=BAL Reg15=DFSCIR FUNC=ITASK return code	ABEND2	DBFICIR0 is the IMS/Fast Path ITASKS initialization module. This module issues DFSCIR FUNC=ITASK to create Fast Path OTHREAD ITASKS and Fast Path ASYNCHRONOUS process ITASKS. The nonzero return code passed from this macro results in an abend.

### For DBFICI10:

DBFICI10 invokes lower-level modules to:

- Create communication router ITASKS, and error message router ITASKS.
- Release lock control ITASKS, notified allocation ITASKS, command process ITASKS, and subordinate command ITASKS.
- Create PST-EPST for common service.

## 1011

The nonzero return code is passed back to DBFICI10 from one of the lower-level modules. The PSW at entry-to-abend points to the instruction label ABEND, where the abend was issued. Register 14 in the abend SVRB is used as the KEY. Register 12 is the base register, and register 15 contains the return code.

Key	Label	Description
Reg1=completion code Register Reg14=BAL Register Reg15=DFSCIR FUNC=ITASK return code or FUNC=CWU return code Return codes in Reg15: -EPS if failure when creating EPST for common services for Fast Path. -PST if failure when creating PST. or Reg15=DFSBCB FUNC=GET return code	ABEND2	DBFICI10 invokes lower-level modules to create communication router ITASKS, error message router ITASKS, release lock control ITASKS, notified allocation ITASKS, command process ITASKS, and dependent command ITASKS and to create PST-EPST for the common service of Fast Path. The nonzero return code passed back to DBFICI10 from one of the lower-level modules results in an abend. The most likely cause is an out of storage condition with CSA.

For DBFBPN10, DBFBPNJ1, DBDBPNS0, DBFBPN50, DBFBPN80, DBFCHKP0, DBFICIR0, DBFICI10, DBFINI20, DBFINI21, DBFINI22, DBRINI2A, DBFINI23, DBFINI24, DBFINI25, DBFINI26, DBFINI27, DBFINI29, or DBFINI2R:

During Fast Path initialization, the DBFINI2x modules obtain storage for various control blocks, load modules and work areas, page fix storage, and reserve pool space. A failure in any of these requests causes IMS to issue abend 1011. In many cases, an associated message is also issued containing additional information.

When an out of storage condition occurs, you can stop other jobs, which in turn might release whatever was preventing DBFINI2x from obtaining the necessary storage. If this tactic fails, you might have to IPL the system.

For page fix failures, you must free up PAGEABLE storage. The other abends are IMS internal errors. You might need to contact IBM Support for assistance.

One of the DBFINI2x modules (DBFINI20, DBFINI27, DBFINI29, DBFINI2R, or DBFBPN80) performs Fast Path initialization. Before the abend, one of the following IMS messages might be issued to provide additional information regarding the failure: DFS2528I, DFS2559I, DFS2627I, DFS2703A, DFS2706A, DFS2707A, DFS2707I, DFS2709W, DFS2710W, DFS2711A, DFS2779I, DFS2840W, DFS3300W, or DFS3704I.

At the time of abend, use the information in the following table to help diagnose the problem:



Register	Content description																
Register 2	<p>Contains one of the following descriptions of the resource being manipulated at the time of the abend:</p> <p><b>AWE</b> Asynchronous work element  <b>AREL</b> Area list  <b>BCNT</b> Module DBFBCNT0  <b>BPN00</b> Unable to load DBFBPN00  <b>BPNK0</b> Unable to load DBFBPNK0  <b>BPNI0</b> Unable to load DBFBPNI0  <b>BPNO0</b>  Unable to load DBFBPNO0  <b>BSIZ</b> BSIZ parameter  <b>CON1</b> DBFCONT1, the storage containing the MSDBs, and ECNTs  <b>CON3</b> DBFCONT3, the storage containing the DBHRs, and BUFFERS  <b>CON4</b> DBFCONT4, the storage containing the DEDB blocks  <b>CON5</b> DBFCONT5, the storage containing the OTHREADs  <b>CON6</b> DBFCONT6, the storage containing the BALGs  <b>CON7</b> DBFCONT7, the storage containing internally used buffers  <b>DMB</b> DBFDMB, the storage used as the DEDB I/O area for ACBLIB reads  <b>DT11</b> Module DBFDT110  <b>DT15</b> Module DBFDT150  <b>FRBM</b> FRBMSDNM, the XRF MSDB work area  <b>HAGU</b> Module DBFHAGU0  <b>HASH</b> Your hash routine  <b>IFIX</b> Module DBFIFIX0  <b>L56X</b> The L56X control block  <b>MMIO</b> Module DBFMMIO0  <b>MMPR</b> Module DBFMMPR0  <b>MSDS</b> DBFMSDSN, the storage used as the MSDB I/O are for ACBLIB reads  <b>64BC</b> 64 Bit Common Storage  <b>64BP</b> 64 Bit Private Storage</p>																
Register 3	<p>Contains one of the following codes that describe the service call that failed:</p> <table border="1"> <thead> <tr> <th>Code</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>04</td> <td>GETMAIN failure. See register 8 for the amount of storage requested</td> </tr> <tr> <td>08</td> <td>Module locate or load failed.</td> </tr> <tr> <td>12</td> <td>Module not found in the expected load module.</td> </tr> <tr> <td>16</td> <td>Page fix error.</td> </tr> <tr> <td>20</td> <td>Miscellaneous errors. See associated messages for descriptions of the error.</td> </tr> <tr> <td>24</td> <td>BCB get failed.</td> </tr> <tr> <td>28</td> <td>Unable to obtain enough 64-bit contiguous storage to satisfy the storage request.</td> </tr> </tbody> </table>	Code	Description	04	GETMAIN failure. See register 8 for the amount of storage requested	08	Module locate or load failed.	12	Module not found in the expected load module.	16	Page fix error.	20	Miscellaneous errors. See associated messages for descriptions of the error.	24	BCB get failed.	28	Unable to obtain enough 64-bit contiguous storage to satisfy the storage request.
Code	Description																
04	GETMAIN failure. See register 8 for the amount of storage requested																
08	Module locate or load failed.																
12	Module not found in the expected load module.																
16	Page fix error.																
20	Miscellaneous errors. See associated messages for descriptions of the error.																
24	BCB get failed.																
28	Unable to obtain enough 64-bit contiguous storage to satisfy the storage request.																
Register 8	The amount of storage requested (applicable when register 3=00000004).																
Register 15	The nonzero return code from the service call that caused the abend.																

**For DBFSYP20:**

DBFSYP20 attempts to obtain a system buffer for sequential dependent segment (SDEP). If the request fails after 10 times, the application abnormally ends with abend U1011.

**Possible cause:** The possible causes for abend 1011 to occur might be one of the following situations:

- An error in the REGION parameter or the BSIZ/DBBF operand in the PARM field on the EXEC job control statement of the IMS control region startup procedure
- An out of storage condition
- Failure to locate or load modules
- Storage page fix failed
- Block get failed

**System action:** The IMS control region is terminated abnormally.

**Programmer response:** Determine the cause of failure as indicated by error messages and dump analysis. If the

## 1012 • 1013

abend occurred because of insufficient storage, increase CSA storage allocations.

**Problem determination:** 3

**Source:** IMS abends

**Module:** DBFBPN10, DBFBPNJ1, DBDBPNS0, DBFBPN50, DBFBPN80, DBFCHKP0, DBFDAC50, DBFICIR0,  
DBFICI10, DBFINI20, DBFINI21, DBFINI22, DBRINI2A, DBFINI23, DBFINI24, DBFINI25, DBFINI26, DBFINI27,  
DBFINI29, DBFINI2R, DBFSYP20

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

### 1012

**Explanation:** IMS control region initialization failed because an error was encountered during Fast Path MSDB load determination process. Fast Path MSDB load determination process failed. For an explanation, refer to any error message issued before this abend occurred.

- If MSDBABND=Y is the first statement of the DBFMSDBx member of IMS.PROCLIB, this abend is issued when no MSDBs can be loaded due to errors in the MSDBINIT data set.
- If MSDBABND=C is the first statement of the DBFMSDBx member of IMS.PROCLIB, this abend is issued when an error occurs during the writing of the MSDBs to the MSDBCPN data set in the initial checkpoint after IMS startup.
- If MSDBABND=I is the first card in the DBFMSDBx member of IMS.PROCLIB, this abend is issued when an error occurs during the initial load of the MSDBs from the MSDBINIT data set as described in option 'Y' above, or there are no segments in the MSDBINIT data set for one or more defined MSDBs.
- If MSDBABND=A is the first card of the DBFMSDBx member of IMS.PROCLIB, this abend is issued as described in options 'C' and 'T' above.
- If MSDBABND=B is the first statement of the DBFMSDBx member of IMS.PROCLIB, this abend is issued as described in options 'Y' and 'C' above.

**Analysis:** The nonzero return code is passed back to DBFXFP10 from DBFINI10 or DBFDDBL0 issues the abend if DBFINI10 has set the ESCDMSDA bit.

The PSW at entry-to-abend points to the instruction label ABEND, where the abend was issued. Register 14 in the abend SVRB is used as the KEY. Register 12 is the base register, and register 15 contains the return code.

**Possible cause:** The causes include:

- MSDB table entries specified in DBFMSDBn where n is a suffix for MSDB member in IMS.PROCLIB.
- A change to the ACBLIB followed by a warm start.

**System action:** The IMS control region is terminated abnormally.

**Problem determination:** 3

**Source:** IMS abends

**Module:** DBFXFP10, DBFDDBL0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

### 1013

**Explanation:** During dependent region create thread (scheduling call), program DBFCPY00 detected that a PCB referenced more segments than were included in the associated DEEDB's DBD. DBFCPY uses the count of segments from the DMCB (DMCBSGMR) to control a loop that attempts to match a PCB's SDB to a SMLTE using segment name.

**Analysis:** An inconsistency exists between the PSB and the DEEDB's DBD. The DBD was changed on DBDLIB, and was used in ACBGEN build PSB processing. A warning was issued from ACBGEN that the DBD was not replaced on ACBLIB. For a DBCTL thread, message DFS0526A is also issued.

Key	Label	Description
Reg14 = address of DMCB		DMCB+8 = DBDNAME for the DBD which has not been rebuilt.

**System action:** The dependent region is terminated abnormally.

**Programmer response:** There is an inconsistency between the PSB and the DBD. The DBD has been changed on DBDLIB. When ACBGEN was run, the DBD was not rebuilt because BUILD PSB was specified, but BUILD DBD was not.

**Source:** IMS abends

**Module:** DBFCPY00

**Related information:**

 DFS0526A

## 1014

**Explanation:** Module DBFXSL30 attempted to deactivate the dependent region using the SUSPEND macro. Register 4 contains the return code.

**Analysis:** This is a standard abend issued by DBFXSL30. Register 4 contains the return code.

Key	Label	Description
Reg4=X'01'		When attempting to deactivate a dependent region, it was found that more than one previous attempt to activate the dependent region was made.
Reg4=X'02'		An attempt was made to deactivate a dependent region when it was already deactivated.
Reg4=X'03'		Program DBFXSL30 was unable to get the LOCAL LOCK.
Reg10=address of EPST		EPST in error.

**Possible cause:** Internal program logic error

**System action:** The Fast Path dependent region terminates abnormally.

**Problem determination:** The contents of register 4, together with a storage dump, should be enough to locate the final cause of the abend.

**Source:** IMS abends

**Module:** DBFXSL30

## 1015

**Explanation:** The IMSAUTH SVC encountered a problem while trying to page fix or page free storage. Register 2 contains FIX or FREE. Register 15 contains the IMSAUTH PGFIX/PGFREE return code. For a description of these codes, see the IMS system services return codes information.

**Analysis:** 1015 is a standard abend issued by DBFCSTS0. Register 2 contains 'FIX'. Register 15 contains the return code.

Register 5 contains the address of the page fix/free list and register 11 contains the address of the SCD.

Key	Label	Description
Reg2=CL4'FIX'		Page-fix operation has failed.

**Possible cause:** Internal program logic or interface error

**For DBFDBDL0:**

The MSDB image copy load module, DBFDBDL0, received a bad return code from the page-fix/free module, DFSV4200, which was invoked by DBFDBDL0 using the IMSAUTH macro.

## 1016

This is a standard abend issued by DBFDL0. Register 2 contains 'FIX' or 'FREE'. Register 15 contains the return code.

Register 5 contains the address of the page fix/free list and register 14 contains the address of the SCD.

Key	Label	Description
Reg2=CL4'FIX'	DBDLFIX	Page-fix operation has failed.
Reg2=CL4'FREE'	DBDLFREE	Page-free operation has failed.

*Possible cause:* Internal program logic or interface error

### For DBFMFLG0:

The IMS Fast Path DEDB set flags module, DBFMFLG0, received a bad return code from the page-fix/free module, DFSV4200, which was invoked by DBFMFLG0 using IMSAUTH macro.

This is a standard abend issued by DFMFLG0. Register 2 contains 'FIX'. Register 15 contains the return code.

Register 5 contains the address of the page fix/free list and register 6 contains the address of the SCD.

Key	Label	Description
Reg2=CL4'FIX'		Page-fix operation has failed.

*Possible cause:* Internal program logic or interface error


**System action:** The IMS control region is terminated abnormally.

**Problem determination:** 4, 35

**Source:** IMS abends

**Module:** DBFCSTS0, DBFDL0, DBFMFLG0

**Related reference:**

 [IMS system services return codes \(Messages and Codes\)](#)

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

## 1016

**Explanation:** The MSDB image copy load module, DBFDL0, has found a mismatch between the checkpoint logical terminal name and the system logical terminal name. (ECNTNAME for checkpoint and system not equal.) The ECNT names in the MSDB checkpoint and in the IMS system do not match because of a control block change.

**Analysis:** This is a standard abend issued by DBFDL0. The checkpoint ECNT can be in one of two places: (1) The input area, pointed to by register 1, (2) the saved area, pointed to by register 7. Register 10 points to the ECNT of the system.

Key	Label	Description
Reg0=End address of this ECNT AREAENDA=Area end address	DBDLPROC	If register 0 is not higher than AREAENDA, the mismatched checkpoint ECNT is in the input area pointed to by register 1. If register 0 is higher than AREAENDA, it is in a saved area pointed to by register 7.

*Possible cause:* A control block change has taken place

**System action:** The IMS control region is terminated abnormally.

**Programmer response:** Restart IMS with the appropriate system.

**Source:** IMS abends

**Module:** DBFDBDL0

### 1017

**Explanation:** The MSDB image copy load module, DBFDBDL0, attempted to convert a virtual to a real address using the load real address (LRA) instruction, encountered an error, , set a nonzero condition code. Register 15 contains a return code.

**Analysis:** This is a standard abend issued by DBFDBDL0. Register 15 contains one of the following return codes:

**Code    Meaning**

X'01'    The CCW area address could not be converted.

X'02'    The input area address could not be converted.

X'03'    The IDAW address could not be converted to a real address.

Key	Label	Description
Reg15=X'01'	DBDLSRD	The CCW area address could not be converted.
Reg15=X'02'	DBDLMRD DBDLSRD	The input area address could not be converted.
Reg15=X'03'	DBDLMRD	The IDAW address could not be converted to a real address.

**Possible cause:** A hardware error, an internal program logic error, or interface error occurred.

**Problem determination:** 35

**Source:** IMS abends

**Module:** DBFDBDL0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

### 1018

**Explanation:** An error occurred in sequential dependent (SDEP) processing during a resynchronization commit request.

**Analysis:** The EPSTUDFI anchor in the extended partition specification table (EPST) contains a queue of sequential dependent update records that are being committed. Module DBFDSRP0 copies these records into the SDEP buffer. The search for the SDEP buffer begins at field DMACXNYW in the DEDB area control list (DMAC) and ends at DMACLYNW. The RBA in the sequential dependent update record should be in the DMACXNYW chain anchor. If it is not in the chain, the module abends with 1018.

Key	Label	Description
Reg1		Address of DMACXNYW, the chain of SDEP buffers
Reg6		Address of LRST, the SDEP update record that DBFDSRP0 is trying to insert
Reg8		DMAC address for this sequential dependent update record
Reg9		Address of current SDEP buffer chain from DMACXNYW

**System action:** The IMS control region terminates abnormally.

**Problem determination:** 1, 4, 5, 6, 15, 36

**Source:** IMS abends

**Module:** DBFDSRP0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

**1019**

**Explanation:** An invalid call was made to the Fast Path Queue Manager. This abend is caused by an internal program logic or interface error.

**Analysis:** Register 3 contains the address of the branch table for ABEND routines. Abend 1019 occurs only when register 3 content is between 20 and 60 or greater than 68.

**Possible cause:** Internal program logic or interface error

**System action:** IMS terminates abnormally.

**Programmer response:** Correct the call interface or contact your IBM Support Center for assistance.

**Problem determination:** 4, 5, 11, 24

**Source:** IMS abends

**Module:** DBFHQMI0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

**1020**

**Explanation:** The Fast Path asynchronous task (or for non-z/OS systems, one of the OTHREAD tasks) has terminated abnormally, and this abend is issued to stop the rest of the IMS system.

**Analysis:** This is a standard abend issued by DBFFATI0. Register 3 contains the address of the TCB that has terminated abnormally.

Key	Label	Description
Reg3=TCB address	FPTCBXIT	This routine is in the ATTACH exit routine of DBFFATI0. If, on return from the asynchronous task, the completion code is nonzero, this abend is issued.

**Possible cause:** Internal program logic or interface error

**System action:** IMS terminates abnormally.

**Operator response:** Contact the IMS system programmer.

**Programmer response:** Correct the problem, and rerun the job.

**Source:** IMS abends

**Module:** DBFFATI0

---

**1021**

| **Explanation:** For modules DBFMGAP0 and DBFMGAP1:

Invalid data was returned from a DEDB randomizing module. One of these problems occurred:

- Register 0 is not a valid anchor point offset within the DEDB.
- The return code from the randomizing module (register 15) was not equal to 0 or 4.
- At time of abend, Register 1 does not contain a valid address. (Register 1 is stored in Register 2 for debugging purposes.)

| For modules DBFDADC0 and DBFDADCS:

| While executing the DEDB Alter utility, ALTER AREA, it was detected that the old and new randomizer returned different AREAs for the same key. It is not allowed to randomize to different AREAs with the same KEY.

| While the DEDB Alter utility was running, the shadow DEDB randomizer returned an AREA (DMAC) which was different from the AREA being altered.

| **Analysis:** This abend is a standard abend that is issued by DBFMGAP0, DBFMGAP1, DBFDADC0, and DBFDADCS. The randomizing module is typically a user-supplied module. If the user did not supply such a module, one is available from the system. The system sample is named DBFHDC40.

| The following keys correspond to modules DBFMGAP0 and DBFMGAP1.

Key	Description
Reg3, DMACFOVF	Register 3 is greater than or equal to the contents of DMACFOVF.
Reg15	Register 15 is not equal to 0 or 4.
Reg3, Reg8	Register 3 contains returned DMAC's pointer to the DMCB. Register 8 contains the DMCB address. Register 3 is not equal to register 8.
Reg6	Register 6 at time of abend contains a reason code:
	01 An AREA for a different DEDB was returned by the randomizer.
	02 Should not occur error while attempting to find next HSSP AREA.
	03 Randomized to a CI outside the RAP section of the AREA.
	04 Randomizer returned return code not equal to 0 or 4.

| The following keys correspond to modules DBFDADC0 and DBFDADCS.

Key	Description
Reg 7	Register 7 points to the ACTIVE (current) DMAC.
Reg 8	Register 8 points to the SHADOW DMAC.

Locate the save area chain at abend, then locate the save area two higher than the current save area. This save area has 'c'rand' in the RET (R14) slot. The remaining registers, except for R5, are those returned from the randomizing routine. You can use these registers, along with the information about the Data Entry Database Randomizing exit routine in *IMS Version 13 Exit Routines*, to determine the cause of the problem in the randomizing routine.

**Possible cause:**

For modules DBFMGAP0 and DBFMGAP1: Randomizing routine error

| For modules DBFDADC0 and DBFDADCS: The new RANDOMIZER in the SHADOW DMCB from the Staging ACBLIB was changed in an invalid manner.

**System action:** The Fast Path dependent region terminates abnormally.

**Programmer response:** For modules DBFMGAP0 and DBFMGAP1: Correct the error in the randomizing module.

| For module DBFDADCS: Correct the randomizer and rerun the DEDB Alter utility.

**Source:** IMS abends

| **Module:** DBFMGAP0, DBFMGAP1, DBFDADC0, DBFDADCS

## 1022

**Explanation:** A LOCESCD macro failed to locate the Fast Path SCD extension identified by ESCD or an IMODULE LOAD for a Fast Path module failed. Register 2 contains the identification used by the macro. All SCD extensions are queued from SCDESCDQ. This abend is issued from the Fast Path modules when the LOCESCD macro fails.

**Analysis:** This abend is issued by Fast Path and non-Fast Path modules. The SCD address should be verified and the SCDESCDQ field in the SCD should be verified.

Key	Label	Description
Reg2=CL4'ESCD'		Fast Path SCD Extension SCD

## 1023 • 1024

**Possible cause:** An error in the link or generation of the IMS system with Fast Path. Check whether the FP=Y parameter is missing, which activated IMSFP for this IMS subsystem.

**System action:** IMS terminates the region abnormally.

**Programmer response:** Probable binder errors in the system definition process or the SCD indicate that the Fast Path is installed erroneously. An IMODULE LOAD failure is probably due to insufficient storage available.

**Problem determination:** 35

**Source:** IMS abends

**Module:** DBFCNDAR0, DBFCDDDB0, DBFCDPS0, DBFCDR0, DBFCHKP0, DBFCPID0, DBFCPRC0, DBFCSTS0, DBFCST00, DBFBAU0, DBFEACL0, DBFENIS0, DBFENOT0, DBFEPSB0, DBFERST0, DBFERS20, DBFATC0, DBFFOR10, DBFHIEL0, DBFQMI0, DBFHRTR0, DBFHTMG0, DBFICIR0, DBFICI10, DBFICLI0, DBFICL20, DBFICL40, DBFINI10, DBFINI20, DBFINI30, DBFINI40, DBFINTE0, DBFIRC10, DBFLHCK0, DBFLIRL0, DBFLRH00, DBFNALC0, DBFNCBS0, DBFNOTM0, DBFNOTX0, DBFNRS0, DBFSBLK0, DBFTERM0, DBFTOFN0, DBFTOPU0, DBFTORS0, DBFXFP10, DFSMINI0, DFSSTB0, DFSXNCL0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

### 1023

**Explanation:** The Fast Path message input edit routine detected that the exit routine had accepted, and moved into the EMHB buffer, an input message that was larger than the maximum length defined in the TERMINAL macro in the IMS system definition.

**Explanation:** Module DBFHIEL0 detected from the message prefix that the message length was greater than that in the EMHB, EMHBMXLN. EMHBMXLN is developed from the buffer size specified on the FPBUF parameter in the TERMINAL macro.

The first two positions in the input message have a value greater than that in EMHBMXLN.

- Register 4=address of EMHB
- Register 5=address of input message buffer

**Possible cause:** The causes include:

- The buffer size parameter on the FPBUF keyword in the TERMINAL macro is too small for the issuing terminal.
- The input message from the terminal is too long.

**System action:** IMS terminates abnormally.

**System programmer response:** Change the user input edit routine to compare the EMHBMXLN field to the input message length and set a return code of X'1C' if the message length is greater than the EMHBMXLN field. Do not move the message to the EMHB buffer.

**Problem determination:** 4, 10, 12, 35

**Source:** IMS abends

**Module:** DBFHIEL0

---

### 1024

**Explanation:** The Fast Path application program issued a GU call to the I/O PCB after receiving a QC status code on the previous GU call.

**Analysis:** The application program should terminate on receiving the QC status code. Correct the program and rerun.

**Possible cause:** Application program error

**System action:** IMS terminates the message-driven application program abnormally with a pseudoabend.

**Programmer response:** Return to the program controller after a QC status code has been issued instead of issuing another GU call to the I/O PCB.

**Source:** IMS abends



**Module:** DBFHGU10, DFSECP10, DFSISI00

## 1025

**Explanation:** Module DBFIRC10 detected an invalid return code after the DL/I call was processed by the Fast Path call analyzer.

**Analysis:** For DBFHSYN0:

Module DBFHSYN0 detected an invalid return code after the DL/I call was processed by the Fast Path call analyzer.

This is a pseudoabend issued by DBFHSYN0. Only a return code of 0, 4, 8, 12, or 16 is valid on return from the Fast Path Call Analyzer. The 1025 abend routine is a common routine. Register 8 indicates the calling routine as shown in the following table:

Key	Label	Description
Reg5=RC Reg8=address of the instruction that detected the error		Routine calling abend. Return code from Fast Path Call Analyzer.

**For DBFIRC10:**

Program DBFIRC10 detected an invalid return code after the DL/I call was processed by the Fast Path call analyzer.

Only a return code of 0, 4, 8, 12, or 16 is valid on return from the Fast Path Call Analyzer. The 1025 abend routine is a common routine. Register 8 indicates the calling routine as shown in the following table:

Key	Label	Description
Reg7=RC Reg8=address of the instruction that detected the error		Routine calling abend. Return code from Fast Path Call Analyzer.

**Possible cause:** Internal program error

**System action:** IMS terminates the Fast Path application program abnormally.

**Programmer response:** Proceed with the Problem Determination.

**Problem determination:** 1, 4, 5, 35

**Source:** IMS abends

**Module:** DBFHSYN0, DBFIRC10

**Related information:**

Chapter 2, "IMS problem determination," on page 13

## 1026

**Explanation:** An IMS Fast Path module has detected a problem condition and has issued the DBFDEBUG macro to describe the condition.

**Analysis:** 1026 is a standard abend issued by several modules.

If the abend is a hard abend, no DBFDEBUG call is made, and thus no DFS2712I message is generated. The registers at the time of the abend are captured in the RTM2WA as with all hard abends.

If the abend is located in the DBFDEBUG macro, the module names and the reason for the abend are displayed by the DBFDEBUG macro.

If the abend is located in the DBFDEBUG macro, see message DFS2712I issued before the abend for the name of the module, the register contents, and (depending on the module) a numeric code in a register. Here is an example of the information that can be displayed by message DFS2712I:

```
DFS2712I  MODULE NAME:  DBFMDLT0
DFS2712I  DEDB PROBABLY HAS ERROR IN IT
DFS2712I  ABEND SUBCODE:  54
```

## 1026

```
DFS2712I   R0-R3      00000006 00000054 0059B126 0061C85C
DFS2712I   R4-R7      0061C9C4 005DCBF4 00000000 00000000
DFS2712I   R8-R11    007FDC30 007AAD98 00927040 00973AC0
DFS2712I   R12-R15   005F6A50 009195E4 00000004 0061C8DC
```

If you want to pursue this problem through IBM Software Support, please retain the DFS2712I message, the dump (if any), the last good image copy, and all log records from the last good image copy up to the time of the abend.

### Subcode summary:

The first digit of the 2-digit hexadecimal subcode identifies the general type of error detected. The second digit identifies the type of error more specifically. The first occurrence of a subcode in a module is X'0xx', the second occurrence is X'1xx', the third occurrence is X'2xx', and so on. For example, the first invalid subcode would be X'033', the second X'133', the third X'233', and so on, as long as subcode X'33' is needed. The subcodes have the following meanings:

#### Subcode (hex)

	Meaning
1x	Interface error (invalid parameter supplied)
10	Zero value supplied, but nonzero value required
11	Invalid type or function code supplied
12	Invalid length or offset supplied
13	Rule broken by caller (for example, caller trying to get a latch it already holds)
14	Caller passed invalid parameters
2x	Invalid control block contents
20	Zero pointer found, but nonzero pointer required
21	Invalid pointer
22	Incorrect control block type
23	Unused
24	Item not found on chain
25	Invalid level encountered
27	DBFSLGE0 detected an invalid data alteration length.
3x	Unexpected return codes
30	Return code from ISWITCH
31	Return code from GETMAIN
32	Return code from IMSAUTH PGFIX
33	General return code error
34	Unexpected status code
5x	Invalid DEDB contents
50	Invalid segment code of 0
51	Incorrect block type for this CI
52	Unable to classify segment code
53	Segment code does not match expected value
54	Segment not found on chain
55	CI contents cannot be scanned (length incorrect)
56	Invalid FSE
57	RBA outside valid range
58	Invalid control CI for independent overflow
59	Invalid segment length
5A	An invalid segment code or pointer error was detected under PROCOPT=GO; PROCOPT was expecting GOx, but received something else.
6x	Invalid MSDB contents
61	Attempt to dequeue bad segment
62	Attempt to enqueue bad segment
7x	HSSP invalid condition detected
70	Number of HSSP buffers requested is greater than the number of HSSP buffers obtained
71	DL/I call is to an invalid area
72	DL/I call is to the wrong UOW
80	DENDRDF and CI RDF do not match.
81	DENDRRBA and DMHRSRBA do not match.

In the following sections, this information is given for each module:

- The DFS2712I message text (A module can issue several messages.)
- Subcodes
- Fields containing useful diagnostic information
- Register contents

For DBFARDC0, DBFCSTS0, DBFDT190, DBHCAS0, DBFSYN10, DBFSYN20, DBFMSRT0, DBFUHDSO, DBFVSOP0, DBFVXOIO:

Message text: DBFSYNL TYPE=RSHR LATCH NOT OWNED

Key	Label	Description
Subcode=X'13' or X'113'		An attempt was made to release the shared sync point latch when it was not owned.

Message text: DBFSYNL TYPE=GSHR ALREADY OWNED

Key	Label	Description
Subcode=X'13' or X'113'		An attempt was made to get shared sync point latch when the task already owned it.

Normal register usage is as follows:

#### Register

	Contents
0	Contents of ESCDSYNL field
1	Subcode
2	Address of ESCD
10	Address of EPST

For DBFCSTS0:

Message text: *no text supplied*

Key	Label	Description
Subcode=X'9D'		Mismatch between DENDRRBA and DMHRSRBA.

Normal register usage is as follows:

#### Register

	Contents
0	DENDRRBA value
3	DMHR address
15	Subcode

For DBFATRM0:

Message text: DBFSYNL TYPE=RSHR LATCH COUNT WAS ZERO

Key	Label	Description
Subcode=X'13' or X'113'		Attempt to release shared sync point latch when latch count is zero (ESCD SYN L).

Message text: DBFSYNL TYPE=RSHR LATCH NOT OWNED

## 1026

Key	Label	Description
Subcode=X'13' or X'113'		An attempt was made to release shared sync point latch when it was not owned.

Normal register usage is as follows:

### Register

	Contents
0	Contents of ESCDSYNL field
1	Subcode
2	Address of ESCD
10	Address of EPST - EPSTR1SL is set, Reg10 + X'B5' = X'80'

### For DBFBENQ0:

Message text: ATTEMPT TO DEQUEUE BAD MSDB RECORD

Key	Label	Description
Subcode=X'61'		An attempt was made to dequeue an invalid MSDB record.

Message text: ATTEMPT TO ENQUEUE BAD MSDB RECORD

Key	Label	Description
Subcode=X'62'		An attempt was made to enqueue an invalid MSDB record.

Normal register usage is as follows:

### Register

	Contents
1	Subcode
6	Address of prefix of segment with invalid decimal field
9	Address of MSNQLIST for this call

### For DBFBGET0:

Message text: ATTEMPT TO ENQUEUE BAD MSDB RECORD

Key	Label	Description
Subcode=X'62'		An attempt was made to enqueue an invalid MSDB record.

Normal register usage is as follows:

### Register

	Contents
1	Subcode
2	Address of prefix of segment with invalid decimal field
9	Address of MSNQLIST for this call

### For DBFCHKP0:

Message text: DMAC CONTENTS ARE INVALID

Key	Label	Description
Subcode=X'20'	CP841000	DMACRAID (DMAC+70) is bad, or DMACOUSZ (DMAC+78) is not equal to X'0078', or DMACHRAF (DMAC+7A) is not equal to X'0002', or DMACDMCB (DMAC+B8) does not point back to the DMCB, or the database name in DMACDBNM (DMAC+4) is not equal to DMCBNAME (DMCB+8).

**Register****Contents**

1	1026 abend subcode
2	Slot for DMAC in 4084 log record
3	4084 log record address
5	DMAC address
9	DMCB address
10	Restart EPST address
11	ESCD address
12	CHKPDMCB subroutine base
15	Area number within database

**For DBFDAC50:**

Message text: BAD RBA

Key	Label	Description
Subcode=X'57'		Unable to find the buffer that contains a segment's RBA.
Reg7=Desired starting RBA		

Normal register usage is as follows:

**Register****Contents**

1	Subcode
2	DAC1 address
3	DAC2 address
8	DMCB address
9	DMHR address
10	EPST address
11	ESCD address

Message text: UNEXPECTED SEGMENT CODE ENCOUNTERED

Key	Label	Description
Subcode=X'53'		Current segment code does not match segment code at the current buffer location.
Reg6=Segment code at current buffer location		
Reg7=Current segment code		
Subcode=X'153'		Current segment code does not match the segment code in the SDBF.
Reg4=SDBF address		
Reg7=Current segment code		
Reg14=Segment code in SDBF		

Normal register usage is as follows:

## 1026

<b>Register</b>	
	<b>Contents</b>
1	Subcode
2	DAC1 address
3	DAC2 address
8	DMCB address
10	EPST address
11	ESCD address

Message text: UNABLE TO GET IOVF BUFFER

Key	Label	Description
Subcode=X'43'	Reg7=Starting RBA for IOVF CI	Unable to get IOVF buffer. Increasing the BUFNO value in your DEDB Alter job might fix this problem.

Normal register usage is as follows:

<b>Register</b>	
	<b>Contents</b>
1	Subcode
3	DBPCB address
8	DMAC address
10	EPST address
11	ESCD address

For DBFDRSC0:

Message text: DBFSYNL TYPE=RSHR LATCH COUNT WAS ZERO

Key	Label	Description
Subcode=X'13'		Attempt to release shared sync point latch when latch count is zero (ESCDSYNL).

Message text: DBFSYNL TYPE=RSHR LATCH NOT OWNED

Key	Label	Description
Subcode=X'13' or X'113'		An attempt was made to release shared sync point latch when it was not owned.

Message text: DBFSYNL TYPE=GSHR ALREADY OWNED

Key	Label	Description
Subcode=X'13' or X'113'		An attempt was made to get shared sync point latch when the task already owned it.

Normal register usage is as follows:

<b>Register</b>	
	<b>Contents</b>
0	Contents of ESCDSYNL field
1	Subcode
2	Address of ESCD
10	Address of EPST

For DBFMCCV9:

Message text: LOGICAL ERROR IN THIS MODULE

Key	Label	Description
Subcode=X'11'		The first character in the EPSTSWAR (Reg10 + X'320') must be either a blank (X'40') or an opening parenthesis (X'4D'), but neither was found. The module does not expect to arrive at this location unless a blank or an opening parenthesis is found.

Normal register usage is as follows:

**Register**

	Contents
1	Subcode
3	EPCB address
4	MLTE address
6	Address of segment, DSEGCODE is first byte
10	EPST address
11	ESCD address

**For DBFMCRP9:**

Message text: UNEXPECTED SEGMENT CODE ENCOUNTERED

Key	Label	Description
Subcode=X'53'		The parent segment code does not match the segment code expected from the parent MLTE. DBFMPCGO0 was called and issued return code X'24'. DMHRBUFP (Reg9 + X'10') contains the address of the buffer. The abend is issued from macro DBFMRCPS.

Normal register usage is as follows:

**Register**

	Contents
1	Subcode
2	Address of parent MLTE
3	EPCB address
4	MLTE address
6	Offset to segment in buffer
7	Address of parent segment, DSEGCODE is first byte
8	Address of parent MLTESGCD
9	DMHR address
10	EPST address
11	ESCD address

**For DBFMCSS9:**

Message text: UNEXPECTED SEGMENT CODE

Key	Label	Description
Subcode=X'53'	DEBUG	The segment code for the retrieved segment does not agree with the code defined for the segment type.

Normal register usage is as follows:

**Register**

	Contents
0	Offset in buffer to segment
1	Subcode

## 1026

- 3 EPCB address (EPCB + X'48' contains DMAC address)
- 4 MLTE address
- 5 ESCD address
- 8 Address of parent MLTESGCD
- 9 DMHR address (DMHR + X'10' is the buffer address)

### For DBFMCTL0:

Message text: LOGIC ERROR IN DBFMCTL0

Key	Label	Description
Subcode=X'11'	LOGERROR	An invalid function type was detected. A Get Next (GN) function, X'40', was expected.
Subcode=X'33'	LOGERROR	The RCR0 value returned from DBFMSSA9 in register 1 was not 0, 4, 8, or 12, as expected.
Subcode=X'53' Reg2=address of segment, DSEGCODE is first byte Reg6=offset in buffer to segment Reg7=RBA of block containing segment Reg8=MLTESGCD (pointed to by Reg4 + X'1E')	DEBUG	MLTESGCD was not equal to DSEGCODE, pointed to by register 2. DBFMPTG00 was called, and the abend was issued from macro DBFMRCSS. DMHRBUFP (Reg9 + X'10') is the address of the buffer.

Normal register usage is as follows:

### Register

#### Contents

- 1 Subcode
- 3 EPCB address
- 4 MLTE address
- 9 DMHR address
- 10 EPST address
- 11 ESCD address

### For DBFMDLT0:

Message text: DEDB PROBABLY HAS ERROR IN IT

Key	Label	Description
Subcode=X'054' Reg15=address of MLTE for previous segment		The previous segment is a parent and does not match the expected segment code.
Subcode=X'154'		The previous segment is a twin and does not match the expected segment code.
Subcode=X'254'		The previous segment is a root and does not match the expected segment code.
Subcode=X'354'		The previous segment is a root and the physical child first (PCF) does not match the pointer to the segment.
Subcode=X'454'		The previous segment is a twin and the physical twin forward (PTF) does not match the pointer to the segment.

Normal register usage is as follows:

### Register

#### Contents

- 0 Length of data to be dumped (6 bytes)



- 1 Subcode
- 2 Address of MLTE for segment to be deleted
- 3 EPCB address
- 8 Address of segment, DSEGCODE is first byte
- 9 DMHR address
- 10 EPST address
- 11 ESCD address

**For DBFMDPT9:**

Message text: LOGICAL ERROR IN ACTION MODULES

Key	Label	Description
Subcode=X'20' Reg0=MLTENRBA, address of the next segment	LOGERROR	The forward pointer in MLTENRBA is 0.

Message text: UNEXPECTED SEGMENT CODE ENCOUNTERED

Key	Label	Description
Subcode=X'53' Reg8=MLTECLOC, address of the parent segment Reg14=MLTEPARP (Reg4 + X'3C'), address of the MLTE of the parent segment	ABND1026	MLTE segment code (Reg14 + X'1E') of the parent segment is not equal to the segment code of the segment pointed to by register 8. The abend is issued from macro DBFMRDPS.

Normal register usage is as follows:

**Register**

	Contents
1	Subcode
3	EPCB address
4	MLTE address
9	DMHR address
10	EPST address
11	ESCD address

**For DBFMDBR0:**

Message text: NO TEXT SUPPLIED

Key	Label	Description
Subcode=X'53'		The value in DSEGCODE (R7 + X'00') was X'01' indicating a root segment, but a different value was found.
Subcode=X'54'		Root pointer (Reg7) is 0, indicating the end of the chain, but the chain should not end.

Normal register usage is as follows:

**Register**

	Contents
1	Subcode
2	MLTE address
3	EPCB address
5	DFSE address
7	Address of segment, DSEGCODE is first byte

## 1026

8	DMAC address
9	DMHR address
10	EPST address
11	ESCD address
15	DMCB address

For DBFMDSG9:

Message text: UNEXPECTED SEGMENT CODE ENCOUNTERED

Key	Label	Description
Subcode=X'53'	Reg5=address of segment, DSEGCODE is first byte	The segment code pointed to by register 5 is not equal to the SDBF segment code in SDBSC (Reg4 + X'11'). The abend is issued from macro DBFMRDTS. DMHRBUF (Reg9 + X'10') contains the address of the buffer.
Subcode=X'153'	Reg2=address of child SDBF Reg6=address of segment, DSEGCODE is first byte	The segment code pointed to by register 6 is not equal to the child SDBF segment code (Reg2 + X'11'). The abend is issued from macro DBFMRDDS.

Normal register usage is as follows:

### Register

#### Contents

1	Subcode
3	EPCB address
4	SDBF address
9	DMHR address
10	EPST address
11	ESCD address

**Attention:** MLTE is not available in DBFMDSG9.

For DBFMFLG0, DBFSDEQ0, DBFSMP10:

Message text: DBFSYNL TYPE=RSHR LATCH COUNT WAS ZERO

Key	Label	Description
Subcode=X'13'		Attempt to release shared sync point latch when latch count is zero (ESCDSYNL).

Message text: DBFSYNL TYPE=RSHR LATCH NOT OWNED

Key	Label	Description
Subcode=X'13' or X'113'		An attempt was made to release shared sync point latch when it was not owned.

Message text: DBFSYNL TYPE=GSHR ALREADY OWNED

Key	Label	Description
Subcode=X'13' or X'113'		An attempt was made to get shared sync point latch when the task already owned it.

Normal register usage is as follows:

**Register**

	<b>Contents</b>
0	Contents of ESCDSYNL field
1	Subcode
2	Address of ESCD
10	Address of EPST

**For DBFMFSE0:**

Message text: LENGTH PARM (R7) INVALID

<b>Key</b>	<b>Label</b>	<b>Description</b>
Subcode=X'12' Reg2=MLTE address Reg3=EPCB address Reg5=DFSE address Reg7=length request in low-order 2 bytes	MFSEDLET	The length in register 0, which is calculated based on the length passed in register 7, is greater than the length in DMACBLKL.

Message text: FSE CHAIN IS BAD

<b>Key</b>	<b>Label</b>	<b>Description</b>
Subcode=X'12' Reg2=MLTE address Reg3=EPCB address Reg5=DFSE address Reg7=length request in low-order 2 bytes	FSEBAD	The length of the FSE is greater than allowed by the physical record length.
Subcode=X'112' Reg2=Address of MLTE Reg5=FSE offset in buffer Reg6=Address in module where error was detected (from BAL instruction) Reg8=Address of DMAC Reg9=Address of buffer		

Message text: FREE IOVF CI HAS WRONG UOW#

<b>Key</b>	<b>Label</b>	<b>Description</b>
Subcode=X'41' Reg2=MLTE address Reg3=EPCB address Reg5=DFSE address Reg8=address of DMAC Reg9=address of buffer	MFSEDLET	The freed IOVF CI UOW not equal to EPCBUOWO. Prevents segments inserted to improper IOVF CI.

Message text: FREE DOVF CI HAS WRONG UOW#

<b>Key</b>	<b>Label</b>	<b>Description</b>
Subcode=X'42' Reg2=MLTE address Reg3=EPCB address Reg5=DFSE address Reg8=address of DMAC Reg9=address of buffer	MFSEDLET	The freed DOVF CI UOW not equal to EPCBUOWO. Prevents segments inserted to improper DOVF CI.

Message text: FSE CHAIN IS BAD

Key	Label	Description
Subcode=X'56' Reg2=Address of MLTE Reg5=FSE offset in buffer Reg6=Address in module where error was detected (from BAL instruction) Reg8=Address of DMAC	FSEBAD	The offset to the FSE is greater than allowed by the physical record length.
Subcode=X'156' Reg2=Address of MLTE Reg5=Address of FSE in buffer Reg6=Address in module where error was detected (from BAL instruction) Reg8=Address of DMAC	FSEBAD	The FSE indicator (X'80') is not set in the next FSE on the FSE chain in the record.
Subcode=X'256' Reg2=Address of MFSEWORK (save area for subroutines) Reg5=FSE offset in buffer Reg7=Address in module where error was detected (from BAL instruction) Reg8=Address of DMAC	FSEERR	The offset to the first FSE is greater than allowed by the physical record length.
Subcode=X'356' Reg2=Address of MFSEWORK (save area for subroutines) Reg5=FSE offset in buffer Reg7=Address in module where error was detected (from BAL instruction) Reg8=Address of DMAC	FSEERR	The offset to the first FSE is greater than allowed by the physical record length.
Subcode=X'456' Reg2=Address of MFSEWORK (save area for subroutines) Reg5=FSE offset in buffer (FSE in error) Reg7=Address in module where error was detected (from BAL instruction) Reg8=Address of DMAC	FSEERR	The FSE indicator (X'80') is not set in an FSE in the buffer.
Subcode=X'556' Reg2=Address of MFSEWORK (save area for subroutines) Reg5=FSE offset in buffer (current FSE) Reg7=Address in module where error was detected (from BAL instruction) Reg8=Address of DMAC	FSEERR	The offset to the next FSE is greater than the value of the current FSE. The chain is in error.
Subcode=X'656' Reg2=Address of MFSEWORK (save area for subroutines) Reg5=FSE offset in buffer (FSE in error) Reg6=FSE address in buffer (FSE in error) Reg7=Address in module where error was detected (from BAL instruction)	FSEERR	The FSE indicator (X'80') is not set in a previous FSE in the chain of FSEs.
Subcode=X'756' Reg2=Address of MFSEWORK (save area for subroutines) Reg7=Address in module where error was detected (from BAL instruction)	FSEERR	The newly allocated independent overflow CI has no FSE at offset 8.

Key	Label	Description
Subcode=X'856' Reg2=Address of MFSEWORK (save area for subroutines) Reg7=Address in module where error was detected (from BAL instruction)	FSEERR	The newly allocated independent overflow CI has no FSE at offset 8.
Subcode=X'956' Reg2=Address of MLTE Reg5=FSE offset in buffer Reg6=Address in module where error was detected (from BAL instruction) Reg8=Address of DMAC	FSEBAD	The first byte of space to be freed is already zeros.
Subcode=X'A56' Reg2=Address of MLTE Reg5=FSE offset in buffer Reg6=Address in module where error was detected (from BAL instruction) Reg8=Address of DMAC	FSEBAD	Previous FSE extends into the space to be freed in this call.

Normal register usage is as follows:

#### Register

	Contents
1	Subcode
9	Address of buffer

Message text: BAD RBA

Key	Label	Description
Subcode=X'57' Reg15=Start RBA of record in buffer		Start RBA of record in buffer is greater than RBA of space to be freed.
Subcode=X'157' Reg15=End RBA of record in buffer		End RBA of record in buffer is less than or equal to RBA of space to be freed.

Normal register usage is as follows:

#### Register

	Contents
1	Subcode
2	Address of MLTE
8	Address of DMAC
9	Address of buffer header (DMHR)

Message text: IOVF CONTROL FSAP WAS BAD

Key	Label	Description
Subcode=X'58' Reg2=MFSEWORK address Reg3=EPCB address Reg6=DFSE address Reg9=DBLK address (pointer to location to be dumped) Reg15=number of bytes to dump	CKFSAP	In checking the DBLKBTDW, space is available, but an error was detected. The DHMR address is located at register 2 + X'2C' (SAVE9).

Message text: IOVF CONTROL WORD WAS BAD

Key	Label	Description
Subcode=X'158' Reg2=MFSEWORK address Reg3=EPCB address Reg6=DFSE address Reg9=DBLK address (pointer to location to be dumped) Reg15=number of bytes to dump	CKFSPT	In checking the DBLKBTDW, space is not available, and a DHMR address is located at register 2 + X'2C' (SAVE9).

Key	Label	Description
Subcode=X'258' Reg2=MFSEWORK address Reg3=EPCB address Reg6=DFSE address Reg9=DBLK address (pointer to location to be dumped) Reg15=number of bytes to dump	CKFSPT or UPCTLCI	In checking the DBLKBTDW, space is available, but an error was detected. The DHMR address is located at register 2 + X'2C' (SAVE9).

Normal register usage is as follows:

#### Register

Register	Contents
1	Subcode
7	If bit0=X'0' (delete), bytes 1—3=number of bytes to delete. If bit0=X'1' (insert), bytes 1—3=UOW ID.
8	DMAC address
10	EPST address
11	ESCD address

#### For DBFMGLA9:

Message text: NO TEXT SUPPLIED

Key	Label	Description
Subcode=X'20'	DEBUG	PTRPMLTE (Reg2) is equal to zero (Reg14). This value is invalid.
Subcode=X'120'	DEBUG	The pointer in register 7 (MLTECRBA - Reg4 + X'14') is zero. A zero pointer is invalid.
Subcode=X'53' Reg2=address of parent MLTE Reg6=offset in buffer to segment Reg7=address of parent segment, DSEGCODE is first byte	DEBUG	DSEGCODE of the parent segment (pointed to by Reg7) is not the same as the parent MLTESGCD (Reg2 + X'1E'). The abend is issued from macro DBFMRCPS. DMHRBUF (Reg9 + X'10') is the address of the buffer.
Subcode=X'33'	LOGGERRO2	An invalid return code was received from DBFMCCS9; 0, 4, 8, and 12 are valid return codes.

Normal register usage is as follows:

#### Register

Register	Contents
1	Subcode
3	EPCB address
4	MLTE address
10	EPST address
11	ESCD address

#### For DBFMGNR0:

Message text: WRONG SEGMENT CODE

Key	Label	Description
Subcode=X'53'		The return code from DBFMPCGO is 24, which indicates an invalid segment code.

#### Register

	Contents
0	MLTE address
1	Subcode
3	EPCB address
4	DMCBARTD address
5	DBLK address
7	EPCBCRBA - RBA of the current segment
8	X'01' A(DSEGCDRT)
10	EPST address
11	ESCD address

Message text: DL/I CALL IS TO AN INVALID AREA

Key	Label	Description
Subcode=X'71'		The DL/I call is to an area that is invalid for this EPCB.
Subcode=X'171'		The DL/I call is to an area that is invalid for this EPCB.

#### Register

	Contents
1	Subcode
7	Address of ESCD
8	Address of DMAC

Message text: SETR area table error

Key	Label	Description
Subcode=X'71'		The return code from DBFHSSR is not zero, and the new DMAC address in register 0 is null, indicating an error in the SETR area table.

#### Register

	Contents
1	Subcode
3	Address of EPCB
7	Address of ESCD
8	Previous address of DMAC

For DBFMGNX0:

Message text: WRONG SEGMENT CODE

Key	Label	Description
Subcode=X'53'		MLTESGCD (stored in Reg8, found at Reg4 + X'1E') is not equal to the DSEGCODE (first byte of segment pointed to by Reg7). The abend is issued from macro DBFMRCCS. DMHRBUFP (Reg9 + X'10') is the address of the buffer.

#### Register

	Contents
1	Subcode
3	EPCB address

## 1026

4	MLTE address
6	Offset in buffer to segment
7	Address of segment, DSEGCODE is first byte
8	MLTESGCD
9	DMHR address
10	EPST address
11	ESCD address

### For DBFMGPD0:

Message text: SEGMENT CODE OF SDEP IS NOT X'02'

Key	Label	Description
Subcode=X'53'		The segment code is different from the expected segment code. The segment being tested is a sequential dependent (SDEP), but the segment code is not X'02'.

### Register

#### Contents

1	Subcode
3	EPCB address
8	DMAC address
9	Address of segment data
10	EPST address
11	ESCD address

### For DBFMGPF0:

Message text: NO TEXT SUPPLIED

Key	Label	Description
Subcode=X'30'		ISWITCH failure
Subcode=X'130'		ISWITCH failure
Subcode=X'31'		IMODULE GETMAIN - A CSA storage shortage might exist.
Subcode=X'32'		IMSAUTH page fix error - A real storage shortage might exist.

### Register

#### Contents

1	Subcode
10	EPST address
11	ESCD address
15	Return code

### For DBFMGRF0:

Message text: DEDB PROBABLY HAS ERROR IN IT

Key	Label	Description
Subcode=X'150'	NEXTITEM	DSEGCODE (Reg2 + X'00'), the first byte of segment, is X'00'
Reg6=address of segment, DSEGCODE is first byte		
Reg7=last segment pointer RBA		
Reg14=return address from subroutine		
Reg15=DMCB address		



Key	Label	Description
Subcode=X'152' Reg6=address of segment, DSEGCODE is first byte Reg7=last segment pointer RBA Reg14=return address from subroutine Reg15=DMCB address	NEXTITEM	This was not a segment. In checking to process it as a scrap, the length in register 2 was zero.
Subcode=X'252' Reg6=address of segment, DSEGCODE is first byte Reg7=last segment pointer RBA Reg14=return address from subroutine Reg15=DMCB address	NEXTITEM	This was not a segment. In checking to process it as a scrap, the length in register 2 was greater than the FSE length, DFSEPLEN (DFSE + X'02').
Subcode=X'352' Reg6=address of segment, DSEGCODE is first byte Reg7=last segment pointer RBA Reg14=return address from subroutine Reg15=DMCB address	NEXTITEM	The DSEGCODE (DSEG + X'00') was tested for a valid segment, for a scrap (X'7x'), and for an FSE (X'Fx'). Because none of these was found, the segment code is invalid.
Subcode=X'153' Reg5=return address from subroutine Reg6=address of segment, DSEGCODE is first byte	RUNCHAIN	DSEGCODE (DSEG + X'00') should be X'01'.
Subcode=X'253' Reg5=return address from subroutine Reg6=address of segment, DSEGCODE is first byte	CHECKBLK	DSEGCODE (DSEG + X'00') should be X'01'. A different value was found, so a branch was taken to DBFMPO0. The return code from DBFMPO0 in register 15 is X'24'.
Subcode=X'155' Reg5=return address from subroutine Reg6=RBA of next segment to be searched Reg7=RBA of first byte beyond data in buffer	SCANAPB	The address in register 6, which is the address of the next segment to be searched, is greater than the address in register 7, the maximum valid pointer address from EPCBRBA (EPCB + X'4C').
Subcode=X'255' Reg5=return address from subroutine Reg6=RBA of next segment to be searched Reg7=RBA of first byte beyond data in buffer	SCANANY	The address in register 6, which is the address of the next segment to be searched, is greater than the address in register 7, the maximum valid pointer address from EPCBRBA (EPCB + X'4C').
Subcode=X'157' Reg5=return address from subroutine	GETBLOCK	The RBA of the segment is not valid. The address in register 7 (from EPCBRBA (EPCB+X'4C')), which is divided by the block length to become the anchor point RBA, is greater than DMACFROW (DMAC+X'84'). Register 7 should be the RBA of the data, and DMACFROW is the beginning of the Reorganization Work Area.

## 1026

### Register

	Contents
1	Subcode
2	MLTE address
3	EPCB address
8	DMAC address
9	DMHR address
10	EPST address
11	ESCD address

### For DBFMIRC9:

Message text: LOGICAL ERROR IN ACTION MODULES

Key	Label	Description
Subcode=X'20'		MLTECRBA (Reg0 loaded from Reg4 + X'14') is equal to zero.
Subcode=X'120'		MLTENRBA (Reg8 loaded from Reg4 + X'04') is equal to zero.
Subcode=X'53' Reg8=address of parent segment, DSEGCODE is first byte Reg14=address of parent MLTE		MLTESEGC (Reg14 + X'1E') of parent is not equal to parent DSEGCODE (Reg8 + X'00'). The abend is issued from macro DBFMRDPS. DMHRBUFP (Reg9 + X'10') is the address of the buffer.

### Register

	Contents
1	Subcode
3	EPCB address
4	MLTE address
10	EPST address
11	ESCD address

### For DBFMIRT0:

Message text: UNEXPECTED SEGMENT CODE ENCOUNTERED

Key	Label	Description
Subcode=X'53'		MLTESGCD (Reg2 + X'1E') of the parent segment is not equal to the segment code of the parent segment pointed to by register 7. DMHRBUFP (Reg9 + X'10') is the address of the buffer. The abend is issued from the DBFMRDPS macro.
Subcode=X'153'		DSEGCODE (Reg6 + X'00') is not equal to X'01'. This should be a root segment.
Subcode=X'253'		MLTESGCD (Reg4 + X'1E') is not equal to DSEGCODE (Reg6 + X'00').

Normal register usage is as follows:

### Register

	Contents
1	Subcode
3	EPCB address
4	MLTE address
6	Address of segment, DSEGCODE is first byte
8	DMAC address
9	DMHR address
10	EPST address
11	ESCD address

### For DBFMLOP0:

Message text: DBFSYNL TYPE=RSHR LATCH COUNT WAS ZERO

Key	Label	Description
Subcode=X'13'		Attempt to release shared sync point latch when latch count is zero (ESCDSYNL).

Message text: DBFSYNL TYPE=RSHR LATCH NOT OWNED

Key	Label	Description
Subcode=X'13' or X'113'		An attempt was made to release shared sync point latch when it was not owned.

Message text: DBFSYNL TYPE=GSHR ALREADY OWNED

Key	Label	Description
Subcode=X'13' or X'113'		An attempt was made to get shared sync point latch when the task already owned it.

Normal register usage is as follows:

#### Register

	Contents
0	Contents of ESCDSYNL field
1	Subcode
2	Address of ESCD
10	Address of EPST

#### For DBFMOC10:

Message text: DMACBLKL EXCEEDS MAXIMUM CI SIZE

Key	Label	Description
Subcode=X'55'	MOCICBUF	DMACBLKL (DMAC+7C) exceeds the maximum DEDB buffer size, ESCDBUFL (ESCD+10C).

#### Register

	Contents
0	ESCDBUFL
1	1026 abend subcode
2	DMAC address
6	Bad buffer size from DMACBLKL
11	ESCD address

#### For DBFMPER9:

Message text: UNEXPECTED SEGMENT CODE ENCOUNTERED

Key	Label	Description
Subcode=X'53'	ABND1026	The parent MLTESGCD (Reg14 + X'1E') is not equal to the parent DSEGCODE (Reg8 + X'00'). The abend is issued from the DBFMRDPS macro. DMHRDMAC (Reg9 + X'20') contains the address of the DMAC. DMHRBUFP (Reg9 + X'10') is the address of the buffer.

Message text: LOGICAL ERROR IN ACTION MODULES

Key	Label	Description
Subcode=X'00'	LOGERROR	The module does not go to this label under any circumstances at this time.

Normal register usage is as follows:

#### Register

	Contents
1	Subcode
3	EPCB address
4	MLTE address
6	Offset of segment in buffer
8	Address of segment, DSEGCODE is first byte
9	DMHR address
10	EPST address
11	ESCD address
14	Address of parent MLTE

#### For DBFMPIO9:

Message text: UNEXPECTED SEGMENT CODE ENCOUNTERED

Key	Label	Description
Subcode=X'53'		The segment code pointed to by register 2 is not equal to the MLTE segment code in MLTESGCD (Reg4 + X'1E'). The abend is issued from macro DBFMRDCS. DMHRBUFP (Reg9 + X'10') is the address of the buffer.

#### Register

	Contents
1	Subcode
2	Address of segment, DSEGCODE is first byte
3	EPCB address
4	MLTE address
6	Offset to segment in buffer
7	RBA of buffer
10	EPST address
11	ESCD address

#### For DBFMPSG9:

Message text: UNEXPECTED SEGMENT CODE ENCOUNTERED

Key	Label	Description
Subcode=X'25' Reg2=address of PCB in PSB pool (from EPCB + X'10')	ABND1026	The value in DBPCBLEV (Reg2 + X'08') is greater than 1. The abend is issued from macro DBFMCLBS.
Subcode=X'53' Reg6=offset of segment in buffer Reg8=address of parent segment, DSEGCODE is first byte Reg14=address of parent MLTE	ABND1026	Parent MLTE segment code (Reg14 + X'1E') is not equal to the expected parent DSEGCODE (Reg8 + X'00'). The abend is issued from macro DBFMRDPS. DMHRBUFP (Reg9 + X'10') is the address of the buffer.

Message text: LOGICAL ERROR IN THIS MODULE

Key	Label	Description
Subcode=X'11' Reg6=offset into table of command codes starting at MLTE + X'57' Offset 0=S, 1=W, 2=Z, 3=M	LOGERROR	An invalid command code was detected. The value in register 6 is negative or greater than 3. The module handles only command codes S, W, Z, or M.

Normal register usage is as follows:

#### Register

	Contents
1	Subcode
3	EPCB address
4	MLTE address
10	EPST address
11	ESCD address

#### For DBFMPSI9:

Message text: UNEXPECTED SEGMENT CODE ENCOUNTERED

Key	Label	Description
Subcode=X'53'		Parent MLTESEGC (Reg14 + X'1E') is not equal to parent DSEGCODE (Reg7 + X'00'). The abend is issued from macro DBFMRDPS. DMHRBUFP (Reg9 + X'10') is the address of the buffer.

#### Register

	Contents
1	Subcode
3	EPCB address
4	MLTE address
6	Offset of segment in buffer
7	Address of parent segment, DSEGCODE is first byte
10	EPST address
11	ESCD address
14	Address of parent MLTE

#### For DBFM PUG0:

Message text: NO TEXT SUPPLIED

Key	Label	Description
Subcode=X'20' Reg8=DMAC address Reg9=0	DEBUG	Previous MLTE pointer is equal to zero, which is invalid.
Subcode=X'120' Reg8=DMAC address Reg9=0	DEBUG	The pointer in MLTEPARP (Reg4 + X'3C') is zero, which is invalid.
Subcode=X'53' Reg6=offset in buffer to segment Reg8=MLTESGCD	DEBUG	MLTESGCD of previous segment (Reg2 + X'1E') is not equal to the DSEGCODE of the previous segment. MLTECLOC (Reg2 + X'00') points to the address of the previous segment, and DSEGCODE is the first byte. The abend is issued from macro DBFMRCPS. DMHRBUFP (Reg9 + X'10') is the address of the buffer.

Key	Label	Description
Subcode=X'153' Reg6=offset in buffer to segment Reg7=address of segment, DSEGCODE is first byte Reg8=MLTESGCD	SDEBUG	MLTESGCD of the previous segment (Reg2 + X'1E') is not equal to the DSEGCODE of the previous segment. The abend is issued from macro DBFMRCPS. DMHRBUFP (Reg9 + X'10') is the address of the buffer.

Normal register usage is as follows:

#### Register

##### Contents

1	Subcode
2	Address of previous MLTE
3	EPCB address
4	MLTE address
9	DMHR address
10	EPST address
11	ESCD address

#### For DBFMRCU0:

Message text: NO TEXT SUPPLIED

Key	Label	Description
Subcode=X'53'		MLTE segment code (Reg4 + X'1E') is not equal to the DSEGCODE of the segment pointed to by register 2. The abend is issued from macro DBFMRCSS. DMHRDMAC (Reg9 + X'20') points to the DMAC. DMHRBUFP (Reg9 + X'10') is the address of the buffer.

#### Register

##### Contents

1	Subcode
2	Address of segment, DSEGCODE is first byte
3	EPCB address
4	MLTE address
6	Offset in buffer to segment
8	MLTESGCD
9	DMHR address
10	EPST address
11	ESCD address

#### For DBFMRPX0:

Message text: UNEXPECTED SEGMENT CODE ENCOUNTERED

Key	Label	Description
Subcode=X'53'		MLTE segment code (Reg4 + X'1E') is not equal to the DSEGCODE of the current segment pointed to by register 2. The abend is issued from macro DBFMRDCS. DMHRBUFP (Reg9 + X'10') is the address of the buffer.

#### Register

##### Contents

1	Subcode
2	Address of segment, DSEGCODE is first byte
3	EPCB address
4	MLTE address
6	Offset of segment in buffer

- 7 RBA of buffer
- 8 DMAC address
- 10 EPST address
- 11 ESCD address

**For DBFMRQC0:**

Message text: NO TEXT SUPPLIED

Key	Label	Description
Subcode=X'33' Reg8=DMAC address	DODEBUG	The return code (from DBFMGPD0 or DBFMCTL0) is not valid. Only 0, 4, 8, and 12 are valid. The invalid return code is in register 0.
Subcode=X'53' Reg6=offset in buffer to segment Reg8=MLTESGCD	DEBUG	MLTE segment code (Reg4 + X'1E') is not equal to the current segment DSEGCODE. MLTECLOC (Reg4 + X'00') points to the current segment. The first byte is DSEGCODE. The abend is issued from macro DBFMRCCS. DMHRBUFP (Reg9 + X'10') is the address of the buffer. DMHRDMAC (Reg9 + X'20') points to the DMAC.

**Register****Contents**

- 0 Address of parameter list
- 1 Subcode
- 3 EPCB address
- 4 MLTE address
- 5 ESCD address
- 9 DMHR address
- 10 EPST address
- 14 EPSTTRRG (EPST + X'3BC') - points to register save area
- 15 ESCDMSG (ESCD + X'3E8')

**For DBFMSEG0:**

Message text: INVALID SEGMENT LENGTH

Key	Label	Description
Subcode=X'59'		The routine that moves a segment from the DEDB buffer to a user I/O area found a segment that was not within the valid limits.

**Register****Contents**

- 1 SDBF address
- 2 Subcode
- 3 Length of fixed-length segment
- 7 Length of variable-length segment
- 10 EPST address

**For DBFMSFI9:**

Message text: LOGICAL ERROR IN THIS MODULE

Key	Label	Description
Subcode=X'25'	LOGERROR	MLTELLIO, the branch table entry from the operator table (at Reg4 + X'2E'), does not contain a valid operator.

## 1026

### Register

	Contents
1	Subcode
3	EPCB address
4	MLTE address
10	EPST address
11	ESCD address

### For DBFMSFO9:

Message text: NO TEXT SUPPLIED

Key	Label	Description
Subcode=X'21'	DEBUG	The SSPP pointer (Reg14) is not zero. To reach this point, there must be a zero pointer in the SSPP. The PCF pointer, checked before checking the SSPP, was zero.
Subcode=X'53' Reg6=offset in buffer to segment Reg7=address of previous segment, DSEGCODE is first byte	DEBUG	MLTESGCD of the previous segment (Reg2 + X'22') is not equal to the DSEGCODE of the segment pointed to by register 7. The abend is issued from macro DBFMRCPS. DMHRDMAC (Reg9 + X'28') points to the DMAC. DMHRBUFP (Reg9 + X'14') is the address of the buffer.
Subcode=X'153' Reg6=offset in buffer to segment Reg7=address of previous segment, DSEGCODE is first byte	DEBUG	MLTESGCD of the segment (Reg4 + X'22') is not equal to the DSEGCODE of the segment pointed to by register 8. DMHRDMAC (Reg9 + X'28') points to the DMAC. DMHRBUFP (Reg9 + X'14') is the address of the buffer.

Normal register usage is as follows:

### Register

	Contents
1	Subcode
2	Address of previous MLTE
3	EPCB address
4	MLTE address
10	EPST address
11	ESCD address

### For DBFMSSA9:

Message text: NO TEXT SUPPLIED

Key	Label	Description
Subcode=X'33'		The return code in register 0 from DBFMCCS9 was invalid; 0, 4, 8, and 12 are valid.
Subcode=X'133'		The return code in register 0 is invalid; 0, 4, 8, and 12 are valid. This return code is generated at label NOUSETOC, and is used as an invalid return from a number of different calls.
Subcode=X'53' Reg2=address of segment, DSEGCODE is first byte Reg6=offset in buffer to segment	DEBUG	MLTESGCD (Reg4 + X'1E') is not equal to the DSEGCODE of the segment pointed to by register 2. The abend is issued from macro DBRMRCSS. DMHRDMAC (Reg9 + X'20') points to the DMAC. DMHRBUFP (Reg9 + X'10') is the address of the buffer.

Normal register usage is as follows:

### Register

#### Contents



1	Subcode
3	EPCB address
4	MLTE address
10	EPST address
11	ESCD address

**For DBFMSSD9:**

Message text: LOGICAL ERROR IN ACTION MODULES

Key	Label	Description
Subcode=X'25'		MLTELLEV (Reg4 + X'54') is X'80', indicating last level, but upon exiting label PATHLOOP the return code in register 15 is not zero. This situation indicates data was moved at this level.

**Register**

Contents	
1	Subcode
3	EPCB address
4	MLTE address
10	EPST address
11	ESCD address

**For DBFMSSR9:**

Message text: LOGICAL ERROR IN ACTION MODULES

Key	Label	Description
Subcode=X'20'		The hold bit is on at EPCBLHLD in EPCBLCID (Reg3 + X'3A'), and the MLTE pointer (Reg4) is zero. Register 4 is loaded from EPCBLKFP (Reg3 + X'6C'), and is set up by the previous call.

**Register**

Contents	
1	Subcode
3	EPCB address
4	Zero
8	DMAC address
9	DMHR address
10	EPST address
11	ESCD address
14	Zero

**For DBFMUHE0:**

Message text: UNEXPECTED OFFSET OR LENGTH PASSED IN

Key	Label	Description
Subcode=X'12'		Either the offset, length, or both, of an update passed to DBFMUHE0 for processing are not acceptable values. Either the offset plus length of the update exceed the CI size or a length of zero was passed.

**Register**

Contents	
0	Length passed by caller
1	Offset passed by caller
2	Abend subcode (X'12')
8	CI size

## 1026

- 9 DMHR address from caller. Address of DMAC for Area is in DMHRDMAC field.
- 10 EPST address
- 11 ESCD address

### For DBFMVSN9:

Message text: LOGICAL ERROR IN ACTION MODULES

Key	Label	Description
Subcode=X'20'		EPCBPRGP (Reg0, loaded from Reg3 + X'60') contains zero and not a pointer to the MLTE having parentage.

### Register

- Contents**
- 0 EPCBPRGP (Reg3 + X'60'), pointer to MLTE having parentage
- 1 Subcode
- 3 EPCB address
- 4 MLTE address
- 10 EPST address
- 11 ESCD address

### For DBFPFDS0:

Message text: ADSC TO BE UNCHAINED NOT FOUND

Key	Label	Description
Subcode=X'24'		The pointer to the next ADSC/IDSC to be unchained is zero.

### Register

- Contents**
- 1 Subcode
- 4 Address of ADSC
- 6 Address of previous ADSC
- 11 Address of ESCD

### For DBFPFPB0:

Message text: DBFSYNL TYPE=RSHR LATCH COUNT WAS ZERO

Key	Label	Description
Subcode=X'13'		Attempt to release shared sync point latch when latch count is zero (ESCDSYNL).

Normal register usage is as follows:

### Register

- Contents**
- 0 Contents of ESCDSYNL field
- 1 Subcode
- 2 Address of ESCD
- 10 Address of EPST

### For DBFPGAP0:

Message text: DL/I CALL TO WRONG UOW

Key	Label	Description
Subcode=X'72'		The call UOW is outside of the root addressable portion of the Area.
Subcode=X'172'		The HSSP end RBA is in the independent overflow or beyond.
Subcode=X'272'		The HSSP end RBA is in the independent overflow or beyond.

**Register**

	Contents
1	Subcode
8	Address of DMAC
11	Address of ESCD

**For DBFSLG20:**

Message text: DBFSYNL TYPE=RSHR LATCH NOT OWNED

Key	Label	Description
Subcode=X'13' or X'113'		An attempt was made to release the shared sync point latch when it was not owned.

Normal register usage is as follows:

**Register**

	Contents
0	Contents of ESCDSYNL field
1	Subcode
2	Address of ESCD
10	Address of EPST

**For DBFSLOG0:**

Message text: DBFSYNL TYPE=RSHR LATCH NOT OWNED

Key	Label	Description
Subcode=X'13' or X'113'		An attempt was made to release shared sync point latch when it was not owned.

Normal register usage is as follows:

**Register**

	Contents
0	Contents of ESCDSYNL field
1	Subcode
2	Address of ESCD
10	Address of EPST

**For DBFSYP20:**

Message text: ERROR IN DMACNXTS OR SEQ DEPENDENT RBA

Key	Label	Description
Subcode=X'57'		The RBA in register 2 is less than the sequential dependent pointer, and the DMAC cycle count does not match.

Message text: ERROR IN SEQUENTIAL DEPENDENT INSERT RBA

Key	Label	Description
Subcode=X'157'		The RBA of the first byte in the CI (Reg6 + X'3E') is equal to the RBA of the first sequential dependent (Reg8 + X'B0').

Message text: DBFSYNL TYPE=RSHR LATCH NOT OWNED

Key	Label	Description
Subcode=X'13' or X'113'		An attempt was made to release shared sync point latch when it was not owned.

Message text: DBFSYNL TYPE=GSHR ALREADY OWNED

Key	Label	Description
Subcode=X'13' or X'113'		An attempt was made to get shared sync point latch when the task already owned it.

#### Register

##### Contents

- 1 Subcode
- 2 Points to RBA of this record
- 3 Data length of this record (halfword)
- 4 Offset of this record (halfword)
- 6 LSRT address
- 8 DMAC address
- 10 EPST address
- 11 ESCD address
- 14 First address of direct dependents (first byte past sequential dependent area)

#### For DBFTOPU0:

Message text: DBFSYNL TYPE=RSHR LATCH COUNT WAS ZERO

Key	Label	Description
Subcode=X'13' or X'113'		Attempt to release shared sync point latch when latch count is zero (ESCDSYNL).

Normal register usage is as follows:

#### Register

##### Contents

- 0 Contents of ESCDSYNL field
- 1 Subcode
- 2 Address of ESCD
- 10 Address of EPST

#### For DBFUHCF7:

Message text: SPACE FOR FSE AVAILABLE IS NEGATIVE

Key	Label	Description
Subcode=X'56'	ABEND1026	Cannot create FSE or SCRAP because the calculated available space is negative.

Message text: END OF FSE IS NOT AT END OF VSAM CSI

Key	Label	Description
Subcode=X'56'	FSEERROR	The old FSE length, plus the offset to that FSE, does not equal the length of a VSAM CI, not including the DEDB CI suffix (the RBA of the CI, VSAM RDF, and VSAM CIDE).

Normal register usage is as follows:

#### Register

	Contents
1	Subcode
2	FSE address in buffer where segment will be moved.
9	UHSW address

#### For DBFUHGS7:

Message text: EXPECTED FREE CI NOT EMPTY

Key	Label	Description
Subcode=X'58'	IOVFPROC	The obtained IOVF CI, which is to be used to copy a segment during HSREORG must be free; however, the IOVF CI was not free.

Normal register usage is as follows:

#### Register

	Contents
1	Subcode
2	UHSW address
9	DMAC Address

#### For DBFUMAF0:

Message text: UTILITY BUFFER MISSING

Key	Label	Description
Subcode=X'20'		The pointer in register 2 to the chain of buffer headers for buffers not related to DEDBs is zeros. The buffer for the utility cannot be located.

#### Register

	Contents
1	Subcode
2	Pointer to unrelated buffer headers
3	TPCB address
4	IOAR address
6	DMAC address
8	UTHR address
9	Buffer address
10	EPST address
11	ESCD address

#### For DBFUMAI0:

Message text: I/O BUFFER CHAIN CONTAINS A RING

Key	Label	Description
Subcode=X'21' Reg0=X'00' Reg5=chain pointer in previous DMR		Register 0 contains zero. It should not decrement to zero before the loop is exited when the forward buffer pointer is 0 in register 14 (UTHRNEXT: Reg14 + X'00').

Message text: DBFSYNL TYPE=RSHR LATCH NOT OWNED

Key	Label	Description
Subcode=X'13' or X'113'		An attempt was made to release shared sync point latch when it was not owned.

Message text: DBFSYNL TYPE=GSHR ALREADY OWNED

Key	Label	Description
Subcode=X'13' or X'113'		An attempt was made to get shared sync point latch when the task already owned it.

#### Register

##### Contents

0	Buffer number counter
1	Subcode
3	EPCB address
4	IOAR address
8	DMAC address
10	EPST address
11	ESCD address
13	PSTSAV1 address
14	UTHR address

Message text: WRONG RBA FROM UTILITY, R7

Key	Label	Description
Subcode=X'57' Reg2 points to start of verify table (EPSTURVR: Reg10 + X'270') Reg5 points to buffer anchor (EPSTURIO: Reg10 + X'284') Reg7=block number Reg9 points to buffer header		The block number in register 7 is greater than or equal to DMACFBAD (Reg8 + X'2C'), which is the block number of the first block beyond the sequential part.

#### Register

##### Contents

0	Buffer number counter
1	Subcode
3	EPCB address
4	IOAR address
8	DMAC address
10	EPST address
11	ESCD address
13	PSTSAV1 address
14	UTHR address

For DBFUMAN0:

Message text: GETMAIN FAILURE

Key	Label	Description
Subcode=X'31'		Register 2 contains the hexadecimal storage size for the GETMAIN. Register 15 contains the return code from the GETMAIN macro.

**Register**

	Contents
1	Subcode
3	TPCB address
4	IOAR address
8	DMAC address
8	UTHR address
10	EPST address
11	ESCD address
13	SAVEAREA address
15	Return code from GETMAIN

**For DBFUMCB9:**

Message text: CALLER BUFFER COUNT IS INCORRECT

Key	Label	Description
Subcode=X'14'		The number of buffers counted by the module (Reg15) is not equal to the anticipated number of buffers in UTDWBUFN (Reg11 + X'A26').

**Register**

	Contents
1	Subcode
3	EPCB address
4	MLTE address
9	DMAC address
10	SDBF address
11	UTDW address
14	DMCB address
15	Count

**For DBFUMCF9:**

Message text: PROGRAM LOGIC ERROR

Key	Label	Description
Subcode=X'56'	ABND1026	Space (Reg0) is a negative number, which is not valid for an FSE.

Message text: FSE LENGTH INCORRECT

Key	Label	Description
Subcode=X'156'	FSEERROR	Register 14 holds the calculated length of remaining space plus the offset to the FSE. The value should equal the precalculated offset to the CI end at UTDWDEN0 (Reg11 + X'8A8'). The two values are not equal.

**Register**

	Contents
1	Subcode
3	CITB address
4	Buffer address
9	DMAC address

## 1026

10 SDBF address  
11 UTDW address

### For DBFUMCW9:

Message text: PROGRAM LOGIC ERROR

Key	Label	Description
Subcode=X'14'		The count of buffers handled (Reg0) exceeded the number of buffers specified (Reg11 + X'A26'- UTDWBUFN).

### Register

	Contents
0	Count of buffers handled
1	Subcode
11	UTDW address

### For DBFUMFR9:

Message text: PROGRAM LOGIC ERROR

Key	Label	Description
Subcode=X'24'		The input buffer is not yet freed. The index in register 2 should be X'01' since only a root remains, but a different value was found.
Subcode=X'124'		The output buffer is not yet freed. The index in register 2 should be X'01' since only a root remains, but a different value was found.

### Register

	Contents
1	Subcode
2	Index
3	EPCB address
4	MLTE address
10	EPST address
11	UTDW address

### For DBFUMGS9:

Message text: NO FREE SPACE IN NEWLY ACQUIRED IOVF CI

Key	Label	Description
Subcode=X'58'	LOGERR1	A free CI should be empty, but it is not.

### Register

	Contents
1	Subcode
4	UTHR address
8	DBLK address
9	DMAC address
10	SDBF address
11	UTDW address

### For DBFUMIM9:

Message text: UNASSIGNED MESSAGE ID ENCOUNTERED



Key	Label	Description
Subcode=X'14'		The ID in register 2 was not yet assigned to a message. The message ID was moved from register 1 to register 2 so that the ID is available for problem analysis.

Message text: NO BUFFER AVAILABLE

Key	Label	Description
Subcode=X'114'		No buffer is available.

Message text: INVALID PCL POINTER

Key	Label	Description
Subcode=X'21'		The PCL pointer is invalid.

Message text: INVALID PCF POINTER

Key	Label	Description
Subcode=X'121'		The PCF pointer is invalid.

Message text: SSPT UNUSED IN PARENT

Key	Label	Description
Subcode=X'221'		The SSPT is not used in the parent.

Message text: GETMAIN FAILED

Key	Label	Description
Subcode=X'31'		The GETMAIN failed.

### Register

	Contents
1	Subcode
2	Message ID
7	IOAR address
9	DMAC address
11	UTDW address

Message text: UNEXPECTED STATUS CODE

Key	Label	Description
X'34'		A DEDB utility received an unexpected status code from a service request (such as read request).

Message text: UNMATCHED SEGMENT CODE

Key	Label	Description
Subcode=X'53'		The Reorganization utility found a segment code that did not match the expected value.

## 1026

Message text: INVALID SEGMENT LENGTH

Key	Label	Description
Subcode=X'59'		The Reorganization utility found a segment length that was not within the valid limits.

For DBFUMPI0:

Message text: NO TEXT SUPPLIED

Key	Label	Description
Subcode=X'14'		UTDWFLG (Reg9 + X'118') contains X'10', indicating that the sequential dependent was already processed.
Subcode=X'114'		DCBOFLGS (Reg10 + X'30') is not set to X'10', which indicates that the DCB was opened successfully. Register 10 is loaded from UTDWPCB (Reg9 + X'124'). An invalid DCB address or an unopened DCB could cause this problem.

### Register

	Contents
1	Subcode
6	Buffer address
9	Points to UTPA, a portion of the utilities work area set by caller. The UTPA begins at UTDW + X'8F0'.
10	DCB address

For DBFUMSC0:

Message text: INVALID SEGMENT CODE FOR SEQ DEPENDENT

Key	Label	Description
Subcode=X'53'	UMSCWORK	The data pointed to by register 8 does not begin with segment code X'02'.

### Register

	Contents
1	Subcode
7	Points to RBA of first buffer
8	Pointer to beginning of segment in buffer
9	Address of last data byte
10	IOAR address
11	UTDW address
14	Return address from UMSCWORK subroutine
15	DMAC address

For DBFUMTQ9:

Message text: PROGRAM LOGIC ERROR

Key	Label	Description
Subcode=X'14'		The count of buffers processed in register 0 exceeded the number of buffers queued in UTDWBUFN (Reg11 + X'A26'). This is a loop.

### Register

	Contents
0	Count of buffers processed
1	Subcode

3 EPCB address  
 4 MLTE address  
 11 UTDW address

**For DBFXCGL0:**

Message text: NO TEXT SUPPLIED

Key	Label	Description
Subcode=X'13'		ESCDRLAT (Reg11 + X'310') is zero if the latch is available. The resource latch is already held. This problem could result from someone failing to free the resource latch.

**Register**

	Contents
1	Subcode
4	Entry point address
10	EPST address
11	ESCD address
13	SAVEAREA address
15	ESCDRLAT (Reg11 + X'310')

**System action:** The Fast Path dependent region terminates abnormally.

**Problem determination:** See the DFS2712I messages issued before the abend for the name of the module, the register contents, and (depending on the module) a numeric subcode in a register.

**Source:** IMS abends

**Module:** DBFARDC0, DBFATRM0, DBFBENQ0, DBFBGET0, DBFCHKP0, DBFCSTS0, DBFDAC50, DBFDRSC0, DBFD190, DBFHCA50, DBFMBED0, DBFMCCV9, DBFMCSS9, DBFMCRP9, DBFMCTL0, DBFMDLT0, DBFMDPT9, DBFMDBR0, DBFMDSG9, DBFMFLG0, DBFMFSE0, DBFMGLA9, DBFMGMR0, DBFMGNX0, DBFMGPD0, DBFMGPF0, DBFMGRF0, DBFMIRC9, DBFMIRT0, DBFMLOP0, DBFMOCIO, DBFMPE9, DBFMPIO9, DBFMPSG9, DBFMPSI9, DBFMUG0, DBFMRCU0, DBFMRPX0, DBFMRQC0, DBFMSEG0, DBFMSEI9, DBFMSFO9, DBFMSRT0, DBFMSSA9, DBFMSSD9, DBFMSSR9, DBFMUHE0, DBFMVSN9, DBFPFDS0, DBFPFPB0, DBFPGAB0, DBFPGAP0, DBFSDEQ0, DBFSLG20, DBFSLOG0, DBFSMP10, DBFSYN10, DBFSYN20, DBFSYP20, DBFTOPU0, DBFUHCF7, DBFUHDS0, DBFUHGS7, DBFUMAF0, DBFUMA10, DBFUMAN0, DBFUMCB9, DBFUMCF9, DBFUMCW9, DBFUMFR9, DBFUMGS9, DBFUMIM9, DBFUMPI0, DBFUMSC0, DBFUMTQ9, DBFVSOP0, DBFVXOIO, DBFXCGL0

**Related information:**

 DFS2712I (Messages and Codes)

**1027**

**Explanation:** This is an internal IMS error. An error has occurred while attempting to enqueue or dequeue a Fast Path resource.

**Analysis:**

**For DBFBENQ0, DBFCBHL0, DBFSLOG0:**

A logical error occurred while attempting to enqueue or dequeue a Fast Path resource. Register 15 contains the following return codes:

Key	Label	Description
Reg15=X'02'	EPSTQBLK	No block on this chain matches the control word in the dequeue request.
Reg15=X'03'		No QBLOCK was found in the EPST before directed enqueue.
Reg15=X'06'	MSNQRCW	Points to the resource control word with a negative count.
Reg15=X'05'		Directed enqueue call was issued by DBFBENQ0 and the return code from DFSLRH00 was X'04'.

Key	Label	Description
Reg15=X'09'		Directed enqueue call was issued by DBFBENQ0 and the return code from DFSLRH00 was X'08'.
Reg15=X'0D'		Enqueue on dequeue call was issued by DBFBENQ0 and the return code from DFSLRH00 was X'12', indicating an invalid call.

**For DBFSLOG0:**

An unacceptable return code was received from the IMS ENQ/DEQ routine DFSLRH00. The registers (R14-R12) at the time of error are saved in PSTSAV15 + 'C'. Register 15 contains the return code from the lock request handler.

*Possible cause:* Internal program logic error or ENQ/DEQ pool is full.

**For DBFMGXC0:**

An unacceptable return code was received from the IMS ENQ/DEQ routine DFSLRH00. There are two routines within DBFMGXC0 that branch to the abend routine, as indicated in the following table:

Key	Label	Description
Reg15=X'10'		SDEP CI lock request duplicate.
Reg15=X'12'	ENQRTN	Error in the call to DFSFXC10.
Reg15=X'04' or X'08'	ENQRTN	Internal logical error; these codes should not occur.

*Possible cause:* Internal program logic error

**For DBFPUC0:**

During UOW contention detection, DBFPUC0 found a UXR in exclusive mode on the UXR DMAC chain.

Key	Label	Description
Reg15=X'11'		A UXR in exclusive mode was found on the UXR DMAC chain during UOW contention detection. The new UXR is in exclusive mode also. This condition should not occur because only one region can access an area with PROCOPT H PCB.
Reg15=X'12'		DBFPUC0 tried to chain a new UXR to the UXR EPST chain, but a UXR for the same area and RBA was already on this chain.

**For DBFSYN20:**

During SYNC, COMMIT is checked to be certain it is valid before known locks are freed.

Key	Label	Description
Reg15=X'14'		DBFSYN20 frees locks through DFSLR but the lock table is either not available or not accessible. An abend 0113 can also occur if DL/I, DC, or Fast Path I/O was processing at the time of the abend 0127.

**For DBFXPIX0, DBFSDEQ0, DBFLRLS0, DBFPUXR0, DBFSLG20:**

A nonzero return code was received from the IMS ENQ/DEQ routine (DFSLRH00). The module terminating abnormally received a nonzero return code from DFSLRH00. The module is dequeuing Fast Path resources and should encounter return codes of zero only. Register 15 contains the invalid return code.

**Possible cause:** Internal program logic error. Contact the IBM Support Center.

**System action:** The dependent region terminates abnormally. If the error occurs while holding critical resources, the control region will abend with abend 0113.

**Programmer response:** For the codes and their meanings see "Analysis" on page 569.

**Problem determination:** 4, 8, 12, 35

**Source:** IMS abends

**Module:** DBFBENQ0, DBFCBHL0, DBFSLOG0, DBFSLOG0, DBFMGX0, DBFPUXC0, DBFSYN20, DBFXPIX0, DBFSDEQ0, DBFLRLS0, DBFPUXR0, DBFSLG20

**Related information:**

Chapter 2, "IMS problem determination," on page 13

## 1028

**Explanation:** During execution of the High Speed Reorganization utility, a severe error was detected in the area being reorganized. Module DBFUHAC7 found a non-ascending key for a segment during area reorganization, and subcode = X'C' was issued. An SDUMP is taken, with the title of the dump corresponding to the subcode *nn* below. Register 3 contains the failure subcode in the symptom dump; register 11 in the save area at entry to the routine UHREDUMP contains the address of the utility work area.

Subcode	Module	Meaning
X'A'	DBFUHAC7	RAP RBA NOT FOUND IN BUFFERS
X'B'	DBFUHRE0	RAP CI INVALID TYPE
X'C'	DBFUHAC7	KEYS ARE NOT ASCENDING
X'E'	DBFUHCS7	INVALID PCL POINTER
X'F'	DBFUHCS7	INVALID PCF POINTER
X'11'	DBFUHRD7	INVALID SSPT POINTERS
X'12'	DBFUHAC7	SEGMENT CODE DOES NOT MATCH SDBF
X'13'	DBFUHAC7	SEGMENT LENGTH NOT WITHIN LIMITS

**Analysis:** At the time of the abend 1028 abend, the general purpose registers contain the following information:

### Register

#### Contents

- 3 Subcode = X'0000000C'
- 4 RBA of root segment
- 5 RBA of segment with non-ascending key
- 7 Address of the input buffer that contains the segment with the non-ascending key value
- 8 Address of the DMAC for the AREA being reorganized
- 9 Address of the DMHR of CI containing the segment with the non-ascending key value
- 10 Address of the current segment type in the SDBT table for the segment with the non-ascending key
- 11 Address of the UHSW table

**Possible cause:** Internal program error

**Source:** IMS abends

**Module:** DBFUHRE0

## 1029

**Explanation:** Module DBFXWU30 attempted to reactivate a dependent region that was not deactivated.

**Analysis:** This is a standard abend issued by DBFXWU30. Register 1, saved at entry to the module, points to the EPST in error.

**Possible cause:** Internal program error

**System action:** The Fast Path dependent region terminates abnormally.

**Operator response:** Enter a /CHECKPOINT ABDUMP command to get a control region dump.

**Problem determination:** Register 1, saved at entry to the module, points to the EPST in error. A control region and a dependent region dump are needed.

**Source:** IMS abends

**Module:** DBFXWU30

## 1030

**Explanation:** Module DBFXTCU0 attempted to reactivate a dependent region that was not deactivated.

**Analysis:** This is a standard abend issued by DBFXTCU0. Register 1, saved at entry to the module, points to the EPST in error.

**Possible cause:** Internal program error

**System action:** The Fast Path dependent region terminates abnormally.

**Operator response:** Enter a /CHECKPOINT ABDUMP command to get a control region dump.

**Problem determination:** Register 1, saved at entry to the module, points to the EPST in error. A control region and a dependent region dump are needed.

**Source:** IMS abends

**Module:** DBFXTCU0

## 1031

**Explanation:** An IMS Fast Path module received a nonzero return code from the page fix or free module, DFSV4200, which was started by the IMSAUTH macro. The IMSAUTH SVC encountered a problem. Register 2 contains FIX or FREE. Register 15 contains the IMSAUTH PGFIX/PGFREE return code. For a description of these codes, see the IMS system services return codes information.

**Analysis:** This is a standard abend issued by any one of the three modules. Register 2 contains 'FIX' or 'FREE'. Register 15 contains the return code. Register 5 contains the address of the page fix/free list and REG 14 contains the address of the SCD.

Key	Label	Description
Reg2=CL4'FIX'	DBDPFIX	Page-fix operation has failed.
Reg2=CL4'FREE'	DBDPFREE	Page-free operation has failed.

The preceding paragraph and table apply only to DBFDBDP0. For DBFERS20, R7=A(SCD) (no register points to the fix/free list). For both the FIX and FREE cases, the abend is issued from label ERS2ABND. For DBFTOPU0, R1=A(page fix/free list), R9=A(SCD). For both the FIX and FREE cases, the abend is issued from label ABND1031.

For DBFERS21, Register 2 contains 'FIX'. Register 15 contains the return code. Register 5 contains the address of the page fix/free list and register 6 contains the address of the SCD.

**Possible cause:** Internal program logic or interface error


**System action:** The IMS control region is terminated abnormally.

**Problem determination:** 4, 35

**Source:** IMS abends

**Module:** DBFDDBDP0, DBFERS20, DBFERS21, DBFTOPU0

**Related reference:**

 IMS system services return codes (Messages and Codes)

**Related information:**

Chapter 2, "IMS problem determination," on page 13

## 1032

**Explanation:** The MSDB Checkpoint/Dump module, DBFDDBDP0, in an attempt to convert a virtual address to a real address using the load real address (LRA) instruction, set a nonzero condition code.

**Analysis:** This is a standard abend issued by DBFDDBDP0. Register 15 contains one of the following return codes:

**Register**

**Contents**

- X'01' The CCW area address could not be translated.
- X'02' The output area address could not be translated.
- X'03' The control record area address could not be translated.
- X'04' The IDAW address could not be converted to a real address.

Key	Label	Description
Reg15=X'01'	DBDPWDTA	A translation failure occurred while attempting to dump a data set area to a Fast Path checkpoint data set.
Reg15=X'01'	DBDPWCTL	A translation failure occurred while attempting to dump control records to a Fast Path checkpoint data set.
Reg15=X'02'	DBDPWDTA	The input data address could not be translated.
Reg15=X'03'	DBDPWCTL	The control record area address could not be translated.
Reg15=X'04'	DBDPWDTA	The IDAW address could not be converted to a real address.

**Possible cause:** The causes include:

- Hardware error
- Internal program logic or interface error

**Programmer response:** No action is required.

**Problem determination:** 4, 35

**Source:** IMS abends

**Module:** DBFDDBDP0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

## 1033

**Explanation:** The application program required more Fast Path database buffers than were specified by the NBA and OBA parameters in the dependent region JCL parameters.

**Analysis:** This is a standard abend issued by DFSECP10, DFSECP20, DFSISI00. The Fast Path data entry database (DEDB) call analyzer, DBFMCL00, or main storage database (MSDB) call analyzer, DBFBCL10, has returned a return code of X'0C' to DBFIRC10, indicating the out-of-resource condition of "no database buffer available." To verify this condition, compare the halfwords at labels EPSTBMAX and EPSTBUSE. If EPSTBMAX=EPSTBUSE, the specified number of buffers has already been used.

DBFIRC10 set the pseudoabend code.

## 1034

**Possible cause:** Application program error, or either the NBA, OBA, or both parameters were specified too small.

**System action:** IMS terminates the message-driven application program.

**Programmer response:** Increase the NBA parameters or the OBA parameters, or both, and then restart the message-driven dependent region.

**Source:** IMS abends

**Module:** DFSECP10, DFSECP20, DFSISI00

---

## 1034

**Explanation:** An ISWITCH from the dependent region to the control region (or from the control region to the dependent region) failed.

**Analysis:** For DBFATRM0, DBFIRC10:

An ISWITCH could not be executed because a dependent region was terminating.

This is a standard abend issued by DBFIRC10 and DBFATRM0. This abend occurs when an ISWITCH is issued in the GIVEBACK routine. This routine switches to CTRL to serialize while returning a message to the BALG.

**For DBFCSTS0, DBFMER00, DBFMFLG0, DBFXVUN0:**

Either an ISWITCH to the CTL TCB or an ISWITCH back to the XFP TCB cannot be executed.

An abend 1034 is a standard abend issued by DBFCSTS0, DBFMER00, DBFMFLG0, and DBFXVUN0 when an ISWITCH to the CTL TCB failed before a page fix request is issued; or when an ISWITCH back to the XFP TCB failed after a page fix request has been issued.

**For DBFDT900:**

Module DBFDT900 is a Fast Path database tracker termination module under Online Recovery Service (ORS). ISWITCH is issued to ensure pagefree for the EDBT is performed under FP CTL TCB. ISWITCH is then issued to return to ORS's RWM TCB.

This is a standard abend issued by DBFDT900. The program status word (PSW) at entry-to-abend points to the instruction where the abend was issued.

Key	Label	Description
Reg2=X'01'		ISWITCH to FPE TCB
Reg2=X'02'		ISWITCH back to RWM TCB (ORS)
Reg14=BALR		
Reg15=ISWITC return code		A nonzero return code was passed from the DFSIDSP0 (ISWITCH entry)

**For DBFMLCL0:**

Module DBFMLCL0 is a Fast Path logical CLOSE processing module that closes the AREA or ADSs under the Fast Path TCB. ISWITCH is issued to control the closing of the TCB switching. Register 14 in the abend SVRB is used as the key, and register 12 is the base register.

This is a standard abend issued by DBFMLCL0. The program status word (PSW) at entry-to-abend points to the instruction where the abend was issued.

Key	Label	Description
Reg1=completion code Reg15=ISWITCH TO=FPE return code Reg15=ISWITCH TO=RET return code	Reg14=BALR	A nonzero return code was passed from the DFSIDSP0 (ISWITCH entry).

**For DBFMLOP0:**



Module DBFMLOP0 is the Fast Path logical OPEN processing module that opens the area and ADSs under the Fast Path TCB. This is triggered by a DL/I call in the dependent region. ISWITCH is issued to control the TCB switching. Register 14 in the abend SVRB is used as the key, and register 12 is the base register.

This is a standard abend issued by DBFMLOP0. The program status word (PSW) at entry-to-abend points to the instruction where the abend was issued.

Key	Label	Description
Reg1=completion code Reg15=ISWITCH TO=FPE return code Reg15=ISWITCH TO=RET return code	Reg14=BALR	A nonzero return code was passed from the DFSIDSP0 (ISWITCH entry).

#### For DBFMPOP0:

Module DBFMPOP0 is a Fast Path physical OPEN processing module that opens the AREA or ADSs under the Fast Path TCB. ISWITCH is issued to control the opening of the TCB switching. Register 14 in the abend SVRB is used as the key, and register 12 is the base register.

This is a standard abend issued by DBFMPOP0. The program status word (PSW) at entry-to-abend points to the instruction where the abend was issued.

Key	Label	Description
Reg1=completion code Reg15=ISWITCH TO= FPE return code Reg15=ISWITCH TO=RET return code	Reg14=BALR	A nonzero return code was passed from the DFSIDSP0 (ISWITCH entry).

#### For DBFMSDP0:

Module DBFMSDP0 is a Fast Path module that sets up the area SDEP buffer. This is a standard abend issued by DBFMSDP0. Module DFSIDSP0 (the ISWITCH entry) passed a nonzero return code.

#### For DBFPICS0:

Module DBFPICS0 is a Fast Path HSSP image copy set-up module that opens the image copy data sets under the Fast Path TCB. The opening of the image copy data sets is triggered by a DL/I call in the dependent region for which HSSP image copy is specified. ISWITCH is issued to control the TCB switching. Register 14 in the abend SVRB is used as the key, and register 12 as the base register.

This is a standard abend issued by DBFPICS0. The program status word (PSW) at entry-to-abend points to the instruction where the abend was issued.

Key	Label	Description
Reg1=completion code Reg14=BALR Reg15=04 (issued by DBFPICS0)		A nonzero return code was passed from the DFSIDSP0 (ISWITCH entry).

#### For DBFSGAB0:

Module DBFSGAB0 is a Fast Path module that gets a buffer from a shared pool. This is a standard abend issued by DBFSGAB0 and DBFSGAB1. This abend occurs when an ISWITCH to a FP TCB is issued to serialize the extend process in module DBFPVTS0 and the later wait in module DBFSGAB0 or DBFSGAB1.

The out-of-resource condition can also occur for the shared VSO buffer if they are exhausted or if the buffer usage EPSTBUSE is greater than the sum of EPSTNBNA and EPSTABA0. The return code of X'0c' is issued by module DBFSGAB0.

## 1035 • 1036

**System action:** The task is terminated abnormally.

**Programmer response:** This error condition should not occur. Contact your IBM Support Center for assistance.

**Source:** IMS abends

**Module:** DBFATRM0, DBFCSTS0, DBFD900, DBFIRC10, DBFMER00, DBFMFLG0, DBFMLCL0, DBFMLOP0, DBFPICS0, DBFMPOP0, DBFMSP0, DBFSGAB0, DBFXVUN0

---

### 1035

**Explanation:** A search of the DEDB area name list (DBFAREA) for a specified ddname resulted in a 'NOT FOUND' condition.

**Analysis:** This is a standard abend issued by DBFMSRH0. Module DBFMSRH0, in performing a binary search for a particular DDNAME in the DBFAREA, was unable to locate the DDNAME.

Key	Label	Description
Reg8=0		DDNAME entry not found in area name list.

**Possible cause:** Internal program logic or interface error

**System action:** The task terminates abnormally.

**Programmer response:** This error condition should not occur.

**Problem determination:** 4, 35

**Source:** IMS abends

**Module:** DBFMSRH0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

### 1036

**Explanation:** The error message number, contained in register 8, is not included in the error message table used by the abnormally terminating module, DBFMER00.

**Analysis:** This is a standard abend issued by DBFMER00. The error message table is an internal table containing error message numbers and their associated texts. This table is contained in and maintained by DBFMER00.

Key	Label	Description
Reg9=MSGTBL address		Error message table address.
Reg8=Error message number		The missing message number.

**Possible cause:** This abend should occur only during the system test cycle, when new messages have been created.

**System action:** Task terminates abnormally.

**Programmer response:** Because this error condition should not occur, save this dump.

**Problem determination:** 1, 2, 3, 4, 11, 35

**Source:** IMS abends

**Module:** DBFMER00

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

**1038**

**Explanation:** During an emergency restart, a VSAM GENCB macro was executed and a nonzero return code was returned from VSAM.

**Analysis:** This abend is a standard abend issued by DBFERST0, when the VSAM GENCB macro fails.

Key	Label	Description
Reg2=VSAM reason code Reg15=Return code from VSAM	ERSTDMHR	The VSAM return codes and reason codes for the GENCB macro.

**Possible cause:** The operands in the GENCB macro were specified incorrectly.

**System action:** IMS is abnormally terminated.


**Programmer response:** Register 15 contains the return code and register 2 contains the reason code from VSAM.

**Problem determination:** 4, 35

**Source:** IMS abends

**Module:** DBFERST0

**Related reference:**

 z/OS: VSAM macro return and reason codes

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

**1039**

**Explanation:** The abend option for FP DEDB online utility was selected.

**Analysis:** The option ERRORACTION=ABEND provides diagnostic information when FP DEDB on-line utility ends with an error message.

**System action:** After processing is complete, the utility ends abnormally with a dump.

**Programmer response:** No action is required.

**Problem determination:** No action is required.

**Source:** IMS abends

**Module:** DBFUMRT0

---

**1040**

**Explanation:** The DFSBCB macro could not obtain an IMS Fast Path related control block.

**System action:** For SYNC POINT LOG processing, this pseudo abend is stored in PSTABTRM, and error message DFS3022W is issued. Processing continues.

In other cases, the IMS control region terminates.

**Programmer response:** Determine the cause of failure as indicated by error messages and the analysis of the memory dump. If the abend resulted from insufficient storage, increase Common Service Area (CSA) storage allocations and emergency restart IMS to resume processing.

**Source:** IMS abends

**Related information:**

 DFS3022W (Messages and Codes)

---

---

1041

**Explanation:** The high level programming interface, HLPI, found a condition caused by a programming error, or DL/I returned a status code to HLPI indicating a programming error.

**Analysis:** 1041 is a standard abend issued by DFSEIPB0. At abend, register 2 contains the address of the control block DFSSDIB. See DLZSDIB DSECT in macro DFSHLPDS. All areas are addressed by this control block, and each is described by a DSECT in DFSHLPDS. DFSEIPB0 uses register 11 as a base register. The actual call parameter list for DL/I may not be complete at time of abend; it is found at label DIBPARM in control block DFSSDIB.

*Possible cause:*

**Register****Contents**

**DHTN** Probably caused by a failure to issue an initialization call (should be done by CICS translator), or DFSDIB control block is overlaid.

**All Other Codes**

Application program must be corrected. HLPI allows only successful status code to be returned to the application program.

**System action:** IMS terminates abnormally with a dump, and message DFS1041 is issued.

**Programmer response:** For an explanation of the status codes, see *IMS Version 13 Application Programming*. Correct the application program.

**Problem determination:** 4, 11, 19, and obtain a listing of macro DFSHLPDS.

**Source:** IMS abends

**Module:** DFSEIPB0

**Related information:**

 DFS1041

Chapter 2, "IMS problem determination," on page 13

---

## 1042

**Explanation:** IMS or an application thread failed due to an IMS internal control block error.

Alternatively, an emergency restart detected a situation where a buffer was being released more than once. This circumstance can cause unpredictable results, so emergency restart was abnormally terminated.

**Analysis:** Check the reason code in register 15.

**Reason Code****Description**

**R15=X'03'**

DBFDBLS0 detected a Fast Path 64 bit buffer being released more than once during ERE.

**R15=X'04'**

DBFDLSR0 detected a Fast Path 64 bit buffer being released more than once during ERE.

**R15=X'05'**

DBFEHSH0 detected a Fast Path 64 bit buffer being released more than once during ERE.

**R15=X'06'**

DBFERDB0 detected a Fast Path 64 bit buffer being released more than once during ERE.

**R15=X'07'**

DBFEROC0 detected a Fast Path 64 bit buffer being released more than once during ERE.

**R15=X'08'**

DBFERST0 detected a Fast Path 64 bit buffer being released more than once during ERE.

**R15=X'09'**

DBFERS10 detected a Fast Path 64 bit buffer being released more than once during ERE.

**R15=X'0A'**

DBFERSY0 detected a Fast Path 64 bit buffer being released more than once during ERE.

**DBFBPNI0, DBFBPNK0:**

While attempting to obtain or release a Fast Path buffer, IMS or an application thread detected an IMS internal control block error. The process was not able to recover from this error and must terminate.

**System action:** The process attempting to obtain a buffer abnormally terminates.

**Programmer response:** Restart or recycle IMS. If the problem persists after restarting or recycling IMS, fall back to the previous buffer manager using the DBBF, BSIZ, and DBFX parameters.

In the abend occurred during an emergency restart, try emergency restart again. If the problem persists, a cold start of the database portion of IMS is required. The cold start of the database portion of IMS might require recovering of affected DEDB areas.

**Problem determination:** Contact IBM Software Support, retaining any supporting documentation such as memory dumps and logs.

**Source:** IMS abends

**Module:** DBFBPNI0, DBFDLS0, DBFDLSR0, DBFEHSH0, DBFERDB0, DBFEROC0, DBFERST0, DBFERS10, DBFERSY0

#### 1043

**Explanation:** You have defined FPBP64=Y in DFSDFxxx, but did not specify FPBP64M.

**Analysis:** For DBFFATC1, during IMS initialization, the FPBP64=Y specification was found. This parameter requires that you also specify FPBP64M.

**System action:** IMS initialization is terminated.

**Programmer response:** Specify FPBP64M with a value between 1 MB and 2047 MB. Then restart IMS.

**Source:** IMS abends

**Module:** DBFFATC0

#### 1044

**Explanation:** An application program issued a GUR DL/I call, but the IMS catalog is not enabled.

**Analysis:** When the IMS catalog is disabled, the GUR call is not supported.

The IMS catalog is disabled by default, so either the default is in effect or CATALOG=N is explicitly specified.

**System action:** The IMS region is terminated.

**Programmer response:** Determine why the IMS catalog is not enabled and correct the problem. After the problem is corrected, restart IMS.

The default value might be in effect for any of the following reasons:

- The DFSDF=xxx parameter is not specified in the startup procedure
- The IMS Catalog Definition exit routine (DFS3CDX0) was not used in a batch environment
- The wrong IMS ID is specified on the <CATALOGxxx> section header in the DFSDFxxx PROCLIB member

The CATALOG=N parameter might be explicitly specified in either the <CATALOG> section of the DFSDFxxx member or in the DFS3CDX0 exit routine.

**Source:** IMS abends

**Module:** DFSCPY00, DFSBINT0

#### 1046

**Explanation:** An IMS Fast Path SDEP utility has detected a condition that should not occur and has issued the DBFDEBUG macro to describe the condition. Message DFS2712I is issued with a text and a subcode to describe the error. The diagnostic information sent to the JOBLOG includes the DMAC, the CI buffer in error, the IOAR control block, the HWM RBA and its time stamp, the start position RBA, the stop position RBA and their options, the working buffer set, and the registers at time of abend.

**Analysis:** This is a standard abend issued from the following modules: DBFUMDL0, DBFUMSC0, and DBFUMRT0.

## 1046

Register 14 contains the address of the instruction before the abend 1046 is issued. The subcodes have the following meanings:

Key	Label	Description
Subcode=X'91'		The time stamp conversion was not successful.
Subcode=X'92'		Conversion to store clock was not successful.
Subcode=X'93'		The utility application failed to connect the area.
Subcode=X'94'		Invalid SDEP CI found.
Subcode=X'95'		Segment code is not SDEP.
Subcode=X'96'		Invalid utility name specified.
Subcode=X'97'		Invalid CI suffix found.
Subcode=X'98'		The offset to the SDEP segment is outside of the CI size.
Subcode=X'99'		Bad CC+RBA found.
Subcode=X'9A'		All new format specified, find old format.
Subcode=X'9B'		The first new format read is not in the first buffer.
Subcode=X'9C'		Bad DMACXVAL found.

**System action:** The Fast Path SDEP utility terminates abnormally.

**Problem determination:** See message DFS2712I, issued prior to the abend, for the name of the module, the register contents, and (depending on the module) a numeric subcode in a register.

**Source:** IMS abends

**Module:** DBFUMDL0, DBFUMDL0, DBFUMRT0

**Related information:**

 DFS2712I

---

## Chapter 23. IMS abend codes 1051 - 2000

This topic identifies and explains the following abend codes issued by IMS. Each code is 4 decimal digits.

---

### 1051

**Explanation:** During on-line change, an unexpected condition occurred processing Fast Path DEDB changes.

**System action:** The control region terminates abnormally.

**Programmer response:** Contact the IBM Support Center.

**Problem determination:** 4, 36

**Source:** IMS abends

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

### 1060

**Explanation:** The DFSCCTL data set contains an error. Accompanying error messages DFS0491A and DFS0492I describe the error.

**Analysis:** For DBFCPY00:

IMS found a problem with the PSB scheduled in the dependent region, and issued one of the following messages: DFS0501A, DFS0504A, or DFS0525A.

**System action:** The region is terminated.

**Programmer response:** Correct the error based on the information in the messages.

**Source:** IMS abends

**Module:** DBFCPY00, DFSSBI00

**Related information:**

 DFS0491A

 DFS0492I

---

### 1061

**Explanation:** The GETMAIN routine was unable to obtain storage for HSSP control blocks.

**Analysis:** The GETMAIN routine failed to obtain segment workareas (EPSTSEG1 and EPSTSEG2) in sp231 ECSA.

For GETMAIN to succeed, you need to free up ECSA. Then you can rerun the abended job. For a DBCTL thread, message DFS0527A is also issued.

**System action:** The region is terminated.

**Programmer response:** Increase the size of the REGION parameters; then resubmit the job. If this abend persists, wait until CSA usage diminishes before resubmitting the job.

**Source:** IMS abends

**Module:** DBFPCAA0, DBFPHI00, DBFCPY00

---

---

**1062**

**Explanation:** The high-speed sequential-processing (HSSP) image copy failed.

**Analysis:** This abend is issued by DFSECP10. It is set in one of two places:

**DBFPICS0**

Image copy set up failed. Register 2 contains the address of the DMAC. This abend follows either message DFS0531I, DFS0532I, or both.

**DBFSIC10**

Image copy I/O failed. Register 2 contains the address of the DMAC. Register 5 contains the address of the ESRB.

**System action:** The region is terminated.

**Programmer response:** Correct the problem based on the information in message DFS0531I or DFS0532I, or both.

**Source:** IMS abends

**Module:** DBFPICS0, DBFSIC10, DFSECP10

---

**1063**

**Explanation:** An internal IMS system error occurred while managing IMS FP buffer pools. These buffers could be SVSO private buffers, common buffers defined by the DBBF and BSIZ parameters, or buffers allocated as a result of specifying the FPBP64=Y parameter. The cause of the error is discussed in the Analysis section.

Area private buffers were on the NBA/OBA buffer chain, or NBA/OBA buffers were on the area private buffer chain. This problem is an internal IMS system error.

**Analysis: For DBFIBUF0:**

While attempting to page fix or page free database buffers, a private buffer was found on the available queue. The IMS control region is terminated. This problem is an internal IMS system error. Contact IBM Software Support for help in determining the specific problem.

**For DBFPFAB0:**

A normal DB buffer was returned to the free private buffer routine. The IMS control region is terminated. This problem is an internal IMS system error. Contact the IBM Software Support for help in determining the specific problem.

**For DBFMDBQ0:**

DBFMDBQ0 found a dummy XCRB pointing to a DMHR while trying to queue buffer headers (DMHRs) and XCRBs to their respective queues.

**For DBFPGAB0:**

Message text: NOT ENOUGH PRIVATE BUFFERS OBTAINED

**Register****Contents**

<b>2</b>	Address of DMAC
<b>11</b>	Address of ESCD

**For DBFSGAB0:**

Csect DBFSGAB1 of module DBFSGAB0 gets a buffer from a shared pool for restart.

1063 is a standard abend issued by DBFSGAB1 when restart cannot get a buffer and the pool is not being extended. This problem is an internal IMS system error. Contact IBM Software Support for help in determining the specific problem.

**For DBFBPNI0:**



During a request for a buffer managed by the FP 64 bit buffer manager, the incorrect buffer was about to be returned to the caller. This problem is an internal IMS system error. Contact IBM Software Support for help in determining the specific problem.

**For DBFBPNK0:**

While making buffers available for reuse by the FP 64 bit buffer manager, the buffer being returned was not an SVSO private buffer or a buffer allocated as part of the 64 bit buffer manager. This problem is an internal IMS system error. Contact IBM Software Support for help in determining the specific problem.

**System action:** The region is terminated.

**Programmer response:** Contact IBM Software Support for help in determining the problem.

**Source:** IMS abends

**Module:** DBFIBUF0, DBFPFAB0, DBFMDBQ0, DBFPGAB0, DBFSGAB0, DBFBPNI0, DBFBPNK0

### 1064

**Explanation:** When HSSP finishes processing an area, it releases the private buffer pool after all private buffers are returned. HSSP was unable to release the private buffer pool because some private buffers had not been returned. This is an internal IMS system error. Register 15 contains a subcode.

**Analysis:** Register 15 contains one of the following subcodes:

Key	Label	Description
Reg15=X'01'		HSSP area termination must wait for OTHREADS to finish writing all the area private buffers and return them to the private buffer pool. These buffers were not returned, thus the OTHREADS did not complete.
Reg15=X'02'		All DMACs with private buffers are placed on the ESCDDMPB chain. This DMAC was not found on the chain.
Reg15=X'03'		Unable to obtain storage for an AWE.

This is an internal IMS system error. Contact the IBM Support Center for help in determining the specific problem.

**System action:** The IMS control region is terminated.

**Programmer response:** Contact the IBM Support Center for help in determining the problem.

**Source:** IMS abends

**Module:** DBFPFPB0

### 1065

**Explanation:** An application made a call with a PCB that is defined with PROCOPT H and PROCOPT GO. This combination of processing options is invalid.

**System action:** The dependent region is terminated.

**Programmer response:** Correct the invalid combination of processing options, or use another PCB.

**Source:** IMS abends

**Module:** DBFIRC10

Correct the invalid combination of processing options, or use another PCB on the call.

### 1070

**Explanation:** The IMODULE GETMAIN request for storage for a private buffer pool in subpool 231 of the IMS control region, or the request to page fix such storage, was unsuccessful.

**System action:** IMS terminates abnormally.

## 1143 • 1352

**Operator response:** After increasing CSA or ECSA storage size, restart IMS.

**Programmer response:** Consult your z/OS system programmer, and increase the CSA or ECSA size.

**Source:** IMS abends

**Module:** DBFPVTS0

---

### 1143

**Explanation:** IMS detected that an ongoing HALDB online alter operation increased the size of a segment. IMS attempted to obtain a larger PSB work area, but could not. Most likely, either the PSB work area pool is too small or IMS is not able to obtain the required storage for the larger PSB work area. However, this abend can also occur as the result of an internal logic error. IMS issues an error message with this abend that provides additional information about the error.

**System action:** The application program terminates abnormally and IMS issues an error message.

**System programmer response:** Look for the error message that IMS issues with this abend for additional information about the cause of this abend.

If the error message does not clarify the problem, check whether the usage of the system storage is abnormally high. An unusually high workload or a problem in IMS or another z/OS subsystem might be temporarily exhausting system storage.

If the amount of available system storage is not the problem, check whether the PSB work area pool is large enough to hold all of the PSB work areas that are required by the number of concurrently scheduled PSBs. After both the HALDB alter operation and Online Change are complete to enable access to the new database changes, you can change the size of the PSB work area pool by changing the PSBW parameter on either the BUFPOOLS system definition macro or the DBC, DCC, or IMS startup procedure.


**Source:** IMS abends


**Module:** DFSDBLP0, DFS5DBB0

**Related concepts:**

 PSB, PSBW, and DMB pool tuning (System Administration)

**Related reference:**

 Parameter descriptions for IMS procedures (System Definition)

 BUFPOOLS macro (System Definition)

---

### 1155

**Explanation:** A MODBLKS data set module was loaded at IMS cold start. Then, the CSVQUERY request was issued to determine the length of the module, but the request failed.

**System action:** IMS abnormally terminates.


**System programmer response:** Use the return code *yyyy* on message DFS4965E to determine why the CSVQUERY request failed and to correct the problem. Then, cold start IMS.

**Problem determination:** 1, 4

**Source:** IMS abends

**Module:** DBFRR00, DFSRDB00, DFSRPG00, DFSRTR00

**Related information:**

 DFS4965E (Messages and Codes)

---

### 1352

**Explanation:** A DBT thread issued a DLI call immediately following a successful commit. However, the SWITCH\_CONTEXT service request failed during the processing of the DLI call.

IMS issues a DFS554A message when this problem is encountered; an example of the message:

```
DFS554A DDLT5J 00001 G PLAPJK04(7) 000,1352 2014 +
/280 09:44:18 RTKN= ODBA00260000000200000001
```

**System action:** The DLI call was not processed and the DBT thread was abnormally terminated.

**System programmer response:** See message DFS554A for additional information to correct the problem.

**Problem determination:** 1, 3, 5, 33, 36

**Module:** DBFIRC10, DFSCPY00

**Related reference:**

 [Parameter descriptions for IMS procedures \(System Definition\)](#)

## 1500

**Explanation:** Recovery was attempted for a database. There was an incompatibility between the task I/O table and the scheduler work areas. A ddname was allocated that could not be located in the task I/O table.

**Analysis:** This is a standard abend issued by module DFSMDA00.

**Possible cause:** IMS system error

**System action:** The IMS control region terminates.

**Programmer response:** No action is required.

**Problem determination:** 4, 35

**Source:** IMS abends

**Module:** DFSMDA00

**Related information:**

Chapter 2, "IMS problem determination," on page 13

## 1501

**Explanation:** Database recovery was attempted for an unsupported DASD device.

**Analysis:** This is a standard abend issued by module DFSMDA00.

**System action:** The IMS control region terminates.

**Programmer response:** Ensure that JCL defining the database is correct, and that you have made no modifications to IMS to attempt to support devices not supported by IMS.

**Problem determination:** 4, 8, 35

**Source:** IMS abends

**Module:** DFSMDA00

**Related information:**

Chapter 2, "IMS problem determination," on page 13

## 1780

**Explanation:** An application call to latch enters a wait while the latch is requested. When the latch is executed, an ITASK posts the application and a check is made to be certain the post code expected by the application matches the post code delivered by the ITASK. When they do not match, the unexpected result causes an ABEND.

**System action:** The dependant region is terminated.

**Programmer response:** No action is required.

**Problem determination:** 4, 35

**Source:** IMS abends

**1780**

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

## Chapter 24. IMS abend codes 2001 - 3000

This topic identifies and explains the following abend codes issued by IMS. Each code is 4 decimal digits.

---

2277

**Explanation:** During OLR commit or backout, buffer purge was called. However, the buffer handler DFSDVBH0 experienced a failure while obtaining ownership for a buffer that is addressed by a buffer queue element BQEL. The block locate from DFSDVBH0 failed and caused a standard abend issued from DFSDVBH0 after label VBHP3160. The reason for the standard abend is that a buffer cannot be written and that the BQEL that points to this buffer cannot be freed during commit or backout.

**System action:** IMS terminates abnormally.

**System programmer response:** Check that the return code from block locate, which can be found in the DL/I trace entry, or in PSTRTCDE. Register two R2 points to the BQEL.

Correct the failure indicated by the block locate return code.

**Problem determination:** 1, 6, 17a through 17e, 11, and 35

**Source:** IMS abends

**Module:** DFSDVBH0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

2476

**Explanation:** DBRC abends when I/O to the RECON data set has been prevented. This abend allows CICS emergency restart (takeover) to proceed.

**System action:** IMS terminates abnormally.

**Source:** IMS abends

---

2478

**Explanation:** The maximum number of waiting PSTs has been exceeded during program isolation processing. The application program abnormally terminates because it must wait for a resource and DFSFXC10 cannot handle any more systems in a WAIT status.

**Analysis:** This is a pseudoabend issued by DFSFXC10.

**System action:** The application program abnormally terminates, and a message processing program (MPP) is automatically rescheduled.

**Operator response:** If the application program was a batch message processing program (BMP), it must be rescheduled at its last sync point.

**Problem determination:** No action is required.

**Source:** IMS abends

**Module:** DFSFXC10

---

2479

**Explanation:** The application program had insufficient storage available for a buffer queue element (BQEL).

**Analysis:** This is a pseudoabend issued by DFSDBH10 and DFSDVSM0.

If the application program is a batch or batch message processing program (BMP), it must be rescheduled from the

## 2480 • 2481

last completed synchronization point. A message processing program (MPP) is automatically rescheduled. If the problem occurs in batch, increase the region size and rerun the job. If the problem occurs repeatedly in a DB/DC environment, specify either a larger CSA, region size, or both, for the IMS control region and restart IMS.

**Possible cause:** CSA or region size was too small.

**System action:** The application program abnormally terminates, and a message processing program (MPP) is automatically rescheduled.

**Operator response:** If the application program was a batch message processing program (BMP), it must be rescheduled at its last sync point.

**Programmer response:** If the problem occurs in batch, increase the region size and rerun the job. If the problem occurs repeatedly in an online environment, specify a larger CSA size or region size, or both, for the IMS control region and restart IMS.

**Problem determination:** No action is required.

**Source:** IMS abends

**Module:** DFSDBH10, DFSDVSM0

---

### 2480

**Explanation:** An internal error in the IMS Database Recovery Control (DBRC) was detected. Message DSP0300I was issued with an error code indicating the nature of the error. Refer to message DSP0300I for a description of the individual error codes.

**System action:** Message DSP0300I indicates whether IMS terminated abnormally (TERM in the message text) or continues (CONT in the message text).

**Programmer response:** Review the error code from message DSP0300I, and if possible correct the problem and rerun the job. Otherwise contact IBM Software Support for problem determination assistance.

**Problem determination:** 2, 3, 4, 8, 35, and your listings of the RECON data sets

**Source:** IMS abends

**Related information:**

DSP0300I

Chapter 2, "IMS problem determination," on page 13

---

### 2481

**Explanation:** The BLDL macro encountered an error while locating the module required to access the Database Recovery Control (DBRC).

**Analysis:** Register 15 contains the return code from the BLDL macro and register 3 contains the BLDL reason code. If the DBRC was installed, determine the cause of the locate error, and be aware that the database log or utility information may need to be entered into the RECON data set since it is not accessible.

The following indicates which module name issued the abend:

**Utility Module Issuing Abend**

**Log Recovery**

DFSULGI0

**DEDB Log Data Set and Check**

DFGLOGC0

**Change Accumulation**

DFSUCMN0

**Online image copy**

DFSUICP0

**System action:** Message DFS2481I precedes this abend. Processing terminates abnormally.

**Programmer response:** Register 15 contains the return code from the BLDL macro and register 3 contains the BLDL reason code. If the Database Recovery Control has been installed, determine the cause of the locate error, and be

aware that the database log or utility information might need to be entered into the RECON data set because it was not accessible.

**Problem determination:** 1, 2, 3, 4, 8, 12, 35

**Source:** IMS abends

**Module:** DFSULGI0, DBFLOGC0, DFSUCMN0, DFSUICP0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

#### 2482

**Explanation:** The DBRC router (DSPCRTR0) has been invoked to perform a function for IMS while a previously invoked functional request is in progress. DBRC detected a programming error on the part of the caller. DBRC cannot perform multiple concurrent functional requests.

**Analysis:** Recursive calls to DBRC are not allowed. While multi-tasking, the caller must ensure that a request for DBRC service completes before another request is initiated.

**System action:** IMS terminates abnormally.

**Programmer response:** Keep all diagnostic information and contact the IBM Support Center for problem determination assistance.

**Problem determination:** 2, 3, 4, 8, 11, 35

**Source:** IMS abends

**Module:** DSPCRTR0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

#### 2483

**Explanation:** The DBRC loader (DSPLOADR) has been invoked to resolve a call to a DBRC module. Either a nonexistent entry point within a known DBRC load module or an entry point within an unknown load module was called. The index of the called module in the DBRC router storage shows an undefined module, so the module invocation is invalid.

**Analysis:** DBRC contains a mechanism that can intercept a call to an entry point in a DBRC load module that is not currently in storage. DBRC ensures that the needed load module is loaded before the call is completed. Abend 2483 indicates that a call to an unknown entry point occurred. This generally occurs as a result of an error in the System Modification Program (SMP) process. The user failed to compile again or properly bind a module affected by a recent change in DBRC's "call intercept" mechanism.

**System action:** IMS terminates abnormally.

**Programmer response:** Keep all diagnostic information and contact the IBM Support Center for problem determination assistance.

**Problem determination:** 2, 3, 4, 8, 11, 35

**Source:** IMS abends

**Module:** DSPLOADR

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

#### 2484

**Explanation:** An IMS Fast Path related control block could not be obtained by using the DFSBCB macro. Register 2 contains the identification of the failing control block.

**Analysis:**

The program status word (PSW) at entry-to-abend points to the instruction within the module from which the abend

## 2485 • 2486

(SVC 13) is issued. Register 12 in the abend SVRB is the base register for the module issuing the abend.

The DFSBCB macro with FUNC=GET issues a nonzero return code indicating that the requested control block could not be obtained. Register 2 in the abend SVRB registers contains an EBCDIC code indicating what control block could not be obtained.

This abend is also issued by the FP notify task module (DBFICLI0).

Key	Label	Description
Reg2=C'AWE'		The asynchronous work element (AWE) could not be obtained.
Reg2=C'ADSC'		The area data set control block (ADSC) could not be obtained.
Reg2=C'FNCFB'		The FNCFB could not be obtained.
Reg2=C'FPCP'		The FPCP could not be obtained.
Reg2=C'FNCFB'		The FNCFB could not be obtained.
Reg2=C'L56X'		The L56X block could not be obtained.
Reg2=C'STOR'		The requested storage could not be obtained

**System action:** The IMS control region terminates.

**Programmer response:** Provide more CSA space and emergency restart IMS to resume processing.

**Source:** IMS abends

**Module:** DBFBNPI0, DBFBPNS0, DBFBPN10, DBFBPN50, DBFBPN70, DBFBN80, DBFDIDT0, DBFDT300, DBFICLI0, DBFMREQ0, DBFMFLG0, DBFMLBI0, DBFMLBR0, DBFMLOP0, DBFNRS0, DBFUMER0, DBFARD10, DBFARD30, DBFARD40, DBFCST00, DBFVXCS0, DBFVXCS0, DBFICL40

---

### 2485

**Explanation:** A DMAC exclusive latch hold using the DBFLATCH macro failed because it already exists.

**Analysis:** This is a standard abend issued from the Fast Path Enhancement DBFLATCH macro. The program status word (PSW) at entry-to-abend points to the instruction within DBFLATCH from which the abend (SVC 13) is issued.

Register 12 in the abend SVRB is the base register for the module issuing the DBFLATCH macro.

When the DBFLATCH macro with TYPE=GEXC is executed, the DMACSLWC field shows that the DMAC exclusive latch is already held. Only one DMAC exclusive latch hold at a time is allowed.

Key	Label	Description
	DMACSLWC	If bit 0 is on, a DMAC exclusive latch is held.

**System action:** The IMS control region terminates.

**Programmer response:** Because this error condition should not occur, save the memory dump.

**Problem determination:** 11, 35

**Source:** IMS abends

**Module:** DBFMLOP0, DBFMLCL0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

### 2486

**Explanation:** A notify request for internal stop area or ADS failed because of a severe error in IMS or IRLM and a return code was received from IRLM.

**System action:** IMS terminates abnormally.

**Programmer response:** Correct the problem determined by the return code, and restart IMS. If you are unable to correct the problem, contact IBM for further assistance.



**Problem determination:** 1, 3, 4, 10, 35

**Source:** IMS abends

**Related information:**

Chapter 2, "IMS problem determination," on page 13

## 2488

**Explanation:** A logic error occurred in either IMS or DBRC when DBRC was called to set/reset the DEDB area data set available flag.

**Analysis:** This is a standard abend issued from the Fast Path emergency restart modules, DBFERST0, DBFEAIS0, or DBFE2CI0. The program status word (PSW) at entry-to-abend points to the instruction within DBFERST0, from which the abend (SVC 13) is issued.

- Register 12 in the abend SVRB is the base register.
- Register 15 contains the following codes and their meanings.

### Code (hex)

	Meaning
X'0C'	The return code set by DBRC indicated that the area was not registered in the RECON data set. Since the area data set status call is only issued if the area is registered to DBRC, an internal error has occurred.
X'10'	The return code set by DBRC indicated that the area data set was not registered in the RECON data set. Since the area data set status call is only issued of the area data set registered to DBRC, an internal error has occurred.
X'2C'	The return code set by DBRC indicated that an internal DBRC error occurred during the processing of the area data set status call. Refer to the message issued by DBRC prior to this abend for further explanation.
X'30'	The return code set by DBRC indicated that a required parameter was not passed to DBRC.

**System action:** IMS terminates abnormally.


**Programmer response:** Correct the problem associated with DBRC, and re-execute the job. Refer to the return codes in DBRC request return codes information to determine the cause of the failure.

**Problem determination:** 1, 4, 6, 35

**Source:** IMS abends

**Module:** DBFERST0, DBFEAIS0, DBFE2CI0

**Related reference:**

 DBRC request return codes (Messages and Codes)

**Related information:**

Chapter 2, "IMS problem determination," on page 13

## 2489

**Explanation:** An asynchronous work element (AWE) enqueue command, issued to the Fast Path common service ITASK routine to stop the DEDB area, failed because of an internal error.

**Analysis:** This is a standard abend issued from the Fast Path Enhancement DEDB second CI update processor (DBFMOCIO), or the Fast Path XRF area cleanup module (DBFEACL0). The program status word (PSW) at entry-to-abend points to the instruction within DBFMOCIO from which the abend (SVC 13) is issued. Register 12 in the abend SVRB is the base register.

Prior to this abend, the following MTO message is issued.

```
DFS3717I DEDB AREA CONTROL BLOCK NOTIFY
         FAILURE - AREA STOPPED. DBD=dbname
         AREA=areaname
```

**System action:** IMS terminates abnormally.

**Programmer response:** Refer to the message explaining the reason for stopping the area issued prior to this abend. Run the Database Recovery utility, DFSURDB0, to recover the area and then rerun the job. If you are unable to determine the problem, proceed with the problem determination to determine the cause of the failure.

**Problem determination:** 1, 4, 6, 35

## 2490

**Source:** IMS abends

**Module:** DBFMOCIO, DBFEACLO

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

## 2490

**Explanation:** An internal DBRC error occurred while running DBRC in a Base Primitive Environment. Register 15 contains a code that identifies the reason for the abend.

*Table 10. Subcodes and explanations on register 15*

Subcode	Explanation
X'0010'	An error occurred during storage allocation. Register 14 contains the return code from BPEGETM.
X'0020'	An error occurred while preloading a module.
X'0024'	An error occurred during a call to BPELOAD.
X'0110'	The BPEPARSE work area could not be allocated.
X'0120'	An error occurred while reading the DSPBIxxx member.
X'0130'	An error occurred while parsing the DBRC runtime parameter.
X'0134'	An error occurred while parsing the DSPBIxxx member.
X'0140'	An error occurred during parameter validation of the DSPBIxxx or DBRC EXEC parameters.
X'0210'	An error occurred during true open of the RECON data set.
X'0220'	The IMSplex name that IMS is using is different from the name that DBRC is using.
X'0230'	An error occurred storing DBRC information into the IMS Resource Manager Parameter list block (IRMP).
X'0240'	An error occurred during System Management Facility (SMF) registration.
X'0250'	The IMS System Contents Directory (SCD) was not found.
X'0260'	The job name does not match the DBRCNM parameter.
X'0270'	DBRC-dependent region is already started.
X'0280'	SVC INIT failure.
X'0290'	Cross memory POST failure.
X'0610'	Group Services latch obtain failure. Register 14 contains the return code from BPELAGET.
X'0620'	Group Services latch release failure. Register 14 contains the return code from BPELAREL.
X'0630'	Group Services error getting an AWE. Register 14 contains the return code from BPECBGET.
X'0640'	Group Services AWE enqueue error. Register 14 contains the return code from BPEAWSRV.
X'0650'	Group Services AWE wait error. Register 14 contains the return code from BPEWAIT.
X'0660'	Group Services query IMSplex failure.
X'0670'	Group Services failure enqueueing the GRP2 AWE for a local quiesce.
X'0680'	Group Services error from call to SCI indicating that DBRC is ready to receive messages.
X'0690'	Group Services error broadcasting message to a specific DBRC or all DBRCs.

**System action:** IMS terminates abnormally.

**Programmer response:** If messages DSP1125A, DSP1140A, or DSP2002E were issued, review the error codes, and if possible correct the problem and rerun the job. Otherwise, contact IBM Software Support for problem determination assistance.

**Problem determination:** 2, 3, 4, 8, 35, and your listings of the RECON data sets. Contact your security administrator to gain authority to the identified DBRC resource, if necessary.

**Source:** IMS abends

**Related information:**

 DSP1125A (Messages and Codes)

 DSP1140A (Messages and Codes)

 DSP2002E (Messages and Codes)

Chapter 2, “IMS problem determination,” on page 13

## 2496

**Explanation:** An internal request for a DEDB area lock failed because of a severe error in IMS or IRLM.

**Analysis:** This is a standard abend issued by DBFARD10, DBFARD20, DBFARD50, DBFCMOC2, and DBFEAIS0. The program status word (PSW) at entry-to-abend points to the instruction following the abend svc in the failing module. Register 12 in the abend SVRB is the base register.

**For DBFARD10, DBFARD20:**

- RC X'14' is returned from IRLM and LMREJECT set. The lock is owned by a failed IMS (retained lock). This bad IRLM return code is an internal stop ads command, and other sharing IMS systems other than the failed lock owner existed. In this case, Register 15 contains PSTLRXRC and PSTLRXFB, and reflects the IRLM notify call and not the initial RC X'14' from IRLM.
- Any other failure return code from IRLM, including X'14' and LMREJECT, is not set (not a retained lock).

**For DBFCMOC2:**

- RC X'14' from IRLM and LMREJECT is set. The lock is owned by a failed IMS (retained lock) or DMACF7F0 (force close and open) is set for an area that contains SDEPs and DBFARD50 is not a deferred close.
- Any other failure return code from IRLM, including X'14' and LMREJECT (retained lock) and DMACF7F0 (force close and open), is not set.

**For DBFEAIS0:**

- RC X'14' from IRLM and LMREJECT are set (retained lock), but the area is not shared and DBFEAIS0 is not an XRF TKO.
- Any other failure return code from IRLM, including when X'14' and LMREJECT, is not set (not a retained lock). See IRLM request return and reason codes information.

**System action:** IMS terminates abnormally.


**Programmer response:** Correct the problem determined by the return code, and restart IMS. If you are unable to correct the problem, contact IBM Software Support for assistance.

**Problem determination:** 1, 3, 4, 10, 35

**Source:** IMS abends

**Module:** DBFARD10, DBFARD20, DBFARD50, DBFEAIS0, DBFCMOC2

**Related tasks:**

 IRLM return and reason codes

**Related information:**

Chapter 2, “IMS problem determination,” on page 13

## 2610

**Explanation:**

An error occurred in the IMS batch dispatcher. This abend is for batch IMS only.

**Analysis:**

This is a standard abend issued by the IMS batch dispatcher (DFSKBDP0) for errors in initialization and SCP waits. Register 15 contains one of the following reason codes.

**Code    Meaning**

## 2763 • 2800

- X'10'** An IMODULE GETMAIN for dispatcher work area failed. Register 3 contains the length requested. Register 5 contains the IMODULE return code.
- X'11'** A call was made to initialize an ECB as waiting, but the batch wait list is full. Register 2 contains the list size. Register 8 points to the ECB.
- X'12'** A call was made to initialize an ECB as waiting, but the ECB was already initialized. Register 3 contains the ECB address.
- X'13'** Unable to obtain a TCB table entry during initialization. Register 3 contains the address of the TCB table prefix from the SCD.
- X'14'** A call was made to the SCP WAIT routine, but an ITASK was already in an SCP WAIT. Register 7 contains the ECB address of the current ITASK trying to WAIT. Register 6 contains the caller's return address.

**Source:** IMS abends

**Module:** DFSKBDP0

---

### 2763

**Explanation:** During an IMS restart (warm start or emergency restart), the Fast Path normal restart processor, DBFNRS0, read the control records from the MSDB checkpoint data sets to determine from which checkpoint to restart the MSDBs, but the selected checkpoint was not found on the IMS log.

**Analysis:** This abend is issued for one of the following conditions:

1. None of the MSDB checkpoints on MSDBCP1 or MSDBCP2 is complete. The ID field in the control record of the MSDB checkpoint data set is not equal to 'MSDBCHPT'.
2. Both of the MSDB checkpoints on MSDBCP1 and MSDBCP2 are earlier than the checkpoint on the X'4001' log record.
3. The checkpoint on MSDBCP2 is equal to or later than the checkpoint on MSDBCP1, and the checkpoint on MSDBCP1 is not complete or is earlier than the checkpoint on the X'4001' log records.
4. No MSDB checkpoint control record is found on the checkpoint data set. A GET for the control record causes an EODAD routine to be entered in DBFNRS0 and abend 2763 is issued. This could occur because the previous IMS cold start was performed without MSDBs specified.

**System action:** IMS terminates abnormally.

**Programmer response:** Look at the IMS console sheet and identify:

1. The earlier of the latest 2 MSDB checkpoints on MSDBCP1 and MSDBCP2, and
2. A system checkpoint that matches or is before that MSDB checkpoint.

Specify the system checkpoint for the restart.

**Problem determination:** 35

**Source:** IMS abends

**Module:** DBFNRS0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

### 2800

**Explanation:** IMS initialization encountered an error during global online change related initialization. The abend subcode provides the reason for the failure.

**Analysis:** IMS initialization terminates abnormally with this abend code. Register 15 contains the abend subcode. The possible return subcodes are:

#### Codes

##### Explanation

#### X'01'

The OLCSTAT data set allocate or deallocate failed. Message DFS2848E, which is issued before this abend, displays associated return codes and information related to the failure.

#### X'02'

The OLCSTAT data set open, read, write, or close failed. Message DFS2843E, which is issued before this abend, displays associated return codes and information related to the failure.

**X'03'**

The OLCSTAT data set access request failed because of an internal error.

**X'07'**

The OLCSTAT data set contents are invalid. Message DFS2844E, which is issued before this abend, displays associated return codes and information related to the failure.

**X'08'**

The operator replied CANCEL to message DFS2845A to cancel IMS initialization after a resource definition inconsistency was detected.

**X'09'**

The OLCSTAT data set name defined by this IMS is inconsistent with the OLCSTAT data set name defined by other IMS systems in the IMSplex. All of the IMS systems in the IMSplex must define the same OLCSTAT data set name.

**X'10'**

Global online change is in progress. IMS cannot initialize while global online change is in progress.

**System action:** IMS initialization terminates abnormally with this abend code.

**Programmer response:** For the OLCSTAT data set abend subcodes (0001, 0002, and 0007), correct the OLCSTAT data set error and retry IMS initialization. For abend subcode 0003, save the IMS dump and contact your IBM support center. For abend subcode 0009, define the correct OLCSTAT data set name to this IMS and retry IMS initialization. For abend subcode 0010, the OLCSTAT data set is locked because global online change is in progress. Wait until an INITIATE OLC PHASE (COMMIT) command or TERMINATE OLC command completes successfully and unlocks the OLCSTAT data set before restarting IMS. In the case where all of the IMS systems abended during global online change, no IMS is up to process an INITIATE OLC command or a TERMINATE OLC command. In this case, you must run the Global Online Change Utility (DFSUOLC0) with the unlock function to unlock the OLCSTAT data set, before attempting to restart IMS.

**Source:** IMS abends

**Module:** DFSIIOC0, DFSCSL40

**Related information:**

 DFS2843E (Messages and Codes)

 DFS2844E (Messages and Codes)

 DFS2845A (Messages and Codes)

**2801**

**Explanation:** IMS restart encountered an error related to global online change. The abend subcode provides the reason for the failure.

**Analysis:** IMS restart terminates abnormally with this abend code. Register 15 contains the abend subcode. The possible return subcodes are:

**Code    Meaning**

**X'01'**    The OLCSTAT data set allocate or deallocate failed. Message DFS2848E, which is issued before this abend, displays associated return codes and information related to the failure.

**X'02'**    The OLCSTAT data set open, read, write, or close failed. Message DFS2843E, which is issued before this abend, displays associated return codes and information related to the failure.

**X'03'**    The OLCSTAT data set access request failed because of an internal error.

**X'07'**    The OLCSTAT data set contents are invalid. Message DFS2844E, which is issued before this abend, displays information related to the failure.

**X'10'**    Global online change is in progress. Global online change was initiated since this IMS initialized.

**X'11'**    The modify ID of this IMS at the end of restart is not in sync with the modify ID in the OLCSTAT data set.

For a tracking IMS, such as an XRF alternate that is taking over, a DBCTL warm standby that is doing an emergency restart, or an FDR region that is doing fast database recovery, this IMS is not in sync with the OLCSTAT data set. This is probably because an X'70' log record was not successfully logged by the last Global Online Change.

For restarting IMS, Global Online Change might have occurred since this IMS initialized.

## 2990

**X'12'** The IMS restart command conflicts with the last global online change.

**X'13'** Global Online Change is not supported. Global Online Change is not supported for RSR tracker.

**System action:** IMS initialization terminates abnormally with this abend code.

**Programmer response:** For the OLCSTAT data set abend subcodes (0001, 0002, and 0007), correct the OLCSTAT data set error and try IMS initialization again. For abend subcode 0003, save the IMS dump and contact IBM software support. For abend subcode 0010, wait until global online change is finished before trying IMS initialization. For abend subcode 0011, try IMS initialization again. If an XRF alternate takeover abended or the DBCTL warm standby abended, perform an emergency restart of the IMS active. For abend subcode 0012, try IMS initialization again. Then issue an IMS restart command that does not conflict with the last global online change. For abend subcode 0013, define DFSCGxxx OLC=LOCAL and restart IMS.

**Source:** IMS abends

**Module:** DFSOLCSO

**Related information:**

[DFS2843E \(Messages and Codes\)](#)

[DFS2844E \(Messages and Codes\)](#)

[DFS2848E \(Messages and Codes\)](#)

---

## 2990

**Explanation:** The IMS Segment Edit/Compression routine DFSCMPX0 has detected an error while attempting to perform compression or expansion services from your edit/compression routine, which encountered an error while attempting compression or expansion services.

**Analysis:** The PSW points to the instruction within the module from which the abend, SVC 13, is issued. Register 14 contains the error reason code set by the detecting module. This error reason code corresponds to the label name at the failing location. Register 10 contains the address of the DMBCPAC (DBFCMPC if Fast Path) control block. Register 9 contains the address of the work area of the routine.

Key	Label	Description
Reg14=D4D7E701	D4D7E701	During a compression request, the input length of the variable length segment is less than 2 bytes. Correct the segment data.
Reg14=D4D7E702	D4D7E702	During an expansion request, the input length of the compressed segment is less than 2 bytes.
Reg14=D4D7E703 Reg15=return code	D4D7E703	During an expansion request, a non-zero return code was returned by the z/OS expansion Service (CSRCEsrv).
Reg14=D4D7E704	D4D7E704	INIT was not specified in the COMPRTN=parameter of the SEGM statement.
Reg14=D4D7E705 Reg6=function code	D4D7E705	Invalid function code.
Reg14=D4D7E706 Reg1=length plus offset of sequence field	D4D7E706	The segments sequence field is not completely within the segment.
Reg14=D4D7E707	D4D7E707	Input length is negative. Correct the segment data.

**System action:** The application ends abnormally.


**Programmer response:** Ensure the edit/compression routine has been built and linked properly.

**Problem determination:** 4, 17a, 17b, 18, 19, and the source/link listings of the edit/compression routine

**Source:** IMS abends

**Module:** DFSCMPX0

**Related reference:**

 z/OS: CSRCEsrv return and reason codes

**Related information:**

Chapter 2, “IMS problem determination,” on page 13

**2991**

**Explanation:** The IMS Segment Edit/Compression routine DFSKMPX0 has detected an error while attempting to perform compression or expansion services. Your edit/compression routine might have encountered an error while attempting compression or expansion services.

**Analysis:** The PSW points to the instruction within the module from which the abend, SVC 13, is issued. Register 14 contains the error reason code set by the detecting module. This error reason code corresponds to the label name at the failing location. Register 10 contains the address of the DMBCPAC (DBFCMPC if Fast Path) control block. Register 4 contains the address of the routine's work area.

Key	Label	Description
Reg14=D4D7E701 Reg6=function code	D4D7E701	Invalid function code.
Reg14=D4D7E702 Reg1=length plus offset of sequence field	D4D7E702	The segments sequence field is not completely with the segment.
Reg14=D4D7E703	D4D7E703	Input length is negative. Correct the segment data.
Reg14=D4D7E704	D4D7E704	During a compression request, the input length of the variable length segment is less than two bytes. Correct the segment data.
Reg14=D4D7E705	D4D7E705	INIT was not specified in the COMPRTN= parameter of the SEGM statement.
Reg14=D4D7E706	D4D7E706	During an expansion request, the input segment contained a sequence field but one is not defined in the SEGM statement.
Reg14=D4D7E707	D4D7E707	During an expansion request, a partial expansion was done.
	D4D7E708	During the expansion of a segment, the expansion length will exceed the maximum segment size. This occurrence may be caused by a DBD mismatch.

**System action:** The application ends abnormally.

**Programmer response:** Ensure the edit/compression routine has been built and linked properly.

**Problem determination:** 4, 17a, 17b, 18, 19, and the source/link listings of the edit/compression routine.

**Source:** IMS abends

**Module:** DFSKMPX0

**Related information:**

Chapter 2, “IMS problem determination,” on page 13

**3000**

**Explanation:** An error occurred while an XRF alternate system or a Fast Database Recovery region was setting up the appropriate data set (ACBLIB, FMTLIB, MODBLKS), or while it was processing an online change record. A DFS33xx or DFS34xx message indicating the error accompanies this abend.

**Analysis:** This abend is a standard abend issued from module DFSICV90.

## 3000

An error occurred on an XRF alternate system while processing a type 4001 or type 70 log record.

DFSICV90 issues the abend because of a nonzero return code passed by a lower-level module. The return code is used to send the accompanying message before the abend is issued. If the abend occurs before the message is received at the IMS or z/OS console, register 7 can be used to locate the message number.

Register 7 at the abend is the pointer to the message AWE. The AWE contains either the readable pre-edit message text beginning at offset X'30' (in other words DFS3488), or a halfword containing the keyed message number at offset X'2E'.

**System action:** IMS terminates abnormally.

**Programmer response:** Issue the /CHECKPOINT command on the active system to take a simple checkpoint; verify from the DFS3804I message that the latest restart checkpoint time stamp is after the online change, then restart the XRF alternate or FDBR region. Alternatively, correct the condition indicated in the accompanying message.

**Problem determination:** 1, 4, 5, 6

**Source:** IMS abends

**Module:** DFSICV90

**Related information:**

 DFS3804I (Messages and Codes)

Chapter 2, "IMS problem determination," on page 13



---

## Chapter 25. IMS abend codes 3001 - 3100

This topic identifies and explains the following abend codes issued by IMS. Each code is 4 decimal digits.

---

### 3006

**Explanation:** A directory block was read from the active format data set library which was greater than 256 bytes in length.

**Analysis:** This is a standard abend issued from module DFSUTL40. Message DFS1033I accompanies this abend. The registers in the abend SVRB should be used for problem isolation.

Key	Label	Description
Reg7=directory block address		The first 2 bytes of the directory block contain the length of the block. These 2 bytes contain a value greater than 256 (X'0100').

**System action:** Phase 2 of the MFS Language utility terminates abnormally.

**Programmer response:** Locate the block that was read in and determine why the length value is greater than 256.

**Problem determination:** 1, 2, 4, 16

**Source:** IMS abends

**Module:** DFSUTL40

**Related information:**

 DFS1033I

Chapter 2, "IMS problem determination," on page 13

---

### 3007

**Analysis:** The MFS Language utility Phase 2 processor has received an unexpected completion code from the host system STOW function.

**Explanation:** This is a standard abend issued from module DFSUTL40. Message DFS1012I, DFS1031I, or DFS1032I accompanies this abend.

The registers in the abend SVRB should be used for problem isolation. The contents of STOWCODE should be examined to determine the return code from the STOW macro instruction.

The STOW macro (SVC 21) has been issued to update a partitioned data set directory by adding, changing, replacing, or deleting an entry in the directory. STOW passes a return code in register 15. This return code is saved in DFSUTL40 within a halfword at label STOWCODE.

#### Codes

##### Explanation

X'00'

Update of the directory was completed successfully.

X'04'

Duplicate block — directory already contains the specified name.

X'08'

Undefined for STOW addition — the specified name could not be found.

X'0C'

There was no space left in the directory.

## 3008

### X'10'

Permanent I/O error occurred when attempting to update the directory.

### X'14'

The specified DCB is not open or is opened incorrectly.

### X'18'

A conditional GETMAIN with STOW was unsuccessful.

The program status word (PSW) at entry-to-abend points to label STOWABND from which abend (SVC 13) is issued.

The message that accompanies this abend is determined by the return code in STOWCODE and the type of STOW being done. If a directory entry is to be added (type A) and the return code is X'08', X'14', or X'18', message DFS1012I is issued. If a directory entry is to be replaced (type R) and the return code is X'04', X'08', X'14' or X'18', message DFS1012I is issued again. Message DFS1031I is issued if the STOW return code is X'0C'. Message DFS1032I is issued if the return code is X'10'.

Key	Label	Description
STOWCODE≠0	STOWIT1	A type A STOW macro has been issued and it passed back a nonzero return code. This is an error, so the abend is issued.
STOWCODE=X'04'	DUPBLOCK	A type A STOW has failed, so a branch is taken to this routine to issue a type R STOW. A nonzero return code was passed from the type R STOW and the abend is issued.
STOWCODE=X'0C'	DIRERR	A type A or type R STOW has failed, message DFS1031I is issued and the abend is taken.
STOWCODE=X'10'	PERMIO	A type A or type R STOW has failed, message DFS1032I is issued and the abend is taken.
STOWCODE=X'08', X'10', X'14', or X'18'	STOWERR	A type A or R STOW failed. Message DFS1012I is issued and the abend is taken.

**System action:** The MFS Language utility terminates abnormally.

**Programmer response:** Examine the "UTPRINT" output from Phase 2 to determine the failure. See also messages DFS1030I, DFS1031I, DFS1032I, and DFS1012I. Determine the MFS Format library status, and resubmit the job.

**Problem determination:** 1, 2, 4, 16

**Source:** IMS abends

**Module:** DFSUTL40

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

## 3008

**Explanation:** Either the MFS Service utility or Phase 2 of the MFS Language utility program received an unexpected return code from the host system function, DEVTYPE or RDJFCB, or the OPEN macro.

At the time of the abnormal termination, register 7 contains a function code indicating the cause of the failure. Register 14 contains the address of the point at which the unexpected return code was encountered. The function codes are as follows:

### Code (hex)

#### Explanation

**04** One of the following error conditions applies to the data set represented by the FORMAT statement:

- The DD statement is missing.
- The DD statement specifies DUMMY.

- 08** This indicates an unsuccessful initial opening of one or more of the data sets represented by the DD FORMAT, UTPRINT, or SEQBLKS DD statements.
- 0C** This indicates an unsuccessful secondary opening of one or more of the data sets represented by the DD FORMAT, or SEQBLKS statements.
- 10** This indicates an unsuccessful final opening of the data set represented by the DD DEQBLKS statement.
- 14** This indicates an unsuccessful DEQ or ENQ of the data set requested by the DD FORMAT statements.

**Analysis:** This is a standard abend issued from module DFSUTL40.

The registers in the abend SVRB should be used for problem isolation. The program status word (PSW) at entry-to-abend points to the instruction within label BADOPEN from which the abend (SVC 13) is issued.

At label CHKDISP1, DFSUTL40 branches and links registers (BALRs) to DFSIDDP0 to validity check the DD statement with the ddname for "FORMAT". If DFSIDDP0 finds an error, it passes the return code back in register 15. Register 7 is set to a value indicating the reason code and should be used to isolate to a label.

The nonzero return codes from DFSIDDP0 are:

#### Codes

##### Explanation

**X'04'**

ddname not found (DEVTYPE)

**X'08'**

DEVTYPE parameter error

**X'10'**

DD DUMMY specified

Key	Label	Description
Reg7=X'04' Reg14=BAL	CHKDISP	This routine is validity checking the DD statement with the ddname of "FORMAT". It branches and links registers (BALRs) to DFSIDDP0, which passes a nonzero return code in register 15. This is an error and the abend is issued.
Reg7=X'08' UTPRINT+DCBOFLGS ^= X'10'	NQDONE:	Validity checking of the FORMAT DD statement DISP is done and this routine does a GETMAIN for subpool 15 and then issues the DEVTYPE macro. It next issues an OPEN for the FORMATIN, FORMATOU, UTPRINT, & SEQBLKS data sets. It then does a series of tests to see if the data sets are opened. If any of them are not, the abend is issued.
FORMATIN+DCBPFLGS ^= X'10'	NQDONE:	Validity checking of the FORMAT DD statement DISP is done and this routine does a GETMAIN for subpool 15 and then issues the DEVTYPE macro. It next issues an OPEN for the FORMATIN, FORMATOU, UTPRINT, & SEQBLKS data sets. It then does a series of tests to see if the data sets are opened. If any of them are not, the abend is issued.
FORMATOU+DCBOFLGS ^= X'10'	NQDONE:	Validity checking of the FORMAT DD statement DISP is done and this routine does a GETMAIN for subpool 15 and then issues the DEVTYPE macro. It next issues an OPEN for the FORMATIN, FORMATOU, UTPRINT, & SEQBLKS data sets. It then does a series of tests to see if the data sets are opened. If any of them are not, the abend is issued.
SEQBLKS+DCBOFLGS ^= X'10'	NQDONE:	Validity checking of the FORMAT DD statement DISP is done and this routine does a GETMAIN for subpool 15 and then issues the DEVTYPE macro. It next issues an OPEN for the FORMATIN, FORMATOU, UTPRINT, & SEQBLKS data sets. It then does a series of tests to see if the data sets are opened. If any of them are not, the abend is issued.
Reg7=X'0C' FORMATOU+DCBOFLGS ^= X'10' SEQBLKS+DCBOFLGS ^= X'10'	REOPEN	The FORMATOU and the SEQBLKS data set are now to be reopened for compressing the FORMAT data set. If the OPEN is unsuccessful, the abend is issued.
Reg7=X'10' SEQBLKS+DCBOFLGS ^= X'10'	EXIT	This routine performs the termination housekeeping. It issues a FREEMAIN to free up subpool 15. A CLOSE is then issued for the FORMATIN, FORMATOU, UTPRINT, and SEQBLKS data set. The SEQBLKS data set is then reopened for output to clear the data set. If the OPEN is unsuccessful, the abend is issued.

### 3009

Key	Label	Description
Reg7=X'14'	NQFMTLIB RETURN	Module DFSMODE0 failed to enqueue on the target format library (ddname=FORMAT) or module DFSMODE0 failed to dequeue the target format library (ddname=FORMAT).

**Programmer response:** Examine the data set represented by the SYSUDUMP DD statement and the JCL for the phase 2 step to locate SYSTEM IECnnn OPEN ERROR MESSAGES.

**Problem determination:** 1, 2, 4, 16

**Source:** IMS abends

**Module:** DFSUTL40

**Related information:**

Chapter 2, "IMS problem determination," on page 13

### 3009

**Explanation:** The MFS Language utility phase 2 processor has encountered an invalid record or record combination in the data set represented by the SEQBLKS DD statement. This data set is constructed by the Phase 1 processor.

**Analysis:** This is a standard abend issued from module DFSUTL40.

Use the registers in the abend SVRB for problem isolation. The register 14 BAL can be used to isolate to a particular label except in the third instance where the abend is detected. The program status word (PSW) at entry-to-abend points to label NOTBLOCK, from which the abend (SVC 13) is issued.

Key	Label	Description
Reg14=BAL STOWBLLEN=Blocksize VBSTYPE ^= X'01' REG= VBSCTL (contains VBSTYPE)	ISRGETC2	The module control table (MCT) indicates a format block is to be obtained from SEQBLKS data set and replaced in the FORMAT library. When SEQBLKS is read, a format block record is not found, indicating an internal logic error. This routine has branches and links to the GET routine to obtain the block record for the SEQBLKS data set. A comparison is made to see if the phase-1 type record byte contains a return code of X'01', the identifier for a format control block record. If it does not, the abend is issued.
Reg10= VBSCTL Reg14=BAL VBSTYPE ^= X'01'	ISRGETC3	The module control Table (MCT) indicates a format block is to be obtained from SEQBLKS data set and replaced in the FORMAT library. When SEQBLKS is read, a format block record is not found, indicating an internal logic error. The GET routine has been branched and linked in order to obtain the block record from the SEQBLKS data set. A comparison is made to ensure that the phase-1 type record byte contains an X'01', the identifier for a format control block record. If it does not, the abend is issued.
Reg11=Phase to Phase table (PPT) PPTSW1 ^= X'40'	ISRTEOF	EODAD was encountered while inserting blocks into the format PDS. A test is made to see if the search was for a control record. If not, the abend is issued.

**System action:** The MFS Language utility terminates abnormally.

**Operator response:** Rerun the job, allocating the SEQBLKS DD statement to a tape unit.

**Programmer response:** No action is required.

**Problem determination:** 1, 2, 4, 16, 35

**Source:** IMS abends

**Module:** DFSUTL40

**Related information:**

Chapter 2, "IMS problem determination," on page 13

**3010**

**Explanation:** An error has been detected during the construction of the ITB tree. The failure could be one of the following:

- Insufficient work storage requirements were calculated
- Invalid hierarchic sequence error in input source

**Analysis:** This is a standard abend issued from module DFSUTL80 at label CATSTROP. The program status word (PSW) at entry-to-abend points to the abend (SVC 13) at this label.

Use the registers in the abend SVRB for problem isolation.

Key	Label	Description
Reg2=hierarchic stack entry size Reg3=end of hierarchic stack Reg4=current position in hierarchic stack Reg4>Reg3	CATSTROP	Insufficient work storage requirements were calculated for a hierarchic stack to build the condensed representation of the hierarchic contents of the module.
Reg5=0 Reg6=current ITB Reg8=current hierarchic level index	CATSTROP	Invalid-hierarchic-sequence error in input source. The ITB module entry for the parent (register 5) had not been processed when the current ITB module entry (register 6) was being processed.

**System action:** The MFS Language utility terminates abnormally.

**Problem determination:** 1, 2, 4, 16, 35

**Source:** IMS abends

**Module:** DFSUTL80

**Related information:**

Chapter 2, "IMS problem determination," on page 13

**3011**

**Explanation:** The contents of the MSG ITB in the IMS.REFERAL library are not consistent with the output message indication in the IMS.REFERAL directory entry.

**Analysis:** This is a standard abend issued from module DFSUTLA0.

Use the registers in the abend SVRB for problem isolation. The program status word (PSW) at entry-to-abend points to the instruction within label HORRORS from which the abend (SVC 13) is issued.

Key	Label	Description
Reg4= Module Control Table Entry (MCTE) in use MCTETYPE=MCTETMSO MSGFLAG1=MSG1INP Reg5= MSGITB in use Reg11= PPT Table PPTWCNAM=MSGITB name PPTMCT= Module Control Table (MCT) in use	HORRORS	The MCTETYPE field created from the REFERAL directory entry indicates that the message is for input, but the check of the message ITB by DFSUTLB0 indicates that the message is for output.

**Programmer response:** Perform the following steps:

1. Run the MFS utility with the SCRATCH REFER=*msgname* control statement.
2. Run the MFS Language utility and reassemble the output MSG. This will place the output MSG ITB into the IMS Referral library and will set output message indication in the directory entry.

**Problem determination:** 1, 2, 4, 16, 35

## 3012 • 3013

**Source:** IMS abends

**Module:** DFSUTLA0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

### 3012

**Explanation:** The contents of the MSG ITB in the IMS REFERAL library are not consistent with the input message indication in the IMS.REFERAL directory entry.

**Analysis:** This is a standard abend issued from module DFSUTLB0.

Use the registers in the abend SVRB for problem isolation. The program status word (PSW) at entry-to-abend points to the instruction within label HORRORS from which the abend (SVC 13) is issued.

Key	Label	Description
Reg5= MSGITB in use Table PPTWCNAM=MSGITB name PPTMCT= Module Control Table (MCT) in use	Reg11= PPT HORRORS	The MCTETYPE field created from the REFERAL directory entry indicates that the message is for input, but the check of the message ITB by DFSUTLB0 indicates that the message is for output.
Reg4= Module Control Table (MCT) in use MCTETYPE=MCTETMSI MSGFLAG1 ^= MSG1INP		

**System action:** IMS terminates abnormally.

**Programmer response:** Perform the following steps:

1. Run the MFS utility with the SCRATCH REFER=*msgname* control statement.
2. Run the MFS Language utility and reassemble the input MSG. This will place the input MSG ITB into the IMS REFERRAL library and will set input message indication in the directory entry.

**Problem determination:** 1, 2, 4, 16, 35

**Source:** IMS abends

**Module:** DFSUTLB0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

### 3013

**Explanation:** The LPAGE, DPAGE, or literal order relocation work stacks have overflowed during the processing of an output message LPAGE statement or input message DPAGE statement.

**Analysis:** This is a standard abend issued from either DFSUTLC0 or DFSUT150.

Use the program status word (PSW) at entry-to-abend to determine which module issued the abend.

**For DFSUTLC0:**

This is a standard abend that can be issued from DFSUTLC0. The program status word (PSW) at entry-to-abend points to the instruction within label HORRORS from which the abend (SVC 13) was issued. Register 6 in the abend SVRB points to the current MODGRP entry in the MOD, register 11 points to the Phase to Phase Table (PPT) and register 12 is the base register.

Key	Label	Description
PPT1CURI=LPAGE stack index PPT1CURI>PPT1MAXI	CALLSEG	The LPAGE stack index is loaded into register 1. It is compared to PPT1MAXI (the number of entries in the stack). If there is not room left for the current entry, the abend is issued.

Key	Label	Description
Reg15=current literal stack index Reg15>PPT3MAXI	STACKLIT	The LPAGE current literal count is loaded into register 15. Register 15 is compared to PPT3MAXI (the number of LPAGE literal entries in the stack). If the LPAGE literal stack index is higher, the abend is issued.

**For DFSUT150:**

This is a standard abend that can be issued by DFSUT150. The program status word (PSW) at entry-to-abend points to the instruction within label HORRORS from which the abend (SVC 13) is issued. Register 5 in the abend SVRB points to the current DIFSECTN (DPAGE section) entry in the DIF, register 11 points to the work table (PPT) that contains all the entries, and register 12 is the base register.

Key	Label	Description
PPT2CURI=current DPAGE index PPT2CURI>PPT2MAXI	CALLDFLD	The current DPAGE index has been loaded into register 1. It is compared with PPT2MAXI (the number of entries in the stack). If the DPAGE stack is higher, the abend is issued.
Reg15=current literal stack index Reg15>PPT4MAXI	STACKLIT	The current DPAGE index is loaded into register 15. Register 15 is compared with PPT4MAXI (the number of literal entries in the stack). If the DPAGE literal stack index is higher, the abend is issued.

**System action:** The MFS Language utility terminates abnormally.

**Programmer response:** No action is required.

**Problem determination:** 1, 2, 4, 16, 35

**Source:** IMS abends

**Module:** DFSUTLC0, DFSUT150

**Related information:**

Chapter 2, "IMS problem determination," on page 13

**3014**

**Explanation:** The MFS Language utility input SEG processor routine (DFSUTLD0) has recognized that the number of fields after a literal count reduction is less than zero, or the MID being constructed will exceed the design size limit of 32,748 when the literal pool is attached to it.

**Analysis:** This is a standard abend issued from module DFSUTLD0.

Use the registers in the abend SVRB for problem isolation. Register 12 contains the entry point address of CSECT DFSUTLD0. The program status word (PSW) at entry-to-abend points to label ABEND, from which the abend (SVC 13) is issued. Register 14 contains the offset from the beginning of the CSECT to the instruction just prior to where the error was detected.

The number of fields and literal counts are developed by DFSUTLF0 and DFSUTLH0, respectively. The MID is being constructed in subpool 20.

Key	Label	Description
Reg15=number of literal orders in MIDSEG entry Reg4=current MIDSEG entry in MID control block Reg2=negative value	MFLDLITS	This routine has returned from the MFLD MID Literal Order processor and now calculates the number of fields minus the literals. The current segment in the MID block entry is loaded into registers 4 and 15 and tested to see if any literals were generated. If literals were generated, register 2 is loaded with the number of fields in MIDSNF and the literal count is subtracted. If register 2 contains a negative value, the abend is issued.

## 3015

Key	Label	Description
Reg4=next available storage in MID control block	CHKSEG2	This routine constructs a SEG entry in the MID control block for each SEG statement specified in the source. The current address in the MID control block is loaded into register 4. Register 4 is then increased by the size of MIDSEG entry (X'0A'). If the new address then exceeds 32,748, the abend is issued.

**System action:** The MFS Language utility terminates abnormally.

**Programmer response:** No action is required.

**Problem determination:** 1, 2, 4, 16, and 35. The number of fields and literal counts are developed by DFSUTLFO and DFSUTLHO, respectively. The MID is being constructed in subpool 20.

**Source:** IMS abends

**Module:** DFSUTLD0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

## 3015

**Explanation:** The literal work stack has exceeded its maximum size during the processing of an output message MFLD statement.

**Analysis:** This is a standard abend issued from module DFSUTLE0 or DFSUTLG0. Use the program status word (PSW) at entry-to-abend to determine which module issued the abend.

**For DFSUTLE0:**

The registers in the abend SVRB can be used for problem isolation. Register 6 is a pointer to the current MODFDE in the MOD control block. The abend is issued from the routine at label LITERR. The program status word (PSW) at entry-to-abend points to the abend (SVC 13) at this label.

Key	Label	Description
PPT3CURI=current stack index PPT3CURI>PPT3MAXI	SETEOP	The current number of literals have been loaded into register 1. It is compared with PPT3MAXI (maximum number of literals). If the current number of literals is higher, the abend is issued.

**For DFSUTLG0:**

Use the registers in the abend SVRB for problem isolation.

The program status word (PSW) at entry-to-abend points to the instruction within label LITERR from which the abend (SVC 13) is issued.

Key	Label	Description
PPT3CURI=current stack PPT3CURI>PPT3MAXI	SETLITS	The current stack index has been loaded into register 1. Register 1 is compared with the number of entries in the stack (PPT3MAXI). If the current stack index is higher, the abend is issued.

**System action:** The MFS Language utility terminates abnormally.

**Programmer response:** No action is required.

**Problem determination:** 1, 2, 4, 16, 35

**Source:** IMS abends

**Module:** DFSUTLE0, DFSUTLG0



**Related information:**

Chapter 2, "IMS problem determination," on page 13

**3016**

**Explanation:** An internal logic error has occurred during the sort of a symbol table (PPTSYM or PPTSYM1) into alphabetical order. User modifications might be the cause.

**Analysis:** This is a standard abend issued from module DFSUTLT0.

Use the registers in the abend SVRB for problem isolation. The error is detected and the abend is issued from the routine at label ARECBREC. The program status word (PSW) at entry-to-abend points to the abend (SVC 13) at this label.

Key	Label	Description
Reg5=BREC pointer Reg6=AREC pointer Reg14=BAL	ARECBREC	This routine compares field FLDEKEY from the FLDEs in register 6 and register 5. If they are equal, the abend is issued because equal fields should not occur.

**System action:** The MFS Language utility terminates abnormally.

**Programmer response:** Ensure that the failure occurs without user modifications to the MFS Language utility.

**Problem determination:** 1, 2, 4, 16, 35

**Source:** IMS abends

**Module:** DFSUTLT0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

**3017**

**Explanation:** An internal logic error has occurred during construction of a binary structured tree for the symbol tables (PPTSYM or PPTSYM1).

**Analysis:** This is a standard abend issued from module DFSUTLT0.

Use the registers in the abend SVRB for problem isolation. The abend is issued from the routine at label TREEERROR. The program status word (PSW) at entry-to-abend points to the abend (SVC 13) at this label.

Key	Label	Description
Reg2=size of wait stack entry Reg3=address of last entry in wait stack Reg4= current entry in wait stack Reg4>Reg3 Reg9=address of wait stack (in subpool 15)	SHFTPWR	The address of the current stack entry (register 4) is increased by 8 (register 2). If register 4 is higher than register 3 (address of the last entry in the stack), the abend is issued.
Reg2=size of wait stack entry Reg3=address of last entry in wait stack Reg4= current entry in wait stack Reg4>Reg3 Reg9=address of wait stack (in subpool 15)	STKNODE	The address of the current stack entry (register 4) is increased by 8 (register 2). If register 4 is higher than register 3 (address of the last entry in the stack), the abend is issued.
Reg5=index to "less than" entry in wait stack Reg6=index to "center"entry in wait stack Reg7=index to "greater than" entry in wait stack	HAVERT	This routine builds an address list for the binary tree in the wait stack and sets "less than" and "greater than" pointers into each entry in the table. If register 6 is negative, the abend is issued.

**Possible cause:** Modifications to the MFS Language utility

**System action:** The MFS Language utility terminates abnormally.

## 3018

**Programmer response:** No action is required.

**Problem determination:** 1, 2, 4, 16, 35

**Source:** IMS abends

**Module:** DFSUTLT0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

## 3018

**Explanation:** An error has been detected during the TABLE and IF statement processing or during a literal reallocation. An incorrect literal offset has been found.

**Analysis:** This is a standard abend issued from module DFSUT0I0, DFSUT020, DFSUT120, or DFSUT130.

Use the program status word (PSW) at entry-to-abend to determine which module issued the abend. Register 12 in the abend SVRB contains the entry point address (base) of the CSECT in control at the time the error occurred.

### For DFSUT0I0:

Use the registers in the abend SRVB for problem isolation.

The error is detected during table name reference resolution in the DIF DFLD FDE entry and the abend is issued from the routine at label TABBRLOC. Or, the error is detected during IF label reference resolution in the IF symbol table at label IFBRLOC. The program status word (PSW) at entry-to-abend points to the abend (SVC 13) at one of these labels and should be used to determine from which label the abend was issued.

Key	Label	Description
Reg3= DIFFTBLO in DIFFDE Reg0=offset from the beginning of DIF block to next available space in the block Reg11=PPT table PPTBLOCK=address of beginning of DIF block PPTAVAIL=address of next available space in block Reg0≥PPTLDRSZ	TABBRLOC	Register 3 has been loaded with the relocation slot in the DIF block. Then, the offset to the table in the DIF is calculated in register 0 and is stored at the address pointed to by register 3. The offset is then compared with the maximum size of the block. If the offset is higher or equal to the maximum space allocated for the DIF block, the abend is issued.
Reg3= IFEFDE in IFE Reg0=offset from the beginning of IF symbol table to next available space in the block Reg11=PPT table PPTBLOCK=address of beginning of IF symbol table PPTAVAIL=address of next available space in block Reg0≥PPTLDRSZ	IFBRLOC	Register 3 has been loaded with the pointer to the IF label reference in the DIF block. The offset to a branch address is calculated in Register 0 and is stored at the address pointed to by register 3. The offset is compared with the maximum size allocated for the DIF block. If the offset is out of range of the DIF block, the abend is issued.

### For DFSUT020:

Use the registers in the abend SVRB for problem isolation.

The error is detected during literal reallocation for DFLD literals. The abend is issued from the routine at label USER3018. The program status word (PSW) at entry-to-abend points to the abend (SVC 13) at this label.

Key	Label	Description
Reg2=offset from beginning of DOF block to origin of literal pool in the block Reg3= DOFFDE for literal reallocation Reg11=PPT table PPTBLOCK= DOF block	RLOC1	The block storage address for the DOF block is subtracted from register 2 to get the offset origin. Register 3 contains the DOF block entry (DOFFDE). The literal offset (DOFFLIT) is obtained and loaded into register 1. Register 2 is added to the literal offset. If the literal offset is now a negative value, the abend is issued.
Reg2= DOFFDE Reg11=PPT table DOFFLIT6>PPTWCOPN	BCTLOOP2	The literal offset has been developed and stored in DOFFLIT. If the offset is greater than the DOF block size (PPTWCOPN), the abend is issued.
Reg0=offset from beginning of DOF block to origin of literal pool in block Reg2= DOFFDE	BCTLOOP2	The address of DOFFDE is in register 2. Register 1 is loaded with DOFFLIT (the literal offset in the literal pool). The block offset (register 0) is added to it. If register 1 is now a negative value, the abend is issued.

**For DFSUT120:**

Use the registers in the abend SVRB for problem isolation.

The error is detected during literal reallocation for DFLD literals. The abend is issued from the routine at label USER3018. The program status word (PSW) at entry-to-abend points to the abend (SVC 13) at this label.

Key	Label	Description
Reg2=offset from DOF block start to the origin of literal pool in the block Reg3= DOFFDE in DOF block for literal reallocation Reg11=PPT table Literal offset>PPTWCOPN	RLOC1	The calculated literal offset (in register 1) is validity checked for a valid value within the DOF block. If the literal offset is a negative value, or is outside of the DOF block limits (PPTWCOPN), the abend is issued.

**For DFSUT130:**

Use the registers in the abend SVRB for problem isolation.

The error is detected during literal reallocation for DPAGE literal. The program status word (PSW) at entry-to-abend points to the instruction within label USER3018 from which the abend (SVC 13) is issued.

Key	Label	Description
Reg2=offset from beginning of DIF block to origin of literal pool in the block Reg3=DIFSECTN entry in DIF block Reg11=PPT table PPTBLOCK= DIF block origin DIFSLIT>PPTWCOPN	RLOC1	The calculated literal offset in register 1 is validity checked for a valid value within the DIF block. If the literal offset (register 1) is a negative value, or is outside of the DIF block limits (PPTWCOPN), the abend is issued.

**System action:** The MFS Language utility terminates abnormally.

**Problem determination:** 1, 2, 4, 16, 35

**Source:** IMS abends

**Module:** DFSUT0I0, DFSUT020, DFSUT120, DFSUT130

**Related information:**

**3019**

**Explanation:** During processing of a DFLD statement, the literal work stack exceeded the maximum size of 32,748 bytes.

**Analysis:** This is a standard abend issued from module DFSUT050.

Use the registers in the abend SVRB for problem isolation. The abend is issued from the routine at label OVERRUN4. The program status word (PSW) at entry-to-abend points to the abend (SVC 13) at this label.

Key	Label	Description
Reg11=PPT Table address PPT04STK=literal stack address PPT4CURI=current index number PPT4MAXI=maximum number of entries allocated for this work stack PPT4CURI>PPT4MAXI	SETLIT	This routine is entered when a literal reference is processed in the DOFFDE. It is trying to save the DOFFDE address in the work stack. A comparison is made to check to see if the maximum number of entries in the work stack has been exceeded. If it has, the abend is issued.

**System action:** The MFS Language utility terminates abnormally.

**Problem determination:** 1, 2, 4, 16, 35

**Source:** IMS abends

**Module:** DFSUT050

**Related information:**

Chapter 2, "IMS problem determination," on page 13

**3020**

**Explanation:** The literal work stack (PPT04STK) has exceeded its maximum size during the processing of a DFLD statement.

**Analysis:** This is a standard abend issued from various modules.

The program status word (PSW) at entry-to-abend should be used to determine from which module the abend was issued. Register 11 points to the Phase to Phase table (PPT), which contains the labels PPTxxxx. Register 12 is the base register.

**For DFSUT090:**

Use the registers in the abend SVRB for problem isolation.

The abend is issued from the routine at label OVERRUN4. The program status word (PSW) at entry-to-abend points to the abend (SVC 13) at this label.

Key	Label	Description
Reg4=PPTAVAIL Reg5=literal stack index Reg5>PPT5MAXI	ENDMOVE	Register 5 contains the literal stack index (number of literals required so far for this DPAGE). A comparison (register 5 with PPT5MAXI) is made to check to see if the maximum number of entries in the stack (for the DIV) has been exceeded. If it has, the abend is issued.
PPT4CURI>PPT4MAXI	BYORGSET	This section of code places the address of literal FDE into stack PPT04STK for later offset resolution. The current literal stack index (PPT4CURI) is acquired. A comparison is made to see if the maximum number of entries in the stack has been exceeded. If so, the abend is issued.
Reg3=number of current literal stack entries	ORGSET	The number of current literal stack entries has exceeded the maximum allowed, so the abend is issued.

**For DFSUT160:**

Use the registers in the abend SVRB for problem isolation.

The abend is issued from the routine at label OVERRUN4. The program status word (PSW) at entry-to-abend points to the abend (SVC 13) at this label.

Key	Label	Description
PPT4CURI>PPT4MAXI	SETLIT1	This routine places a literal on stack PPT04STK for later offset resolution. The current literal stack index is obtained and compared with PPT4MAXI to see if an overflow condition exists on the stack. If the current index is greater than the maximum number of stack entries, the abend is issued.

**For DFSUT180:**

Use the registers in the abend SVRB for problem isolation.

The abend is issued from the routine at label OVERRUN4. The program status word (PSW) at entry-to-abend points to the abend (SVC 13) at this label.

Key	Label	Description
PPT4CURI>PPT4MAXI	SETLIT1	This routine places the address of literal FDE into stack PPT04STK for later offset resolution. The current literal stack index (PPT4CURI) is obtained and compared with PPT4MAXI to see if an overrun condition exists on the stack. If the current index is greater than the maximum number of stack entries, the abend is issued.

**For DFSUT190, DFSUT200, DFSUT300:**

Use the registers in the abend SVRB for problem isolation.

The abend is issued from the routine at label OVERRUN4. The program status word (PSW) at entry-to-abend points to the abend (SVC 13) at this label.

Key	Label	Description
PPT4CURI>PPT4MAXI	OVERRUN4	This section of code places the address of literal FDE into stack PPT4STK for later offset resolution. The current literal stack index (PPT4CURI) is acquired. A comparison is made to see if the maximum number of entries (PPT4MAXI) in the stack has been exceeded. If so, the abend is issued.

**For DFSUT260:**

Use the registers in the abend SVRB for problem isolation.

The abend is issued from the routine at label OVERRUN4 or OVERRUN5. The program status word (PSW) at entry-to-abend points to the abend (SVC 13) at one of these labels.

Key	Label	Description
PPT4CURI > PPT4MAXI	SETLIT1 (OVERRUN4)	The current literal stack index (PPT4CURI) is obtained and compared with PPT4MAXI to see if an overrun condition exists on the stack. If the current index is greater than the maximum number of stack entries, the abend is issued.

## 3021

Key	Label	Description
PPT5CURI > PPT5MAXI	SETCATR1 (OVERRUN5)	The current DFLD index (PPT5CURI) is obtained and compared with PPT5MAXI to see if an overflow condition exists on the stack. If the current DFLD index is greater than the maximum number of stack entries, the abend is issued.

### For DFSUT280:

Use the registers in the abend SVRB for problem isolation.

The abend is issued from the routine at label OVERRUN4. The program status word (PSW) at entry-to-abend points to the abend (SVC 13) at this label.

Key	Label	Description
PPT4CURI>PPT4MAXI	SETLIT1	This routine places the address of literal FDE into stack PPT4STK for later offset resolution. The current literal stack index (PPT4CURI) is obtained and compared with PPT4MAXI to see if an overflow condition exists on the stack. If the current stack index is greater than the maximum number of stack entries, the abend is issued.

### For DFSUT290:

Use the registers in the abend SVRB for problem isolation.

The abend is issued from the routine at label OVERRUN4. The program status word (PSW) at entry-to-abend points to the abend (SVC 13) at this label.

Key	Label	Description
PPT4CURI>PPT4MAXI	SETLIT1	The current literal stack index (PPT4CURI) is obtained and compared with PPT4MAXI to see if an overrun condition exists on the stack. If the current index is greater than the maximum number of stack entries, the abend is issued.

**System action:** The MFS Language utility terminates abnormally.

**Problem determination:** 1, 2, 4, 16, 35

**Source:** IMS abends

**Module:** DFSUT090, DFSUT160, DFSUT180, DFSUT190, DFSUT200, DFSUT260, DFSUT280, DFSUT290, DFSUT300

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

## 3021

**Explanation:** Each DPAGE must have at least one literal Field Description Element (FDE) for each internal physical page to be created. If there is no LLLLZZ literal in this stack, an internal logic error has occurred.

**Analysis:** This is a standard abend issued from module DFSUT090, the 3270 printer output DFLD Processor.

Use the registers in the abend SVRB for problem isolation. The program status word (PSW) at entry-to-abend points to the instruction within label USER3021 from which the abend (SVC 13) was issued. Register 11 points to the PPT work table where PPT05STK contains the address of LLLLZZ work stack and PPT5CURI contains the index to current DPAGE work stack. Register 12 is the base register.

Key	Label	Description
Reg3=number of literals in current DPAGE. Reg5=address of literal stack for current DPAGE	SETME4	The DFLDs in the current DPAGE were prescanned to determine the internal physical page breakdown. For each internal physical page, an LLLLZZ literal is required. This routine obtains a work stack for each LLLLZZ literal reference. The number of required literals (register 3) is equal to zero; this is incorrect and the abend is issued.

**System action:** The MFS Language utility terminates abnormally.

**Problem determination:** 1, 2, 4, 16, 35

**Source:** IMS abends

**Module:** DFSUT090

**Related information:**

Chapter 2, "IMS problem determination," on page 13

### 3022

**Explanation:** The FMT or MSG descriptor currently being processed is too large. The resultant online block being constructed in SUBPOOL 20 has exceeded the design limit size of 32,748 bytes.

**Analysis:** This is a standard abend issued from various modules.

Because of the number of modules issuing this abend, analysis of each module has not been done. Only a generalized description of the error condition is provided.

Use the registers in the abend SVRB to determine from which module the abend was issued. Register 12 contains the base register used by these modules.

In all cases, the abend is issued from the abend routine at label SIZABEND. There might be multiple references to this label and the module that issued the abend (SVC 13) should be checked for this situation. Register 11 points to the Phase to Phase Table (PPT), which contains all PPT labels.

Key	Label	Description
PPTBLKEN<PPTAVAIL		The routine detecting the error compares the address of the last byte of block storage with the address of the next available byte in block storage to be used. If the address of the last byte of block storage is the lower, the maximum block size has been exceeded and the abend is issued.

**System action:** The MFS Language utility terminates abnormally.

**Programmer response:** The description cannot be processed in its current form. If possible, redefine the descriptor or reconstruct the descriptor as multiple descriptors, and resubmit the job.

**Problem determination:** 1, 2, 4, 16

**Source:** IMS abends

**Module:** DFSUTLA0, DFSUTLB0, DFSUTLC0, DFSUTLD0, DFSUTLE0, DFSUTLF0, DFSUTLH0, DFSUTLJ0, DFSUTLN0, DFSUTLT0, DFSUT0A0, DFSUT0I0, DFSUT020, DFSUT030, DFSUT040, DFSUT050, DFSUT060, DFSUT090, DFSUT120, DFSUT130, DFSUT140, DFSUT150, DFSUT160, DFSUT170, DFSUT180, DFSUT190, DFSUT260, DFSUT280, DFSUT290

**Related information:**

Chapter 2, "IMS problem determination," on page 13

### 3025

**Explanation:** An internal logic error has occurred while processing the literal stack (PPT055STK) for TABLE description. The error might have been caused by user modifications.

**Analysis:** This is a standard abend issued from module DFSUT0I0, the TABLE and IF Processor.

## 3026

Use the registers in the abend SVRB for problem isolation. The program status word (PSW) at entry-to-abend points to the instruction within label `OVERRUN5` from which the abend (SVC 13) is issued.

This module is called by the DIF Block Building routines to process tables referenced in any DFLDs and to incorporate the table information into the block.

Key	Label	Description
Reg4= current DIFTBLE entry in DIF block Reg11=address of PPT Table PPT5CURI + 1=current literal index PPT5MAXI=maximum number of entries allocated for PPT05STK PPT5CURI>PPT5MAXI Reg14=BAL DIFTOP2=offset of literal from beginning of literal table	PROCIFB3	This routine branches and links to DFSUTLU0 to enter the literal in the table. On return, the offset of the literal in the literal pool is developed. The current literal index is acquired and compared with PPT5MAXI, the number of entries in the literal stack. If the current index is greater, the abend is issued.

**Possible cause:** Internal logic error in allocating storage for PPT055TK or user modification to the MFS Language utility.

**System action:** The MFS Language utility terminates abnormally.

**Problem determination:** 1, 2, 4, 16, 35

**Source:** IMS abends

**Module:** DFSUT010

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

## 3026

**Explanation:** An internal logic error has occurred while processing a TABLE symbol table for DFLD table name.

**Analysis:** This is a standard abend issued from modules DFSUT060 and DFSUT170. Use the program status word (PSW) at entry-to-abend to determine from which module the abend was issued.

**Possible cause:** Internal logic error; the maximum reference was probably incorrectly calculated.

**For DFSUT060:**

Use the registers in the abend SVRB for problem isolation.

The abend is issued from the routine at label `OVERUNTB`. The program status word (PSW) at entry-to-abend points to the abend (SVC 13) at this label.

Key	Label	Description
Reg3=TBLE TBL1CURI>TBL1MAXI Reg14=BAL	NOTDET	This routine has branched and linked to DFSUTLX0, which detects the error, to locate the entry for the tablename symbol table. Register 3 contains the address of the tablename symbol table descriptor (TBLE). The current reference index number is acquired and compared with the maximum reference index. If the current index is larger, the abend is issued.

**For DFSUT170:**

Use the registers in the abend SVRB for problem isolation.



The abend is issued from the routine at label OVERUNTB. The program status word (PSW) at entry-to-abend points to the abend (SVC 13) at this label.

Key	Label	Description
Reg3=TBLE TBL1CURI>TBL1MAXI Reg14=BAL	GETFLDE	This routine has branched and linked to DFSUTLX0, which detects the error, to locate the entry for the tablename symbol table descriptor (TBLE). The current reference index number is acquired and compared with the maximum reference index. If the current index is larger, the abend is issued. The maximum reference index is calculated based on the number of times the tablename is referenced by DFLDs under this DIV statement.

**System action:** The MFS Language utility terminates abnormally.

**Problem determination:** 1, 2, 4, 16, 35

**Source:** IMS abends

**Module:** DFSUT060, DFSUT170

**Related information:**

Chapter 2, "IMS problem determination," on page 13

### 3027

**Explanation:** An internal logic error has occurred while processing an IF symbol table for a TABLE description. This error might be caused by user modifications.

**Analysis:** This is a standard abend issued from module DFSUT0I0, the TABLE and IF processor.

Use the registers in the abend SVRB for problem isolation. The program status word (PSW) at entry-to-abend points to the SVRB at label OVERUNIF.

Key	Label	Description
Reg5=address IF label symbol table entry IF1CURI-current reference index number IF1MAXI=address IF label symbol table entry described by dsect IFE Reg14=BAL	CHKSERCH	This routine has returned from DFSUTLX0 (using a BAL) and acquired the current label symbol table entry. It obtains the current index (IF1CURI), adds one (1) to it and compares the index with IF1MAXI to see if the maximum reference index has been exceeded. If it has, the abend is issued.

**Possible cause:** Internal logic error in allocating storage for the work stack or user modification to MFS Language utility.

**System action:** The MFS Language utility terminates abnormally.

**Problem determination:** 1, 2, 4, 16, 35

**Source:** IMS abends

**Module:** DFSUT0I0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

### 3030

**Explanation:** The MFS Language utility phase 1 processor has detected an invalid member in the IMS.REFERRAL library. This is an internal logic error; user modifications might be the cause.

**Analysis:** This is a standard abend issued from module DFSUPB70.

## 3030

Use the registers in the abend SVRB for problem isolation. Register 3 has the address of the member name for which the failure occurred; register 2 has the function code indicating the cause of the failure. The function codes are as follows:

Code	Meaning
X'04'	First record not an ESD record.
X'08'	Member-name on ESD record does not match directory member-name.
X'0C'	END record not found.
X'10'	Block size in ESD record does not match directory block size.
X'14'	Requested member-name not found in the directory.

Use the contents of register 2 to isolate to the particular label where the error was detected.

The abend is issued from the common abend routine at label MYABEND. The program status word (PSW) at entry-to-abend points to the abend (SVC 13) at this label.

A block of the IMS.REFERAL Library member is read into storage (subpool 15). BUFFADDR (addressed by register 12) contains the address of the beginning of the block. Register 4=address of the BLDL list. DSECT BLDLLIST is used to describe this list. Register 5=address of current record in the block. DSECT CRDIMAGE is used to describe each record in the block.

Key	Label	Description
Reg2=X'04' CRDTYPE ^= X'02C5E2C4'	STARTBLK	The member has been located on the REFERAL library. The first block of REFERAL has been read. A comparison is made to verify that the first record is an ESD record. If it is not, the abend is issued.
Reg2=X'08' CRDNAME ^= BLDLNAME	STARTBL1	This subroutine verifies that the member name on the ESD record is the same as the one specified in the BLDL list. If not, the abend results.
Reg2=X'0C' CRDTYPE ^=X'02C5D5C4'	ENDCARD	This subroutine verifies that this is the END record. If it is, the return parameters are set and storage is freed. If it is not, the abend is issued.
Reg2=X'10' Reg6 ^=BLKSIZE	ENDCARD1	An output area has been obtained and the data for TXT record has been moved there. The end flags are moved into an output area pointed to by register 6. The length of the block is then developed in register 6. Register 6 is then compared with BLKSIZE on the ESD record. If the block sizes don't match, the abend is issued.
Reg2=X'14' TBLESW ^=X'01' Reg15=X'04'	LOCATE NOTFOUND	A BLDL has been issued to locate the member on the REFERAL library. On return from BLDL, register 15 is tested for an X'04' indicating that the member was not found. If register 15=X'04', routine LOCATE branches to NOTFOUND to see if the request was for TABLE-NAME. It was not, so the abend is issued.

**Possible cause:** User modification to the IMS.REFERAL library or the MFS Language utility.

**System action:** The MFS Language utility terminates abnormally. At time of abend, register 3 has the address of member name for which the failure occurred; register 2 has the function code indicating the cause of the failure.

**Programmer response:** Ensure that the failure occurs without modifications to the IMS REFERRAL library or the MFS Language utility.

**Problem determination:** 1, 2, 3, 4, 8, 12, 16, 35

**Source:** IMS abends

**Module:** DFSUPB70

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

**3040**

**Explanation:** The initialization of an IMS control or batch region failed. The module and associated reason follow.  
**DFSRRRA30**

Unable to acquire subpool 231 storage for building a system queue header control block or an external subsystem work area.

**DFSRRRA70**

Unable to acquire subpool 0 storage for building a system queue header control block for a batch region.

**DFSRRRA70**

Unable to acquire subpool 0 for loading the HLPI bootstrap routine for a batch region.

**DFSRRRA80**

Unable to acquire subpool 0 for loading the HLPI bootstrap routine for a batch region.

**Analysis:** This is a standard abend issued by DFSRRRA30 (control region), DFSRRRA70 (batch region), or DFSRRRA80.

Register 14, at the time of abend, contains the address of the location where the error was detected. This identifies which of the possible conditions caused this abend. Register 15 contains an IMODULE GETMAIN return code indicating the cause of the failure.

Register	Label	Module
Reg14=BAL Reg15=IMODULE return code	RACTL4	DFSRRRA30
Reg14=BAL Reg15=IMODULE return code	RADL13	DFSRRRA70
Reg14=BAL Reg15=IMODULE return code	RADSVCI	DFSRRRA70
Reg14=BAL Reg15=IMODULE return code	UTLHLP	DFSRRRA80

**Possible cause:** The GETMAIN request (using IMODULE) exceeds the amount of available storage in the requested subpool (that is, 231 (CSA)). Increase the size of CSA available to the job. If the request was for subpool 0, increase the region size on the job/execute statement.

**System action:** IMS terminates abnormally.


**Programmer response:** Register 14 at the time of abend contains the address of the location where the error was detected. This register identifies which of the possible conditions caused the abend. Register 15 contains an IMODULE GETMAIN return code indicating the cause of the IMODULE failure. For a description of these codes, see the IMS system services return codes.

**Problem determination:** 23, 35. Dumps in CSA

**Source:** IMS abends

**Module:** DFSRRRA30, DFSRRRA70, DFSRRRA80

**Related reference:**

 IMS system services return codes (Messages and Codes)

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

**3041**

**Explanation:** An unsupported return code was received from the external subsystem attachment package (ESAP). The subsystem connection between IMS and the specified subsystem is terminated.

**Analysis:** This is a standard abend issued by DFSESI00, DFSESI20, DFSESI30, DFSESI50, DFSESI60, DFSIESI0, and DFSXESI0.

Refer to message DFS3603I, DFS3606I, or DFS3607I for information to determine the nature of the problem.

Register 15 is in the following format to indicate the module that detected the nonzero return code:

bytes 1-2 | bytes 3-4

## 3042

### Bytes 1 and 2

Contain a hexadecimal value that represents the module name that detected the nonzero return code. To determine the module name, refer to the table in the information about IMS system services return codes or to the DFSESF macro.

### Bytes 3 and 4

Contain the hexadecimal return code that is passed back by the external subsystem exit routine (that is, X'08').

Register 14 contains the address of the location where the error was detected. See contents of register 15 as described earlier.

Label ABEND—an unsupported return code received from an external subsystem exit routine.

**Possible cause:** For more information, see message DFS3603I, DFS3606I, or DFS3607I. Consult the external subsystem documentation to determine the reason for the return code as indicated in register 15.

**System action:** IMS will terminate the TCB representing the external subsystem. The subsystem connections between IMS and the specified subsystem is terminated and processing continues. The master terminal operator is informed of the failure by receiving message DFS3603I, DFS3606I, or DFS3607I.

**Operator response:** Notify the system programmer to correct the external subsystem resource.


**System programmer response:** Refer to the external subsystem documentation to find the information concerning the subsystem failure.

**Problem determination:** 4, 5, 6, 7

**Source:** IMS abends

**Module:** DFSESI00, DFSESI20, DFSESI30, DFSESI50, DFSESI60, DFSIESI0, DFSXESI0

**Related reference:**

 [IMS system services return codes \(Messages and Codes\)](#)

**Related information:**

Chapter 2, “IMS problem determination,” on page 13

 [DFS3603I \(Messages and Codes\)](#)

 [DFS3606I \(Messages and Codes\)](#)

 [DFS3607I \(Messages and Codes\)](#)

---

## 3042

**Explanation:** Module DFSESPR0 issues this abend after detecting error. DFSESPR0 also issues this abend due to errors detected in lower level modules. The module detecting the error sets the abend code for DFSESPR0 to issue. Use registers 14 and 15 to further analyze the error condition detected.

**Analysis:** Register 15 is in the following format to indicate the module that detected the nonzero return code:

bytes 1-2 | bytes 3-4

### Bytes 1 and 2

Contain a hexadecimal value representing the module name that detected the nonzero return code. To determine the module name, refer to the table in the information on IMS system services return codes or to the DFSESF macro.

### Bytes 3 and 4

Contain the hexadecimal return code passed back by the external subsystem exit routine or the lower level module.

Register 14 contains the address of the location where the error was detected. See contents of register 15 as described above.

**Possible cause:** Consult the external subsystem documentation for the exact reason for the return code.

**System action:** The TCB in the dependent region, under which these modules run, is terminated.

**Operator response:** Notify the system programmer.

**Programmer response:**


Bytes 1 and 2 of Register 15 contain a hexadecimal value representing the module name that detected the nonzero return code. To determine the module name, refer to the categories describing the various conditions detected by IMS modules in IMS Return codes and function codes information. Bytes 3 and 4 of Register 15 contain the hexadecimal return code passed back by the external subsystem exit or the lower level module.


**Problem determination:** 4, 5, 6, 7

**Source:** IMS abends

**Module:** DFSESD70, DFSESPR0, DFSVES00

**Related reference:**

 Return and function codes associated with the external subsystem support (Messages and Codes)

 IMS system services return codes (Messages and Codes)

**Related information:**

Chapter 2, "IMS problem determination," on page 13

### 3043

**Explanation:** This abend is issued when the external subsystem mother tasks (TCB) end-of-task-exit-routine (ETXR) is scheduled and has passed an invalid daughter task (TCB) address.

**Analysis:** This is a standard abend issued by the EXTR routine, DFSIESI0. The external subsystem mother task (TCB-(ESSM)) is abnormally terminated.

Register 3 contains the address of the questionable TCB. Register 9 contains the address of the external subsystem entry table (ESET) prefix (mapped by DSECT DFSESETP) which in turn points to the defined external subsystem entries (mapped by DSECT DFSGESE). Each entry contains the address of its associated TCB. Register 11 contains the address of the SCD, at label ETXRFTCB, where the condition is detected.

**Possible cause:** Probably an IMS logic error

**System action:** The external subsystem mother task (TCB-(ESSM)) abnormally terminates. This is probably an IMS logic error.

**Operator response:** Notify the system programmer to correct any unique recovery procedures.

**Programmer response:** Use the IMS /START SUBSYS command to restart the mother and daughter external subsystem tasks (that is, subsystem connections).

**Problem determination:** 1, 2, 4, 5, 6, 7, 8

**Source:** IMS abends

**Module:** DFSIESI0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

### 3044

**Explanation:** A should-not-occur situation was encountered in communicating with an external subsystem. The dependent region terminates abnormally and must be restarted if application programs will be scheduled in the terminated region.

**Analysis:** This is a pseudoabend issued by DFSESPR0, DFSFESP0, or DFSTMS00.

See message DFS3624I describing the problem. Register 15 contains the error code and register 14 the error address. Correct the problem and use the IMS /START command to restart the failed external subsystem.

Register 15 is in the following format to indicate the module that detected the nonzero return code:

bytes 1-2	bytes 3-4
-----------	-----------

**Bytes 1 and 2**

Contain a hexadecimal value representing the module name that detected the nonzero return code. To determine the module name, refer to the information on IMS system services return codes or to the DFSESF macro.

**Bytes 3 and 4**

Contain the hexadecimal return code passed back by the external subsystem exit routine (that is, X'20').

Register 14 contains the address of the location where the error was detected. See contents of register 15 as described above.

**Possible cause:** Consult the external subsystem documentation for the exact reason for the return code.

**System action:** The abend will be recognized and handled by IMS during the application termination cleanup processing. As a result, IMS will do the following:

1. Terminate the application program. If this error is detected during sync point processing, terminate the region.
2. Log the relevant data areas.
3. If the transaction is PSTOPPED, send message DFS554A to the master terminal operator and DFS555I to the input terminal.
4. Allow the external subsystem to log relevant information from both the control and the dependent regions.
5. Invoke the external subsystems ECHO exit for determining the disposition of the input message. If the external subsystem does NOT respond to the ECHO request, IMS will save the input message and PSTOP the transaction. If it DOES respond to the ECHO request, the input message will be re-queued for future processing.
6. Back out all uncommitted changes during application termination.
7. Send message DFS3624I to the master terminal operator.

**Operator response:** Check the status of the subsystem connection using the /DISPLAY command. The connection might be terminated. You can start or restart the connection.

**Programmer response:** See message DFS3624I sent to the master terminal. Message DFS3624I contains the function and return code identifying the problem and the module that detected it.


Options include starting the external subsystem or changing the region error option in the external subsystem PROCLIB member (SSM).

**Problem determination:** 4, 5, 6, 7

**Source:** IMS abends

| **Module:** DFSESPR0, DFSFESP0, DFSTMS00

**Related reference:**

 [IMS system services return codes \(Messages and Codes\)](#)

**Related information:**

 [DFS3624I](#)

Chapter 2, "IMS problem determination," on page 13

**Explanation:** This is an IMS pseudoabend caused by a failure in the external subsystem during one of the must-complete phases of connection management. A nonzero return code was passed back by one of the following external subsystem exits indicating a failure condition.

- Sign-on failed in the external subsystem.
- Create thread failed in the external subsystem.
- Terminate thread with the abort option failed in the external subsystem.
- Abort continue failed in the external subsystem.

| **Analysis:** This is a pseudoabend issued by DFSESPR0, DFSFESP0, or DFSTMS00.

Refer to message DFS3624I for the function (FC) identifying the module that detected the error and return code (RC) identifying the error condition. Refer to the IMS system services return codes information to determine the IMS module (function code value) that detected the problem. This information includes a brief description of the module function (that is, the SIGNON and CREATE-THREAD). Using the return code, you can now consult the external subsystem documentation for debugging information to determine why the exit routine encountered the condition that failed.

Register 15 = function code (FC) and return code (RC).

**Possible cause:** Consult to the external subsystem documentation for possible cause.

**System action:** The abend will be recognized and handled by IMS during application cleanup processing. IMS will do the following:

1. Terminate the application program. If this error is detected during sync point processing, terminate the region.
2. Log the relevant information from both the control and dependent regions.
3. Discard the input message.
4. STOP the transaction.
5. Back out the database changes done by the application program.
6. Send message DFS555I to the input terminal.
7. Send message DFS554I to the master terminal.
8. Send message DFS3624I to the master terminal.

**Operator response:** The operator should notify the system programmer.


**Programmer response:** Refer to message DFS3624I sent to the master terminal. It contains the function code (FC) identifying the module that detected the error and the return code identifying the error condition. Using the return code, you can now consult the external subsystem manuals for debugging information to determine why the exit encountered the condition which failed.

**Problem determination:** 1, 4, 5, 6, 7 and 14

**Source:** IMS abends

**Module:** DFSESPR0, DFSFESP0, DFSTMS00

**Related reference:**

 [IMS system services return codes \(Messages and Codes\)](#)

**Related information:**

 [DFS3624I](#)

Chapter 2, "IMS problem determination," on page 13

3046

**Explanation:** This is an IMS pseudoabend initiated by IMS during commit processing. It is necessary to abort the application program when a return code of 'X'04' is received from the external subsystem commit-continue exit or terminate-thread with the commit option. The commit crossover record has been written on the log, and the input messages are processed by the application program. However, application programs connected to the external subsystems are marked with recovery outstanding for the thread; that is, residual recovery elements (RREs) until they are resolved either by resolve-in-doubt processing (part of the normal connection protocol) or by IMS command deleting the RREs (/CHANGE SUBSYS).

**System action:** IMS will do the following:

1. Terminate the application program. If this error is detected during sync point processing, terminate the region.
2. Log the appropriate data areas.
3. Send message DFS554A to the master terminal operator.
4. Send message DFS555A to the input terminal.
5. Allow the external subsystems involved with the application program to log their relevant information from the appropriate region (that is, control region or dependent region).
6. Send message DFS3624I to the master terminal. This message will indicate which of the above processes encountered the problem.
7. STOP the transaction.

**Operator response:** Refer to message DFS3624I, which contains the function and return code identifying the problem and the module that detected it.

**Programmer response:** Determine the reason why the external subsystem commit process failed. Refer to the external subsystem documentation for help in this process. Restarting the subsystem (/START SUBSYS xxxx) should not destroy any debugging information, because most of the required information is logged or externalized using a diagnostic message.

**Problem determination:** 4, 5, 6 and 7

## 3047

**Source:** IMS abends

**Module:** DFSESPR0, DFSFESP0

**Related information:**

 DFS3624I (Messages and Codes)

Chapter 2, "IMS problem determination," on page 13

---

## 3047

**Explanation:** This abend is issued because an abend was specified as the region error option in the subsystem definition in IMS PROCLIB. If one of the following situations occurs, the application program abends and the input message is discarded.

- The external subsystem connection cannot be established due to resource constraints, because it was the first request directed to the subsystem.
- The connection to the external subsystem failed either in the external subsystem attachment package (ESAP) or in the external subsystem.
- The external subsystem name specified in the SSN field of the PROCLIB member pointed to by the SSM=parameter contains an invalid subsystem name. Although the PROCLIB member contains the REO option of 'r' (send back a return code), the dependent region terminates with abend 3047 and discards the input message.

Review the REO= option specified for either the SSM member or Resource Translation Table (RTT). When an RTT is used, the OPTION specified in the RTT overrides the REO option in the SSM member. See the DB2 for z/OS Installation and Migration Guide for further information. Abend 3047 can occur if REO=A is set.

**Analysis:** IMS abends when the external subsystem has detected a conflict in resource definitions between the two participating subsystems (that is, IMS or other subsystems). Possibly the user ID is unknown to the external subsystem and the user ID is passed as part of the input parameters to the signon exit routine.

Most likely the problem is that the application program name or the PSB name passed at CREATE-THREAD does not match a corresponding resource (for example, plan ID) in the external subsystem.

If this is not the case, then the external subsystem probably encountered an internal resource conflict. It makes the appropriate debugging information available.

Abend 3047 is a pseudoabend issued by DFSESPR0. Register 15 is formatted in the following manner to indicate the module that detected the nonzero return code;

bytes 1-2 | bytes 3-4

### Bytes 1 and 2

Contain a hexadecimal value representing the module name that detected the nonzero return code. To determine the module name, refer to the table in the information on IMS system services return codes or to the DFSESFC macro.

### Bytes 3 and 4

Contain the hexadecimal return code passed back by the external subsystem exit routine (that is, X'1C').

Field RCSSOB2 in PXPARMS contains the address of the SSOB that contains the address of the parameters (SSESPMA) passed to the exit routine.

Label PRABEND issues the abend.

**Possible cause:** The problem is probably the PSB name versus similar resource type do not agree. Ensure that the PSB names that access external subsystem resources are properly defined to the external subsystem. Refer to message DFS3624I, which contains the function code identifying the module that detected the problem and the return code identifying the problem. Consult the external subsystem documentation to determine the reason for the return code as indicated in message DFS3624I.

**System action:** IMS will do the following:

1. Terminate the application program.
2. Back out all uncommitted changes.
3. Stop the transaction.
4. Discard the input message.
5. Send message DFS555I to the IMS master terminal operator and the input terminal.
6. Log the relevant data areas.
7. Allow the external subsystem to log error information from both the control and dependent regions.



8. Send message DFS3624I to the master terminal.
9. Make sure that the subsystem name specified in the PROCLIB member pointed to by the SSM=*parameter* is a valid Db2 for z/OS subsystem name. Correct the name, if necessary.

**Operator response:** Start or restart the failing subsystem.

Refer to message DFS3624I which contains the function and return code identifying the problem and the module that detected it.


Options include starting the external subsystem or changing the region error option in the external subsystem PROCLIB member (SSM).

**Problem determination:** 4, 5, 6, 7

**Source:** IMS abends

**Module:** DFSESPRO

**Related reference:**

 IMS system services return codes (Messages and Codes)

**Related information:**

 DFS3624I (Messages and Codes)

Chapter 2, "IMS problem determination," on page 13

3048

**Explanation:** This pseudoabend is issued because the external subsystem indicated that a temporary failure or a temporary resource constraint has occurred.

**Analysis:** This is a pseudoabend issued by DFSESPRO or DFSTMS00 .

This pseudoabend allows the input message to be placed on the transaction suspend queue. By recognizing this abend from others, the synchronization point process does not discard the input but ensure it is available for future processing. Use the /DEQ SUSPEND command to remove all suspended transactions. Refer to message DFS3624I, which contains the function code (FC) identifying the module that detected the error and return code (RC) identifying the cause of the problem.

Register 4 contains the address of the SSOB, which contains the function and return code from the module that detected the error. SSOB field SSESFRC contains the function return code; SSESMRC contains the pseudoabend code.

Register 15 also contains the function code and return code respectively. Each code length is 2 bytes.

Label PRABEND is the location in DFSESPRO where this abend is issued.

**Possible cause:** The external subsystem exit routine probably received a return code indicating that the appropriate resource is not available and is making this information available to IMS. IMS must assume the reason for the return code is justified and this temporary condition will pass.

**System action:** IMS will do the following:

1. Terminate the application program.
2. Back out all uncommitted changes.
3. Place the current input message on the IMS suspend queue. The input message is once again released and reprocessed when the failure is recovered or the master terminal operator enters the /DEQ SUSPEND command to release the suspend queue.
4. Send message DFS3324I to the master terminal operator to indicate that the input message is being placed on the suspend queue.
5. Send diagnostic message DFS3624I to the master terminal to help determine the cause of the abnormal termination.
6. Send message DFS554A indicating that the program controller task (TCB) has been abnormally terminated.

**Operator response:** Notify the system programmer to correct the external subsystem resource allocation problem.

**Programmer response:** Refer to message DFS3624I, which contains the function code identifying the module that detected the error and the return code identifying the cause of the problem.

Options include starting the external subsystem or changing the region error option in the external subsystem PROCLIB member identified by the SSM parameter.

## 3049

**Problem determination:** 4, 5, 6, 7

**Source:** IMS abends

**Module:** DFSESPR0, DFSTMS00

**Related information:**

 DFS3624I

Chapter 2, "IMS problem determination," on page 13

---

## 3049

**Explanation:** One of the following conditions was detected by IMS:

- An invalid return code was detected upon return from an external subsystem exit. Either the return code was not within the acceptable range valid for the exit (that is, greater than X'20') or, not having exceeded the range check, the return code was not supported for this exit. Message DFS3608I will also appear if the return code (R15 = FF) in register 15 indicates an exit is missing. The RC = xx value in message DFS3624I is the return code in question; this message will accompany this abend. Register 15 will also contain the function code (FC - module identifier) and return code (RC) value.
- The required data (that is, the parameter list, the exit addresses, or RRE) did not pass validity checking. These conditions can be recognized by the contents of the RC = field.

**Analysis:** For diagnostic purposes, message DFS3624I accompanies this abend. The RC = xx value in the message is the return code in question. Register 15 also contains either the function code (FC - module identifier), the return code (RC) value, or both.

Another possibility is that required data (that is, parameter list and exit addresses) did not pass validity checking. These conditions can be recognized by the contents of the RC= field. The following list indicates the condition for each field.

### R15=FC

IMS is attempting a dependent region SIGNON to the subsystem without the required RRE. This usually occurs after an abend of the subsystem's IMS ESI TCB.

### R15 = FF

A required external subsystem exit routine was not supplied by the external subsystem package or the address was erroneously zeroed out.

### R15 = FE

The function code passed by the DFSESGO macro, which passed it to module DFSESGLO. IMS system service was invalid. The caller of DFSESGLO has a logic error that probably invalidated the function.

### R15 = FD

The count in the parameter list passed to DFSESGLO indicating the number of parameters within the parameter list is invalid. The caller of this system service has a logic error and is invalidating the parameter count (that is, making it negative).

**System action:** IMS takes the following action:

1. Terminate the application program. If this error is detected during sync point processing or subsystem SIGNON processing, terminate the region.
2. Back out all uncommitted changes.
3. Place the input message being processed at the time back on the IMS input message queue for future reprocessing, provided that the application has not reached a synchronization point. In such a case, the application will logically resume processing after the synchronization point.
4. Send a DFS555I message to the input terminal.
5. Send a DFS554A message to the master terminal.
6. STOP the transaction. If the error is detected during subsystem SIGNON processing, the transaction is not stopped.
7. Send message DFS3624I to the master terminal for diagnostic purposes.
8. Send message DFS3608I to the master terminal if R15 = FF (exit missing).

**Operator response:** The operator should notify the system programmer and then take standard recovery/restart actions.

**Programmer response:** To quickly determine the cause of the problem, you should first refer to message DFS3624I and examine the RC = xx value. The FC = yy value indicates the ID of the module that detected the error.

If the RC = xx value does not begin with Fx (that is, 0C), the problem encountered is probably an invalid return code,

as described in the first portion of the abend explanation. This is an indication of a possible logic error in the external subsystem exit. Consult the IBM Support Center for aid in problem resolution.

If the RC = xx value **does** begin with Fx (that is, FD), use the following list of acceptable return code installation actions as a guide:

**R15 = FF**

The external subsystem module table (ESMT) specified for the subsystem member specified by the SSM parameter might not include the required exit. Message DFS3608I will indicate the missing exit ID. If the exit was not included, consult the external subsystem documentation to determine how to include the exit. However if the exit was defined, a logic error in IMS overlaid its address or never inserted its address. Refer to message DFS3608I for additional diagnostic information. Consult the IBM Support Center for assistance in problem resolution. Restart IMS.

**R15 = FE**

This problem probably stems from an IMS logic error. Consult the IBM Support Center for assistance in problem resolution. Restart IMS.

**R15 = FD**

An invalid parameter count detected by this system service is probably the result of an IMS logic error. Consult the IBM Support Center for aid in problem resolution. Restart IMS.

**R15 = FC**

IMS is attempting a dependent region SIGNON to the subsystem without the required RRE. This usually occurs after an abend of the subsystems IMS ESI TCB.

**Problem determination:** 1, 4, 5, 6

**Source:** IMS abends

**Module:** DFSRRA00, DFSFESP0, DFSESPRO, DFSTMS00

**Related information:**

 DFS3624I (Messages and Codes)

Chapter 2, "IMS problem determination," on page 13

3050

**Explanation:** This abend is issued when the module that initializes IMS dispatching blocks detects an error that cannot be corrected.

**Analysis:** This abend is issued for all abnormal conditions. Register 15 contains the following reason codes for both modules DFSKDP00 and DFSKDS00.

**For DFSKDP00:**

**Code    Meaning**

- X'01'    The requested dispatcher type was invalid. Register 2 contains the invalid dispatcher type.
- X'02'    DFSBCB for IDSPWRK section 1 failed. Register 2 contains the current dispatcher type and register 3 contains the DFSBCB return code.
- X'03'    DFSBCB for IDSPWRK section 2 failed. Register 2 contains the current dispatcher type and register 3 contains the DFSBCB return code.
- X'04'    The dependent region requested DYN save sets. Register 2 contains the requesting dispatcher type.
- X'05'    The dispatcher type requires a special INIT and the routine has not been added to the module. Register 2 contains the requesting dispatcher type.
- X'06'    Unable to obtain TCB entry for the maximum pages generated. Register 2 contains the requesting dispatcher type.
- X'07'    Unable to obtain TCB entry. Register 2 contains the requesting dispatcher type. DFSBCB GET failed and register 3 contains the DFSBCB return code.
- X'08'    RCF TCB initialization failed. Unable to locate matching TCB address for terminating ECB in ITRCFTCB table. Register 5 contains the TCB address that is the subject of the search.
- X'09'    Release was requested. TCB used DYN SAPs. Register 2 contains an invalid dispatcher type.
- X'0A'    A GETMAIN failed for the dynamic SAP control block for the IMS CTL TCB dispatcher. Register 3 contains the IMODULE GETMAIN return code.
- X'0A'    This reason code is issued by module DFSKDP00. An IMODULE GETMAIN for dynamic SAP control blocks failed. Register 2 contains the current dispatcher type. Register 3 contains the IMODULE GETMAIN return code. For a description of these codes, see the IMS system services return codes information.

## 3050

- X'0B' DFSBCB for DPXB failed. Register 2 contains current dispatcher type. Register 3 contains the DFSBCB return code.
- X'10' IMODULE GETMAIN failed for the common terminate ECB control block. Register 3 contains the IMODULE return code. Register 14 contains the address within DFSKDP00 where the error was detected.
- X'11' DFSBCB GET failed for save areas for common terminate ITASK. Register 3 contains the DFSBCB return code. Register 14 contains the address within DFSKDP00 where the error was detected.
- X'12' ITASK CREATE (DFSCIR) failed for common terminate ITASK. Register 3 contains the DFSCIR return code. Register 14 contains the address within DFSKDP00 where the error was detected.

### For DFSKDS00:

<b>Code</b>	<b>Meaning</b>
-------------	----------------

- |              |   |
|--------------|---|
| <b>X'13'</b> | Unable to get the initial set of dynamic SAPs. BP1003. Register 3 contains the DFSKDP10 return code; register 8 contains the DISP WK area address.  |
| <b>X'13'</b> | The DFSCIR macro failed for a dynamic save area prefix (SAP) wakeup ITASK. BP1003. Register 3 contains the DFSCIR return code; register 8 contains the DISP WK area address.                            |
| <b>X'16'</b> | The calling dispatcher was not set up for DYN SAPS. Register 2 contains the requesting dispatcher type.   |
| <b>X'17'</b> | The requested number of dynamic SAPs was zero. Register 2 contains the requesting dispatcher type.  |
| <b>X'18'</b> | DFSBCB GET for SAPS failed. Register 2 contains the requesting dispatcher type and register 3 contains the DFSBCB return code.  |
| <b>X'19'</b> | DFSBCB GET for the save areas failed. Register 2 contains the requesting dispatcher type and register 3 contains the DFSBCB return code.  |
| <b>X'1A'</b> | DFSBCB GET for SAP work areas failed. Register 2 contains the requesting dispatcher type and register 3 contains the DFSBCB return code.  |
| <b>X'1B'</b> | A DFSBCB GET for a LATE (latch) failed. Register 2 contains the requesting dispatcher type and register 3 contains the DFSBCB return code. For an explanation, see the DFSBCB return codes information. |

### DFSKDS10:

Return code from the failed service is in the high halfword of R15; the subcode is in the low halfword.

<b>Code</b>	<b>Meaning</b>
-------------	----------------

- |              |   |
|--------------|---|
| <b>X'01'</b> | Illegal function was specified.                       |
| <b>X'10'</b> | Get for SAPs failed.                                  |
| <b>X'11'</b> | Get for the save areas failed.                        |
| <b>X'12'</b> | Get for QMWAs failed.                                 |
| <b>X'13'</b> | Get for the Common Latch List Elements (CLLE) failed. |
| <b>X'14'</b> | Get for SVPLs fail – BN70022.                         |
| <b>X'15'</b> | Get for XMCI failed.                                  |

**System action:** The control region terminates abnormally.

**Operator response:** Perform IMS emergency restart.

**System programmer response:** This abend is caused by a defective IMS code.

The subcode in register 15 at the time of the abend indicates the type of defect.

<b>Subcode (hex)</b>	<b>Meaning</b>
----------------------	----------------

A possible CSA shortage requires an IPL of the operating system to relieve storage fragmentation.

A possible private address space storage shortage requires an increase in the region size for the IMS job.

**X'01'** IMODULE GETMAIN failed for KDS20 ECB.

**X'01'** DFSCIR failed for KDS20.

The abend is issued by modules DFSKDP00 and DFSKDS00. The subcodes are internal to IMS and are documented in the module prologues. Follow the procedure for correcting IMS software problems.

**Problem determination:** 1, 4, 5, 12, 35

**Source:** IMS abends

**Module:** DFSKDP00, DFSKDS00

**Related reference:**

➤ IMS system services return codes (Messages and Codes)

➤ DFSBCB return codes (Messages and Codes)

**Related information:**

Chapter 2, “IMS problem determination,” on page 13

### 3051

**Explanation:** This is an IMS pseudoabend. IMS has allocated this abend for external subsystems use. The abend is issued when the external subsystem not operational (SNO) exit passes back a X'10' return code. It is returned by the subsystem exit when it requires an IMS abend for debugging purposes, or when the external subsystem is not operational and “Region Error option Q” was selected. It is usually accompanied by a X'55' log record, written to the IMS log and containing additional debugging information.

Review the REO= option specified for either the SSM member or Resource Translation Table (RTT). When an RTT is used, the OPTION specified in the RTT overrides the REO option in the SSM member. See the DB2 for z/OS Installation and Migration Guide for further information. Abend 3051 might occur if REO=Q is set.

**Analysis:** This is a pseudoabend issued by DFSESPR0. The installation problem determination should consist of the following:

- The external subsystem documentation should define the reasons for requesting this abend. Determine the nature of the problem from this documentation and correct it.
- Start the external subsystem connection using the /STA SUBSYS command.

Register 2 points to the environmental controller's work area, which uses DSECT DFSECP to determine the offset of field ECSAVE2 in the work area. The beginning of the parameter list passed to the system not operational (SNO) exit routine is at this location. Using the DSECT parameter list defined in module DFSESPR0, examine the list to ensure accuracy. The 3051 abend code should be in field PARMsrc. If it is there, the SNO exit routine placed it there for problem determination reasons.

Label ESPX010 sets up the abend.

**System action:** IMS does the following:

1. Terminate the application program.
2. Log the relevant data areas.
3. PSTOP the transaction.
4. Send a DFS554A message to the MTO and a DFS555I message to the input terminal.
5. Allow the external subsystem to log relevant information from both the control and the dependent regions.
6. Queue the input message again for future processing. IMS will *not* invoke the external subsystems ECHO exit to determine the input message disposition. However, the input message will be re-queued for future processing.
7. Back out all uncommitted changes during application termination processing.

**Operator response:** Start or restart the failing subsystem connection.

**Programmer response:** Installation options include:

- The external subsystem documentation should define the reasons for requesting this abend. Determine the nature of the problem from this documentation and correct it.
- Start the external subsystem connection using the /START SUBSYS command.

**Problem determination:** 4, 5, 6 and 7

**Source:** IMS abends

**Module:** DFSESPR0

**Related information:**

Chapter 2, “IMS problem determination,” on page 13

### 3052

**Explanation:** This IMS abend is issued when an internal required function did not complete successfully. One of the following problems occurred:

- An invalid function code was passed to the external subsystem synchronization point manager (DFSFPSP0) during an application synchronization point.

- The 1-byte code is passed by the IMS synchronization point manager, DFSFXC30, in PST field PSTSYNFC.

**Analysis:** This is a pseudoabend issued by DFSFESP0, code resident in the dependent region SSOB save area. This code is only run when one or both of the conditions occur. Normally the program request handler (DFSESPR0) gets control and process the application request, but since the external subsystem support was never initialized, the program request handler was not loaded.

**Possible cause:** The fact that this abend occurred is justification to either prohibit the application that issued the external subsystem request from running (that is, /PSTOP *tran*) or insert the SSM parameter after building the IMS.PROCLIB member defining the subsystems.

**For DFSXLRM0:**

This abend is issued when an IMS service failed during the execution of DFSXLRM0. The service which failed can be determined by the subcode that is contained in Register 15 as follows:

Code	Meaning
X'04'	DFSBCB get qsav failed.
X'08'	DFSCDSP failed.
X'0C'	IMODULE load failed.
X'10'	DFSCWU failed.
X'14'	IPOST failed.
X'18'	IMODULE GETMAIN failed.
X'1C'	DFSCIR failed.
X'20'	DFSBCB get 1sav failed.
X'24'	IMODULE LOAD for DFSTFRG0 failed.

This is a user abend issued by DFSXLRM0 when an IMS service returns a nonzero code. The return code from the service is contained in R8 and the address of the next sequential instruction is contained in R14. For the return codes for IMODULE and DFSBCB, see the IMS system services return codes information. The return codes for DFSCIR can be obtained from the prologue for the DFSCIR macro.

**Possible cause:** Possible cause depends on the service that failed and the reason for the failure. Most problems are related to a storage shortage.

**System action:**

1. Terminate the application program. If this error is detected during sync point processing, terminate the region.
2. Log the relevant information from both the control and dependent regions.
3. Discard the input message.
4. STOP the transaction.
5. Back out the database changes done by the application program.
6. Send message DFS555I to the input terminal.
7. Send message DFS554A to the master terminal.
8. Send message DFS3624I to the master terminal.

**Operator response:** Notify the system programmer, then perform the prescribed recovery/restart actions.


**Programmer response:** Refer to message DFS3624I that is sent to the master terminal that contains the function (FC) and return code (RC) identifying the module that detected the error and the error condition (return code). With the return code, see the IMS Return codes and function codes information. An invalid function code could denote a logic error.


**Problem determination:** 1, 2, 3, 4, 5, 6, 7

**Source:** IMS abends

**Module:** DFSFESP0

**Related reference:**

 IMS system services return codes (Messages and Codes)

 Return and function codes associated with the external subsystem support (Messages and Codes)

**Related information:**

Chapter 2, "IMS problem determination," on page 13

## 3053

**Explanation:** This is an IMS pseudoabend issued when either the external subsystem sign-on or commit prepare exit passes back return code X'18' in register 15. This is an indication that the recovery token (NID) associated with the application program already exists in the external subsystem. This condition is most likely to occur when IMS is repeatedly cold started after abnormal terminations while an active external subsystem connection exists.

**Analysis:** This is a pseudoabend issued by DFSESPR0, DFSFESP0.

| This is a pseudoabend issued by DFSESPR0, DFSFESP0, or DFSTMS00

Register 15 is in the following format to indicate the module that detected the nonzero return code:

bytes 1-2 | bytes 3-4

**Bytes 1 and 2**

Contain a hexadecimal value representing the module name that detected the nonzero return code. To determine the module name, refer to either the table in the information on IMS system services return codes or DSECT DFSESEFC.

**Bytes 3 and 4**

Contain the hexadecimal return code passed back by the external subsystem exit routine (that is, X'18').

The above information is provided in the event that abend 3053 becomes a recurring problem. Anticipate an abend 3053 in the event IMS is cold started frequently.

Label RCSOFA in module DFSESS00 (signon) and RETURN in module DFSESP10 (commit phase 1).

**Possible cause:** This condition is most likely to occur when IMS is repeatedly cold started after abnormal terminations while an active external subsystem connection exists. The external subsystem has retained the recovery token (NID) associated with the application program because resolve-in-doubt processing was not completed for that NID.

**System action:** IMS takes the following action:

1. Terminate the application program. If this error is detected during sync point processing, terminate the region.
2. Back out all uncommitted changes.
3. Place the input message being processed at the time back on the IMS input message queue for immediate rescheduling and reprocessing. The act of rescheduling will increase the origin application sequence number (OASN) portion of the NID, thereby making it unique.
4. Send a DFS555 message to the input terminal when the application was scheduled into a BMP.


The transaction or program will not be PSTOPPED.

**Programmer response:** None required. The transaction will be immediately reprocessed without installation/operator intervention. However, if this abend recurs each time the external subsystem connection is established (that is, after /START SUBSYS command), the installation should consult the external subsystem application program documentation to determine the mechanism for manually deleting the recovery tokens in question.

**Source:** IMS abends

| **Module:** DFSESPR0, DFSFESP0, DFSTMS00

**Related reference:**

 IMS system services return codes (Messages and Codes)

## 3054

**Explanation:** 3054 is an IMS abend, issued by either DFSESI40 or DFSESPL0 when an error is detected while attempting to release resources (that is, storage work areas). Register 15 contains the return code passed back by the IMODULE service and a hexadecimal value representing the module that detected the nonzero return code (see below for register 15 content). Register 14 contains the address of the module that detected the nonzero return code.

**Analysis:** 3054 can be either a standard abend detected in the control region or a pseudoabend detected in the dependent region. This abend is issued by modules DFSESI40 and DFSESPL0.

Register 15 contains the return code passed back by the IMODULE service and a hexadecimal value representing the

## 3054

module that detected the nonzero return code (see below for R15 content). Register 14 contains the address of the exact IMODULE DELETE request that failed within the module.

Using the address in register 14, determine which of the above modules issued the abend and the reason the IMODULE delete failed.

Register 15 contains the following:

bytes 1-2 | bytes 3-4

### Bytes 1 and 2

Contain a hexadecimal value representing the module name that detected the nonzero return code. To determine the module name, see either the table in the information about IMS system services return codes or DSECT DFSESFC.

### Bytes 3 and 4

Contain the hexadecimal return code passed back by IMODULE (DFSMODU0).

See contents of register 15 as described above for the key. Go to the address in register 14 to determine which delete failed.

### Label In Module

#### TEMPA00

DFSESI40 / DFSESPL0

#### TEMPC00

DFSESI40 / DFSESPL0

#### TEMPE00

DFSESI40 / DFSESPL0

#### TEMPG00

DFSESI40 / DFSESPL0

#### ESIEX30

DFSESI40

#### ESIEX20

DFSESI40

#### FREETAB

DFSESPL0

**System action:** IMS terminates the TCB representing the external subsystem and prevent a connection between itself (IMS) and the specified external subsystem from being established. If the 3054 occurs during IMS initialization processing, all other IMS tasks (TCBs) are permitted to continue their initialization processing. If it occurs as a result of the /START SUBSYS command, only the external subsystem task is terminated abnormally and other IMS processing continues.

**Operator response:** Notify the system programmer, then perform the prescribed installation recovery/restart actions.

**Programmer response:** For an explanation of the IMODULE DELETE return codes, see the IMS system services return codes information.

Use the address in register 14 to determine which module issued the abend and the reason the IMODULE DELETE failed.


Use the IMS /START command to restart the failing external subsystem.

**Problem determination:** 4, 5, 6, 7

**Source:** IMS abends

**Module:** DFSESI40, DFSESPL0

**Related reference:**

 IMS system services return codes (Messages and Codes)

**Related information:**

Chapter 2, "IMS problem determination," on page 13



---

3055

| **Explanation:** The external subsystem (Db2 for z/OS) was unable to complete a phase 1 commit request, and passed back return code 4 during normal application termination. This abend can also occur when the phase 1 commit request passes back return code X'10' while processing either a commit request during normal application termination or a non-terminating commit request, such as a Message Get Unique request.

| These return codes (RC=X'04', RC=X'10') from DB2 indicates that a DB2 module has voted no to the commit request.

| **Analysis:** This is a pseudoabend issued by modules:

| **DFSPCC20**

|       When RC X'04' or RC X'10' is received for a normally terminating application during phase 1 commit.

| **DFSCPY00**

|       When RC X'10' is received during phase 1 commit for a non-terminating scenario.

| **DFSTMS00**

| The return code is passed back from DB2 to DFSESP10, which in turn passes the return code back to DFSFESP0. DFSFESP0 sets up the pseudoabend.

**System action:** The dependent region terminates with abend 3055.

**Programmer response:** Check the Db2 for z/OS subsystem for abends or messages, or both, that indicate why Db2 for z/OS is unable to process the phase 1 commit request.

One possible cause of an ABEND 3055 might be a suspended user ID; this condition is indicated by a message from RACF or other security product.

**Source:** IMS abends

| **Module:** DFSFESP0, DFSTMS00

---

## 3056

**Explanation:** The DB2 RRSAP interface module, DFSD2AF0, encountered a severe error in the function identified by register 15.

**Analysis:** This is a pseudoabend that is issued by module DFSD2AF0. Message DFS3628I is issued along with the abend for function 00. For functions 01-04, message DFS3629I is issued. Register 2 and register 3 contain the return code and reason code for the failed Db2 for z/OS RRSAP function.

Code	Meaning
X'00'	Initialization
X'01'	RRSAF Identify
X'02'	RRSAF Signon
X'03'	RRSAF Create Thread
X'04'	RRSAF Terminate Thread

**System action:** The dependent region which issued the abend is terminated.

**Programmer response:** Contact the system programmer.

**Problem determination:** 3, 4

**Source:** IMS abends

**Module:** DFSD2AF0

**Related information:**

 DFS3628I

 DFS3629I

Chapter 2, "IMS problem determination," on page 13

---

---

3057

**Explanation:** This abend occurs when an application program attempts to connect to an external subsystem and a connection is not established. One of the following conditions can be the cause of the abend:

- The SSM execute parameter was not specified for the control region defining the external subsystem to IMS.
- The dependent region SSM execute parameter did not include the definition for the external subsystem.
- A batch application program issued an SQL call to a Db2 for z/OS external subsystem and a connection had not been established between IMS and the Db2 for z/OS subsystem. In this case, the responsibility for establishing the connection lies with Db2 for z/OS. See the Db2 for z/OS manuals on how to establish a connection between IMS and Db2 for z/OS in a batch environment.

**Analysis:** This is a pseudoabend issued by DFSLI000, code resident in the dependent region SSOB save area. This code is only executed in the event one or both of the aforementioned conditions occur. Normally the program request handler (DFSESPR0) gets control and process the application request. However, in this case, because the external subsystem support never initialized, the program request handler was not loaded.

This abend also occurs when an application program in a batch region makes an SQL call without specifying DB2, either using DDITV02 or SSM=member.

**Possible cause:** The fact that this abend occurred should prohibit the application that issued the external subsystem request from running (that is, a /PSTOP transaction should be issued). Alternatively, you should insert the appropriate SSM parameter after building the IMS.PROCLIB member defining the subsystem.

**System action:** Application program initiated, external subsystem calls are directed to the external subsystem Program Request Handler (DFSESPR0). Since a connection was not established with the external subsystem, the call could not be processed. The call was intercepted so that the application program could be terminated. IMS will initiate the following action depending on whether the application program is running in an online or batch environment:

1. Terminate the application program.
2. Back out all uncommitted changes.
3. Place the input message being processed at the time on the IMS input message queue for reprocessing.
4. Send a DFS555 message to the inputting terminal.
5. The transaction will be PSTOPPED to prevent it from being rescheduled.

**Operator response:** Notify the system programmer.

**Programmer response:** Determine the reason for the connection failure and make the necessary corrections. If the application program is running in a batch environment, rerun the batch application. If the application program is running in an online environment, restart the dependent region application.

**Problem determination:** 4, 5, 6, 7

**Source:** IMS abends

**Module:** DFSLI000

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

## 3058

**Explanation:** The base cluster block (BCB) storage manager detected a failure.

Modules DFSBCB00, DFSBCB30, and DFSBCB90 support get or release requests for blocks in specific storage pools, referred to as CBT pools. Modules DFSBCB00 and DFSBCB90 are the interface modules for the get and release requests; Module DFSBCB30 provides services in support of the requests. Module DFSBCB60 provides the online storage compression function for these pools.

Requests for CBT storage use macro DFSBCB, which results in a call to DFSBCB00, DFSBCB80 (in module DFSBCB30), or DFSBCB90, depending on the characteristics of the block being requested. Macro DFSBCB can also process the request with an inline expansion for certain block types.

The CBT pools are defined in DFSCBT00 COPY. The SCD field, SCDCBTA, points to the control block table header (DFSCBTHD), which is mapped by DFSCBTS FUNC=DSECT. The field CBTFE in the header points to the first storage pool entry (load module DFSCBT10).

For a given pool, storage is obtained in units termed IPAGES. Each IPAGE begins with a prefix (mapped by

DFSPPRE) that includes the chain to the next IPAGE (PRENP), and the IPAGE free element queue (PRENAB). Blocks are not chained across IPAGES. IPAGES can have different lengths (PREIPL, which includes the prefix) in the same storage pool.

The storage pool entry (mapped by DFSCBTS FUNC=DSECT) points to the oldest IPAGE (CBTEPA) and the latest (CBTCURR). To follow the IPAGE chain, proceed from CBTEPA using PRENP. Each IPAGE includes the four byte name of its associated storage pool (PRENAME). The offset in the block to be used for chaining purposes is found in the storage pool entry (CBTCOFF) and in the IPAGE prefix (PRECO).

If additional IPAGE storage is needed as part of processing a get block request, module DFSBCB30 enqueues a work element to the CSS ITASK running under the job step TCB. Module DFSCSS00 services this queue and starts module DFSSTM00 to issue the GETMAIN.

Blocks for a given storage pool are either address type or block offset type (CBTFLG1:CBTBOFF).

In the first case, the available blocks for the IPAGE are chained from the free queue using storage addresses. For blocks using block offsets, the IPAGE free queue contains the offset to the *first* available block (from the start of the IPAGE). The chain field in the first available block contains the block offset (from the start of the IPAGE) to the *next* available block. The high 16-bits of the block offset are the IPAGE ID, and the low 16-bits are the offset from the start of the IPAGE. To release block offset type blocks, the caller must pass the block offset (that is, not the actual storage address). The get block function returns both values.

When blocks are allocated, they can be formatted using a Block Formatter routine. The name of the associated routine is specified in DFSCBT00 COPY, on the pool definition macro.

When blocks are allocated, they are removed from the IPAGE free queue. Based on a storage pool option (CBTFLG1:CBTACTQ), these allocated blocks can then be threaded on the ACTIVE QUEUE anchored in the storage pool entry (CBTALLOC). The blocks are chained on this queue in the same way that they are chained off the IPAGE Free queue.

In the second case, for block offset type pools with active queue chaining, CBTALLOC contains the block offset of the first block on the queue. It is necessary to locate the IPAGE that contains this block and access its chain field to obtain the block offset of the next element on the queue.

It might be necessary to allocate storage during the processing of a request (that is, the caller provided only a single save area, and additional ones are needed). The storage manager has a queue of such elements for this purpose anchored in the SCD (SCDBCQB). Module DFSXCB00, at initialization time, obtained BXQE storage. If an element is unavailable, the storage manager GETMAINS a BCPE element to enter its logic to obtain more BXQE storage. It then proceeds to complete the request. The blocks referred to are mapped using DFSBCB FUNC=EQU. If there is no more ECSA storage to allocate for the additional storage request, then an abend 3058 can be issued by DFSBCB00.

For performance reasons, certain blocks have a storage pool entry, but the associated storage is not in IPAGE format. These blocks are on queues anchored in the SCD: asynchronous work element (AWE) anchor is SCDAWEQ, BXQE (storage manager internal elements) anchor is SCDBCQB, and SRBC (Data Sharing SRB's) anchor is SCDDSSRB. Each such block has a prefix (mapped by DFSBCB FUNC=EQU), which is used by the storage manager for release validation and chaining.

Module DFSBCB60 is the online storage compression function for the CBT pools. Module DFSXBC60 provides timer services for the BCB60 ITASK. This ITASK locates IPAGES that have no in-use blocks and free that storage. If the IPAGES have been fixed, IMSAUTH is used to first unfix the storage.

**Analysis:** This is a standard abend issued by DFSBCB00, DFSBCB30, DFSBCB60, DFSXBC60, and DFSBCB90.

**For DFSBCB00, DFSBCB30:**

**Register**

**Description**

**Register 5**

Contains qualifying information as indicated

**Register 6**

Abend subcode

**Register 7**

For release errors, register 7 contains the block address/block offset of the block to be released.

**Register 9**

Address of the storage pool entry mapped by DFSCBTS FUNC=DSECT

**Register 10**

Points to DFSBCB00 callers' AWE mapped by DFSDAWE TYPE=BCB) or zero indicates no asynchronous work element (AWE) was passed.

**Register 11**

The SCD address

**Register 13**

Points to the DFSBCB00 callers' save area if it is indicated. Callers original 0, 1, and 15 might have been used.

**For DFSXBC60:****Register 6**

Abend subcode:

Key	Label	Description
Reg6=X'01'	DFSBCB00 RBLK000	Address type block to release was not found on any IPAGE.
Reg6=X'02'	DFSBCB00 RBLK900	The caller passed a chain of blocks to be released. The chain ended before the caller count was zero. Register 5 has the remaining count. The address of the first block in the chain is in the AWBLKA (DFSDAWE) field. The original count is in AWBCNT.
Reg6=X'03'	DFSBCB00 RELBOFF	No IPAGE exists with the IPAGE ID specified in the block offset passed by the caller.
Reg6=X'04'	DFSBCB00 FMT60	The block formatter routine returned a nonzero return code in register 5.
Reg6=X'05'	DFSBCB30 RELBACTQ	Caller is attempting to release a block offset type block. Active queue chaining is indicated for the storage pool and the block offset passed is not on the active queue.
Reg6=X'06'	DFSBCB30 RBACT300	Caller is attempting to release a block offset type block. Active queue chaining is indicated for the storage pool. While following the active queue chain to locate the caller passed block offset (register 7), a block offset was found in register 5 that was not on any IPAGE, indicating that the active queue chain had been destroyed.
Reg6=X'07'	DFSBCB30 RACT400	Caller is attempting to release address type blocks. Active queue chaining is indicated. The block to release (register 7) was not found on the active queue.
Reg6=X'08'	DFSBCB30 RACT705A	A chain of BLOCKS was to be removed from the active queue (address type blocks). The first block was located (register 7). The active queue was exhausted before the user-specified block count (AWBCNT) was reached. The remaining count was in register 5.
Reg6=X'09'	DFSBCB30 RACT210	An attempt was made to locate a block in register 7 on the active queue. There were no IPAGES for this storage pool.
Reg6=X'0A'	DFSBCB30 REL20	The caller is ready to release the storage pool entry latch and additional register space is needed. There were no BXQEs available. The IMODULE GETMAIN failed for a BCBPE element. The high-order 16 bits of register 15 contain the IMODULE return code. The low-order 16 bits contain X'0C'.
	DFSBCB00 FMTIMERR	Additional storage is needed to start the block format routine. IMODULE GETMAIN failed for a BCBPE element. Refer to abend code X'0A'.
Reg6=X'0B'	DFSBCB30 REL20	Refer to the description for subcode X'0A'. The BCBPE was successfully obtained. DFSBCB30 was entered at entry DFSBCB50 to obtain more BXQE storage using module DFSSTM00. The IMODULE GETMAIN for BXQEs issued by DFSSTM00 failed. The high-order 16 bits of register 15 contain the IMODULE return code. The low-order 16 bits contain X'1C'.

Key	Label	Description
	DFSBCB00 FMTBXERR	Additional storage is needed to start the block format routine. Refer to abend code X'0B'.
Reg6=X'0C'	DFSBCB60 FREE00	An error occurred attempting to IMODULE DELETE an IPAGE. Register 3 = IPAGE address Register 5 = subpool passed to IMODULE Register 15 = IMODULE return code IPAGE prefix (DFSPPRE:PRECUB) contains the address of the associated storage pool (DFSCBTE). Message DFS3623E is issued.
Reg6=X'0D'	DFSBCB60 QP40	An error occurred attempting to unfix pages using IMSAUTH FUNC=PGFREE. Register 15- IMSAUTH return code Free list is at label BCBFIX in module DFSBCB60. The original register 6 was saved in register 2. In each IPAGE prefix, DFSPPRE, PRECUB is the address of the associated storage pool. Message DFS3625E is issued.
Reg6=X'0E'	DFSXBC60 INT05	The storage compression timer (DFSXBC60) could not find the address of DFSBCB60. DFSBCB60 is loaded by XSTM0 for the control address space, and DFSXDL00 for the DL/I address space.
Reg6=X'0F'	DFSBCB30 SWITCH	The caller is in cross-memory mode. The caller's save area chain was followed to locate an ITASK ECB. The ECB was not found. Register 13 is the address of the caller's save area chain.
Reg6=X'10'	DFSBCB30 GETECB0	The caller is in cross-memory mode. The caller's save area chain was followed to locate an ITASK ECB. The chain was followed BCBLPCNT times (DFSBCB FUNC=EQU) and the chain did not terminate. Most likely, a save area is chained to itself. Register 13 is the address of the caller's save area chain.

For subcodes 12 and 13, IMS terminates abnormally. It is likely that the IPAGE prefix has been overlaid.

For subcodes 1, 2, 3, 5, 7, 8, and 9, it is likely that the caller of DFSBCB00 passed an invalid input.

For subcode 4, examine the block formatter routine associated with the storage pool to determine why it returned a nonzero return code. The block formatter name is specified on the pool definition macro in DFSCBT00 COPY.

Subcode 6 indicates that an internal IMS logic error occurred.

For subcode 10, IMODULE GETMAIN was issued for 184 bytes (Subpool 228 z/OS online, else 251). For code 11, IMODULE GETMAIN was issued for 4096 bytes (Subpool 231 z/OS online else 251).

Subcode 14 or 16 indicates an internal IMS logic error.

For subcode 15, all DFSBCB requestors who are in cross-memory mode must be ITASKS.

For subcodes other than 12 and 13, IMS can terminate abnormally depending on whether the abend is issued in the control region and whether the TCB that took the abend is reattachable.

#### For DFSBCB90:

Blocks managed by DFSBCB90 have an eight byte suffix immediately following the block. This suffix is used for validation and overlay detection on a release call. The first word of the suffix contains the block type name as a character string of the block being released. The first byte of the second word contains a single character indicating whether the block is allocated ('A'), free ('F'), or initialized but never allocated ('I'). If validation of this suffix fails when a block is released, a 3058 abend can be issued with one of the following subcodes:

Key	Description
Reg15=X'01'	A DFSBCB release call was made, but the block that was released failed validity checking. The block type in the block suffix did not match the block type specified on the release. At time of abend, register 2 points to the block being released, register 14 points to the block suffix, and register 9 points to the CBTE for the block type of the block being released. The first four bytes in the CBTE contain the name of the block type.
Reg15=X'02'	A DFSBCB release call was made, but the block being released had not been allocated. At time of abend, register 2 points to the block being released, register 14 points to the block suffix, and register 9 points to the CBTE for the block type of the block being released. The byte at offset X'04' in the block suffix should have been 'A', indicating that the block was allocated. This abend is issued if the byte is not 'A'.
Reg15=X'04'	The block being obtained has a block formatter routine, but the routine could not be found.
Reg15=X'0A'	A BXQE was needed in order to call a block formatter routine for the block, but the BXQE could not be obtained. IMODULE GETSTOR failed in GET5000 (in DFSBCB30) for a BCBPE block.
Reg15=X'0B'	A BXQE was needed in order to call a block formatter routine for the block, but the BXQE could not be obtained. DFSBCB50 was not able to get a new BXQE IPAGE.

**System action:** IMS terminates abnormally, depending on whether the abend is issued in the control region and whether the TCB which took the abend is reattachable.

**Programmer response:** If IMS terminates, perform appropriate restart procedures.

**Problem determination:** 4, 5, 6, 7

**Source:** IMS abends

**Module:** DFSBCB00, DFSBCB30, DFSBCB60, DFSBCB90

**Related information:**

Chapter 2, "IMS problem determination," on page 13

## 3059

**Explanation:** Module DFSSTM00 was entered to obtain an IPAGE of storage. The generalized IPAGE formatter, DFSSPF00 was called during part of this processing. DFSSPF00 invokes the pool page formatter, if one is specified. On return from the call to DFSSPF00, DFSSTM00 found that the number of blocks (DFSPPRE:PRENOB) on the IPAGE was 0. Refer to abend 3058 for further explanation and analysis.

**Analysis:** This is a standard abend issued by DFSSTM00.

Key	Label	Description
Reg3	DFSSTM00 NEWPG300	Address of the storage pool entry (mapped by DFSCBTS).
Reg5	DFSSTM00 NEWPG300	Address of the IPAGE (prefix mapped by DFSPPRE).

This abend is caused by a defective IMS code. Values in the storage pool entry are suspect (in general, CBTEBL- the block length and CBTBLKN - the number of blocks should not be zero). The storage pool page formatter is also suspect. To find the pool page formatter, DFSCBT00 COPY contains the pool definitions, indicating the page formatter name for this pool (PFN=), if any.

**System action:** IMS terminates abnormally.

**Programmer response:** Refer to the registers to determine the cause of the problem.

**Problem determination:** 4, 5, 6, 7

**Source:** IMS abends

**Module:** DFSSTM00

**Related information:**

3058

Chapter 2, "IMS problem determination," on page 13

---

**3062**

**Explanation:** An error occurred during memory allocation, obtaining storage, or module loading in one of several modules. Reason codes further identify and isolate the type of error:

**Reason code**

	<b>Description</b>
4	A call to obtain storage to SQL10 for DFSSQLA failed. Module: DFSSQL00
15	A call to obtain storage for ISQLA or DFSSOLA failed. Module: DFSAERG0, DFSPDLI0
	<i>or</i>
	PNSQA (IMS Native SQL anchor) does not exist. Module: DFSPDLI0
100	Obtain storage call for Parser work area failed. Module: DBFDT180
104	Obtain storage call for Parser working stack area, Parse Tree work area, or sqlstr failed. Module: DFSSQL10
116	Obtain storage call for Stack block failed. Module: DFSSQL10
120	Detect routine call failed. Module: DFSSQL10
304	PCRE Locate for DFSCAT00 failed. Module: DFSDPSQL
308	Locate_PCRC failed. Module: DFSSQL30

**System action:** IMS terminates abnormally with a dump.

**System programmer response:** Contact IBM Software Support.

**Source:** IMS abends

---

**3090**

**Explanation:** A logic error was detected during Fast Path database tracker initialization. Reason codes further identify the type of error:

**Reason code**

	<b>Description</b>
01	An error return code was returned from IMODULE GETMAIN macro while attempting to get storage for EDBT. The return code from R15 has been saved in R2. Module: DBFDT100
02	An error return code was returned from IMODULE LOAD macro while attempting to load RSR FP related module DBFDTL. The return code from R15 has been saved in R2. Module: DBFDT100
03	An error return code was returned from MVS SETLOCK request. The return code from R15 has been saved in R2. Module: DBFDT100
04	An error return code was returned from MVS TCBTOKEN request. The return code from R15 has been saved in R2.

- Module: DBFDT100  
**05** An error return code was returned from DSPSERV macro, while attempting to create a data space. The return code from R15 has been saved in R2.
- Module: DBFDT100  
**06** An error return code was returned from ALESERV macro, while attempting to create a data space. The return code from R15 has been saved in R2.
- Module: DBFDT100  
**07** An error return code was returned from IMODULE LOAD macro while attempting to load data space management module DFSDSM0L. The return code from R15 has been saved in R2.
- Module: DBFDT100  
**08** An error occurred while getting the address of ESCD.
- Module: DBFDT110  
**09** An error return code was returned from DFSCWU macro while attempting to create an ITASK (FPQS ITASK) for FP DB tracker. The return code from R15 has been saved in R2.
- Module: DBFDT110  
**0A** An error return code was returned from DFSCWU macro while attempting to create an ITASK (FPTI ITASK) for FP DB tracker. The return code from R15 has been saved in R2.
- Module: DBFDT110  
**0B** An error return code was returned from IMODULE macro while attempting to get storage for FP DB tracker's control block (IOTI). The return code from R15 has been saved in R2.
- Module: DBFDT110  
**0C** An error return code was returned from DFSCWU macro while attempting to create an ITASK (IOTI ITASK) for FP DB tracker. The return code from R15 has been saved in R2.
- Module: DBFDT110  
**0D** An error return code was returned from DFSBCB macro while attempting to get storage for SRB (SRB-A). The return code from R15 has been saved in R2.
- Module: DBFDT110  
**0E** An error return code was returned from DFSBCB macro while attempting to get storage for SRB (SRB-B). The return code from R15 has been saved in R2.
- Module: DBFDT110  
**0F** An error return code was returned from IMODULE macro while attempting to get storage for EMAC hash table. The return code from R15 has been saved in R2.
- Module: DBFDT150  
**10** An error return code was returned from DFSBCB macro while attempting to get storage for AWE. The return code from R15 has been saved in R2.
- Module: DBFDT150  
**11** An error return code was returned from IMODULE macro while attempting to get storage for TUR hash table. The return code from R15 has been saved in R2.
- Module: DBFDT150  
**12** An error return code was returned from IMODULE macro while attempting to get storage for OFR\_TUR hash table. The return code from R15 has been saved in R2.
- Module: DBFDT150  
**13** An error return code was returned from DFSCWU macro while attempting to create ITASK (LRQS ITASK). The return code from R15 has been saved in R2.
- Module: DBFDT100  
**14** An error return code was returned from IMODULE macro while attempting to get storage for DB buffer header. The return code from R15 has been saved in R2.
- Module: DBFDT110  
**15** An error return code was returned from IMODULE macro while attempting to get storage for DB buffer. The return code from R15 has been saved in R2.
- Module: DBFDT150



- 16 An error return code was returned from IMSAUTH macro while attempting to page fix DB buffer. The return code from R15 has been saved in R2.  
Module: DBFDT150
- 24 The Fast Path tracker is unable to load DFSLRIQS from the libraries specified in the JOBLIB and STEPLIB DD cards. Examine the libraries and make sure DFSLRIQS exists in one of them. If it does, report this problem to IBM. If not, determine why the module was not installed with the rest of IMS. If it was installed, determine why it was removed from the library.

**Source:** IMS abends

### 3091

**Explanation:** A logic error was detected during Fast Path database tracking running under Log Router TCB. Reason codes further identify the type of error:

#### Codes

##### Explanation

- 01 An invalid request code was detected by FP database tracking AWE queue server.  
Module: DBFDT250
- 03 During restart of OFR, AREA control block could not be found for the AREA that was OFR-in-progress when IMS terminated.  
Module: DBFDT270
- 04 An error return code was returned from IMODULE GETSTOR, indicating that storage for an AWE could not be obtained in CSA. The return code from IMODULE has been saved in R2.  
Module: DBFDT250, DBFDT260
- 05 An invalid type of FP database redo record was encountered during tracking redo-queue processing.  
Module: DBFDT250

**System action:** IMS terminates abnormally with a dump.

**System programmer response:** Contact IBM.

**Source:** IMS abends

### 3093

**Explanation:** A logic error was detected during FP database tracking running under FP TCB. Reason codes further identify the type of error:

#### Reason code

##### Description

- 01 An error return code was returned from DFSBCB GET, indicating that AWE could not be created. The return code from DFSBCB has been saved in R2.  
Module: DBFDT300
- 02 An error return code was returned from DFSBCB GET indicating that AWE could not be created when attempting to create DBRC request area. The return code from DFSBCB has been saved in R2.  
Module: DBFDT300
- 03 An invalid request code was detected by FP database tracking AWE queue server.  
Module: DBFDT300
- 04 An invalid type of AWE was detected by FP database tracking AWE queue server. The storage of the AWE cannot be released.  
Module: DBFDT300
- 05 A nonzero return code was returned by DBRC stream complete notification macro. See message DFS2966A.  
Module: DBFDT300

**System action:** IMS terminates abnormally with a memory dump.

## 3094 • 3100

**System programmer response:** Contact IBM.

**Source:** IMS abends

**Related information:**

 DFS2966A (Messages and Codes)

---

### 3094

**Explanation:** A logic error was detected during Fast Path database tracking. Reason codes further identify the type of error:

**Reason code**

- |    | Description   |
|----|---|
| 01 | An invalid function code was detected by FP database status change processing.<br>Module: DBFDT180  |
| 02 | An error return code was returned from DFSBCB GET indicating that AWE could not be created by FP database status change processing. The return code from DFSBCB has been saved in R2.<br>Module: DBFDT180 |
| 03 | A nonzero return code was returned by DBRC early end tracking macro. See message DFS2969A.<br>Module: DBFDT180  |

**System action:** IMS terminates abnormally with a dump.

**System programmer response:** Contact IBM.

**Source:** IMS abends

---

### 3100

**Explanation:** An internal error was detected in the structure of the parse table.

**Analysis:** This is a standard abend issued by the MFS preprocessor parse error analysis routine, DFSUPAM0.

Each source statement is processed by a separate module of the preprocessor. Each statement processor contains a table that defines the syntax of the statement. The statement processors call module DFSUPAM0 to perform syntax checking on the current statement using the syntax table (addressed using PARSTABL in MFSGBL) provided by the statement processor.

The program status word (PSW) at entry-to-abend points to the instruction within label XFERBUMP from which the abend (SVC 13) is issued. Register 11 in the abend SVRB points to the MFS preprocessor global dictionary (MFSGBL) passed by a statement preprocessor module. The statement preprocessor module that called DFSUPAM0 can be found by obtaining register 14 from the save area set, pointed to by register 13 in the abend SVRB, and using it as the return address. Register 9 points to the parse stack entry (PSE) for the current state of the parse stack.

Key	Label	Description
Reg1=X'80000C1C' Reg9 Parse Stack Entry (PSE) Reg10 to the PARSTABL	XFERBUMP	A match between the current entry and the transfer entry was not found prior to looping through the entire parse table. This indicates an internal error. Register 14 contains the return address of the caller who passed this table.

**Possible cause:** MFSGBL area destroyed. Parse table of calling module destroyed, or Parse Stack Entry destroyed.

**System action:** The MFS preprocessor terminates abnormally.

**Problem determination:** 1, 2, 8, 11, 12, 16, 35

**Source:** IMS abends

**Module:** DFSUPAM0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

**640** Messages and Codes, Volume 3: IMS Abend Codes

---

## Chapter 26. IMS abend codes 3101 - 3300

This topic identifies and explains the following abend codes issued by IMS. Each code is 4 decimal digits.

---

### 3101

**Explanation:** An internal logic error has occurred while trying to provide error recovery on a source statement.

**Analysis:** This is a standard abend issued by the MFS preprocessor parse error analysis routine, DFSUPAM0.

Module DFSUPAM0 calls module DFSUPBM0 if any syntax errors are detected. DFSUPBM0 attempts to correct the error by assigning a valid default. Should DFSUPBM0 encounter algorithm trouble, the error recovery logic is in error. Return is to DFSUPAM0, indicating a catastrophic error.

The abend is issued within label CATSTROP. The program status word (PSW) at entry-to-abend points to the instruction within this label from which the abend (SVC 13) was issued. Use register 14 in the abend SVRB to isolate to a specific label. Register 12 is the base register.

Key	Label	Description
Reg14=BAL to DFSUPBM0	LAB4	The return from DFSUPBM0 indicates an algorithm trouble was encountered. This is a critical error.
Reg14 ≠BAL to DFSUPBM0 Reg10+X'CO'≠X'0001'	PAMEND	The parse operation is complete; there should be only a single entry on the parse stack. The compare within label PAMEND failed resulting in a branch to CATSTROP.

**System action:** The MFS preprocessor terminates abnormally.

**Programmer response:** After performing problem determination, examine the source statement causing the failure, correct the errors or reorder the operands, and resubmit the job.

**Problem determination:** 1, 11

**Source:** IMS abends

**Module:** DFSUPAM0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

### 3102

**Explanation:** An internal logic error has occurred while attempting to create or write an ITB prior to being written to the IMS.REFERAL data set.

**Analysis:** This is a standard abend issued by the MFS Preprocessor Referral Manager, DFSUPAX0.

DFSUPAX0 is called by the various statement preprocessors to acquire storage for intermediate text blocks (ITBs). ITB storage is allocated in 4K blocks called text buffers.

The program status word (PSW) at entry-to-abend points to the instruction within label USER3102 from which the abend (SVC 13) is issued. Register 3 in the abend SVRB contains the size for a LOCATE space request. The caller's return address can be found by using register 14 in the save area set pointed to by register 13 in the abend SVRB. Register 12 is the base register.

Key	Label	Description
Reg1=X'80000C1E' Reg3>X'0FF0'	NEEDSTG	The size of the space request exceeds the size of one text block.

## 3103

**Possible cause:** Invalid parameters were passed by the calling module.

**System action:** The MFS preprocessor terminates abnormally.

**Problem determination:** 1, 2, 8, 11, 16, 35

**Source:** IMS abends

**Module:** DFSUPAX0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

## 3103

**Explanation:** An internal logic error has occurred while trying to classify the character currently being processed.

**Analysis:** This is a standard abend issued from Get Item processor module, DFSUPAD0.

Use the registers in the abend SVRB for problem isolation. The register 14 BAL can be used to determine where the error was detected. Register 11 in the abend SVRB points to the MFSGBL.

The abend is issued from the common abend routine at label ABEND001. The program status word (PSW) at entry-to-abend points to the abend (SVC 13) at this label.

Key	Label	Description
Reg14=BAL Reg11=MFSGBL Reg15>'X'1C' SOURCE=current source character	CHECKVAL	This routine gets a character from the area CARD of MFSGBL. It continues to obtain characters until an item is formed. Register 15 contains the current source character value, and is checked to see if this indexing value is too high for this item list. If it is, the abend is issued.
Reg14=BAL Reg15>'X'1C' SOURCE=current source character	ALPHA	This routine has obtained the characters for an alphameric item. Register 15 contains the current source character value that is checked to see if this indexing value is too high for this item. If it is, the abend is issued.
Reg14=BAL Reg15>'X'1C' SOURCE=current source character	NUMERIC	The characters for a numeric item are obtained. Register 15 contains the current source character value that is checked to see if this indexing value is too high for this item. If it is, the abend is issued.
Reg14=BAL Reg15>'X'1C' SOURCE=current source character	AMERIC	The characters for this item are obtained. Register 15 contains the current source character value that is checked to see if this indexing value is too high for this item. If it is, the abend is issued.
Reg14=BAL Reg15>'X'1C' SOURCE=current source character	QUOTE	The characters for a quote (literal) item are obtained. Register 15 contains the current source character value that is checked to see if this indexing value is too high for this item. If it is, the abend is issued.
Reg14=BAL Reg15>'X'1C' SOURCE=current source character	QUOQUO	The character for this item has been obtained. Register 15 contains the current source character value that is checked to see if this indexing value is too high for this item. If it is, the abend is issued.

**Possible cause:** The causes include:

- An error in a source statement
- VALUETBL in MFSGBL contains an incorrect translate table for source record translation. The ALPHA statement processor (DFSUPAN0) might have incorrectly modified VALUETBL.

**System action:** The MFS preprocessor terminates abnormally.

**Programmer response:** After performing problem determination, examine the source statement causing the failure, correct the error, and resubmit the job.

**Problem determination:** 1, 2, 8, 11, 16, 35

**Source:** IMS abends

**Module:** DFSUPAD0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

### 3104

**Explanation:** An internal logic error has occurred while the MFS preprocessor was processing a delimiter. It was trying to append the source character delimiter to a stack of characters representing the item just scanned. The ITEM area within MFSGBL overflowed.

**Analysis:** This is a standard abend issued from Get Item processor module, DFSUPAD0.

Use the registers in the abend SVRB for problem isolation. The abend is issued from the routine at label DELMOVE. The program status word (PSW) at entry-to-abend points to the abend (SVC 13) at this label.

Key	Label	Description
SOURCE ^=C.' Reg15=SDW SDWCURLT>SDWMAXLT	DELIMIT	The item list has been formed and based on the translated value of the current source character. The delimiter character cannot be appended to the end of the current item because the item's length exceeds the maximum (SDWMAXLT). The current length of the string descriptor word (SDW) exceeds the maximum length.

**System action:** The MFS preprocessor terminates abnormally.

**Programmer response:** Examine the source statement causing the failure, correct the error, and resubmit the job.

**Problem determination:** 1, 2, 8, 11, 16, 35

**Source:** IMS abends

**Module:** DFSUPAD0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

### 3105

**Explanation:** An internal logic error in the branch vector table has been detected, after finding a continuation character.

**Analysis:** This is a standard abend issued from module DFSUPAE0.

Use the register in the abend SVRB for problem isolation. GETCFUNC is a 1-byte cell indicating the current function being performed while obtaining an input character from SYSIN. GETCFUNC is assigned a value that is used as an index value when one state is completed and the next state is to be executed. This abend is issued when the GETCFUNC index value is greater than the permitted value.

The abend is issued from the common abend routine at label RTN8. The program status word (PSW) at entry-to-abend points to the abend (SVC 13) at this label.

Key	Label	Description
Reg15=GETCFUNC Reg15>X'18'	MOVESOR	This routine obtains a source character from the CARD area of MFSGBL and branches to perform the function defined by GETCFUNC. GETCFUNC is put in register 15 and then checked to see if the range of the indexing value is too high for this list. If it is, a branch is taken to label RTN8 to issue the abend (SVC 13).
Reg15=GETCFUNC Reg15>X'18' or Reg15=0	CHKPOS72	This routine checks column 72 for a nonblank character and processes the continuation record. If 72 is a blank, there is no continuation record. GETCFUNC (the indexing value) is set in register 15 and register 15 is checked to see if the indexing value is too high. If the indexing value is 0, the abend is also issued.

Key	Label	Description
Reg15=GETCFUNC Reg15=0 or Reg15>X'18'	BLNK72	Column 72 is blank so this routine processes the end-of-statement item. GETCFUNC (the indexing value) is set in register 15 and register 15 is checked to see if the indexing value is too high. If it is, the abend is issued. The abend is also issued if the indexing value is 0.

**System action:** The MFS preprocessor terminates abnormally.

**Programmer response:** After performing problem determination, examine the source statement causing the failure, correct the error, and resubmit the job.

**Problem determination:** 1, 2, 8, 11, 16, 35

**Source:** IMS abends

**Module:** DFSUPAE0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

### 3107

**Explanation:** An internal logic error has occurred on a PUT call, while attempting to create or write an ITB to the IMS.REFERAL data set. The storage area most recently located is not the same area supplied on a PUT call.

**Analysis:** This is a standard abend issued by the MFS preprocessor REFERAL library manager, DFSUPAX0.

A PUT call from one of the various statement preprocessor modules results in the movement of an ITB to a text buffer. A LOCATE call for the ITB space must precede the PUT call.

The program status word (PSW) at entry-to-abend points to the instruction within label USER3107 from which the abend (SVC 13) is issued. Register 3 in the abend SVRB contains the size of the ITB to be moved. The value in register 3 is compared to the size of the space requested by the previous LOCATE call. The size requested by the previous LOCATE call is located in the MFSGBL at label STGLOC. Register 11 in the abend SVRB is the pointer to the MFSGBL. The caller's return address can be found by using register 14 in the save area set pointed to by register 13. Register 12 is the base register.

Key	Label	Description
Reg1=X'80000C23' label STGLOC	Reg3=address at PUT	The PUT call was not for the area most recently located.

**Possible cause:** Invalid parameters were passed by the calling module, or an invalid call sequence of LOCATE and PUT was issued.

**System action:** The MFS preprocessor terminates abnormally.

**Problem determination:** 1, 2, 8, 11, 16, 35

**Source:** IMS abends

**Module:** DFSUPAX0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

### 3108

**Explanation:** An internal logic error has occurred on a PUT call, while attempting to create or write an ITB to the IMS.REFERAL data set.

**Analysis:** This is a standard abend issued by the MFS preprocessor REFERAL library manager, DFSUPAX0.

A PUT call from one of the various statement preprocessor modules results in the movement of an ITB to a text buffer. A LOCATE call for the ITB space must precede the PUT call and must be for the same amount.

The program status word (PSW) at entry-to-abend points to the instruction within label USER3108 from which the abend (SVC 13) is issued. The PUT area length is contained in the MFSGBL at label STGLOCS. Register 11 in the abend SVRB is a pointer to the MFSGBL, register 5 contains the requested PUT area length, and register 12 is the base register. The caller's return address is pointed to by register 14 in the save area set pointed to by register 13 in the abend SVRB.

Key	Label	Description
Reg1=X'80000C24' Reg5>size in STGLOCS	PUT	The size of the call is greater than the size of the LOCATE call.

**Possible cause:** The calling module has increased the amount of storage to be PUT to a greater amount than that which was previously LOCATED, or the MFSGBL area has been destroyed.

**System action:** The MFS preprocessor terminates abnormally.

**Problem determination:** 1, 2, 8, 11, 16, 35

**Source:** IMS abends

**Module:** DFSUPAX0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

### 3109

**Explanation:** An internal logic error has occurred while attempting to create or write an ITB to IMS.REFERAL.

**Analysis:** This is a standard abend issued by the MFS Preprocessor REFERAL Library Manager, DFSUPAX0.

While searching a text block for a specific ITB it was ascertained that the ITB did *not* reside wholly within one text block. This is an illogical condition. This condition would be detected if a PRINT of the ITB is requested. Printing of an ITB is controlled by two statements; the PDIAG=DIAG in the parameter field of the EXEC JCL statement for the MFS utility program, or the compilation statement PRINT=ON,GEN is coded (these operands are the defaults).

The program status word (PSW) at entry-to-abend points to the instruction within label USER3109 from which the abend (SVC 13) was issued. Register 10 in the abend SVRB points to the requested ITB; register 4, the pointer to the next text block, contains zero. DSECT BLK using register 4 describes the text block linkage.

Key	Label	Description
Reg1=X'80000C25' Reg4=zero	NOTHERE	While processing a PRINT ITB operation the requested ITB could not be located.

**Possible cause:** MFSGBL has been destroyed, or text blocks have been destroyed.

**System action:** The MFS preprocessor terminates abnormally.

**Problem determination:** 1, 2, 8, 11, 16, 35

**Source:** IMS abends

**Module:** DFSUPAX0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

### 3110

**Explanation:** An internal logic error has occurred while attempting to create or write an ITB to IMS.REFERAL.

**Analysis:** This is a standard abend issued by the Library Write Function module, DFSUPB60.

As new intermediate text blocks (ITBs) are generated, module DFSUPAX0 calls DFSUPB60 to write the ITBs to the IMS.REFERAL data set. DFSUPB60 validity checks field ITBTYPE within the ITB to ensure it was created by only a TABLE, FORMAT, or MESSAGE statement.

## 3111 • 3115

The program status word (PSW) at entry-to-abend points to the instruction within label USER3110 from which the abend (SVC 13) is issued. Register 4 in the abend SVRB is the pointer to the current ITB being processed, and the register 12 is the base register. Valid ITBTYPES are:

Code	Meaning
X'20'	TABLE ITB
X'01'	FORMAT ITB
X'11'	MESSAGE ITB

Key	Label	Description
Reg2=BAL Reg4 to the ITB ITBTYPEN=valid type	DFSUPB60	An ITB is scheduled to be written to IMS.REFERAL but the validity check operation failed to identify it as a valid ITB.

**Possible cause:** The calling module did not passed a valid ITB.

**System action:** IMS terminates abnormally.

**Problem determination:** 1, 11

**Source:** IMS abends

**Module:** DFSUPB60

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

### 3111

**Explanation:** The MFS Language utility preprocessor tried to analyze a source statement and an internal logic error occurred in the preprocessor. The processor was analyzing the device type in the DEVITB (DEVOCCR4). The device code did not match any valid device type. Register 6 contains the function code showing the statement processor that took the abend. Register 15 contains the invalid value that caused the abend.

**Analysis:** Register 6 contains one of the following error codes to indicate the module that issued the abend.

Code	Meaning
X'04'	DFSUPAZ0, the DEV statement processor.
X'08'	DFSUPA10, the DPAGE statement processor.
X'0C'	DFSUPA30, the DFLD statement processor.

Register 15 contains the invalid-device code from the field DEVOCCR in DEVITB. For DFSUPA10 and DFSUPA30, register 3 contains a pointer to the DEVITB. For DFSUPAZ0, register 2 contains a pointer to the DEVITB.

**Possible cause:** An incorrect DEVITB address or overlaid data within the DEVITB exists.

**System action:** The MFS preprocessor terminates abnormally.

**Problem determination:** 1, 2, 8, 11, 16, 35

**Source:** IMS abends

**Module:** DFSUPAZ0, DFSUPA10, DFSUPA30

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

### 3115

**Explanation:** An internal logic error occurred during the stacking or unstacking of records for the MFS preprocessor source input.

**Analysis:** This is a standard abend issued from module DFSUPAF0, DFSUPBH0, DFSUPBJ0, or DFSUPBK0.

The program status word (PSW) at entry-to-abend determines which module detected an error and issued the abend.

**For DFSUPAF0:**



Use the registers in the abend SVRB for problem isolation. Register 10 contains the address of the point at which the abend was initiated. Register 12 is the base register. Register 11 points to MFSGBL, which contains the stack pointers and ID.

The program status word (PSW) at entry-to-abend points to the abend (SVC 13) within the routine which detected the error.

Key	Label	Description
Reg10=BAL Reg14=BAL to DFSUPBK0 Reg15 ≠0 STACKID= stack just unstacked STKIDQCB= stack ID Queue Control Block	ENDSTACK	The end of the stack has been reached and UNSTACKing is terminated. This routine deletes the stack that was just UNSTACKed (QE and blocks). DFSUPBK0, the symbol table search routine, is BALed to check for a STACKID entry. A nonzero return code is passed in register 15, indicating that the STACKID entry was not found in the symbol table, and the abend is issued.
Reg10=BAL Reg14=BAL to DFSUPBK0 Reg15≠0 STACKID=ID of stack to that record is to be added	STACKCRD	A record is to be added to stack that corresponds to the identifier in the STACKID of MFSGBL. A BAL is made to DFSUPBK0 to search for the STACKID entry in the symbol table. A nonzero return code is passed in register 15 and the abend is issued because a QE must exist if stacking is in progress.

#### For DFSUPBH0, DFSUPBK0:

The registers in the abend SVRB should be used for problem isolation. Register 10 contains the address of the point at which the abend was initiated.

The program status word (PSW) at entry-to-abend points to the abend (SVC 13) within the routine that detected the error.

Key	Label	Description
Reg10=BAL READSWS=X'10'	TMUNSTK2	This routine processes the 'STACK ON, ID' statement. UNSTACK is active (READSWS=X'10'). This is an error, because there should not be a 'STACK ON, ID' statement while unstacking is taking place. Unstacking is the process of obtaining the next source statement from an incore stack. The statement had been placed on the stack by a previous STACK ON, ID statement. The label at which the abend is issued is UABEND1.
Reg10=BAL Reg14=BAL to DFSUPBK0 Reg15≠0	DLETCARD	The records from the storage stack associated with a specific ID are to be deleted. On return from DFSUPBK0, where a search for the stack ID entry was made in the symbol table, register 15 is nonzero, indicating that the entry was not found. The abend is then issued at label UABENDDD1.
Reg10=BAL Reg4=STKIDQE FIRSTBLK=0 Reg1= first block in queue=0	HAVEQE	This routine is to find the last block on the current stack. Register 4 is set up to establish addressing to the QE. Register 1 is loaded with the address of the first block for the ID (FIRSTBLK). Register 1 is tested and found to be zero, so the abend is issued at label UABENDDD2.

#### For DFSUPBJ0:

Use the registers in the abend SVRB for problem isolation. Register 10 contains the address of the point at which the abend was initiated.

## 3116

The program status word (PSW) at entry-to-abend points to the abend (SVC 13) within the routine that detected the error.

Key	Label	Description
Reg10=BAL READSWS=X'10'	TMUNSTK1	This routine processes the 'UNSTACK ID, KEEP/DELETE' statement. UNSTACK is active (READSWS=X'10'). Nested STACK and UNSTACK statements are not permitted. It is, therefore, an illogical condition to find an UNSTACK statement in a stack while an UNSTACK statement is in progress. Therefore, an abend is issued from label UABEND1.
Reg10=BAL Reg14=BAL to DFSUPBK0 Reg15=0	DLETCARD	The statements from the incore stack associated with a specific ID are to be deleted. On return from DFSUPBK0, where a search for the stack ID entry was made in the symbol table, Register 15 is nonzero, indicating that the entry was not found. The abend is then issued from label UABENDDD1.
Reg4=STKIDQE FIRSTBLK=0 Reg10=BAL	HAVEQE	This routine is to find the last block on the current stack. Register 4 is set up to establish addressing to the Queue Element (QE). Register 1 is loaded with the address of the first block for the ID (FIRSTBLK). Register 1 is tested and found to be zero, so the abend is issued from label UABENDDD2.
Reg2=CARDSPER Reg2>1 Reg10=BAL	NOPREVBL	No more blocks containing source statements to be unstacked exist. Yet CARDSPER (statements/records left to delete) is >1. The statements/records left to be deleted are loaded into register 2 from CARDSPER. Register 2 is then compared to see if there really are any statements/records left. If there are, the abend is issued from label UABENDDD3.

**System action:** The MFS preprocessor terminates abnormally. Register 10 contains the address of the point at which the abend was initiated.

**Problem determination:** 1, 2, 8, 11, 16, 35

**Source:** IMS abends

**Module:** DFSUPAF0, DFSUPBH0, DFSUPBJ0, DFSUPBK0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

## 3116

**Explanation:** An internal logic error occurred in a general sort routine used by the MFS preprocessor.

**Analysis:** This is a standard abend issued from module DFSUPAZ0.

Use the registers in the abend SVRB for problem isolation. Register 14 contains the return address immediately following the BAL instruction for invocation of EXCHSORT or NOZESORT subroutine.

The program status word (PSW) at entry-to-abend also points to the code which detected the error.

The registers of the caller of the subroutine EXCHSORT or NOZESORT are saved (register 14 through register 12) at label TEMPSAVE. Within these registers, register 1 contains a parameter list address. The parameter list has the format:

**halfword**

Number of entries to be sorted

**halfword**

Size of each entry

**fullword**

Pointer to first entry

Key	Label	Description
Reg9=number of entries	EXCHSORT	The number of entries to be sorted is obtained from the parameter list and loaded into register 9. Register 9 is compared to see if the number of entries is greater than one (1). If it is not, the abend is issued for an invalid number of entries for sort.
Reg5=first entry Reg8=0 Reg9=last entry	GOODNUM	The number of entries to be sorted is greater than one (1). Register 8 is loaded with the size of an entry (in bytes) and tested to see if it is greater than zero. If it is not, the abend is issued.
Reg9=number of entries	NOZESORT	The number of entries to be sorted is obtained from the parameter list and loaded into register 9. Register 9 is compared to see if the number of entries is greater than one (1). If it is not, the abend is issued for an invalid number of entries for sort.
Reg5=first entry Reg8=0 Reg9=last entry	GOODNUMZ	The number of entries to be sorted is greater than one (1). Register 8 is loaded with the size of an entry (in bytes) and tested to see if it is greater than zero. If it is not, the abend is issued.

**System action:** The MFS preprocessor terminates abnormally.

**Problem determination:** 1, 2, 8, 11, 16, 35

**Source:** IMS abends

**Module:** DFSUPAZ0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

### 3120

**Explanation:** During IMS restart, module DFSCMR00 is needed to reinstate the MSC control blocks, but is not available.

**System action:** The ABEND was preceded by an IMS DFS3120I message.

**Programmer response:** Make sure that DFSCMR00 is linked into DFSRST00 load module.

**Source:** IMS abends

**Module:** DFSRLP00

**Related information:**

 DFS3120I

### 3141

**Explanation:** An error was detected while an XRF alternate system or a Fast Database Recovery region was reading the log data set during emergency restart.

**Analysis:** SCDDKDCB contains the address of the DCB, and SCDRSTEB contains the address of the PST used for the OSAM read operation. This abend is issued from label ABND3141 in module DFSRLP00.

**Possible cause:** Some possible causes are:

- An I/O error occurred on the restart data set (RDS).
- Extended recovery facility (XRF) specifications cannot be changed in this restart. See message DFS3851I.
- The other system is not the active system. See message DFS3852I.
- The IMSID name does not match the checkpoint recoverable service element (RSE) name. See message DFS3868I.
- The HSBID for the XRF primary and active systems cannot be equal. See message DFS3894I.
- An attempt was made either to start a DB/DC system with a Database Control (DBCTL) log or to start a DBCTL system with a DB/DC log.

## 3265

**System action:** IMS terminates abnormally with abend 3141.

**Operator response:** After reloading, restart IMS using the /ERE command. If an I/O error occurred, a restart from an archived log might be necessary.

**Problem determination:** 1, 2, 3, 4, 6, 35

**Source:** IMS abends

**Module:** DFSRLP00

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

## 3265

**Explanation:** DFSFDLW0 tries to find the log blocks following the last log block in the OLDS input, but the WADS had detected one or more than one I/O errors. The log recovery utility encountered the error.

**Analysis:** This is a standard abend that issued by DFSULG20. The log recovery utility, DFSULTR0, a composite module, and DFSULG10 are the entries to CSECT. DFSULG10 branches and links registers (BALRs) to module DFSULG20 if the DUP or CLS mode is specified. When the end-of-data is encountered or the number of errors reaches the specified user number, DFSULG20 calls DFSFDLW0 to search the following log blocks in WADS, which also contains errors. When this abend is issued the program status word (PSW) at entry-to-abend points to the instruction from which the abend (SVC 13) is issued in DFSULG20.

Register 12 in the abend SVRB registers in the base register for DFSULG20. Register 11 points to the last read log block. Register 8 points to a WADS process subroutine (WADSPROC). Register 15 contains the following abend codes issued by DFSFDLW0.

One of the following errors can be issued during a WADS selection:

<b>Code</b>	<b>Meaning</b>
X'01'	OPEN failure
X'02'	Device type error or DEVTYP macro failure
X'03'	CLOSE failure
X'04'	WADS read error
X'05'	End-of-file detected in WADS
X'06'	No WADS is available
X'07'	TRKCALC macro failure

One of the following errors can be issued during block rebuilding:

<b>Code</b>	<b>Meaning</b>
X'08'	OPEN failure
X'09'	The next block was unavailable but a following block is available
X'0A'	X'FF' segment count has been found
X'0B'	Sequence error in the time stamp in the OLDS suffix
X'0C'	Invalid block size (BDW)
X'0D'	Invalid RDW (record length is less than 5 or length is too large)
X'0E'	RDW of X'FFNN' has been found
X'0F'	Error in log record sequence
X'10'	Read error
X'11'	End-of-file detected in WADS
X'12'	Invalid segment count
X'13'	Floating segment has been found
X'14'	Logic error

The following labels are issued by module DFSFDLW0:

Key	Label	Description
Reg15=X'01' RLWABC=X'01' DCBOFLGS = X'10' PDCB - primary WADS DCB	OPEN	DFSFDLW0 issues an OPEN (SVC19) for the primary DCB while trying to locate the last used WADS. The DCB is tested to determine if OPEN was successful. If OPEN is not successful, it returns a return code of X'12' and reason code of X'01'.
Reg15=X'02' RLWABC=X'02' DAREA1,DAREA2 - device type save area	LTA1240	DFSFDLW0 issues a DEVTYPE macro for the primary DCB while trying to locate the last used WADS. If a nonzero code was returned from the DEVTYPE macro and was not the same as the previously processed WADS device type, DFSFDLW0 returns a return code of X'12' and a reason code of X'02'.
DAREA1 - device type save area	LTA1250	DFSFDLW0 issues a DEVTYPE macro for the primary DCB while trying to locate the last used WADS. If a nonzero code in register 15 was returned from the DEVTYPE macro, DFSFDLW0 returns a return code of X'12' and a reason code of X'02'.
Reg15=X'03' RLWABC=X'03' DCBOFLGS= X'10' PDCB - primary WADS DCB	CLOSE	DFSFDLW0 issues a CLOSE (SVC 20) for the primary DCB while trying to locate the last used WADS. The DCB is tested to determine if CLOSE was successful. If CLOSE is not successful, it returns a return code of X'12' and a reason code of X'03'.
Reg15=X'04' RLWABC=X'04' DECBA - DECBA address	LTG210	DFSFDLW0 issues a BSAM READ for the primary DCB while trying to locate the last used WADS. DECBA is tested to determine if the READ was successful. If READ is not successful, (second error if dual WADS mode), it returns a return code of 12 and a reason code of X'04'.
Reg15=X'05' RLWABC=X'05' IOEI=X'40'	LTG20R	DFSFDLW0 issues a BSAM READ for the primary DCB while trying to locate the last used WADS. If the end-of-data condition was detected, it returns a return code of X'12' and a reason code of X'05'.
Reg15=X'06' RLWABC=X'06' WADSAM - WADS availability matrix save area	LTA120E	If WADS at the time of the last IMS failure was not available, DFSFDLW0 returns a return code of X'12' and a reason code of X'06'.
Reg15=X'07' RLWABC=X'07'	SEGMENTC	DFSFDLW0 issues a TRKCALC macro for the primary DCB while trying to locate the last used WADS. If the macro returned a nonzero return code in register 15, DFSFDLW0 returns a return code of X'12' and a reason code of X'07'.
Reg15=X'08' RLWABC=X'08' DCBOFLGS = X'10' PDCB - primary WADS DCB	LTA2100	DFSFDLW0 issues an OPEN (SVC19) for the primary DCB while rebuilding an OLDS block. DCB is tested to determine if OPEN was successful. If OPEN is not successful, it returns a return code of X'12' and a reason code of X'08'.
SDCB - secondary WADS DCB	LTA2110	DFSFDLW0 issues an OPEN (SVC19) for the secondary DCB while rebuilding an OLDS block. DCB is tested to determine if OPEN was successful. If OPEN was not successful, it returns a return code of X'12' and a reason code of X'08'.
Reg15=X'09' RLWABC=X'09' BLKTABA - block table address	LTA260E	When DFSFDLW0 cannot find an OLDS block in the WADS contiguous to the last block in the OLDS, but found a higher sequence block in the WADS, DFSFDLW0 returns a return code of X'12' and a reason code of X'09'.
	LTA2640	When DFSFDLW0 found an OLDS block in the WADS contiguous to the last block in the OLDS and a noncontiguous sequence block was found in the following blocks in the WADS, DFSFDLW0 returns a return code of X'12' a reason code of X'09'.

Key	Label	Description
Reg15=X'0A' RLWABC=X'0A' WADSBA - WADS buffer address	LTA320R	When DFSFDLW0 detected a segment-ID of X'FF' in the WADS buffer, DFSFDLW0 returns a return code of X'12' and a reason code of X'0A'.
Reg15=X'0B' RLWABC=X'0B' SEG TABA - segment table address	LTA330R	When DFSFDLW0 could not find a complete set of segments for an OLDS block and the block was not the last one in WADS, DFSFDLW0 returns a return code of X'12' and a reason code of X'0B'.
	LTA3310	When DFSFDLW0 could not find segment 0 but found following segment, it returns return code of X'12' and a reason code of X'0B'.
Reg15=X'0C' RLWABC=X'0C' RLWPBA - OLDS buffer address	TRUNC INCOMSEG	When DFSFDLW0 detected an invalid BDW in the rebuilt OLDS block, DFSFDLW0 returns a return code of X'12' and a reason code of X'0C'.
Reg15=X'0D' RLWABC=X'0D' RLWPBA - OLDS buffer address	LTA3410 LTA3540	When DFSFDLW0 detected an invalid RDW in the rebuilt OLDS block, it returns return code of X'12' and a reason code of X'0D'.
Reg15=X'0E' RLWABC=X'0E' RLWPBA - OLDS buffer address	LTA350R	When DFSFDLW0 detected X'FFnn' in the RDW of the rebuilt OLDS block and the block was not the last block in the WADS, DFSFDLW0 returns a return code of X'12' and a reason code of X'0E'.
Reg15=X'0F' RLWABC=X'0F' RLWPBA - OLDS buffer address	LTA3550	When DFSFDLW0 detected an invalid log sequence in the rebuilt OLDS block, it returns return code of X'12' and a reason code of X'0F'.
Reg15=X'10' RLWABC=X'10' DECBA - DECBA address	LTJ220 LTJ230	DFSFDLW0 issues a BSAM READ for the WADS DCB while rebuilding an OLDS block. DECBA is tested to determine if READ was successful. If READ was not successful, DFSFDLW0 returns a return code of X'12' and a reason code of X'10'.
Reg15=X'11' RLWABC=X'11' DECBA - DECBA address	LTJ20R	DFSFDLW0 issues a BSAM READ for the WADS DCB while rebuilding an OLDS block. If an end-of-data condition was detected, DFSFDLW0 returns a return code of X'12' and a reason code of X'11'.
Reg15=X'12' RLWABC=X'12' WADSBA - WADS buffer address	LTN210R LTN320R	When DFSFDLW0 detected an invalid segment-ID (segment-ID is larger than number of segments per block - 1), DFSFDLW0 returns a return code of X'12' and a reason code of X'12'.
Reg15=X'13' RLWABC=X'13' WADSBA - WADS buffer address	LTN230R LTN3320	When DFSFDLW0 detected a segment written noncontiguously in the track, DFSFDLW0 returns a return code of X'12' and a reason code of X'13'.
Reg15=X'14' RLWABC=X'14' RLWPBSN - OLDS block sequence number	READTG	When DFSFDLW0 detected an invalid block sequence number in the input parameter, DFSFDLW0 returns a return code of X'12' and a reason code of X'14'.
EXITSAVE - REG save area	EXIT	When DFSFDLW0 detected an internal logic error, DFSFDLW0 returns a return code of X'12' and a reason code of X'14'.

**System action:** The Log Recovery utility terminates abnormally.

**Problem determination:** 1, 2, 3, 5, 8, 11, 17a, 17g, 35

**Source:** IMS abends

**Module:** DFSULG20

**Related information:**

Chapter 2, "IMS problem determination," on page 13

## 3271

**Explanation:** The Log Recovery utility could not correctly close (terminate) the OLDS. An end-of-file was encountered in the input OLDS, but the closing point had not yet been found. Refer to the explanation of message DFS3271I for additional detail.

**Analysis:** This is a standard abend that can be issued by DFSULG10. When this abend is issued, the program status word (PSW) at entry-to-abend points to the instruction from which the abend (SVC 13) is issued.

Register 12 in the abend SVRB register is the base register for DFSULG10.

**Prior OLDS was specified**

Key	Label	Description
Reg15=X'32'	EODADUMP	An end-of-data condition was encountered while trying to read the last block written to the prior OLDS. The last block sequence number written to the prior OLDS is gotten from DBRC and stored in field POLDMBSN.

**Next OLDS was specified**

Key	Label	Description
Reg11-address of last read block	DUPE0350	When DFSULG20 read the log blocks in the input OLDS, DFSULG20 encountered an end-of-data condition, but did not find the closing point. The closing point is the log block whose block sequence number is 1 less than the first block sequence number in the next OLDS. Label CBSEQ contains the first block sequence number in the next OLDS.

**System action:** Log Recovery utility terminates abnormally.

**Problem determination:** 1, 2, 3, 5, 8, 11, 17a, 17g, 35

**Source:** IMS abends

**Module:** DFSULG10

**Related information:**

 DFS3271I

Chapter 2, "IMS problem determination," on page 13

## 3272

**Explanation:** An internal processing error occurred in the Log Recovery utility. DFSULG30 has detected an empty PSTTBL entry. Message DFS3272E accompanies this abend.

**Analysis:** This is a standard abend issued from DFSULG30. It is issued due to an internal processing error. The corresponding PSTTBL entry is empty. A DFS3272E message is issued and the abend occurs with a dump.

Key	Label	Description
Reg5=A(Pointer to the PSTTBL)	PSTPOINT	An empty PSTTBL entry has been detected.
Reg6=A(PSTTBL)		
Reg4=PST#		

**System action:** The Log Recovery utility terminates processing.

**Programmer response:** Contact the IBM Support Center for assistance.

**Source:** IMS abends

**Module:** DFSULG30

**Related information:**

## 3274

**Explanation:** The Log Archive utility or Recovery utility received an error return code from a DBRC exit. Message DFS3274I is issued when this error occurs and shows the type of DBRC exit that failed and the DBRC return code. The type of DBRC exit invoked is indicated by one of the following values:

ARCHIVE INIT  
 ARCHIVE COMPLETE  
 RECOVERY INIT  
 RECOVERY OPEN  
 RECOVERY EOJ  
 RECOVERY CLOSE

See the DBRC request return code information for an explanation of any DBRC return codes received.

An additional error message might be issued to SYSPRINT by the DBRC exit.

**Analysis:** This is a standard abend that can be issued by the Log Archive utility, DFSUARC0, or the Log Recovery utility, DFSULG10 and DFSULG20. When this abend is issued, the program status word (PSW) at entry-to-abend points to the instruction from which the abend (SVC 13) is issued.

Before issuing this abend, both utilities issue error message DFS3274I, which indicates the exit routine name and the return code. Register 12 in the abend SVRB registers is the base register for the issued module and register 15 contains a return code from DBRC.

**ARCHIVE INIT exit routine (input is OLDS)**

Key	Label	Description
Reg15=X'44'	DINI010 (DFSUARC0)	An internal DBRC failure was encountered.
Reg15=X'48'	DINI010 (DFSUARC0)	A required parameter was not specified on the exit routine invocation.

**ARCHIVE INIT exit routine (input is batch SLDS)**

Key	Label	Description
Reg15=X'44'	ILOPEN (DFSUARP0)	An internal DBRC failure was encountered.
Reg15=X'48'	ILOPEN (DFSUARP0)	A required parameter was not specified on the exit routine invocation.

**ARCHIVE ARCOMPL exit routine**

Key	Label	Description
Reg15= X'0C'	ARCP10 (DFSUARC0)	You specified an invalid volume stop time.
Reg15= X'44'	ARCP10 (DFSUARC0)	An internal DBRC failure was encountered.
Reg15= X'48'	ARCP10 (DFSUARC0)	A required parameter was not specified on the exit routine invocation.

**ARCHIVE EOJ exit routine**

Key	Label	Description
Reg15= X'44'	ENDJ10 (DFSUARC0)	An internal DBRC failure was encountered.
Reg15= X'48'	ENDJ10 (DFSUARC0)	A required parameter was not specified on the exit routine invocation.



**LOG RECOVERY INIT exit routine (input log is OLDS)**

Key	Label	Description
Reg15= X'0C'	DINI010 (DFSULG10)	A DSPINIT macro failure or DSPRCLOS macro failure occurred.
Reg15= X'44'	DINI010 (DFSULG10)	An internal DBRC failure was encountered.
Reg15= X'48'	DINI010 (DFSULG10)	A required parameter was not specified on the exit routine invocation.

**LOG RECOVERY OPEN exit routine (input log is OLDS)**

Key	Label	Description
Reg4= A(DBRCRMI1) Reg15=X'04'	DPRI020 (DFSULG10)	An internal DBRC failure was encountered at the primary input log OPEN exit routine. Register 3 indicates an address of DCB.
Reg4=A(DBRCRMI2) Reg15= X'04'	DSEC020 (DFSULG10)	An internal DBRC failure was encountered at the secondary input log OPEN exit routine. Register 3 indicates an address of DCB.
Reg4= A(DBRCRMI1) Reg15=X'44'	DPRI020 (DFSULG10)	An internal DBRC failure was encountered at the primary input log OPEN exit routine. Register 3 indicates an address of DCB.
Reg4=A(DBRCRMI2) Reg15= X'44'	DSEC020 (DFSULG10)	An internal DBRC failure was encountered at the secondary input log OPEN exit routine. Register 3 indicates an address of DCB.
Reg4=A(DBRCRMO1) Reg15= X'44'	OPOL020 (DFSULG10)	An internal DBRC failure was encountered at the primary output log OPEN exit routine. Register 3 indicates an address of DCB.
Reg4=A(DBRCRMO2) Reg15= X'44'	OPOL020 (DFSULG10)	An internal DBRC failure was encountered at the secondary output log OPEN exit routine. Register 3 indicates an address of DCB.
Reg4=A(DBRCRMN1) Reg15= X'44'	NOLD030 (DFSULG10)	An internal DBRC failure was encountered at the primary next OLDS OPEN exit routine. Register 3 indicates an address of DCB.
Reg4=A(DBRCRMN2) Reg15= X'44'	NOLD030 (DFSULG10)	An internal DBRC failure was encountered at the secondary next OLDS OPEN exit routine. Register 3 indicates an address of DCB.
Reg4=A(DBRCRMI1) Reg15= X'48'	DPRI020 (DFSULG10)	A required parameter was not specified at primary input log OPEN exit routine.
Reg4=A(DBRCRMI2) Reg15= X'48'	DSEC020 (DFSULG10)	A required parameter was not specified at secondary input log OPEN exit routine.
Reg4=A(DBRCRMO1) Reg15= X'48'	OPOL020 (DFSULG10)	A required parameter was not specified at primary output log OPEN exit routine.
Reg4=A(DBRCRMO2) Reg15= X'48'	OPOL020 (DFSULG10)	A required parameter was not specified at secondary output log OPEN exit routine.
Reg4=A(DBRCRMN1) Reg15= X'48'	NOLD030 (DFSULG10)	A required parameter was not specified at primary next OLDS OPEN exit routine.
Reg4=A(DBRCRMN2) Reg15= X'48'	NOLD030 (DFSULG10)	A required parameter was not specified at secondary next OLDS OPEN exit routine.

**LOG RECOVERY CLOSE exit routine (input log is OLDS)**

Key	Label	Description
Reg4=A(DBRCRMI1) Reg15=X'44'	ENDO010 (DFSULG10)	An internal DBRC failure was encountered at primary input log CLOSE exit routine. Register 3 indicates an address of DCB.
Reg4=A(DBRCRMI2) Reg15=X'44'	ENDO010 (DFSULG10)	An internal DBRC failure was encountered at secondary input log CLOSE exit routine. Register 3 indicates an address of DCB.
Reg4=A(DBRCRMO1) Reg15=X'44'	ENDO010 (DFSULG10)	An internal DBRC failure was encountered at primary output log CLOSE exit routine. Register 3 indicates an address of DCB.
Reg4=A(DBRCRMO2) Reg15=X'44'	ENDO010 (DFSULG10)	An internal DBRC failure was encountered at secondary output log CLOSE exit routine. Register 3 indicates an address of DCB.
Reg4=A(DBRCRMI1) Reg15=X'48'	ENDO010 (DFSULG10)	A required parameter was not specified at primary input log CLOSE exit routine.
Reg4=A(DBRCRMI2) Reg15=X'48'	ENDO010 (DFSULG10)	A required parameter was not specified at secondary input log CLOSE exit routine.
Reg4=A(DBRCRMO1) Reg15=X'48'	ENDO010 (DFSULG10)	A required parameter was not specified at primary output log CLOSE exit routine.
Reg4=A(DBRCRMO2) Reg15=X'48'	ENDO010 (DFSULG10)	A required parameter was not specified at secondary output log CLOSE exit routine.

#### LOG RECOVERY EOJ exit routine (input log is OLDS)

Key	Label	Description
Reg8=A(DBRC0300) Reg15=X'44'	ENDO010 (DFSULG10)	An internal DBRC failure was encountered at normal EOJ exit routine.
Reg8=A(DBRC0400) Reg15=X'44'	EXIT (DFSULG10)	An internal DBRC failure was encountered at abend EOJ exit routine.
Reg8=A(DBRC0300) Reg15=X'48'	ENDO010 (DFSULG10)	A required parameter was not specified at normal EOJ exit routine.
Reg8=A(DBRC0400) Reg15=X'48'	EXIT (DFSULG10)	A required parameter was not specified at abend EOJ exit routine.

#### Input LOG is SLDS

Key	Label	Description
Reg15=X'44'	DINI010 (DFSULG10)	An internal DBRC failure was encountered
Reg15=X'48'	DINI010 (DFSULG10)	A required parameter was not specified on the exit routine invocation.


**System action:** The utility terminates abnormally.

**Problem determination:** 1, 2, 3, 8, 11, 35

**Source:** IMS abends

**Module:** DFSUARC0, DFSULG10, DFSULG20

**Related reference:**

 DBRC request return codes (Messages and Codes)

**Related information:**

DFS3274I

Chapter 2, "IMS problem determination," on page 13

3275

**Explanation:** A logic error occurred in either IMS or DBRC. In all but two cases, message DFS030I accompanies the abend and contains a reason code describing the failure. For the other two cases, message DFS030I is not issued, but the two low-order bytes of register 15 contain a reason code in character format. These are:

**Code Meaning**

- X'03'** A return code 12 was received from DBRC during the database authorization process. A return code of X'0C', which indicated that the subsystem was not signed on to DBRC, was received upon return. The condition should not have occurred unless there was an error in the subsystem initialization or in DBRC. Message DFS030I will also be issued and will contain reason code 3 when the abend is issued. (DBFDBAU0, DFSDBAU0)
- X'04'** A database authorization call was made to DBRC. A correct area lock scope initiated by the SHARELVL in an encoded state was returned from DBRC and the internal lock scope table, which is located in DBFDBAU0. This condition should not have occurred unless there was an error in the DBRC or in the DBFDBAU0. (DBFDBAU0)
- X'05'** A signoff call was made to DBRC while IMS was terminating. The return code indicated that either there was no subsystem entry found, that an internal error was encountered during the unauthorized process, that the subsystem entry was found but terminated abnormally, or that the recovery processing had been started. See message DFS030I. (DFSPCCC0)
- X'0A'** The database open call to DBRC was issued during DL/I open processing. The return code from DBRC indicated that the database being processed was not registered to DBRC. This situation should not occur since DBRC is not called during open processing unless authorization for the database was previously obtained from DBRC. Non-authorization is not the cause because both the open call to DBRC and DL/I open processing are issued only after access to the database is authorized. If the subsystem record did not exist in DBRC, check whether the SSID information was removed from the database record in DBRC when the subsystem was deleted. (DFSDDUI0, DBFMLOP0)
- X'0B'** The return code set by DBRC when the database open call was processed indicated that the subsystem was not registered to DBRC. This situation should not occur since DBRC is called during DL/I open processing only if the subsystem previously obtained authorization for the database, at which time it had to have been registered to DBRC. (DFSDDUI0, DBFMLOP0)
- X'0C'** The database open call to DBRC was issued during DL/I open processing. The return code from DBRC indicated that the subsystem did not have authorization for the database. (DFSDDUI0, DBFMLOP0)
- X'0D'** The database open call to DBRC was issued during DL/I open processing. The return code from DBRC can indicate that an internal DBRC error was detected. A return code of zero can indicate that the RECON Initialization Time (RIT) is zero. (DBFMLOP0)
- X'0F'** An end HSSP image copy call was made to DBRC during image copy termination. The return code set by DBRC indicated that the image copy record specified was not found. Since the begin HSSP image copy call should result in the creation of that image copy record, an error occurred. (DBFPIC0)
- X'10'** An end HSSP image copy call was made to DBRC during image copy termination. The return code set by DBRC indicated that the database or area was not registered in the RECON data set. Since HSSP image copy can only be processed against a registered database or area, an internal error occurred. (DBFPIC0)

When 3275 is issued from DFSDBDR0 there is an inconsistency between the DMB Directory in IMS (DDIR) and DBRC. An UNAUTHL call was made for a DB that was not authorized to DBRC. This can occur when a previous update to the RECONS indicated ABNORMAL termination of a subsystem that was still active. The system action is to terminate the subsystem.

**X'11'** An ADSC control block for the DEDB area did not have a corresponding entry in the ADS list returned by DBRC.

**X'14'** DBRC detected an inconsistency for one of the following situations:

**Within the RECONS:** Either the database or subsystem record indicates that the subsystem has not been authorized to the database; if it is authorized, both the database and subsystem records must indicate this condition.

**Between the RECONS and IMS:** IMS indicates that the database is authorized, but the RECONS indicate that it is not. This situation can occur if external commands, separate from IMS commands, are used to modify the RECONS.

**X'20'** An end HSSP image copy call was made to DBRC during image copy termination. The return code set by

## 3275

DBRC indicated that the end HSSP image copy call was already made for this image copy process. This abend is issued if this return code was received and the system is not currently being emergency restarted. (DBFPICT0)

- X'30'** An end HSSP image copy call was made to DBRC during image copy termination. The return code set by DBRC indicated that an internal DBRC error occurred. (DBFPICT0)
- X'40'** An end HSSP image copy call was made to DBRC during image copy termination. The return code set by DBRC indicated that an invalid parameter was found. (DBFPICT0)
- X'FF'** An invalid or unexpected return code was received from DBRC (DBFMLOP0).

The logic error might be caused by the invalid use of the DELETE.SUBSYS or CHANGE.SUBSYS commands; if this occurs, register 15 might contain one of the reason codes associated with message DFS030I.

**Analysis:** For DFSPCCC0:

Signoff to DBRC return code > 4.

Key	Label	Description
Reg15=3	AL340	An unexpected return code was received from DBRC upon database authorization request. Reg9 contained the return code. See the section on return codes for authorization calls in the DBRC request return codes information for further description. Reg15=3 indicates that the abend was issued by DFSDBAU0.
Reg15=6, 7, 8 or 9	UNAUERR	The abend was issued by module DFSDLOC0 after a return code of 0 was returned on the UNAUTH call to DBRC. Register 2 contains the PST address and register 8 has the DDIR address for the DMB for which the UNAUTH request was issued.
Reg15=A, B, C or D	VERABEND	DFSDDU10

**For DBFARD30, DBFARD40, DBFSTAP0, DBFHDEP0, DBFMLCL0:**

When DBRC returns non-zero return code for UNAUTH request, DFS0019I and DSP0230I are written. When the return code is greater than X'10', abend 3275 is issued. See the section on return codes for authorization calls in the DBRC request return codes information for further description.

**Possible cause:** A breakdown occurred in communication between IMS and DBRC. The status information might have been lost on the RECON data set.

**System action:** The IMS control region terminates abnormally.


**Programmer response:** This is an internal IMS system error. Contact the IBM Support Center for help in determining the problem.

**Problem determination:** 1, 4, 10, 35

**Source:** IMS abends

**Module:** DBFDDBAU0, DFSDBAU0, DFSDBDR0, DFSDLOC0, DFSPCCC0, DFSDDU10, DBFMLCL0, DBFMLOP0, DBFPICT0

**Related reference:**

 DBRC request return codes (Messages and Codes)

**Related information:**

 DFS030I

Chapter 2, "IMS problem determination," on page 13

---

**3276**

**Explanation:** While performing I/O processing, Fast Path detected the inflight I/O count (DMACWTCT) to be invalid. The inflight I/O count is already zero or negative prior to the decrement.

**Analysis:** This is a standard abend issued by:

- DBFFOR10: Fast Path Output Thread Router
- DBFVXOW0: VSO XES Output Processor
- DBFXCNX0: XES Complete Exit Routine
- DBFVXOE0: Shared VSO I/O Error Routine

The program status word (PSW) points to the instruction within the module from which abend (SVC 13) is issued. This error indicates an IMS internal logic error.

**System action:** IMS issues the abend, produces a dump, and terminates abnormally.

**Operator response:** Emergency restart IMS to resume processing.

**Problem determination:** 1, 2, 3, 4, 6, 35

**Source:** IMS abends

**Module:** DBFFOR10, DBFVXOE0, DBFVXOW0, DBFXCNX0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

**3287**

**Explanation:** The application program was terminated by the DL/I call analyzer because one of the databases referenced by the application program was stopped by a global command or had an I/O error.

**Analysis:** PSTCSFLG byte was set to a hexadecimal value of 'D7' by the IMS NOTIFY exit routine. This value was set in order to acquire the application programs that had access to the bad or stopped databases.

**Possible cause:** The master terminal operator entered a global command to stop the shared database or the database had an I/O error.

**System action:** The application program terminates abnormally.

**Programmer response:** Contact the master terminal operator or system programmer to recover the database, and re-execute the job.

**Problem determination:** 1, 4, 11

**Source:** IMS abends

**Module:** DFSDLA00

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

**3289**

**Explanation:** The command lock request failed. An attempt was made to notify other subsystems of the database backout failure. A message listing the database names that had backout failures is sent to all other sharing subsystems. Prior to sending the message, the command lock is acquired to serialize the commands across the subsystems.


**System action:** IMS terminates abnormally.

**Programmer response:** Check the IRLM LOCK request and other IRLM sharing subsystem failures. See the IRLM request return and reason codes information to determine the cause of the failure.

**Problem determination:** 1, 4, 10, 35

**Source:** IMS abends

**Related tasks:**

 IRLM return and reason codes

**Related information:**

Chapter 2, “IMS problem determination,” on page 13

**3290**

**Explanation:** During sync point processing, the lock request handler was called to release all locks held by the application program. The request to release these locks failed. This abend can occur for any application when an unlock request to the IRLM fails because of an internal IRLM problem or when the IRLM experiences an out-of-storage condition.

**Analysis:** This is a standard abend issued out of DFSFXC50. The address of the IRLM parameter can be located in the PSTIRLMA. Register 15 contains the return code and register 14 contains the reason code of the UNLOCK request.

**System action:** IMS terminates abnormally.


**Programmer response:** Correct the problem determined by the UNLOCK request return code. Refer to the IRLM request return and reason code information to determine the cause of the failure.

**Problem determination:** 1, 4, 10, 35

**Source:** IMS abends

**Module:** DFSFXC50

**Related tasks:**

 IRLM return and reason codes

**Related information:**

Chapter 2, “IMS problem determination,” on page 13

**3300**

**Explanation:** This abend can occur for any application when a lock request to the IRLM fails because of an internal IRLM problem or when the IRLM experiences an out-of-storage condition. It is also issued for batch applications (DLI or DBB region type) when the IRLM terminates abnormally.

A lock could not be granted by the IRLM for one of the following reasons:

1. The IRLM failed or was not available.
2. A system error occurred that prevented the IRLM from completing the lock request.
3. An out-of-storage condition was experienced by the IRLM on a GETMAIN request.

**Analysis:** 3300 is a pseudoabend issued from DFSLRH00, DBFLRH00, or DBFNOTM0. The lock trace entry for the unsuccessful lock request has the return code and feedback information passed from the IRLM.

The following return codes and feedback codes can be found in the trace.

**LOCK Request**

The following table shows the reason codes for return code X'08':

Reason code	Description
X'80'	System error
X'01'	Out-of-storage

RC=20 indicates that IRLM was not available.

**Possible cause:** Examine the return code and feedback to determine the cause. The out-of-storage condition can indicate that the IRLM region size was exceeded.

**System action:** The application program terminates abnormally.

**Programmer response:** If the IRLM terminates abnormally, run database backout and restart the job after the IRLM is reinitialized.

If this abend is caused by an abnormal termination other than IRLM, the IMS lock trace table contains the return and reason codes from the IRLM on the lock request. See the IRLM LOCK request information to determine the cause of the failure.

**Problem determination:** 1, 17, 34

**Source:** IMS abends

**Module:** DFSLRH00, DBFLRH00, DBFNOTM0

**Related tasks:**

 IRLM LOCK request

**Related information:**

Chapter 2, "IMS problem determination," on page 13





---

## Chapter 27. IMS abend codes 3301 - 4100

This topic identifies and explains the following abend codes issued by IMS. Each code is 4 decimal digits.

---

### 3301

**Explanation:** This pseudoabend is issued when the number of locks acquired on behalf of the application program has exceeded the maximum specified by the installation or application programmer. The maximum number of locks an application program can accumulate before committing is specified by the PSBGEN LOCKMAX= parameter or the LOCKMAX= override parameter in the DL/I or DBB batch, or dependent region JCL.

**System action:** The application program abends with user abend 3301.

**Programmer response:** You can override the PSBGEN LOCKMAX= specification at execution time by specifying the LOCKMAX= parameter in the batch or dependent region JCL. An override parameter of LOCKMAX=0 will turn off all locking limitations. The LOCKMAX value for the PSBGEN and override cases is in units of 1000. Each time the LOCKMAX value is increased by 10, the application is allowed to acquire 10,000 more locks.

Each application program is allowed to have its own limit on the maximum number of locks it can acquire, that is each application program can be individually constrained to a specific amount of virtual storage resources to be used for locking.

An errant application program that is not constrained can consume all locking storage and cause significant disruption to other application programs using the same locking services.

**Source:** IMS abends

---

### 3302

**Explanation:** The LOCK request handler received an unexpected return code from the IRLM or from the program isolation (PI) lock manager if the IRLM is not being used. An invalid condition was detected by the IRLM on a LOCK, UNLOCK, or NOTIFY request. This indicates an internal IMS error.

**Analysis:** This is a pseudoabend issued from DFSLRH00, DBFLRH00, and DBFNOTM0.

The lock trace entry for DFSLRH00 has the return code and feedback information from the lock manager. If the lock trace is on (LOCK=ON), you must extract the X'67FF' log records using DFSERA10. From these records, locate the reason code at PST + X'344' and the return code at PST + X'347'.

Refer to DFSLR MACRO (DL/I LOCK REQUEST) to map these codes.

The following return codes and feedback codes can be found in the trace.

#### LOCK Request

RC=X'08'

Code	Meaning
------	---------

X'40'	The LOCK request specified MODE=COND and the request would have to wait in order to obtain the requested state.
-------	---

X'10'	An unconditional request that did not wait for an incompatible lock.
-------	--

RC=X'0C'

Code	Meaning
------	---------

X'80'	The specified parent token was not owned by the work unit.
-------	--

RC=X'10'

Code	Meaning
------	---------

X'40'	Invalid class
-------	---------------

X'20'	Invalid state
-------	---------------

## 3303

X'10' Invalid parent token  
X'08' Invalid scope  
X'04' Invalid token  
X'02' Invalid resource name length

### UNLOCK Request

RC=X'0C'

Code	Meaning
------	---------

X'40'	Lock not held by work unit
X'20'	No lock exists for resource name specified
X'08'	Lock is not held in the specified state
X'04'	Lock is not held in the specified class

RC=X'10'

Code	Meaning
------	---------

X'40'	Invalid class
X'20'	Invalid state
X'04'	Invalid token
X'02'	Invalid resource name length

*Possible cause:* An internal logic error has occurred.

**System action:** The application program terminates abnormally.

**Programmer response:** No action is required.

**Problem determination:** 1, 17, 30, 34

**Source:** IMS abends

**Module:** DFSLRH00, DBFLRH00, DBFNOTM0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

## 3303

**Explanation:** One of the following conditions occurred:

1. A lock request to the IRLM failed because of a conflict with the retained lock for a failed IMS subsystem. Message DFS3304I is issued with this abend.
2. The IRLM entered IRLM FAILED or COMM FAILED state, or abnormally terminated. IRLM could also have failed if some or all databases could not be authorized or verified in batch mode.
3. An IMS system terminated all applications when the IRLM terminated abnormally.
4. The application tried to use a database or HALDB partition that was not available for access or update. In addition, it did not issue an INIT STATUS GROUPA call that indicates that the application wanted to receive database status information in DB PCB instead of a U3303 abend. Message DFS3303I is issued with this condition.
5. A "read any" request was rejected because an I/O Toleration extended error queue element (EEQE) exists for the control interval (CI) specified in message DFS2524I. For this condition, message DFS2524I is issued with abend 3303.
6. A /START, /STOP, or /DBRECOVERY database command was issued against a full-function database that is referenced by an active PSB in a Fast Path IFP region. The regions end abnormally and reschedule with access to the database changed (removed or added).
7. A DB2 subsystem that was connected to IMS is shutting down, and IFPs are waiting in the load balancing group control block (BALG) with access to the DB2 resource. IMS handles this situation in the same manner as it would if a /DBR command were issued.
8. An application attempted to access data that is part of a recoverable indoubt structure (RIS) established in a CCTL - IMS connection. Message DFS0693I displays the PSB and the recovery token.
9. A ROLS call was issued:

- For a DB PCB that received one of the data-unavailable status codes
  - For an I/O PCB without an I/O area or token in the call
  - By an application program that received incorrect input data
10. A coupling facility connection failure or structure failure occurred during a read-and-register operation to the coupling facility in a data-sharing environment.
  11. A data-sharing DL/I batch job was running when a structure rebuild occurred. The batch job is then terminated.
  12. The application tried to access a specific version of a full-function or Fast Path database, but either the IMS systems could not retrieve the requested version or database versioning is not enabled in the IMS system.

**Analysis:** This is a problem with either an IRLM failure or unavailable data. Determine why IRLM, the database, or data blocks of the database, are unavailable.

Possible causes are:

- An IRLM failure
- Failure of a data-sharing partner IMS
- Full-function database is offline
- Application has not requested availability status codes (INIT STATUS GROUP A)
- In-doubt data (as a result of a CCTL or RRS issue)

#### For DBFSYN10:

IRLM failed during commit of Fast Path resources.

*Possible cause:* The IRLM has failed.

#### For DBFMBED0:

A read against a shared VSO DEDB area failed, but data within the CI had already been modified and an EEQE exists against this CI.

*Possible cause:* A read error occurred against the DEDB area.

#### For DBFLRH00:

1. A DBCTL thread terminated during the lock request.
2. A retained lock for a failed subsystem exists for the lock being requested.
3. A lock request was rejected.
4. A lock was rejected with return code 20.

To determine the cause of failure for 3 or 4 in this list, see the description of the LOCK request in the information about IRLM request return and reason codes. For failure 3, a DFS3304 message will be issued with the abend.

*Possible cause:* Can be any of the following:

- DBCTL terminated unexpectedly.
- A retained lock exists for the lock being requested.
- IRLM rejected the lock request.
- IRLM has abnormally terminated.

#### System action:

The application program terminates abnormally. The message being processed is placed on the SUSPEND QUEUE unless the SERIAL=YES option was specified on the TRANSACT macro or was involved in a protected conversation with RRS/MVS. For SERIAL=YES, the message is put back on the queue as the next message to be processed for this transaction, and the transaction is stopped. For the protected conversation, IMS discards the input message but does not stop the transaction.

If the abend is caused by an unavailable data condition (DFS3303I was issued) and it occurs 10 times, IMS takes one of the following actions:

- In a DB/DC environment, IMS stops the transaction with a USTOPPED condition.
- In a DBCTL environment, IMS stops the PSB, preventing further scheduling of the application program.

**Important:** For IFP regions, a GU call to the I/O PCB must be issued before a DL/I call against a DB PCB. If a DL/I call is issued against a DB PCB before the GU call to the I/O PCB, the IFP region might loop during rescheduling of the IFP. The application must check the database status or issue an INIT STATUS GROUPA call. If a bad status code is received from the call, the program must exit. If the application does not check the status code and exit, it will continue to receive transactions from the BALG. Each transaction will trigger the U3303 pseudoabends on subsequent messages, which results in a failure or rescheduling loop in the IFP region.

**Operator response:** Use the following list to determine the appropriate action for each condition that caused the abend. The numbers in the list correspond to the numbers in the Explanation section.

#### Condition

##### Action

- 1 Run the database backout or emergency restart for the failed subsystem. The database backout or emergency restart automatically notifies this IMS to release the transactions on the suspend queue. The DFS3304I message that precedes abend 3303 identifies the failed IMS.
- 2 The IRLM can be restarted or the communication between the IRLM reestablished, or both. The databases can be restarted by using the /START command, and the suspended transactions can be released by using the /DEQUEUE SUSPEND command. Any BMPs might also need to be restarted.
 

**Note:** For IRLM, PSTCSHAB indicates an IRLM problem and is set to the value ON in module DFSSDLB0 or DBFNOTM0, and is detected in modules DBFHGU10, DBFIRC10, DBFLMGR0, and DBFPCC30, where the value of field PSTABTRM is set.
- 3 The IRLM can be restarted and the MVS MODIFY command can be used to specify RECONNECT to IMS. The databases can be restarted by using the /START command. Any BMPs might also need to be restarted.
- 4 See message DFS3303I to determine the reason why the database was unavailable. Follow the directions in DFS3303I to correct the problem.
- 5 The database can be unlocked by using the /UNLOCK command on the alternate system. This command purges the EEQEs. The suspended transactions can be released by using the /DEQUEUE SUSPEND command on the data-sharing system. Any BMPs might also need to be restarted.
- 6 Abend 3303 forces a rescheduling so that the status change of the full-function database can be picked up.
- 7 The IFP regions no longer have access to the external subsystem until it is restarted.
- 8 Issue the /DISPLAY CCTL command to find the indoubt thread. Then, issue the /CHA CCTL command to remove the RIS.
- 9 No action is necessary.
- 10 Correct the problem with the coupling facility or cache structure.
- 11 Correct the problem with the coupling facility or cache structure, and perform any required batch backout.
- 12 Message DFS3303I indicates the reason why the version was unavailable.

**Programmer response:** Use the following list to determine the appropriate action for each condition that caused the abend. The numbers in the list correspond to the numbers in the Explanation section.

#### Condition

##### Action

- 4 Add an INIT STATUS GROUPA call to the application. Adding this call causes the DL/I call to receive status code BA instead of U3303. The application can then take an appropriate action. For a description of the INIT call, see *IMS Version 13 Application Programming APIs*. For formats and parameters for the INIT call, see *IMS Version 13 Database Administration*.
- 9 Determine the cause of the ROLS and correct, if necessary.
- 12 You can also issue the INIT call to receive a status code instead of abend 3303 for problems retrieving a specific version of a full-function database.

**Problem determination:** 1, 17


**Source:** IMS abends

**Module:** DBFLRH00, DBFDBAC0, DBFHGU10, DBFIRC10, DBFLRH00, DBFMBED0, DBFSYN10, DFSDVSM0, DFSDBH20, DFSDBLM0, DFSDLA00, DFSLRH00, DFXES0

**Related tasks:**

 IRLM LOCK request

**Related reference:**

 INIT call (Application Programming APIs)

**Related information:**

 DFS3303I (Messages and Codes)

Chapter 2, “IMS problem determination,” on page 13

### 3304

**Explanation:** This abend is issued when a database read error occurs and data has been modified on this call. This prevents potential database integrity problems.

**System action:** The application program terminates abnormally. IMS processing continues.

**Programmer response:** Message DFS2571I or DFS3712A, displayed on the master terminal, gives details about the read error. Before rerunning the application, it might be necessary to perform database recovery for the DEDB area specified.

**Source:** IMS abends

### 3305

**Explanation:** One of the following conditions occurred:

- IRLM drove the IMS status exit after being informed that the active IMS system was being taken over by the alternate system.
  - IRLM drove the IMS status exit informing IMS that the IRLM terminated abnormally while the alternate IMS system was either in tracking mode or in takeover processing, but before takeover completed.
  - IRLM drove the IMS status exit in a Fast Database Recovery region.
  - IRLM drove the IMS status exit on the active IMS and informed IMS that the IRLM terminated.
- IRLMFAIL=ABEND was coded in the DFSFDRxx PROCLIB member.

**Analysis:** 3305 is a standard abend issued by DFSSTAX0 or DFSFDRC0.

**System action:** IMS terminates abnormally.

**Programmer response:** Take the appropriate action to regain XRF capability:

- For the first condition, bring up the former active IMS system as the alternate after IMS terminates.
- For the second condition, bring up the IRLM and restart the alternate IMS system.
- For the third condition, restart the Fast Database Recovery region after connectivity is reestablished to the IRLM lock structure.
- For the fourth condition, restart IRLM, and then restart IMS (after FDBR recovery completes, if applicable).

**Source:** IMS abends

**Module:** DFSFDRC0, DFSHTRM0, DFSSTAX0

### 3306

**Explanation:** This abend is issued to prevent non-APF authorized programs from gaining PSW (program status word) supervisor state, which violates z/OS integrity. The IRPM pointer is invalid. The key 8 IRPM does not point to its paired key 7 IRPM. The paired key 7/key 8 IRPMs are formatted during SVC initialization to point to each other.

This abend can occur when faulty programs cause storage overlays, though there are attempts to use the PC number of IMS. The PSTIRLMA pointer in the PST pointing to IRLM RLPL storage within the IRPM block is overlaid, or the key 8 IRPM storage itself is overlaid.

**System action:** IMS terminates abnormally.

**Programmer response:** Check for an overlaid IRPM block or an application that is unauthorized to use the PC number of IMS.

## 3307 • 3310

**Problem determination:** 1, 4, 6, 13

**Source:** IMS abends

**Module:** DFSLMGR0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

### 3307

**Explanation:** When an IRLM out-of-storage condition occurs in the lock table during coupling facility use, an application terminates abnormally and issues this abend.

**System action:** The application program terminates abnormally. The IMS control region might terminate with abend 0113.

**Programmer response:** Consider increasing the size of the lock table in the coupling facility.

**Problem determination:** 1, 17, 30, 34

**Source:** IMS abends

**Related information:**

IMS abend 0113

Chapter 2, "IMS problem determination," on page 13

---

### 3308

**Explanation:** The system terminated for one of the following reasons:

- A CSL request error was encountered.
- The operator replied CANCEL to a DFS4454A message.

**System action:** IMS terminates abnormally.

**Programmer response:** When the termination is the result of a CSL request error, message DFS3308E is issued before this abend. Refer to this message to determine the reason for the termination.

When the termination is the result of an operator replying CANCEL to a DFS4454A message, refer to this message to determine the reason for this termination.

**Source:** IMS abends

**Related information:**

 [DFS3308E \(Messages and Codes\)](#)

 [DFS4454A \(Messages and Codes\)](#)

---

### 3310

**Explanation:** A lock request timed out by IRLM; a LOCKTIME was specified in either a DFSVSMxx PROCLIB member or DFSVSAMP DD. The number of seconds the lock is held before being timed out is specified by LOCKTIME at IMS initialization. You can change the timeout value by issuing the command `MODIFY irlmproc,SET,TIMEOUT=nn`.

| **System action:** The application program terminates abnormally. No memory dump is created, the transaction and  
| the program are not stopped, and the transaction input message is always discarded unless you are using the  
| Non-Discardable Messages user exit (NDMX) to tell IMS what to do with the input message associated with the  
| abend.

**Programmer response:** Investigate why the lock request was timed out. RMF™ record 79.15 indicates what task was blocking this request from being granted. Resource Measurement Facility™ (RMF) can also be used online to format the IRLM LONG LOCK information. See Enabling the IRLM lock timeout function (System Definition) for information about how this parameter is set.

Message DFS2291I also contains the blocker and waiter information and can be useful in the analysis of the cause of the timeout.

**Source:** IMS abends

**Related information:**

 DFS2291I (Messages and Codes)

Chapter 2, "IMS problem determination," on page 13

3312

**Explanation:** An error occurred during DBRC processing for an internal system error. Register 15 contains one of the following reason codes:

**Code    Meaning**

X'04'    DBRC signon had a nonzero return code.  
 X'08'    BLDL for DSPCRTR0 failed.  
 X'0C'    First call for DBRC initialization (INIT-0) had a nonzero return code.  
 X'10'    Second call for DBRC initialization (INIT-1) had a nonzero return code.  
 X'14'    DBRC signoff had a nonzero return code.

**Analysis:** This is a standard abend issued by DFSUDMP0.

**Code    Meaning**

X'04'    Register 3 contains the address of the subsystem name. Register 5 contains the DBRC return code.  
 X'08'    Register 5 contains the BLDL return code.  
 X'0C'    Register 5 contains the DBRC return code.  
 X'10'    Register 5 contains the DBRC return code.  
 X'14'    Register 3 contains the address of the subsystem name. Register 5 contains the DBRC return code.

*Possible cause:*

**Code    Meaning**

X'04'    An invalid subsystem name specification. Consult the appropriate DBRC documentation.  
 X'08'    The required module is not in the correct library or the JOB/STEPLIB DD statement does not specify the correct library.  
 X'0C'    A failure occurred in the DBRC processing. Consult the appropriate DBRC documentation.  
 X'10'    A failure occurred in the DBRC processing. Consult the appropriate DBRC documentation.  
 X'14'    A failure occurred in the DBRC processing. Consult the appropriate DBRC documentation.

**System action:** Processing terminates abnormally.

**System programmer response:** Take appropriate action based on the return code.

**Problem determination:** 1, 2, 3, 4, 8, 12

**Source:** IMS abends

**Module:** DFSUDMP0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

3313

**Explanation:** An IMS utility program encountered a severe error requiring analysis by an IBM Support Center representative. A return code is placed in register 15. The return codes have the following meanings:

**Code (hex)**

**Meaning**  
 00    The Log Merge utility (DFSLTMG0) encountered a deadlock situation.  
 04    The Log Merge utility (DFSLTMG0) failed to reach EOF on one or more of its input files.

**System action:** The job is terminated with a dump.

**Programmer response:** Rerun the job with a SYSUDUMP DD statement. Retain all input and output data sets. Contact your IBM Support Center for assistance.

**Source:** IMS abends

## 3314

**Explanation:** An unrecoverable error was detected by DFSPCC40 during processing because:

- Changed data exit routines or changed data logging were defined for the database.
- The changed data exit routine returned an abend return code.

The reason code for the abend on the z/OS message and in register 15 contains the following reasons:

**1st Byte:**

Identifies the module that detected the error.

Code	Meaning
04	DFSPCC40.

**2nd Byte:**

Identifies the routine that detected the error.

Code	Meaning
00	Set up.
04	CAPD chain loop.
08	CAPD routine.
12	Build XPCB control blocks.
16	Build XSDB control blocks.
20	Invoke user exit.
24	Restore exit interface and clear exit storage.
28	Load exit.
32	Get storage.
36	Free storage.
40	Delete exit.
44	Send message.

**3rd Byte:**

Contains the return code.

Code	Meaning
04	The user exit defined for the segment was not found. Message CSVOO3I contains the load module name.
08	Storage obtain or release error
12	Exit LOAD error.
16	Exit DELETE error.
20	Exit return code requests an abend.
24	Invalid return code from the exit routine. The return code is less than or equal to 20 or is not a multiple of 4.

**4th Byte:**

Contains the reason code, which is based on the return code in Byte 3.

**Return code**

Return code	Reason code
04	00—No additional qualification.
08	The low byte of the return code was passed back by MVS GETMAIN or FREEMAIN (see MVS GETMAIN/FREEMAIN return codes).
12	00—No additional qualification.
16	The return code returned was passed back by z/OS on a module request DELETE (see MVS DELETE return codes).
20	Any value. Value is user-defined by user exit.
24	Any value that is not a multiple of 4 or greater than 20.

**System action:** The application program terminates abnormally.

**Problem determination:** For return code 4, EXIT NOT FOUND, ensure that the exit load module is available. For return code 20 or exit return code 24, see the exit routine listing.

**Source:** IMS abends

**Module:** DFSPCC40



## 3315

**Explanation:** The Hardware Data Compression Dictionary utility (DFSZLDU0) detected an error while attempting to build or validate the dictionary.

**Analysis:** The program status word (PSW) points to the instruction within the module from which the abend (SVC13) is issued. Register 15 contains the reason code.

Key	Label	Description
Reg15=1		Unable to get storage in the dictionary initialization module, or an internal error occurred in that module.
Reg15=2		Invalid parameters were passed to the pre-dictionary builder module.
Reg15=3		Invalid parameters were passed to the dictionary build module.
Reg15=4		Invalid parameters were passed to the dictionary validation module, or a bad dictionary has been created.
Reg15=5		An open or close error has been detected on a data set allocated to ddname HDCIN, HDCDIT, HDCDCTL, HDCDOUT, or SYSPRINT. See <i>z/OS DFSMS Macro Instructions for Data Sets</i> .
Reg15=6		A logic error has been detected in the statistics/data integrity module during data compression or expansion.

**System action:** The job terminates abnormally.

**Problem determination:** 1, 3, 4, 5, 8, 11, 12, 15, 27

**Source:** IMS abends

**Module:** DFSZLDU0

**Related reference:**

➤ z/OS: OPEN--Connect Program and Data (BDAM, BISAM interface to VSAM, BPAM, BSAM, QISAM interface to VSAM, and QSAM)

➤ z/OS: CLOSE--Disconnect Program and Data (BDAM, BISAM, BPAM, BSAM, QISAM, and QSAM)

**Related information:**

Chapter 2, "IMS problem determination," on page 13

## 3325

**Explanation:** The Hardware Data Compression exit routine detected an error while attempting to perform compression or expansion services.

**Analysis:** The program status word (PSW) points to the instruction within the module from which abend (SVC 13) is issued. Register 14 contains the reason code for the error. This reason code corresponds to the label name at the failing location. Register 10 contains the address of the DMBCPAC (DBFCMPC if FP) control block. Register 9 contains the address of the routine work area.

Key	Label	Description
Reg14=D3C4E701 Reg6=function	D3C4E701	Invalid function code.
Reg14=D3C4E702 Reg6=length plus offset of sequence field	D3C4E702	The segment sequence field is not completely within the segment.
Reg14=D3C4E703 Reg6=input length	D3C4E703	Input length is negative.
Reg14=D3C4E704	D3C4E704	INIT was not specified in the COMPRTN= parameter of the SEGM statement.
Reg14=D3C4E705	DEC4E705	There was no dictionary address within the DFSZLDX0 CSECT. Examine the compression/expansion link step.

Key	Label	Description
Reg14=D3C4E706 Reg2=dictionary address	D3C4E706	The HDC Dictionary is not on a page boundary. The compression/expansion exit routine did not specify 'page DFSZHDCCD' on its linkage step.
Reg14=D3C4E707	D3C4E707	Invalid compression/expansion routine. Eyecatcher was not DFSZHDCCD. Examine the compression/expansion link step.
Reg14=D3C4E708 Reg3=CVT address	D3C4E708	Hardware Data Compression services are unavailable.
Reg14=D3C4E709 Reg15=return code from GETMAIN	D3C4E709	A GETMAIN for the z/OS compression simulation routine work area failed (full-function database). Increase storage size.
Reg14=D3C4E70A Reg15=return code from GETMAIN	D3C4E70A	A GETMAIN for the z/OS compression simulation routine work area failed (fast path DEDB call). Increase storage size.
Reg14=D3C4E70B	D3C4E70B	During a compression request, the input length of the variable length segment is less than 2 bytes. Correct the segment data.
Reg14=D3C4E70C	D3C4E70C	During an expansion request, the input length of the compressed segment is less than 2 bytes.
Reg14=D3C4E70D	D3C4E70D	During an expansion request, the hardware instruction found that the end of the first operand was reached but the end of the second operand was not reached.
Reg14=D3C4E70E Reg15=return code from expansion simulator	D3C4E70E	During an expansion request, a nonzero return code was returned by the z/OS expansion simulator.
Reg14=D3C4E70F	D3C4E70F	At database close time, INIT was not specified on the COMPRTN= parameter of the SEGM statement.
Reg14=D3C4E711	D3C4E711	A Function Recovery Routine (FRR) has intercepted a Data Exception interrupt during compression or expansion of a segment. Examine any recent changes to the database or the dictionary to determine the cause of the exception, such as compressing a database that was not expanded during unload, or using a dictionary that was created for a different database.

**System action:** IMS terminates abnormally.

**System programmer response:** Register 14 contains the reason code for the error. Also see *IMS Hardware Data Compression Extended for z/OS, User's Guide (SC27-1005)*.

**Source:** IMS abends

**Module:** DFSZLDX0

### 3368

**Explanation:** An error occurred during IMS initialization when IMS was trying to perform one of the following actions:

- Trying to obtain storage for automatic import or export processing
- Trying to read the resource definitions from the resource definition data set (RDDS) or IMSRSC repository for automatic import processing

**Analysis:** The module DFSRDD10 issues abend 3368 if there are not enough valid system resource definition data sets (RDDSs) to perform automatic import or export.

Abend 3368 is issued by DFSIP000 during automatic import processing if an error occurs trying to obtain a 64-bit storage buffer. DFSIP000 also issues this abend during automatic import processing if you specified IMPORTERR=ABORT in the DFSDFxxx IMS.PROCLIB member and an error occurs trying to access a system resource definition data set. Message DFS3368E or DFS3397E contains a return code indicating the cause of the failure.

Abend 3368 is issued by module DFSIP300 during autoimport from repository when there was an error reading the resource definitions from the IMSRSC repository. The DFS4401E identifies the error that occurred when reading the resource definitions.

On abend register 15 contains a subcode indicating the reason for the abend. Possible values are as follows:

Reason code in R15 (hex)	Description	Module
04	I/O error occurred during AUTOIMPORT from RDDS and RDDSERR=ABORT was specified.	DFSIP000
08	64-bit storage error occurred during AUTOIMPORT.	DFSIP000
0C	RDDSs are defined, but less than 2 are available.	DFSRDD10
10	I/O block storage failure occurred.	DFSRDD10
14	One or more RDDSs are not owned by this IMS.	DFSRDD10
20	Error during AUTOIMPORT from IMSRSC repository and REPOERR=ABORT was specified.	DFSIP300

**System action:** IMS terminates abnormally.

**Programmer response:** Use the information in message DFS3368E for the name of system resource definition data set and to identify the problem and correct it. Cold start IMS using the corrected resource definition data set.

Use information in the DFS4401E message to identify the error for the 3368 RC=32 abend. Correct the error and cold start IMS.

**Problem determination:** Determine the name of the system resource definition data set being processed from the DFS3368E message. Use the DFS3368E and DFS3397E messages to determine the cause of the failure.

Use information in the DFS4401E message to identify the cause of the abnormal termination.

**Source:** IMS abends

**Module:** DFSIP000, DFSRDD10, DFSIP300

**Related information:**

 [DFS3368E \(Messages and Codes\)](#)

 [DFS3397E \(Messages and Codes\)](#)

 [DFS4401E \(Messages and Codes\)](#)

3397

**Explanation:** During automatic import processing, IMS encountered an error in a resource or descriptor definition. IMPORTERR=ABORT was specified in the DFSDFxxx PROCLIB member. Refer to message DFS3395I for the name of the system resource definition data set being processed by automatic import. One or more copies of messages DFS3398E or DFS3423E or both should have been issued prior to the abend to indicate the type of error that occurred.

**System action:** IMS terminates abnormally.

**Programmer response:** Identify the problem using the accompanying DFS3398E and DFS3423E messages and correct it. Cold start IMS using the corrected definitions.

**Problem determination:** Determine the name of the resource definition data set being processed from the DFS3395I message. Use the DFS3398E and DFS3423E messages to determine the definitions in error.

**Source:** IMS abends

**Module:** DFSRIMP0

**Related information:**

 DFS3395I

 DFS3398E

 DFS3423E

### 3398

**Explanation:** This abend code only appears in the DFS554 message for Database Control (DBCTL) threads. It means that the thread was actively scheduled and then collapsed because the interface between DBCTL and the Coordinator Control (CCTL) subsystem terminated.

**System action:** No action results specifically from this abend.

**Programmer response:** The transaction did not complete. All changes were backed out. You can rerun the transaction after the DBCTL-CCTL connection is established again.

**Source:** IMS abends

### 3399

**Explanation:** Each time DBRC allocates an RSR-covered, shareable DEDB area or FF database, the counter SDCRSR\_CVCNT increases by 1. Each time DBRC deallocates an RSR-covered, shareable DEDB area or FF database, the counter decreases by 1.

If the SDCRSR\_CVCNT counter equals zero before DBRC deallocates an RSR-covered shareable DEDB, the IMS control region returns user abend 3399. The count must be positive before a deallocation. The count can only reach zero after DBRC deallocates the last remaining RSR-covered, shareable DEDB area or FF database.

**Analysis:** At the time of the abend, register 7 contains the SDCRSR\_CVCNT value of zero. This error is indicative of an IMS internal logic error.

**System action:** IMS terminates abnormally.

**Programmer response:** Because this situation should never occur, save the IMS dump and contact the IBM Support Center.

**Source:** IMS abends

**Module:** DBFLBEU0

### 3400

**Explanation:** This abend is issued by the sample status code error handling routine DFS0AER. Abnormal termination was requested by the sample application program.

Possible reasons are:

- Operator requested abend as a result of error status code (DFS3125A).
- An application program requested abend as a result of error status code.
- The sample status code error handling routine DFS0AER was invoked more than 20 times during the application program execution.

**Analysis:** This application abend is issued in a dependent region by the Primer Function sample status-code error-handling routine DFS0AER. The uses of key registers at the time of the abend are described below:

#### Register

	Usage
7	DIB address (if CICS command level programming)
8	DB-PCB address.
9	I/O PCB address.
11	Base register.

**System action:** IMS processing continues.

**Programmer response:** Check the application program, its invoking of DFS0AER, and the status code error listing produced by DFS0AER.

**Source:** IMS abends

**Module:** DFS0AER

**Related information:**

 [DFS3125A \(Messages and Codes\)](#)

### 3411

**Explanation:** IMS encountered an error when trying to open the data set with the ddname of MODSTAT. Either the DD statement with the ddname of MODSTAT is nonexistent or the operating system has encountered an I/O error during open processing.

**Analysis:** This is a standard abend issued by DFSIIOC0.

**Possible cause:** Either the DD statement with the ddname of MODSTAT does not exist or the operating system has encountered an I/O error during open processing. To correct this problem, either provide the missing DD statement, or create the data set referenced by the MODSTAT DD statement and execute IMS.

**System action:** IMS terminates abnormally.

**System programmer response:** Either provide the missing DD statement or create the data set referenced by the MODSTAT DD statement, or do both, and then execute IMS.

**Problem determination:** 11

**Source:** IMS abends

**Module:** DFSIIOC0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

### 3412

**Explanation:** An I/O error was detected while attempting to read the MODSTAT data set.

**Analysis:** This is a standard abend issued by DFSIIOC0. In order to overcome this problem, reconstruct the MODSTAT data set with the ddnames indicated on the last successful messages, DFS3410I or DFS3499I, and execute IMS.

**System action:** IMS terminates abnormally.

**System programmer response:** Reconstruct the MODSTAT data set with the ddnames indicated in message DFS3410I or DFS3499I, and execute IMS.

**Problem determination:** 11

**Source:** IMS abends

**Module:** DFSIIOC0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

### 3413

**Explanation:** The MODSTAT data set contains an invalid ddname.

**Analysis:** This is a standard abend issued by DFSIIOC0. To overcome this problem, reconstruct the MODSTAT data set with the ddnames indicated on the last successful message, DFS3410I or DFS3499I, and execute IMS.

**System action:** IMS terminates abnormally.

**System programmer response:** Reconstruct the MODSTAT data set with the ddnames indicated in message DFS3410I or DFS3499I, and execute IMS.

## 3414 • 3416

**Problem determination:** 11

**Source:** IMS abends

**Module:** DFSIIOC0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

### 3414

**Explanation:** IMS was unable to allocate storage.

**Analysis:** This is a standard abend issued by DFSIIOC0. IMODULE GETMAIN failed for the MODSTAT work area (MSWA). To overcome this problem, increase the amount of storage available to the region and execute IMS.

**System action:** IMS terminates abnormally.

**System programmer response:** Increase the amount of storage available to the region, and execute IMS.

**Problem determination:** 11

**Source:** IMS abends

**Module:** DFSIIOC0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

### 3415

**Explanation:** IMS was unable to ENQ on the data set associated with the data set described in the ddname of message DFS3415X.

**Analysis:** This is a standard abend issued by DFSIIOC0. To overcome this problem, wait until the other task has released the library and execute IMS.

**System action:** IMS terminates abnormally.

**System programmer response:** Wait until the other task has released the library and execute IMS.

**Source:** IMS abends

**Module:** DFSIIOC0

**Related information:**

 [DFS3415X](#)

---

### 3416

**Explanation:** IMS was unable to find the job file control block (JFCB) for the data set described in the ddname files of message DFS3416X. Either the DD statement with the ddname referenced does not exist, or the operating system encountered an I/O error during open processing.

**Analysis:** This is a standard abend issued by DFSIIOC0. To overcome this problem, either provide the missing DD statement, or create the data set referenced by the specified DD statement; then execute IMS.

**System action:** IMS terminates abnormally.

**System programmer response:** Either create the data set referenced by the MODBLKSA or MODBLKSB DD or provide the missing DD statement, or do both, and then execute IMS.

**Problem determination:** 11

**Source:** IMS abends

**Module:** DFSIIOC0

**Related information:**

 [DFS3416X](#)

---

Chapter 2, "IMS problem determination," on page 13

---

**3419**

**Explanation:** IMS was unable to load the DL/I control block modules. Message DFS3419X identifies the module.

**Analysis:** This is a standard abend issued by DBFRRT00, DFSRDB00, DFSRPG00, and DFSRTR00.

**Possible cause:** There is not enough storage available in the control region or the active MODBLKSA or MODBLKSB data set was not authorized for APF.

**System action:** IMS terminates abnormally.

**System programmer response:** Increase the amount of storage available to the region and execute IMS. Ensure that the active MODBLKSA(B) data set is authorized for APF.

**Source:** IMS abends

**Module:** DBFRRT00, DFSRDB00, DFSRPG00, DFSRTR00

**Related information:**

 DFS3419X

---

**3420**

**Explanation:** IMS detected messages for a transaction that does not exist. The transaction definition does not match the message activity indicated on the IMS system log.

**Analysis:** This is a standard abend issued by DFSQRST0. The log data set is inconsistent with the definition in the MODBLKS data set. The SMB blocks that were loaded from the MODBLKS during initialization are not the same as those that were active during the prior IMS execution. This can occur if the MODSTAT data set was reconstructed specifying an incorrect ddname of the active MODBLKS data set or by changing the contents of the MODBLKS data set while IMS was inactive. Use the same copy of the MODBLKS in use during the last IMS execution and restart IMS with a /ERE from the last checkpoint that was taken with a DUMPQ.

**System action:** IMS terminates abnormally.

**System programmer response:** The log data set is inconsistent with the definition in the MODBLKS data set. The SMB blocks which were loaded from MODBLKS during initialization are not the same as those which were last active during the prior IMS execution. This can occur if the MODSTAT data set was reconstructed specifying an incorrect ddname of the active MODBLKS data set or by changing the contents of the MODBLKS data set while IMS was inactive. Use the same copy of MODBLKS that was in use during the last IMS execution and restart IMS with a /ERE from the last checkpoint that was taken with a DUMPQ.

**Problem determination:** 1, 2, 3, 4, 5, 6, 8, 10

**Source:** IMS abends

**Module:** DFSQRST0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

**3421**

**Explanation:** If IMS detects incompatible control blocks during /MODIFY COMMIT command processing, it issues this abend. The DFSSMB0x or DBFRCTEx control block modules are incompatible with the DFSPDIRx or DFSDDIRx control block modules. The PSBs or routing codes referenced do not exist.

If this abend occurs during an IMS cold start, a PDIR or RCTE could not be found because either no PDIRs exist or no RCTEs exist.

**Analysis:** This is a standard abend issued by DBFCPRC0 or DFSCPSM0 during /MODIFY COMMIT processing. The DFSSMB0x or DBFRCTEx control block modules are inconsistent with the DFSPDIRx or DBFRCTEx control block modules. PSBs or routing codes are referenced that do not exist.

**Possible cause:** Check for errors in the IMS system definition stage 1 source statements. Check that IMS.MODBLKSA(B) (active MODBLKS data set) contains the proper set of control block modules, DFSSMB0x,

## 3422 • 3423

DFSPDIRx, DFSDDIRx or DFSRCTEx where x is the nucleus suffix

**System action:** IMS terminates abnormally.

**System programmer response:** If dynamic resource definition (DRD) is not enabled, check that IMS.MODBLKSx contains the correct control block modules, DFSSMB0x, DFSPDIRx, DFSDDIRx, and DBFRCTEx. If the above checks do not indicate a problem, contact the IBM Software Support for assistance.

If DRD is enabled, and resource definitions are being imported from the IMS.MODBLKSx data set during a cold start, check that IMS.MODBLKSx contains the correct control block modules, DFSSMB0x, DFSPDIRx, DFSDDIRx, and DBFRCTEx. If resource definitions are being imported from a Resource Definition Data Set (RDDS) during a cold start, check that the correct RDDS is being used and that the RDDS has a complete set of resource definitions. If the above checks do not indicate a problem, contact the IBM Software Support for assistance.

**Problem determination:** 1, 2, 3, 4, 5, 6, 8, 10

**Source:** IMS abends

**Module:** DBFCPRC0, DFSCPSM0, DFSRTR00, DBFRRT00

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

### 3422

**Explanation:** Either IMS was unable to allocate storage or the BLDL list did not match the incore blocks during /MODIFY processing.

This abend can also be issued by DFSCPDM0 during MODIFY processing.

**Analysis:** This is a standard abend issued by DBFRRT00, DFSRDB00, DFSRPG00 and DFSRTR00 if IPAGE storage cannot be obtained for the PDIR, DDIR, SMB or RCTE control blocks. This abend is also issued if sufficient storage cannot be obtained to hold the resource extension block or to add the resource to the hash table or alphabetic chain.

Abend 3422 is also issued by DFSCPDM0 and DFSCPPS0 if, during /MODIFY processing, the BLDL list does not match the incore blocks.

**Possible cause:** Enough storage was not available for the IMS control region to increase the storage and execute IMS.

**System action:** IMS terminates abnormally.

| **System programmer response:** For subpool 231, increase the size of ECSA by specifying a larger value for the CSA=  
| parameter of member IEASYSxx in PARMLIB and IPL. For subpool 0, increase the size of e-private storage by  
| specifying a larger value for the REGION= parameter of the IMS control region startup procedure or JCL, and restart  
| IMS.

**Problem determination:** 1, 2, 3, 4, 5, 6, 8, 10

**Source:** IMS abends

**Module:** DBFRRT00, DFSCPDM0, DFSCPPS0, DFSRDB00, DFSRPG00, DFSRTR00

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

### 3423

**Explanation:** This abend is the result of an error that occurred while processing a resource during an IMS cold start or restart.

**System action:** IMS issues message DFS3423E and then terminates abnormally.

**System programmer response:** Refer to message DFS3423E to determine the reason for the abend.

**Problem determination:** 1, 11

**Source:** IMS abends

**Related information:**

 DFS3423E



Chapter 2, "IMS problem determination," on page 13

---

**3430**

**Explanation:** During processing of the TMS procedure, the module ELXKSTAR is not APF-authorized.

**System action:** The TMS subsystem is terminated with abend code and dump.

**Operator response:** Contact the system programmer to authorize the ELXKSTAR module.

**Source:** IMS abends

**Module:** ELXKSTAR

---

**3431**

**Explanation:** While loading the module ELXKALNK, the LOAD macro returns a nonzero return code.

**System action:** The TMS subsystem is terminated with abend code and dump.

**Operator response:** Contact the system programmer to analyze the return code.

**Source:** IMS abends

**Module:** ELXKCDSP

---

**3432**

**Explanation:** While scanning the Wait list for the STASK ECB, the dispatcher found there is no ECB for S-task. This is a logic error.

**System action:** The TMS subsystem is terminated with abend code and dump.

**Operator response:** Contact the system programmer to analyze the problem.

**Source:** IMS abends

**Module:** ELXKCDSP

---

**3433**

**Explanation:** During the creation of S-task, the dispatcher found that the number of S-tasks created are larger than the maximum allowed by TMS subsystem. This is a logic error.

**System action:** The TMS subsystem is terminated with abend code and dump.

**Operator response:** Contact the system programmer to analysis the problems.

**Source:** IMS abends

**Module:** ELXKCDSP

---

**3434**

**Explanation:** While processing the TMS initialization, the ELXKCINT found the current SCP(VS1) is not supported by the TMS subsystem.

**System action:** The TMS subsystem is terminated with abend code and dump.

**Operator response:** Contact the system programmer to analyze the problem.

**Source:** IMS abends

**Module:** ELXKCINT

---

---

3435

**Explanation:** While loading the module, the LOAD macro returns a nonzero return code. R5 contains the address of module name. R6 contains the original abend code from LOAD. R7 contains the original reason code from LOAD.

**System action:** A dump is produced for the abend. The TMS subsystem either continues processing or is terminated.

**Operator response:** Contact the system programmer to analyze the problem.

**Source:** IMS abends

**Module:** ELXKCINT

---

3436

**Explanation:** While processing command input buffer (CIB), the QEDIT macro returns a nonzero return code. R6 contains the logic code which indicates where the abend occurred. R7 contains the original return code from QEDIT.

**System action:** The TMS subsystem is terminated with abend code and dump.

**Operator response:** Contact the system programmer to analyze the problems.

**Source:** IMS abends

**Module:** ELXKCSM

---

3439

**Explanation:** This abend is occurred by operator issuing ABDUMP command.

**System action:** The TMS subsystem is terminated with abend code and dump.

**Source:** IMS abends

**Module:** ELXCABN0

---

3440

**Explanation:** The explanation and actions vary depending on the return code as follows:

**REASON CODE 1 (X'01')**

The return code from STIMERM SET indicates a logic error. The return code was copied to register 5 before the abend was issued. This abend is issued by module ELXKAPS1.

**System action:** The TMSS is terminated with abend code and dump.

**Operator response:** Contact the system programmer to analyze the problem.

**REASON CODE 2 (X'02')**

The return code from STIMERM CANCEL indicates a logic error. The return code was copied to register 5 before the abend was issued. This abend is issued by module ELXKAPS1.

**System action:** The TMSS is terminated with abend code and dump.

**Operator response:** Contact the system programmer to analyze the problem.

**REASON CODE 3 (X'03')**

A call was made to ELXKAPS2 when the skeleton subsystem was in the process of stopping (SVD1STOP set). APS services are not available and the call is invalid. This abend is issued by module ELXKAPS2.

**System action:** The TMSS is terminated with abend code and dump.

**Operator response:** Contact the system programmer to analyze the problem.

**REASON CODE 4 (X'04')**

A call was made to ELXKAPS2 with invalid parameters.

**System action:** The TMSS is terminated with abend code and dump.

**Operator response:** Contact the system programmer to analyze the problem.

**REASON CODE 10 (X'0A')**

FREEMAIN nonzero return code to ELXKAFRM.

**System action:** The TMSS is terminated with abend code and dump.

**Operator response:** Contact the system programmer to analyze the problem.

**REASON CODE 11 (X'0B')**

SETLOCK nonzero return code to ELXKAFRM.

**System action:** The TMSS is terminated with abend code and dump.

**Operator response:** Contact the system programmer to analyze the problem.

**REASON CODE 12 (X'0C')**

GETMAIN nonzero return code to ELXKAGTM.

**System action:** The TMSS is terminated with abend code and dump.

**Operator response:** Contact the system programmer to analyze the problem.

**REASON CODE 13 (X'0D')**

SETLOCK nonzero return code to ELXKAGTM.

**REASON CODE 20 (X'14')**

FREEMAIN nonzero return code to ELXKASSI.

**System action:** The TMSS is terminated with abend code and dump.

**Operator response:** Contact the system programmer to analyze the problem.

**REASON CODE 21 (X'15')**

SETLOCK nonzero return code to ELXKASSI.

**System action:** The TMSS is terminated with abend code and dump.

**Operator response:** Contact the system programmer to analyze the problem.

**REASON CODE 22 (X'16')**

SUBSYSTEM FOUND TO BE ALREADY STARTED - ELXKASSI.

**System action:** The TMSS is terminated with abend code and dump.

**Operator response:** Contact the system programmer to analyze the problem.

**REASON CODE 23 (X'17')**

LOAD failure to ELXKASSI while loading SSI listening post modules.

**System action:** The TMSS is terminated with abend code and dump.

**Operator response:** Contact the system programmer to analyze the problem.

**REASON CODE 24 (X'18')**

SSI already exists - ELXKASSI or ELXKCDSP.

The TMS instance name matches an existing subsystem SSI name.

**System action:** The TMSS is terminated with abend code and dump.

**Operator response:** Contact the system programmer to analyze the problem.

**Programmer response:** The SSI must be unique. Do one of the following:

- Change the INSTANCE name in the TMS SET command statement.
- Change the IMSID parameter in member DFSPBxxx in IMS.PROCLIB, or the IMSID parameter in the IMS procedure library for the IMS system with the duplicate subsystem SSI name.
- Change the duplicate subsystem SSI name in member IEFSSNxx in SYS1.PARMLIB.

**REASON CODE 32 (X'20')**

GETMAIN nonzero return code to ELXKABSK (ELXKABSK and ELXKABGS).

**System action:** The TMSS is terminated with abend code and dump.

**Operator response:** Contact the system programmer to analyze the problem.

## 3469 • 3476

### REASON CODE 33 (X'21')

SETLOCK nonzero return code to ELXKABSK (ELXKABSK and ELXKABGS).

**System action:** The TMSS is terminated with abend code and dump.

**Operator response:** Contact the system programmer to analyze the problem.

### REASON CODE 35 (X'23')

Invalid input passed to ELXKABSK to free a stack (ELXKABFS).

**System action:** The TMSS is terminated with abend code and dump.

**Operator response:** Contact the system programmer to analyze the problem.

### REASON CODE 44 (X'2C')

Size of the subsystem Entry Table Descriptor is too small (ELXKAXMI).

**System action:** The TMSS is terminated with abend code and dump.

**Operator response:** Contact the system programmer to analyze the problem.

### REASON CODE 45 (X'2D')

Could not make the subsystem address space non-swappable (ELXKAXMI).

**System action:** The TMSS is terminated with abend code and dump.

**Operator response:** Contact the system programmer to analyze the problem.

**Source:** IMS abends

**Module:** ELXKAPS1, ELXKAPS2, ELXKAFRM, ELXKAGTM, ELXKASSI, ELXKCDSP, ELXKABSK, ELXKABGS, ELXKAXMI

---

## 3469

**Explanation:** While acquiring stack storage during prologue processing, the stack manager abends due to unavailable storage.

**System action:** The TMS Subsystem terminates with an abend code and dump.

**System programmer response:** Contact the system programmer to analyze the problem.

**Source:** IMS abends

**Module:** ELXKALNK

---

## 3476

**Explanation:** The control region initialization detects that the Extended Terminal Option (ETO) feature has not been either installed or licensed as was requested with the execution parameter ETO=Y. A DFS3676I message indicating the error accompanies this abend.

**Analysis:** This is a standard abend issued by module DFSAINB0. If you do not plan to use the ETO feature, change the execution parameter to ETO=N. If you plan to use the ETO feature, be sure that it is installed or licensed.

**System action:** IMS terminates abnormally.

**Programmer response:** Correct the condition indicated in the accompanying message.

**Problem determination:** 1, 4, 5, 6

**Source:** IMS abends

**Module:** DFSAINB0

**Related information:**

 DFS3676I

Chapter 2, "IMS problem determination," on page 13

---

---

**3477**

**Explanation:** The TM initialization detects that the primary master logical terminal (PMTO) is not defined in the system. IMS requires the PMTO to be properly specified in the system definition. A DFS0507I message indicating the error accompanies this abend.

**Analysis:** This is a standard abend issued by DFSAINB0. Verify your system definition to ensure that the PTMO is properly specified.

**System action:** IMS terminates abnormally.

**System programmer response:** Correct the condition indicated in the accompanying message.

**Source:** IMS abends

**Module:** DFSAINB0

**Related information:**

 DFS0507I

---

**3498**

**Explanation:** An unexpected user environment error was detected during DL/I processing of a HALDB. The error might have been detected during processing for one of these events:

- Selection of a HALDB partition
- Validation of an extended pointer set
- Correction of an extended pointer set

**Analysis:** This occurred when an unexpected user environment error was detected during DL/I processing of a HALDB for one of the following functions:

- Partition selection
- Validation of an extended pointer set
- Correction of an extended pointer set

The contents of registers 14 through 12 at the time of abend have been saved starting at offset X'C' in the last save area in the PST. This save area starts at label PSTSAVL. The error reason code can be found at offset X'1C' in this save area.

The following reason codes are possible:

**Codes****Explanation****X'8001'**

Prereorganization utility or DFSUPNT0 utility has not been executed prior to load.

**X'8010'**

Target partition was not found.

**X'8031'**

Open DMB failure.

**X'8051'**

User partition selection failed.

**System action:** The application program terminates abnormally.

**Problem determination:** 1, 3, 6, 8, 17a through 17d, 35, 40

**Source:** IMS abends

**Module:** DFSDLR00, DFSDDLE0, DFSDXMT0, DFSDDLDC0

**Related information:**

Chapter 2, "IMS problem determination," on page 13

---

**3499**

**Explanation:** During processing of a DL/I call for a HALDB, one of two situations occurred:

- A request for the abend was encountered from a prior partition structure INIT call or a prior partition structure REBUILD call.
- In the partition selection call (FIRST, NEXT, or TARGET partition), the request for the abend from the Partition Selection exit was received.

In both cases, the abend is assumed to be the result of return code 8 or 12 from the Partition Selection exit.

**System action:** The application program terminates abnormally.

**Programmer response:** Examine the Partition Selection Exit that requested the abend and determine why the request was made:

- If the reason for the prior partition structure call was Initialization or Rebuild and the return code passed to IMS from the exit was 8, analyze the exit to determine why the return code was 8. If the reason for the prior partition structure call was Termination, the return code from the exit is ignored.
- If the reason for the partition selection call was FIRST or TARGET and the return code passed to IMS from the exit was 8 or 12, analyze the exit to determine why the return code was 8 or 12. If the reason for the partition selection call was NEXT and the return code passed to IMS from the exit was 8, analyze the exit to determine why the return code was 8.

The call reason can be found in the PECACT field of the DFSPECA control block, and the return code can be found in the PECRC field. For more information about the Partition Selection exit, see *IMS Version 13 Exit Routines*.

**Source:** IMS abends

---

**3500-3998**

**Explanation:** Reserved for customer use.

**Source:** IMS abends

---

**3999**

**Explanation:** A code path was taken that was coded as "this should never occur."

**System action:** The IMS subsystem terminates with an abend code and dump.

**Operator response:** Contact the system programmer to analyze the problem.

**Source:** IMS abends

**Module:** DBFLBEU0, DBFCRAR0

---

**4000-4093**

**Explanation:** Reserved for customer use.

**Source:** IMS abends

---

**4095**

**Explanation:** This code indicates that an IMS TCB is being terminated abnormally because a different IMS TCB in the control region has terminated abnormally. The initial abend will contain the abend code that describes the problem. The 4095 abend is normal under these circumstances.

**System action:** IMS terminates abnormally.

**Programmer response:** Obtain the SDUMP or SYSUDUMP data set.

**Problem determination:** The dump data set contains information about the original problem. Problem analysis should continue, using the original abend conditions.

Message DFS629I, showing the TCB abend codes other than 4095, and the other related error messages, can be helpful for problem determination. See the message for more information.

**Source:** IMS abends

**Module:** DFSFCST0, DFSFMOD0

**Related information:**

 DFS629I





---

## Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

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  
US*

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*

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 jurisdictions 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.

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 provide 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 Director of Licensing  
IBM Corporation  
North Castle Drive, MD-NC119  
Armonk, NY 10504-1785  
US*

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.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

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.

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 actual people or business enterprises 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.

Each copy or any portion of these sample programs or any derivative work must include a copyright notice as follows:

© (your company name) (year).

Portions of this code are derived from IBM Corp. Sample Programs.

© Copyright IBM Corp. \_enter the year or years\_.

---

## Trademarks

IBM, the IBM logo, and [ibm.com](http://www.ibm.com)<sup>®</sup> are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at <http://www.ibm.com/legal/copytrade.shtml>.

Adobe, the Adobe logo, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Linux is a registered 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.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

UNIX is a registered trademark of The Open Group in the United States and other countries.

---

## 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.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

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 <http://www.ibm.com/privacy> and IBM's Online Privacy Statement at <http://www.ibm.com/privacy/details> the section entitled "Cookies, Web Beacons and Other Technologies" and the "IBM Software Products and Software-as-a-Service Privacy Statement" at <http://www.ibm.com/software/info/product-privacy>.

---

## Bibliography

This bibliography lists all of the publications in the IMS Version 13 library, supplemental publications, publication collections, and accessibility titles cited in the IMS Version 13 library.

<b>Title</b>	<b>Acronym</b>	<b>Order number</b>
<i>IMS Version 13 Application Programming</i>	APG	SC19-3646
<i>IMS Version 13 Application Programming APIs</i>	APR	SC19-3647
<i>IMS Version 13 Commands, Volume 1: IMS Commands A-M</i>	CR1	SC19-3648
<i>IMS Version 13 Commands, Volume 2: IMS Commands N-V</i>	CR2	SC19-3649
<i>IMS Version 13 Commands, Volume 3: IMS Component and z/OS Commands</i>	CR3	SC19-3650
<i>IMS Version 13 Communications and Connections</i>	CCG	SC19-3651
<i>IMS Version 13 Database Administration</i>	DAG	SC19-3652
<i>IMS Version 13 Database Utilities</i>	DUR	SC19-3653
<i>IMS Version 13 Diagnosis</i>	DGR	GC19-3654
<i>IMS Version 13 Exit Routines</i>	ERR	SC19-3655
<i>IMS Version 13 Installation</i>	INS	GC19-3656
<i>IMS Version 13 Licensed Program Specifications</i>	LPS	GC19-3663
<i>IMS Version 13 Messages and Codes, Volume 1: DFS Messages</i>	MC1	GC19-4240
<i>IMS Version 13 Messages and Codes, Volume 2: Non-DFS Messages</i>	MC2	GC19-4241
<i>IMS Version 13 Messages and Codes, Volume 3: IMS Abend Codes</i>	MC3	GC19-4242
<i>IMS Version 13 Messages and Codes, Volume 4: IMS Component Codes</i>	MC4	GC19-4243
<i>IMS Version 13 Operations and Automation</i>	OAG	SC19-3657
<i>IMS Version 13 Release Planning</i>	RPG	GC19-3658
<i>IMS Version 13 System Administration</i>	SAG	SC19-3659
<i>IMS Version 13 System Definition</i>	SDG	GC19-3660
<i>IMS Version 13 System Programming APIs</i>	SPR	SC19-3661
<i>IMS Version 13 System Utilities</i>	SUR	SC19-3662

### Supplementary publications

<b>Title</b>	<b>Order number</b>
<i>Program Directory for Information Management System Transaction and Database Servers V13.0</i>	GI10-8914
<i>Program Directory for Information Management System Transaction and Database Servers V13.0 Database Value Unit Edition V13R1</i>	GI10-8966
<i>Program Directory for Information Management System Transaction and Database Servers V13.0 Transaction Manager Value Unit Edition V13R1</i>	GI10-9001
<i>IRLM Messages and Codes</i>	GC19-2666

## Publication collections

Title	Format	Order number
IMS Version 13 Product Kit	CD	SK5T-8864

## Accessibility titles cited in the IMS Version 13 library

Title	Order number
<i>z/OS TSO/E Primer</i>	SA22-7787
<i>z/OS TSO/E User's Guide</i>	SA22-7794
<i>z/OS ISPF User's Guide Volume 1</i>	SC34-4822





Product Number: 5635-A04  
5655-DSM  
5655-TM2

Printed in USA

GC19-4242-03





Spine information:

IMS Version 13

Messages and Codes, Volume 3: IMS Abend Codes

